

بسم الله الرحمن الرحيم

قال الله تعالى ((سبحانك لا علم لنا إلا ما علمتنا إنك أنت
العليم الحكيم))

و قال أيضا : ((و قل رب زدني علما))

و قال أيضا : ((و قل اعملوا فسيرى الله عملكم ورسوله
و المؤمنون))

أهدي هذا الكتاب إلى الوالدين الكريمين اللذان تعبوا من
أجلي عند صغري و أطلب من المولى عز و جل أن
يرحمهما و يحفظهما كما قال تعالى :
((...و اخفض لهما جناح الذل من الرحمة و قل رب
ارحمهما كما ربياني صغيرا))

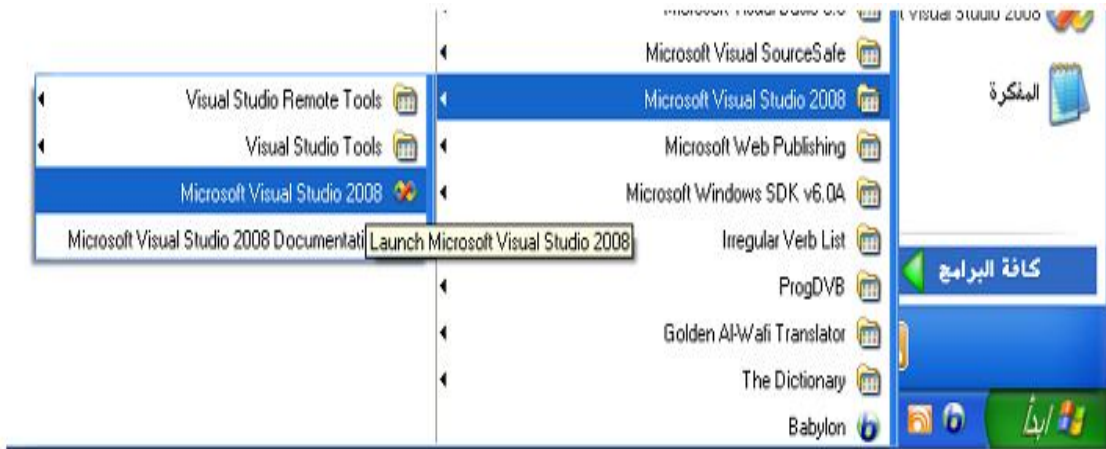
الفصل الأول : التعرف على بيئة التطوير المتكاملة و إنجاز المشاريع

1. فتح و تشغيل برنامج Visual Studio 2008.....3
2. صفحة البداية Start Page.....3
3. إنشاء مشروع جديد Create New Project.....4
4. أنواع المشاريع Project Types.....5
5. نظرة عامة عن بيئة التطوير.....7
6. تصميم نموذج وتخصيص الأدوات.....8
7. مشروع الأول Your First Program.....9
 - 1.7 - تشغيل البرنامج.....9
 - 2.7 - إنشاء مشروع جديد.....9
 - 3.7 - إنشاء الواجهة.....10
 - 4.7 - كتابة التعليمات أو الكود.....11
 - 5.7 - تنفيذ و تجريب البرنامج.....12
 - 6.7 - حفظ البرنامج.....12
8. مكونات بيئة التطوير.....13
 - 1.8 - مستكشف الحل Solution Explorer.....13
 - 2.8 - صندوق الأدوات Toll Box.....16
 - 3.8 - نافذة الخصائص Properties Window.....21
 - 4.8 - نافذة التصميم Designer Form.....24
 - 5.8 - محرر الكود Editor Code.....26
 - 6.8 - الشريط القياسي Standard Bar.....34
 - 7.8 - شريط العنوان Title Bar.....35
 - 8.8 - القوائم Menus.....35
 - 36..... قائمة ملف File ▶
 - 39..... قائمة تحرير Edit ▶
 - 43..... قائمة عرض View ▶
 - 45..... قائمة المشروع Project ▶
 - 51..... قائمة البناء Build ▶
 - 52..... قائمة البيانات Data ▶
 - 56..... قائمة الأدوات Tools ▶
 - 56..... قائمة النوافذ Windows ▶
 - 58..... قائمة المساعدة Help ▶

تشغيل البرنامج

لتشغيل برنامج Microsoft Visual Studio 2008 اتبع الخطوات التالية

1. من قائمة ابدأ Start
 2. اختر كافة البرامج All Programs
 3. اختر حزمة Microsoft Visual Studio 2008
 4. اختر برنامج Microsoft Visual Studio 2008
- كما يمكنك إنشاء اختصار على سطح المكتب ، بالنقر المزدوج عليه يتم فتح البرنامج



الشكل 1 / 1 : فتح برنامج Visual Studio 2008

صفحة البداية Start Page

بعد تشغيل البرنامج تظهر أمامك صفحة البداية Start Page تساعدك على إنشاء مشروع جديد أو فتح مشاريع تم حفظها و تقدم لك آخر و أحدث الأخبار عن Visual Studio في صفحات الويب و تعليمات مساعدة سريعة تساعدك للانطلاق من مكتبة MSDN



الشكل 1/2 : صفحة البداية Start Page


و تتكون من النوافذ التالية :

<p>Recent Projects Open : Project Creat: Project</p>	<p>تعرض قائمة المشاريع المستعملة مؤخرا لفتح المشاريع الموجودة و المحفوظة في جهازك للإتشاء مشروع جديد</p>
<p>Getting Started</p>	<p>تعرض قائمة من المواضيع الخاصة بالتعليمات تساعدك و يمكنك من الانطلاق مثل : Creat Your First Application : يعرض لك تعليمات و يعرفك عن كيفية إنجاز مشروعك الأول Learn Visual Basic : تعليمات تساعدك على تعلم Visual Basic</p>
<p>Get New From Microsoft</p>	<p>تعرض معلومات و آخر الأخبار و كل ما هو جديد في صفحات ويب من Microsoft</p>

إنشاء مشروع جديد

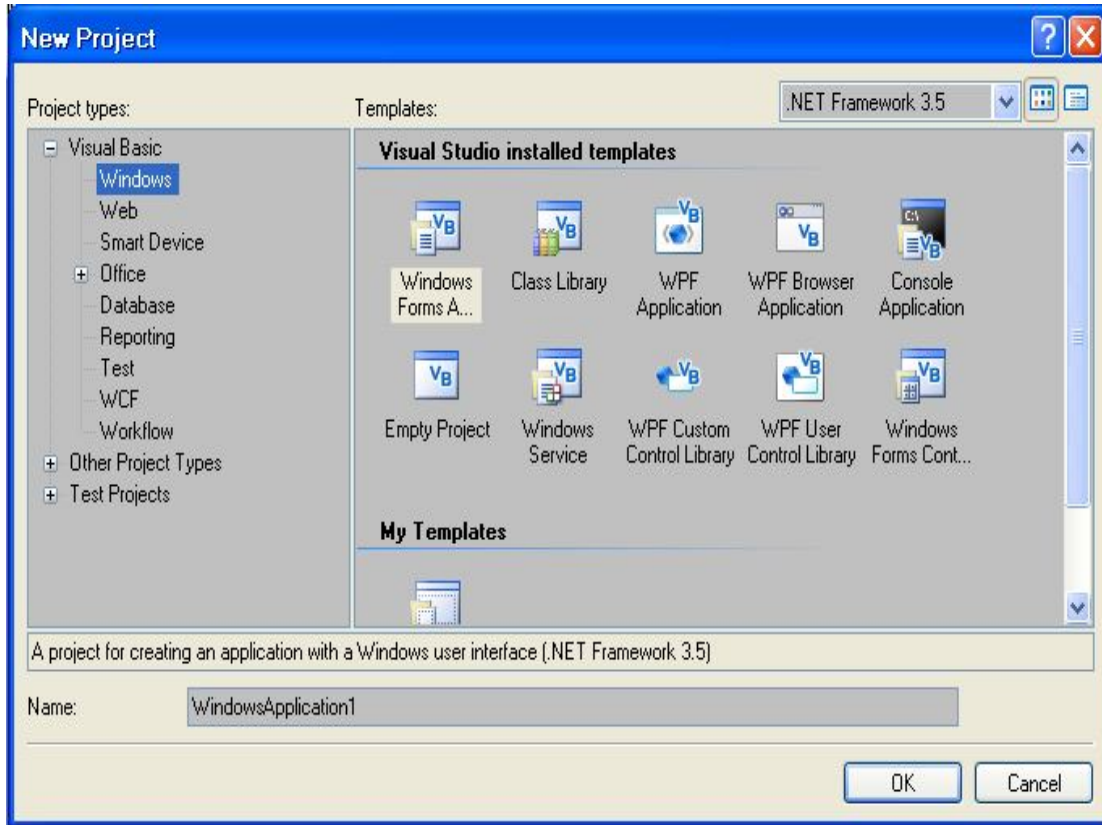
لإنشاء مشروع جديد اتبع الخطوات التالية :

1. من قائمة File اختر الأمر New Project

أو من الشريط القياسي اختر المفتاح  New Project

أو من لوحة المفاتيح **Ctrl + N**

2. يظهر مربع حوار New Project



الشكل 1/3: مربع حوار إنشاء مشروع جديد و أنواع المشاريع

هذا المربع يتكون من قسمين :

قسم أنواع المشاريع **Project Type** : و يحتوى على العديد من المشاريع مثل مشاريع

Windows و مشاريع الويب Web و مشاريع خاصة بقواعد البيانات Data Base و

مشاريع أخرى ..

قسم التطبيقات **Templates** : و يشتمل على أنواع التطبيقات المتاحة لكل مشروع محدد

كما تلاحظ

نوع المشروع Windows الافتراضي يتكون من التطبيقات الهامة التالية :

الفصل الأول : بيئة التطوير المتكاملة IDE

Windows Forms Application تطبيق ويندوز	يستخدم لإنشاء تطبيقات ويندوز
Class Library مكتبة الفئات	يستخدم لإنشاء ملفات DLL و مكتبات خاصة لفئات كثيرة يمكنك الاستفادة منها في مشاريع أخرى
Console Application تطبيق الاختبار	وهو تطبيق ذو واجهة استخداماته محدودة جداً، هي أقرب ما تكون لشاشة DOS السوداء، حيث يتم فيها إخراج النواتج وإدخال البيانات، على سبيل اختبارها.
Windows From Control library مكتبة أدوات تحكم ويندوز	يستخدم إنشاء أدوات خاصة UserControl لاستخدامها في مشروعك مباشرة او تحويلها الى ملفات Dll واستخدامها كملفات مرفقة
Web Control library مكتبة أدوات تحكم الانترنت	ينشئ أدوات مثل User Control ولكنها تعمل على صفحات Html وتطبيقات الويب
Windows Service خدمة ويندوز	تقوم بإنشاء إحدى خدمات Windows التي تعمل بخلفية البرنامج عند بدء تشغيل النظام للقيام بمهام معينة.

و هناك مشاريع أخرى بتطبيقاتها الخاصة

- مشاريع ويب
- ومشاريع التوزيع و النشر
- و مشاريع قواعد البيانات
- وغيرها

3. اختر نوع التطبيق الافتراضي Windows Forms Application

4. في خانة الاسم Name نكتب اسم المشروع

نتركه كما هو افتراضيا نحن الآن بصدد أخذ نظرة عامة عن كيفية إنشاء مشروع و

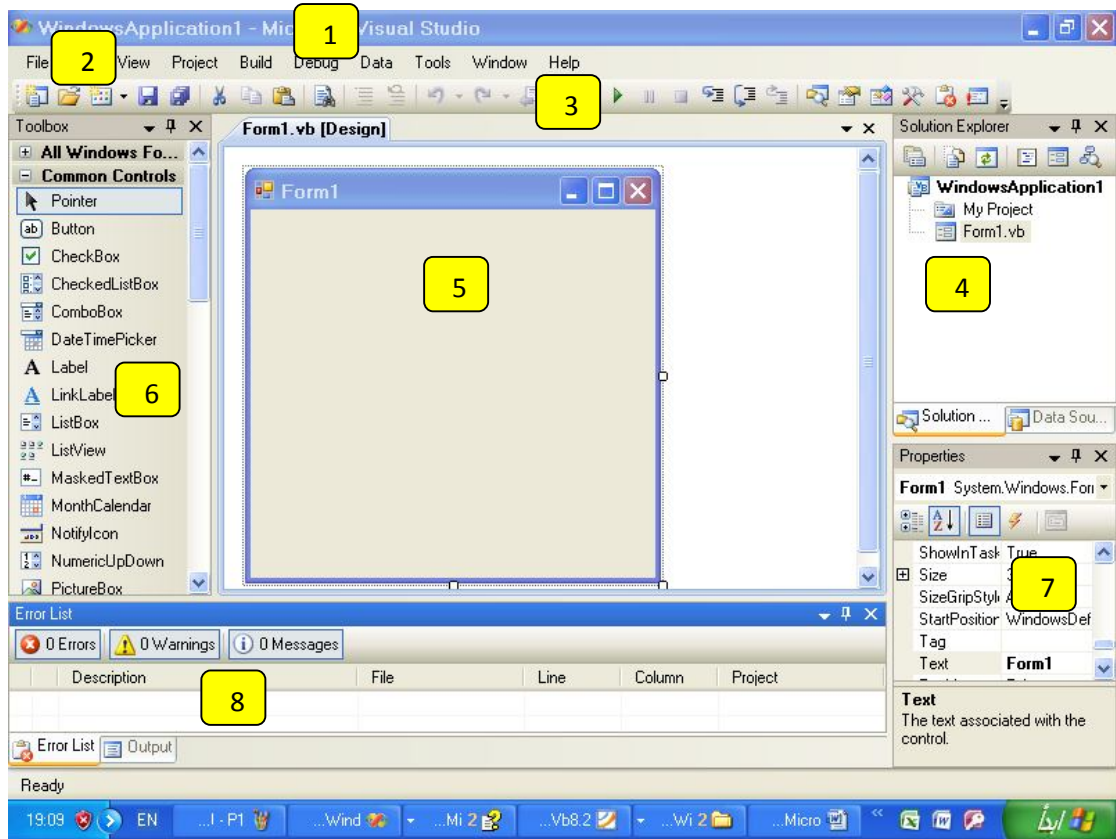
التعرف على بيئة التطوير بصورة عامة

5. انقر Ok أخيرا

ليتم فتح بيئة مليئة بالنوافذ و الأدوات و القوائم هذه البيئة تعرف ببيئة التطوير المتكاملة .

بيئة التطوير المتكاملة IDE

و هي اختصارا Integrated Development Environment و هي وسط عمل وتطوير يحتوي على الكثير من الأدوات و النوافذ بصورة مرئية في قمة الروعة يمكنك من إنجاز مشاريع هامة بعمليات بسيطة في ظرف قياسي و توفر لك كل ما تحتاجه لتصميم نوافذ وبرامج ، كتابة أكواد إدارة ملفات و تحرير قوائم إنشاء قواعد البيانات
 يمكنك من إنجاز مشاريعك بأي لغة من لغات (Visual Studio 2008 ، C# ، C++ ، Vb) في بيئة واحدة مما يغنيك من تعلم بيئة كل برنامج على حدى وتجد نفسك تستعمل نفس الأدوات و الخصائص لإنجاز مشاريعك بمختلف اللغات .



شكل 1/4 : بيئة التطوير المتكاملة IDE ونوافذها

الرمز	ماذا يمثل	توضيح
1	شريط العنوان	Title Bar
2	القوائم Menus	و تشمل كل الأوامر لإنجاز مشاريعك و هي كثيرة
3	الشريط القياسي Standard Bar	و يشتمل على مفاتيح للقيام بالأوامر الهامة و المستعملة كثيرا و هذا للوصول السريع إليها.

الفصل الأول : بيئة التطوير المتكاملة IDE

نافذة تحتوي على كل ملفات المشاريع الحالية	مستكشف الحل Solution Explorer	4
نافذة نرسم و نضيف إليها الأدوات التي نحتاجها في مشروعنا ومنها يتم تنفيذ البرنامج و الأوامر	نافذة التصميم Form Designer	5
يحتوي كل الأدوات التي تحتاجها لإنجاز مشاريعك	صندوق الأدوات Tool Box	6
نافذة يمكنك من إسناد خصائص للأدوات وقت تصميم البرنامج	نافذة الخصائص Properties	7
تعرض الأخطاء المتعلقة بالبرنامج في قائمة و عند النقر المزدوج على عنصر من القائمة يتم الانتقال مباشرة إلى الخطأ لتتقيقه	قائمة الأخطاء Error List	8

والآن تعال معي لتتعرف كيف نبدأ في إنجاز مشاريعنا ؟

تصميم النموذج و تخصيص الأدوات

نقوم بتصميم برنامجنا في نافذة مصمم النماذج Form Designer و هذا بإضافة الأدوات من صندوق الأدوات Tools Box لإنجاز واجهة التطبيق Application interface ثم كتابة التعليمات (الكود) اللازمة لكل أداة ثم نقوم بتنفيذ و تجريب البرنامج. سنتعلم الآن كيف نضيف الأدوات إلى النموذج ؟

1. حلق فوق لافتة Toolbox الموجودة أعلى يسار البيئة

يظهر صندوق الأدوات يحتوي على كل الأدوات التي تستعملها في مشاريعك في مجموعات حسب وظائفها . و نحن بصدد إنجاز مشروع بسيط لهذا نستخدم

أدوات Common Controls

2. انقر نقرا مزدوجا على أيقونة الأداة سيتم إضافتها في أعلى يسار النموذج أو انقر على الأداة ثم قم بسحبها و ارسمها بالحجم و الأبعاد المناسبة كما في برنامج الرسم و إذا أردت حذف الأداة حدها بالنقر عليها ثم اضغط مفتاح Delete كيف نحدد خصائص الأداة ؟

1. نحدد الأداة و نغير خصائصها من نافذة الخصائص Properties Window الموجودة

في أسفل الشاشة على اليمين و إذا لم تكن ظاهرة اضغط F4

مشروعك الأول Your First Project

لإنشاء أي مشروع يجب إتباع الخطوات التالية

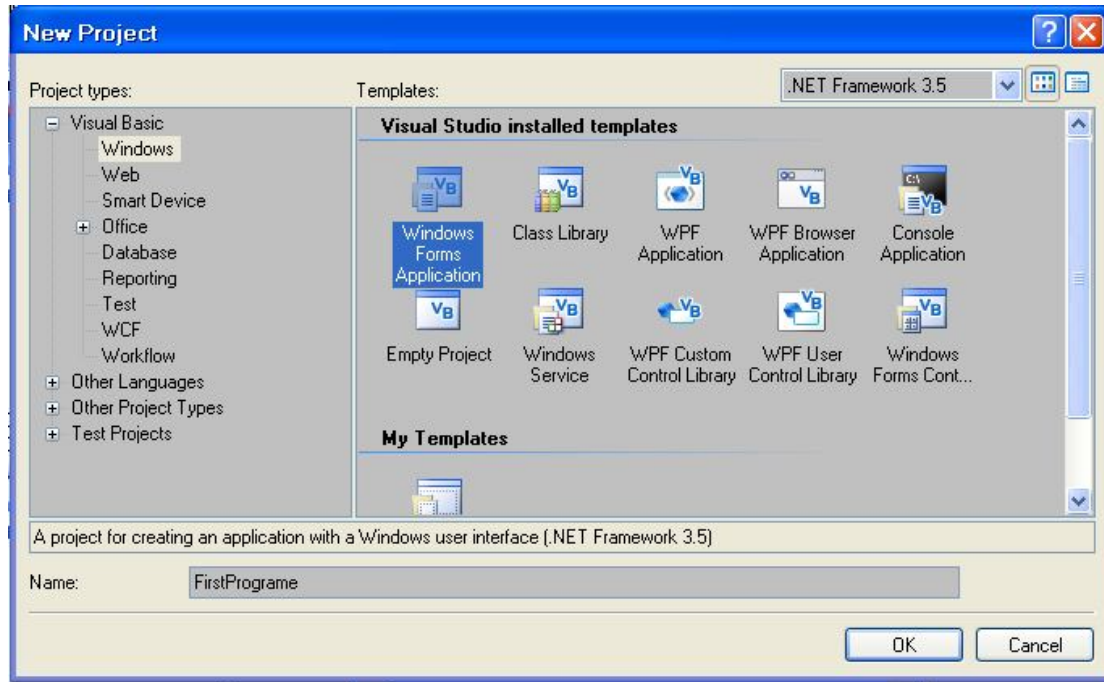
1. تشغيل برنامج Visual Studio 2008
2. إنشاء مشروع جديد Create New Project
3. بناء واجهة التطبيق Create a User Interface
4. كتابة التعليمات Add Visual Basic Code
5. تنفيذ و تجريب البرنامج Run and Test Your Program
6. حفظ المشروع Save Project
7. نشر المشروع Build Project

أولا : تشغيل البرنامج

1. من قائمة ابدأ Start اختر كافة البرامج ثم حدد حزمة Microsoft Visual Studio 2008
ثم أيقونة Microsoft Visual Studio 2008
تظهر أمامك صفحة البداية

ثانيا: إنشاء مشروع جديد

1. من قائمة ملف اختر الأمر New Project
يظهر أمامك مربع حوار مشروع جديد
2. اختر التطبيق Windows Application
3. اكتب في خانة Name اسم مشروعك و ليكن FirstPrograme



4. انقر Ok للدخول إلى بيئة التطوير المتكاملة

ثالثا : بناء الواجهة **Create a User Interface**

1. إضافة الأدوات إلى برنامجك To add controls to your application

▶ انقر تبويب صندوق الأدوات Tool Box

▶ انقر على مجموعة Common Controls

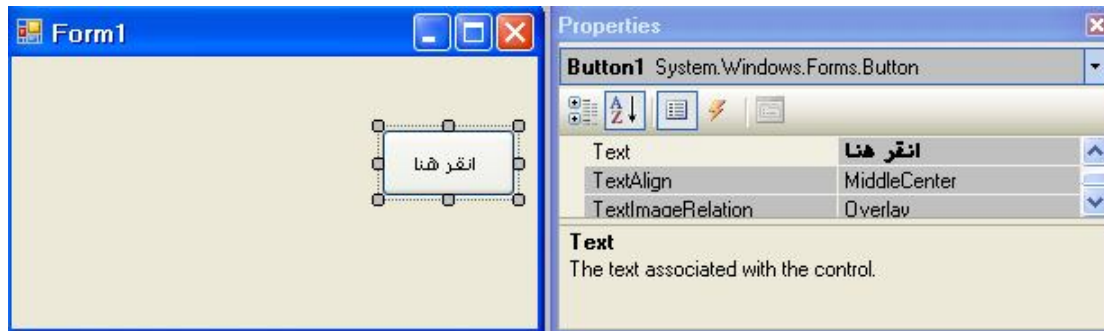
▶ أضف زر Button و أداة عنوان Lable إلى نافذة التصميم

وهذا بعملتي السحب و الفلت

2. ضبط الخصائص Catch Properties

▶ حدد أداة الزر بالنقر عليها الموجودة في نافذة التصميم

▶ من نافذة الخصائص اختر الخاصية Text و اكتب في الخانة المجاورة لها انقر هنا




▶ بنفس الخطوات نكتب في خاصية Text للنموذج برنامج الترحيب

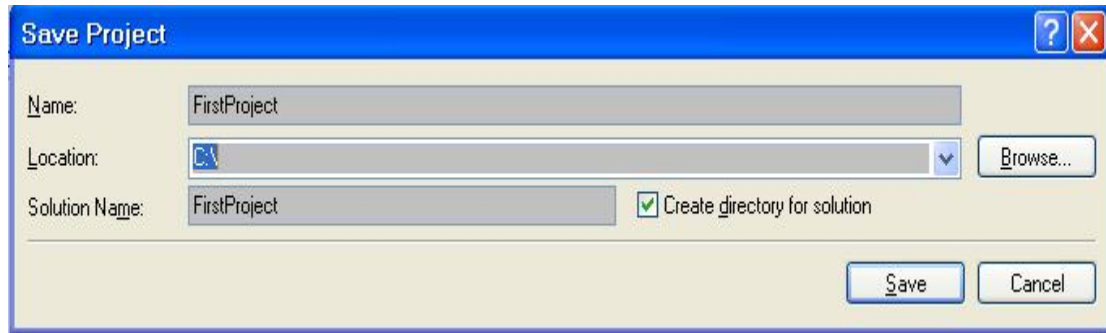
بسم الله الرحمن الرحيم



سادسا : حفظ المشروع

لحفظ المشروع السابق نتبع ما يلي :

من القائمة ملف اختر File Save All أو المفتاح  من الشريط القياسي تظهر نافذة حفظ المشروع Save Project



في خانة Name اكتب اسم المشروع لاحظ أنه يظهر نفس الاسم في خانة Solution Name

Browse : لتحديد مكان لحفظ المشروع

Location : اكتب مسار الملف الذي سيتم فيه حفظ مشروعك

في النهاية انقر Save لحفظ مشروعك







ملاحظة : ما هو الفرق بين الاسم الذي نكتبه في خانة Name و خانة Solution Name

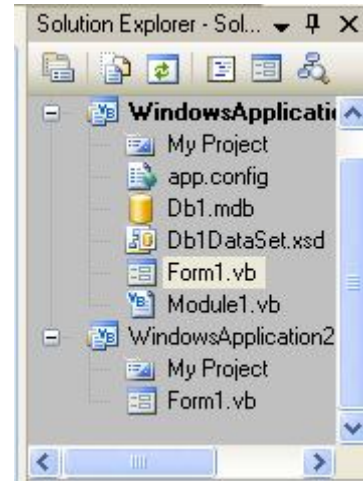
الاسم الذي نكتبه في خانة Name هو الذي يظهر في قائمة All Programs

مكونات بيئة التطوير

أولاً : مستكشف الحل Solution Explorer

نافذة تشمل كل ملفات المشروع الحالي من مشاريع و نماذج و تقارير و قوالب و فئات و صور و قواعد البيانات ...
فهو يعرضها على شكل منظم شجري يسهل لك الوصول إليها بالنقر المزدوج على العنصر يتم فتحه أو عرضه
يوجد في الأعلى على يمين الشاشة و إذا كان غير ظاهر يمكنك عرضه من قائمة View ثم اختر Solution Explorer أو اضغط **Ctrl+r**.

عرض رسم تخطيطي لمكونات البرنامج	View Class Diagram	
عرض نافذة التصميم	View Designer	
عرض محرر الكود	View Code	
تحديث	Refresh	
عرض الكل	Show All	
عرض خصائص النموذج	Properties	

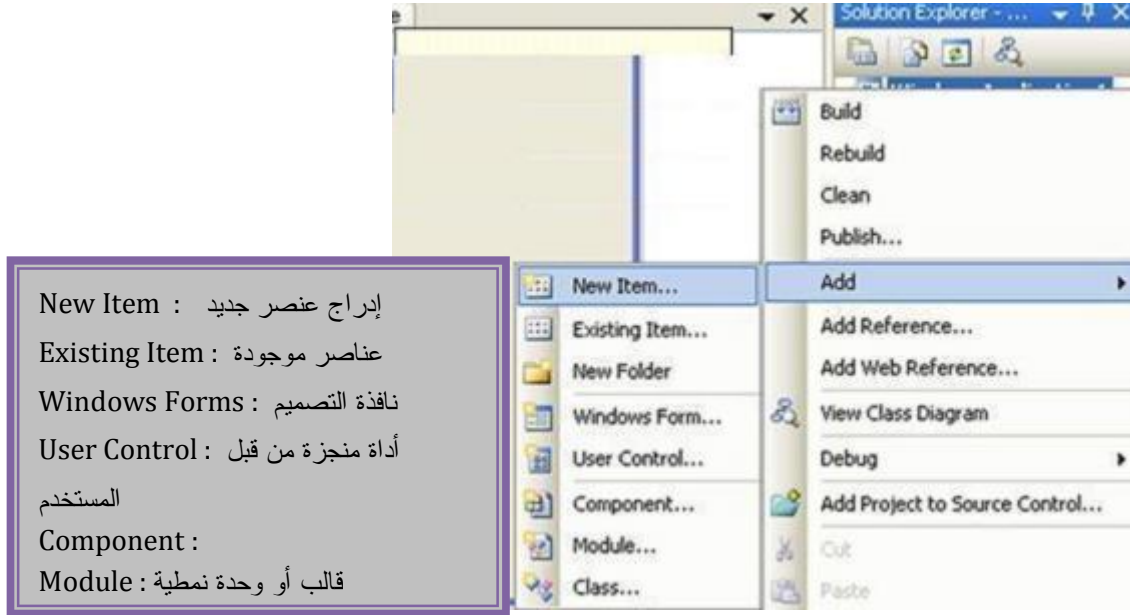


الشكل 1/5 : مستكشف الحل

عند النقر المزدوج على **Form1.vb** سيتم عرض نافذة التصميم . أما عند النقر المزدوج على **MyProject** سيتم عرض نافذة خصائص المشروع

استعمالات مستكشف الحل : Using Solution Explorer

يمكنك من مستكشف الحل إضافة عنصر جديد أو عنصر موجود أو نافذة أو أداة أنجزتها أو أدوات أخرى أو وحدة نمطية أو فئة وهذا بالنقر أعلى المستعرض في اسم المشروع (**Windows Application1** ثم انقر بالزر الأيمن على الفأرة تظهر القائمة المنسدلة التالية



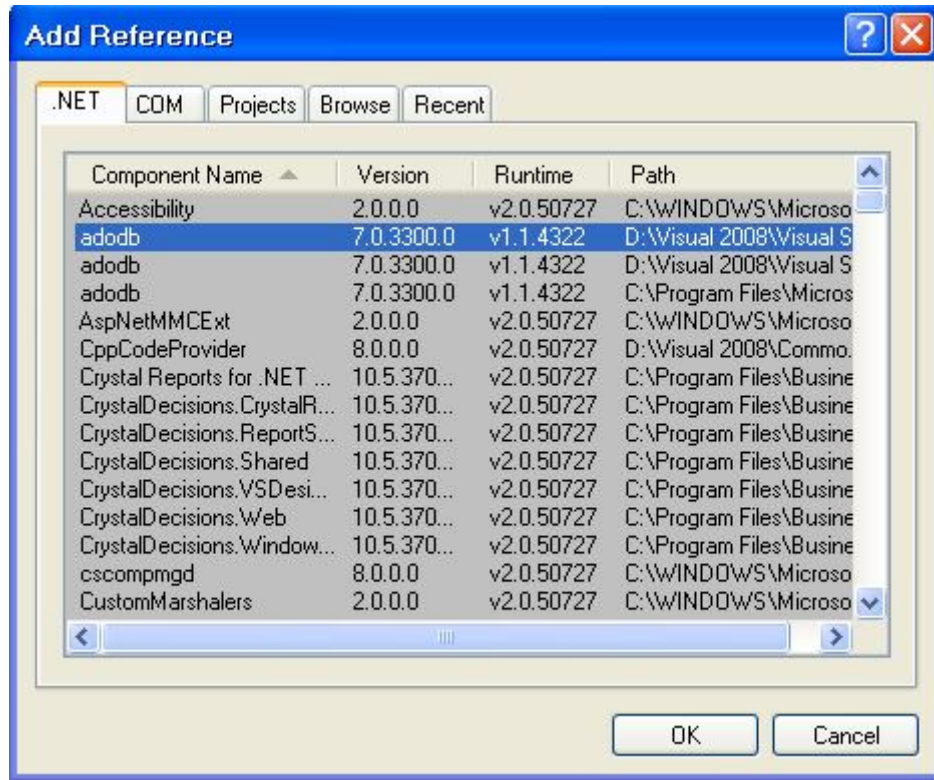
الشكل 1/6 إضافة عناصر من مستكشف الحل

إضافة نموذج جديد Add New Form

1. حدد اسم المشروع مثلا : Windows Application1
 2. انقر بزر الفأرة الأيمن ثم اختر الأمر Add ثم الأمر Windows Form
- يظهر مربع حوار Add New Item ثم حدد العنصر Windows Form
ثم اكتب اسم النموذج في خانة Name ثم انقر Add
سيتم إضافة النموذج إلى مستكشف الحل
ملاحظة : نفس الطريقة نتبعها لإضافة عناصر أخرى موجودة أو قالب أو فئة

إضافة مرجع Add Reference

1. انقر بالزر الأيمن على اسم المشروع (WindowsApplication1)
2. اختر Add Reference
3. يظهر مربع حوار Add Reference كما يبدو في الشكل



الشكل 1/7 : إضافة مراجع

4. حدد الخيار الذي تريد ثم انقر Ok

إضافة مشروع جديد : Adding New Project

1. من قائمة ملف File اختر الأمر Add New Project

2. يظهر مربع حوار يخبرك هل تريد حفظ التغييرات في المشروع الموجود سابقا

اختر Save للحفظ

3. يظهر مربع حوار Add New Project اختر التطبيق Windows Application

واختار اسما للمشروع في خانة Name أو اتركه كما كان. ثم انقر Ok

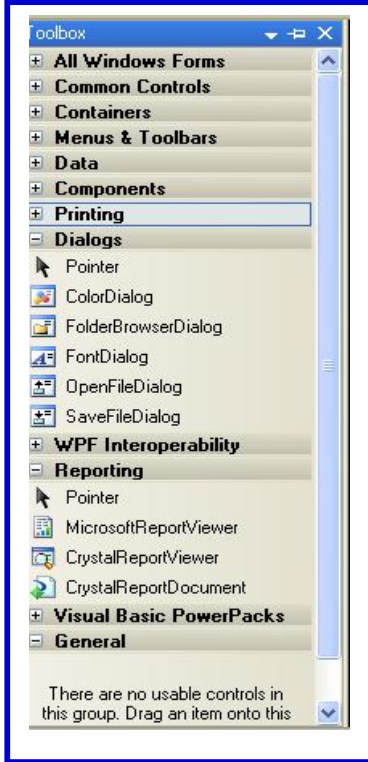
و لاحظ في مستكشف الحل قد تم إضافة هذا المشروع

حذف عناصر من مستكشف الحل To remove a solution item

1. في مستكشف الحل حدد العنصر الذي تريد حذفه

2. انقر على الزر الأيمن ثم اختر حذف Delete

ثانيا : صندوق الأدوات Toolbox



نافذة تحتوي على كل الأدوات التي تستعملها لبناء واجهة التطبيق Application interface. ويعرضها على شكل مجموعات حسب وظائفها و كل مجموعة أمامها علامة (+) عند النقر عليها يتم إظهار الأدوات التي تحويها و تتحول إلى (-) و عند الضغط مرة أخرى على (-) تختفي الأدوات . في الحالة الافتراضية يكون على الجانب العلوي الأيمن من البيئة IDE وفي حالة عدم ظهور الصندوق Toolbox يمكنك عرضه من قائمة عرض View ثم اختر Ctrl+Alt+X .

عند تمرير الفأرة على لوحة Panel هذه يظهر صندوق الأدوات و يختفي عند النقر في أي موضع آخر و يمكنك تثبيت موضع صندوق الأدوات بالنقر على المربع الموجود في عنوان شريط الصندوق ليصبح شكله رأسيا و إذا أردت الرجوع إلى الوضع افتراضي أنقر مرة أخرى على المربع أو انقر نقرا مزدوجا على شريط عنوان صندوق الأدوات الأدوات الخاصة بكل تبويب

الشكل 1/8 صندوق الأدوات

ليصبح شكله رأسيا و إذا أردت الرجوع إلى الوضع افتراضي أنقر مرة أخرى على المربع أو انقر نقرا مزدوجا على شريط عنوان صندوق الأدوات

Tab name	Description
All Windows Forms	Contains all the standard Windows Forms controls that are available in the Toolbox. يحتوي كل الأدوات القياسية المتاحة في صندوق الأدوات
Common Controls	Contains the most common controls that are typically used in Windows-based applications. يحتوي على الأدوات الهامة المستعملة كثيرا في تطبيقات ويندوز
Containers	Contains controls that hold other controls, such as GroupBox controls and Panel controls. Containers can help isolate a set of controls from other controls on the form. يحتوي على الأدوات الحاضنة لأدوات أخرى مثل : Panel ، GroupBox
Menus and ToolBars	Contains controls that enable you to create menus and toolbars for your application, such as the MenuStrip and ToolStrip controls. يحتوي على الأدوات الخاصة بالقوائم و أشرطة الأدوات مثل : MenuStrip ، ToolStrip
Data	Contains controls that help you easily work with data in your application, such as the DataGridView control.

الفصل الأول : بيئة التطوير المتكاملة IDE

	يحتوي على الأدوات التي يمكنك من التعامل مع قواعد البيانات مثل : DataGridView
Components	Contains components that typically do not have a user interface, such as the Timer and the ErrorProvider. يحتوي على أدوات أخرى لا يمكن ظهورها على الواجهة مثل : Timer ، ErrorProvider.
Printing	Contains controls that enable you to provide printing capabilities to an application. يحتوي على أدوات الطباعة
Dialogs	Contains controls that let you use common dialog boxes in your application, such as the SaveFileDialog, FolderBrowserDialog and FontDialog controls. يحتوي على أدوات مربعات الحوار القياسية مثل : SaveFileDialog ...
Reporting	Crystal Report ، Report : يحتوي على أدوات التقارير مثل :
General	فارغ خصص لإضافة أدوات أخرى غير موجودة في صندوق الأدوات

التعامل مع التبويبات Tabs

للتعامل مع التبويبات انقر أولاً على اسم التبويب الذي تريد التعامل معه . ثم انقر بزر الفأرة الأيمن تظهر أمامك قائمة منسدلة مختصرة يمكنك بالقيام بالعمليات

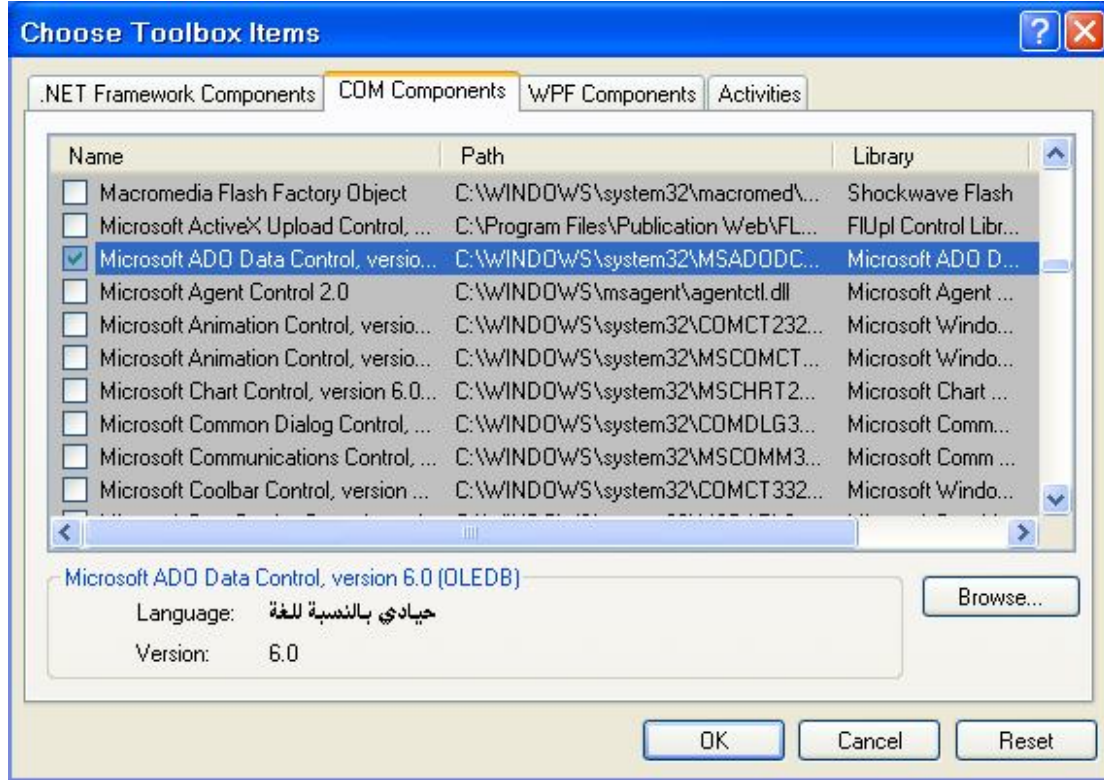
- لإضافة تبويب اختر الأمر Add Tab
- لحذف تبويب اختر الأمر Delete Tab
- لإعادة تسمية تبويب اختر الأمر Rename Tab
- لنقل تبويب إلى أسفل اختر **Move Down**
- لنقل تبويب إلى أعلى اختر **Move Up**
- لترتيب و فرز الأدوات في التبويب حسب الترتيب الأبجدي اختر Sort Items
alphabetically
- لاستعادة الافتراضيات لصندوق الأدوات انقر Reset Tool box
- لعرض كل الأدوات الخاصة بصندوق الأدوات و كل التبويبات اختر Show All
- لإدراج عنصر أو أداة جديدة اختر Choose Items

إضافة تبويب مخصص To add a custom Toolbox tab

- انقر بزر الفأرة الأيمن في صندوق الأدوات اختر Add Tab من القائمة المختصرة
- في مربع النص الخاص بالتبويب اكتب اسم التبويب و اضغط Enter سيتم إضافة هذا التبويب الجديد و يمكنك التعامل معه مثل باقي التبويبات

إضافة و حذف أدوات أخرى إلى صندوق الأدوات

1. انقر بزر الفأرة الأيمن على صندوق الأدوات ثم اختر Choose Items



الشكل 1/9 : إضافة أدوات أخرى

2. حدد الأداة بنقر خانة الاختيار ثم انقر OK سيتم إضافة الأداة

3. و لحذفها اعد الخطوتين السابقتين و عند نقر خانة الاختيار سيتم نزع التحديد وعند

النقر OK سيتم حذف الأداة

4. أو بالنقر على الأداة في صندوق الأدوات بالزر الأيمن ثم اختيار Delete سيتم

حذف الأداة

التعامل مع الأدوات :

نقل أداة في الصندوق :

▪ انقر على الأداة ثم قم بسحبها Drag باستمرار ثم أفلتها في الموضع المختار

إعادة تسمية عنصر في الصندوق rename an item on a Toolbox tab

▪ انقر بيمين الفأرة على العنصر (أداة)

▪ اختر من القائمة المنسدلة Rename Item

▪ ثم اكتب اسم العنصر الجديد في صندوق النص ثم اضغط ENTER

نقل عنصر من تبويب إلى آخر move an item from one Toolbox tab to another

- قم عمليتي السحب و الإفلات من تبويب إلى آخر
- أو بعملتي القص و النسخ

إضافة أدوات إلى النموذج Add Controls To a Form

قم بإحدى الخطوات التالية :

- ▶ بالنقر المزدوج على الأداة في صندوق الأدوات سيتم إضافة الأداة
- ▶ أو بالنقر على الأداة و سحبها ثم فلتها على النموذج كما في برنامج الرسام
- ▶ في حالة إضافة عدد كبير من الأدوات من نفس النوع كأزرار في الآلة الحاسبة

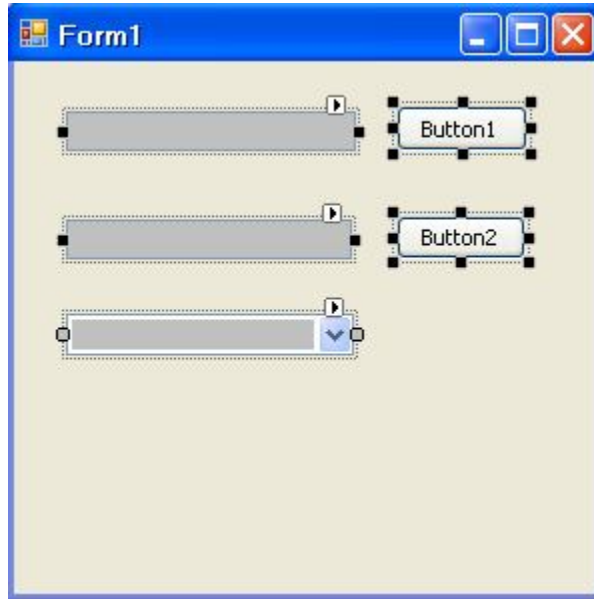
اضغط Ctrl ثم انقر على الزر في صندوق الأدوات ثم انقر على النموذج Form

في كل نقرة يتم إضافة زر و إذا أردت إيقاف عملية الإدراج افلت مفتاح control

أو باستعمال طريقة لقص و اللص

تحديد الأدوات : Selecting Controls

قم بالضغط على الماوس ثم ارسم مستطيلا يحوي كل الأدوات ثم افلت زر الماوس سيتم تحديد كل الأدوات

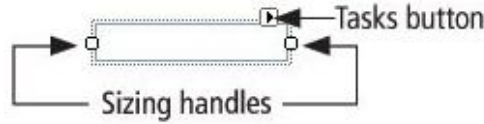
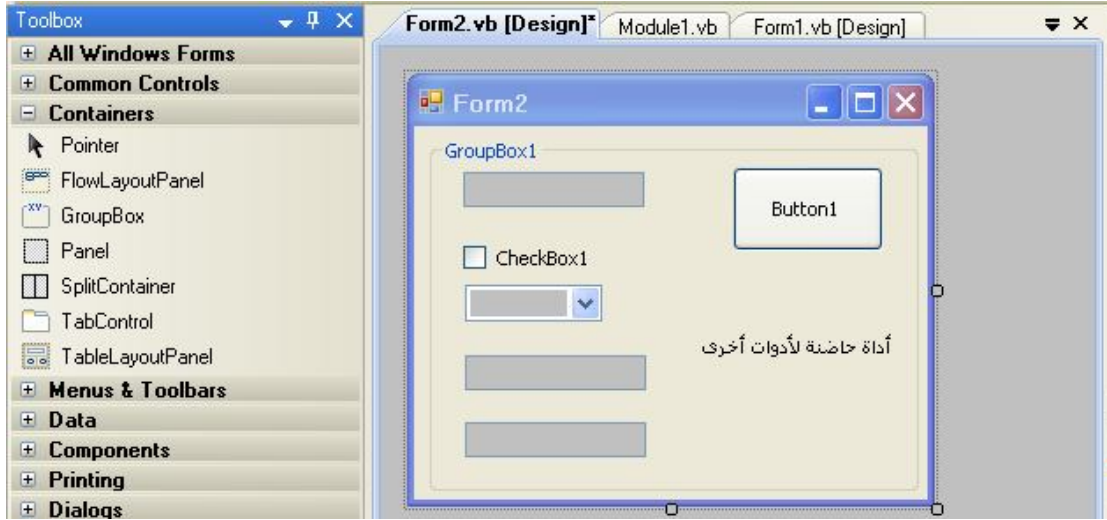


تجميع الأدوات في حاوية Container Controls

يمكنك تجميع الأدوات في حاويات مثل : TabControl, GroupBox, Panel باتباع إحدى

الخطوات التالية :

- ▶ عن طريق القص و اللص
- ▶ أو السحب و الفلت
- ▶ بتحديد الحاوية و النقر المزدوج في صندوق الأدوات



تحجيم ونقل أداة عن طريق الفأرة

- ▶ حدد الأداة التي تريد زيادة حجمها
- ▶ تظهر نقاط على شكل مستطيل
- ▶ قم بتمرير الماوس فوق هذه النقطة عندما يتغير شكل المؤشر إلى سهم من الجهتين
- ▶ اسحب مع الضغط لزيادة طول أو عرض الأداة
- ▶ عند تمرير الفأرة فوق النقطة عندما يتغير شكل المؤشر إلى 4 أسهم
- ▶ انقل الأداة بالضغط المستمر عليه إلى الموضع الذي تريد ثم افلتها
- ▶ كما يمكنك نقلها عن طريق لوحة المفاتيح باستعمال الأسهم

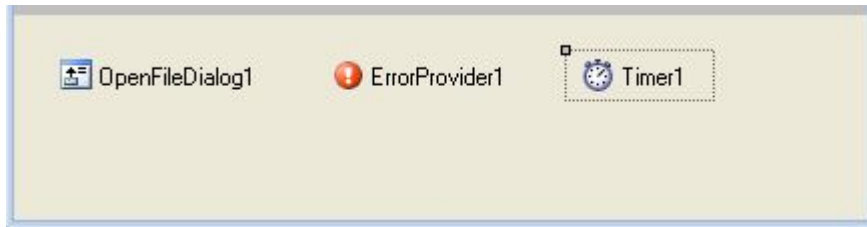
تصنيف الأدوات

Control : هو العنصر البرمجي الذي يأخذ شكل رسومي ويوضع على النموذج

Form مثل : Button, Label, TextBox

Component : وهو العنصر البرمجي لا يظهر على النموذج و إنما في شريط

خاص مثل :



الفصل الأول : بيئة التطوير المتكاملة IDE



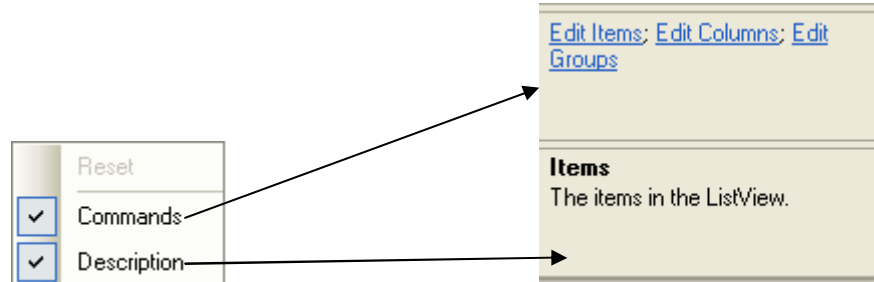
الشكل 1/10 : نافذة الخصائص وتخصيص قيم لها

القائمة المنسدلة فهي تحتوي على أسماء كل الأدوات الموجودة على النموذج والتي يمكنك التنقل بينها لاستعراض خصائصها، أو يمكنك النقر عليها مباشرة بالفأرة على النموذج تحت القائمة المنسدلة شريط قوائم قصير يحوي بعض الأزرار كما بالشكل التالي:

Events : لعرض الأحداث الموافقة لأداة المحددة
 Properties : لعرض نافذة الخصائص
 Alphabetic : ترتيب الخصائص أو الأحداث ترتيباً أبجدياً
 Categorizes : ترتيب الخصائص أو الأحداث حسب الوظائف

الشكل 1/11 نافذة الخصائص

هذا الجزء مخصص لعرض تلميحات و معلومات سريعة عن كل خاصية أو حدث محدد عند النقر على نافذة الخصائص بالزر الأيمن للفأرة نحصل على القائمة المختصرة



Commands يعرض الخصائص المتعلقة بالمجموعات collection انظر الشكل ب
 Description : عرض التلميحات و المعلومات السريعة لكل خاصية محددة

الفصل الأول : بيئة التطوير المتكاملة IDE

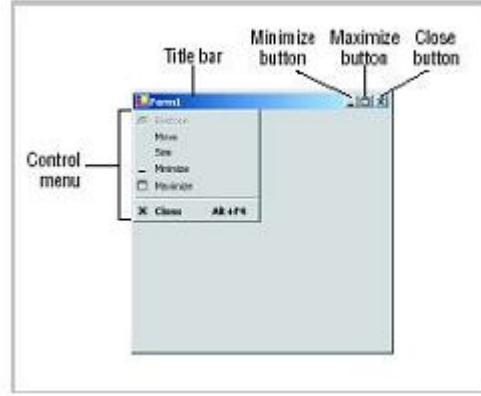
Control Menu : أوامر قائمة التحكم

Title Bar : شريط العنوان

Minimize : للتصغير النموذج

Maximize : لتكبير النموذج

Close : للإغلاق



الشكل 1/12 : نافذة التصميم

وظيفة	أوامر قائمة التحكم
يستعيد هذا الأمر حجم النموذج الأصلي، حينما يكون في أقصى حجم له .Maximized	استعادة Restore
يسمح للمستخدم بتحريك النموذج على الشاشة باستخدام لوحة المفاتيح لإنهاء التحريك اضغط زر الإدخال Enter ليستقر النموذج في الموضع الجديد، أو زر الإلغاء Esc ليبقى النموذج في موضعه القديم.	تحريك Move
يشبه الأمر السابق، إلا إن ضغط الأسم من لوحة المفاتيح يؤدي لتغيير حجم النموذج.	تغيير الحجم Size
يجعل النموذج يصل لأقصى حجم له ليملاً الشاشة.	تكبير Maximize
يجعل النموذج يختفي من على الشاشة، ليصير مجرد أيقونة على شريط مهام	تصغير Minimize
يغلق النموذج نهائياً.	إغلاق Close



إذا نقرت بزر الفأرة الأيمن على الفورم تظهر القائمة المنسدلة التالية :

للقيام بقتل الأدوات و منع التغيير في أطوالها وأحجامها ومواقعها	Lock Controls		View Code
للانتقال إلى خصائص الأداة المحددة .	Properties		Lock Controls
			Paste
			Properties

لاحظ وجود الخيار View Code وهو ما يسمح أيضا بالانتقال إلى طور معالجة الكود.

هو عبارة عن مستند Document لتحرير التعليمات البرمجية (كود) الخاصة ب Visual Studio development language

يمكن الوصول إلى نافذة محرر الكود بإحدى الطرق التالية :

Solution Explorer	انقر على الأيقونة View Code  الموجودة في شريط عنوان مستكشف الحل
Windows form or Web form	انقر نقرا مزدوجا على الأداة أو النموذج أو انقر بيمين الفأرة على النموذج ثم اختر View Code من القائمة المختصرة . يمكنك التنقل بين نافذة النموذج و محرر الكود بالتنقل بين التبويبات كما في الشكل  Form1.vb : عرض نافذة الكود Form1.vb[Desing]
View Menu	من قائمة View اختر الأمر Code

محرر الكود ومكوناته

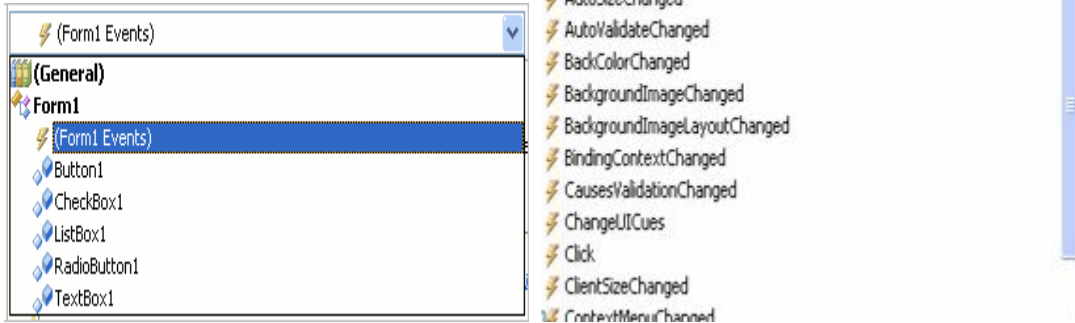


الشكل 1/13 : نافذة محرر الكود و أجزائها

- 1- لإظهار نافذة محرر الكود Form.vb
 - 2- لإظهار نافذة النموذج Form .Vb [Desin]
 - 3- لسرد الأدوات المستعملة في المشروع
 - 4- لسرد أحداث الأداة المحددة في القائمة 3
- بمحرر الكود وفي القائمة (4) تحتوي على كل الأحداث التي يمكنك قنصها كما يلي

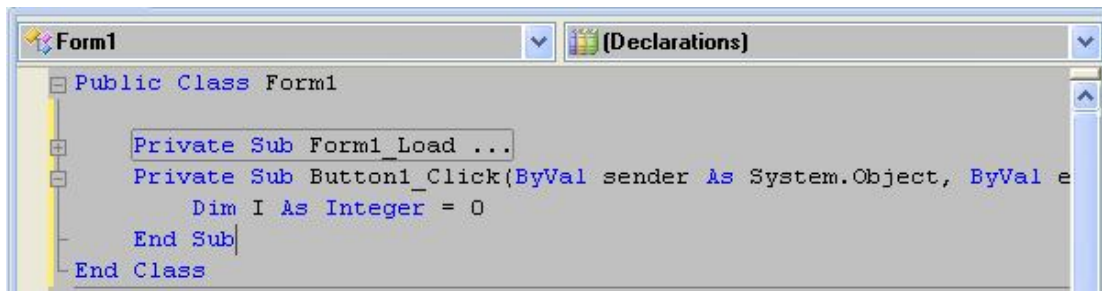
الفصل الأول : بيئة التطوير المتكاملة IDE

أما القائمة التي على يسار محرر الكود فتشمل كل الأدوات التي استعملتها في مشروعك

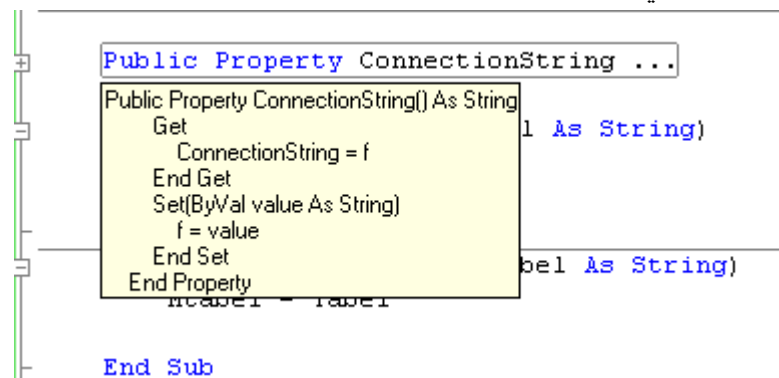


خصائص و مزايا محرر الكود

1- إخفاء و إظهار الإجراءات :



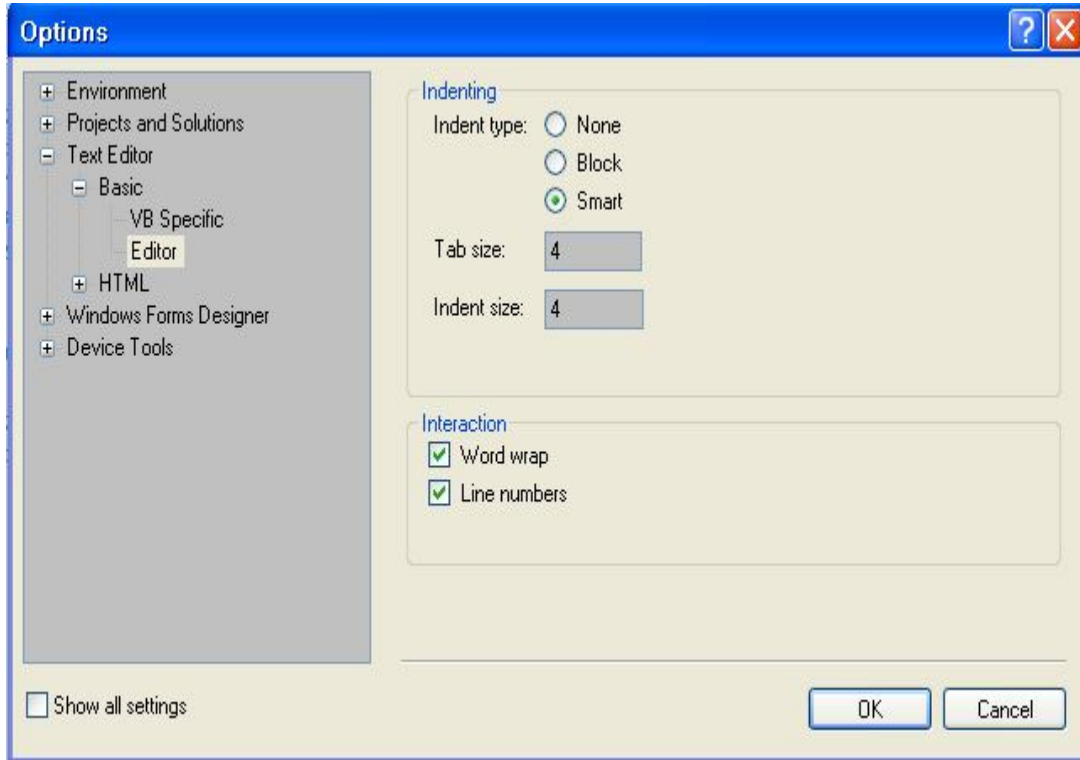
مع بداية كل إجراء توجد علامة - و تعني أن الإجراء معروض و لإخفائه انقر على العلامة - و تصبح + كما في الشكل السابق . هذا يمكنك من قراءة و رؤية التعليمات بكل سهولة و خاصة إذا كان المحرر يحتوي على العشرات من الإجراءات . كما يمكنك معاينة و قراءة إجراء مخفي بتمرير الفأرة فوقه انظر الشكل



2 ترقيم الأسطر: Outlining

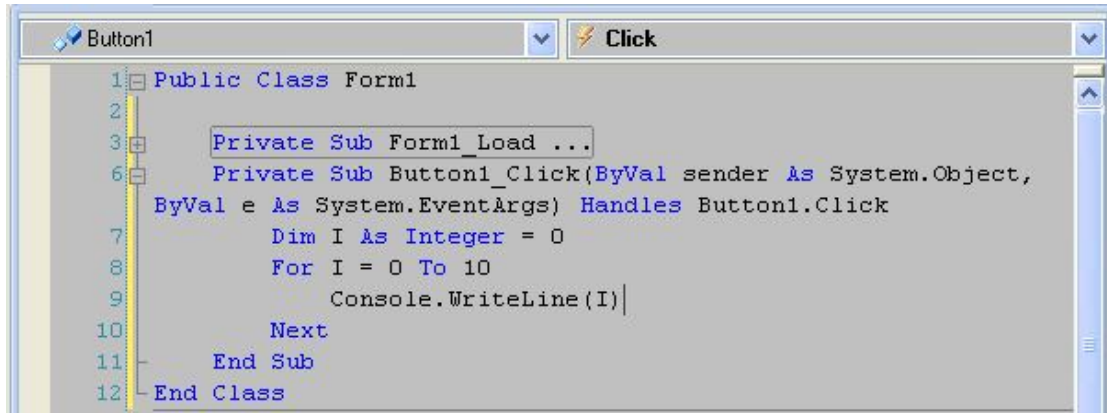
اتباع الخطوات التالية لترقيم صفحة محرر الكود لترقيم صفحة محرر الكود :

Tools -> Option -> Text Editor -> Basic -> Editor -> Line Numbers



الشكل 1/14 : نافذة الخيارات لتنسيق محرر الكود

ملاحظة : كما يوجد الخيار Word Warp فعند تحديده يتم عرض الأسطر آليا بحيث يراها المستخدم دون استعمال العلامة _ لكسر السطر .



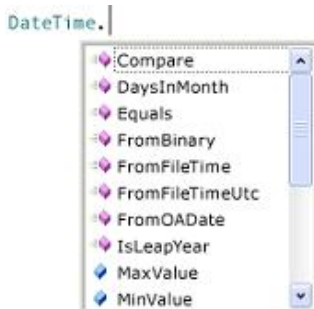
3-الوظيفة IntelliSense

من أهم ميزات Vs.Net 2008 السهولة في كتابة التعليمات البرمجية وذلك بفضل الوظيفة IntelliSense تقدم هذه الوظيفة عدة وظائف تساعد المبرمج على كتابة تعليمات ببساطة وبدون أخطاء و تتمثل في :

1-سرد الأعضاء List Members

عند كتابة اسم كائن أو أداة متبوعا بمؤشر نقطة point

Operator تظهر قائمة بجميع خصائص و أحداث ووظائف الكائن



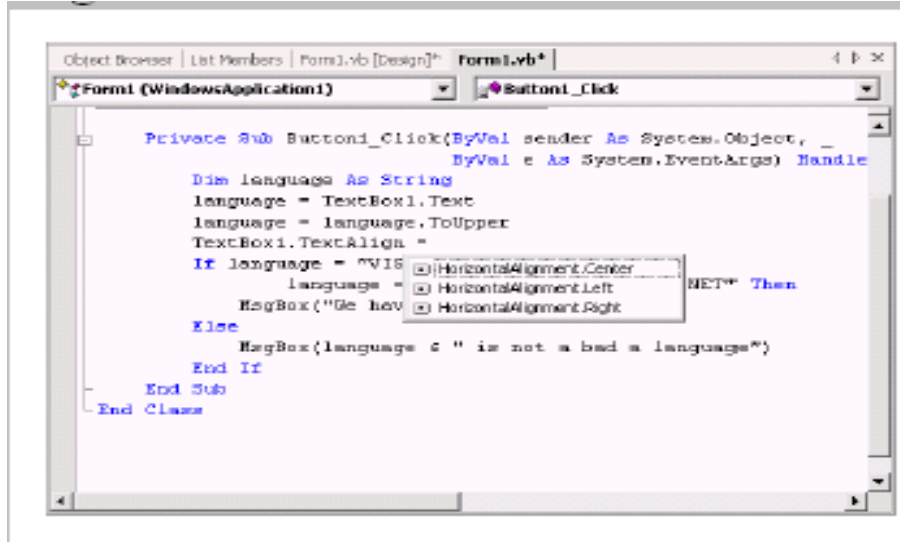
ما يعرف بأعضاء الكائن List Members كما في الشكل وهذا يتيح لك أمورا في غاية الأهمية:

- يريحك من عناء تذكر كل أسماء أعضاء هذا الكائن، وهي كثيرة بالفعل لكل كائن
- يسهل لك الانتقال بين الأعضاء Members بالفأرة أو بلوحة المفاتيح، وفي كل مرة تحدد فيها أحد الأعضاء، يظهر لك تلميح على الشاشة يشرح وظيفة العضو بطريقة مختصرة، وبهذا يمكنك أن تتعلم بسرعة العديد من خصائص ووظائف الكائن، وتستطيع العثور على الوظيفة التي تريدها، حتى لو لم تكن تعرفها من قبل.
- بالضغظ المزدوج بالفأرة، أو ضغظ المسافة أو زر الإدخال من لوحة المفاتيح، يمكنك كتابة اسم العضو المحدد حاليا في القائمة، في الموضع الحالي من الكود، وبهذا توفر وقت كتابة الكلمة بنفسك، وترحم نفسك من كثرة الأخطاء الهجائية.
- وحتى لا تضطر للتنقل بين الأعضاء طويلا، للوصول إلى عضو متأخر في الترتيب، اكتب حرفاً أو حرفين من اسمه، وستجد أن القائمة تنتقل تلقائياً لتحديد العضو الذي يبدأ بما كُتِب من حروف.. وعندما يظهر لك اسم العضو على الشاشة اضغظه بالفأرة مرتين، أو صل إليه بالأسهم من لوحة المفاتيح واضغظ مسافة، عندئذ سيتم إكمال كتابة الاسم في الكود.
- وإذا كانت هناك خاصية لا تقبل أية قيمة إلا من بين مجموعة محددة من القيم، ستجد أن قائمة بهذه القيم ستظهر لك فور كتابتك لعلامة "=" بعد اسم الخاصية.. فلو كتبت:

TextBox1.TextAlign =

فستظهر لك قائمة تحتوي على القيم التالية – أو ما يعادلها من أرقام – المسموح بها للخاصية:

HorizontalAlignment.Center
HorizontalAlignment.Right
HorizontalAlignment.Left



4- معلومات الإعداد Parameter Info

Console.Write(|

▲1 of 18 ▼ void Console.Write (bool value)
value: The value to write.

عند كتابة طريقة ما و تعقبها بفتح قوس يظهر تلميح يقدم معلومات عن الإعدادات الموجودة كما يبدو في الشكل التالي

5- المعلومات السريعة Quick Info

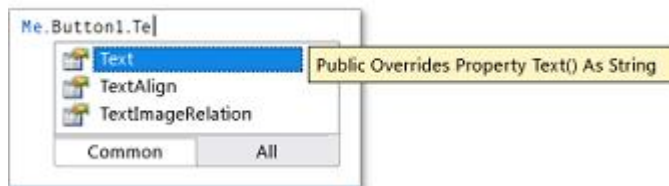
عند تمرير مؤشر الفأرة فوق أي كائن أو متغير سيظهر تلميح يقدم لك معلومات سريعة عنه كما في الشكل

DateTime.Compare(FirstDate, SecondDate):

int DateTime.Compare (DateTime t1, DateTime t2)
Compares two instances of System.DateTime and returns an indication of their relative values.

6- إكمال الكلمات Complete Word

عند كتابة طريقة أو خاصية بمجرد كتابة الأحرف الأولى تظهر لك قائمة تستكمل كتابة الكلمات بمجرد الضغط على مفتاح Space أو Enter يتم كتابتها .



تدريب نموذجي :

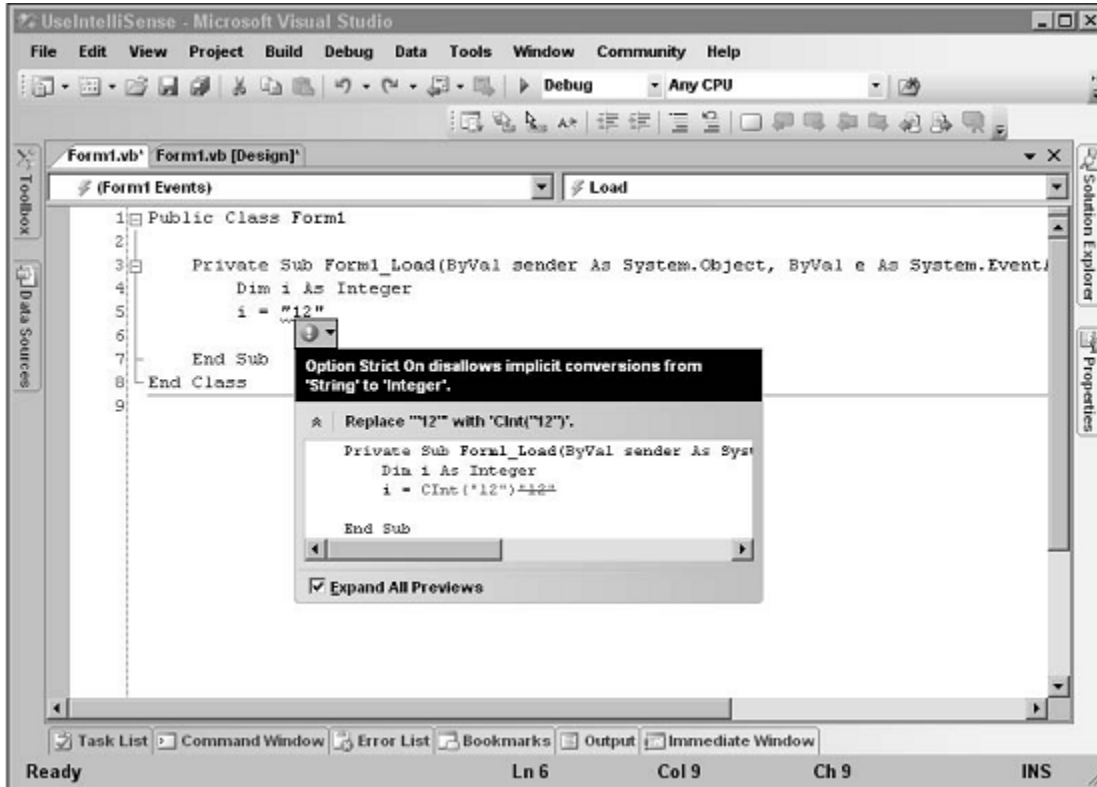
1. من قائمة ملف اختر New Project
تظهر نافذة إضافة مشروع جديد
2. حدد التطبيق Windows Application ثم اختر Ok
3. ضف أداة زر إلى النموذج
4. انقر نقرا مزدوجا على الزر لتنتقل إلى محرر الكود و حدث النقر للزر
5. اكتب button1 متبوعة بنقطة لاحظ ظهور قائمة طويلة List Members

الفصل الأول : بيئة التطوير المتكاملة IDE

6. اكتب T ولاحظ تقليص أعضاء القائمة إلى أعضاء تبدأ بحرف T فقط
7. حدد الخاصية Text ثم انقر على Enter أو مفتاح المسافة أو بالنقر المزدوج سيتم إدراج الخاصية ثم اكتب = متبوعة بالنص " موافق "
8. ثم انقر مفتاح Run و اضغط زر Button

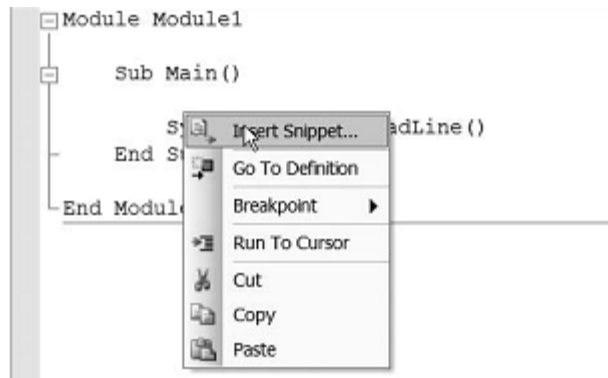
7- استعمال الموجه Snippets

في حالة حدوث خطأ أحيانا تظهر قصاصة Snippet كما في الشكل التالي و توضح لنا كيفية معالجة الخطأ

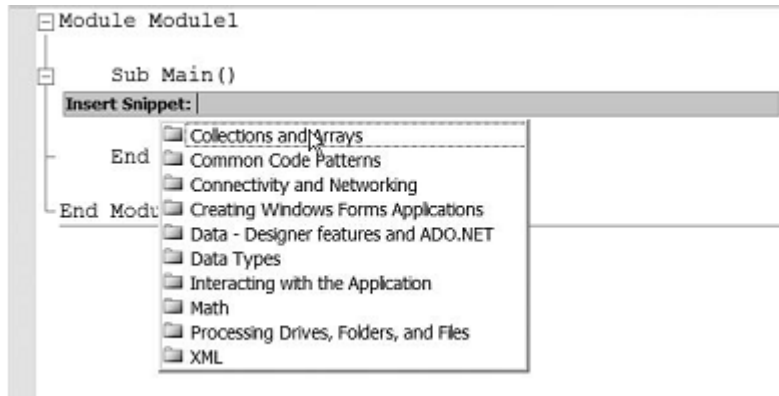


لعرض و استعمال Snippets اتبع الخطوات التالية :

- 1- قم بإدراج Snippets و ذلك بضغطه يمين على الفأرة في محرر الكود أو من قائمة تحرير : Edit - IntelliSense - Insert Snippets

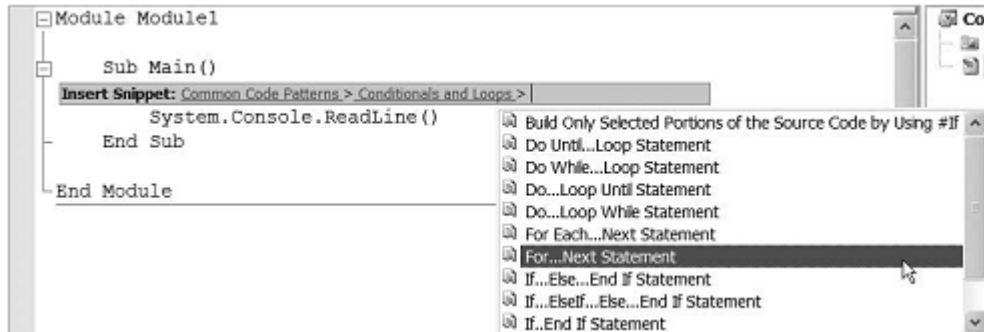


- 2- انقر الخيار Insert Snippet يظهر ما يلي :



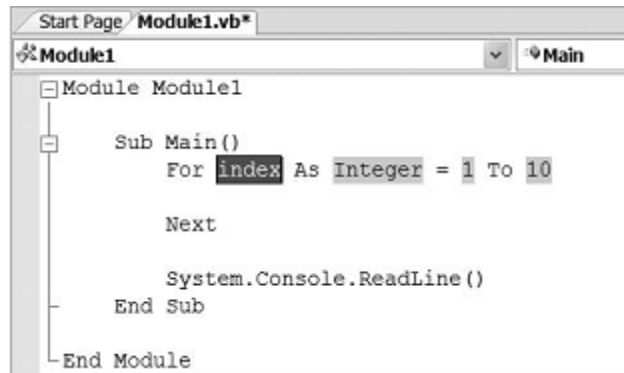
3- اختر الأمر Common Code Patterns

4- ثم اختر For ...Next Statment



تحصل على الشكل التالي الذي يوضح لنا الصيغة العامة لحلقة For ...Next التي تساعدك لكتابة الكود

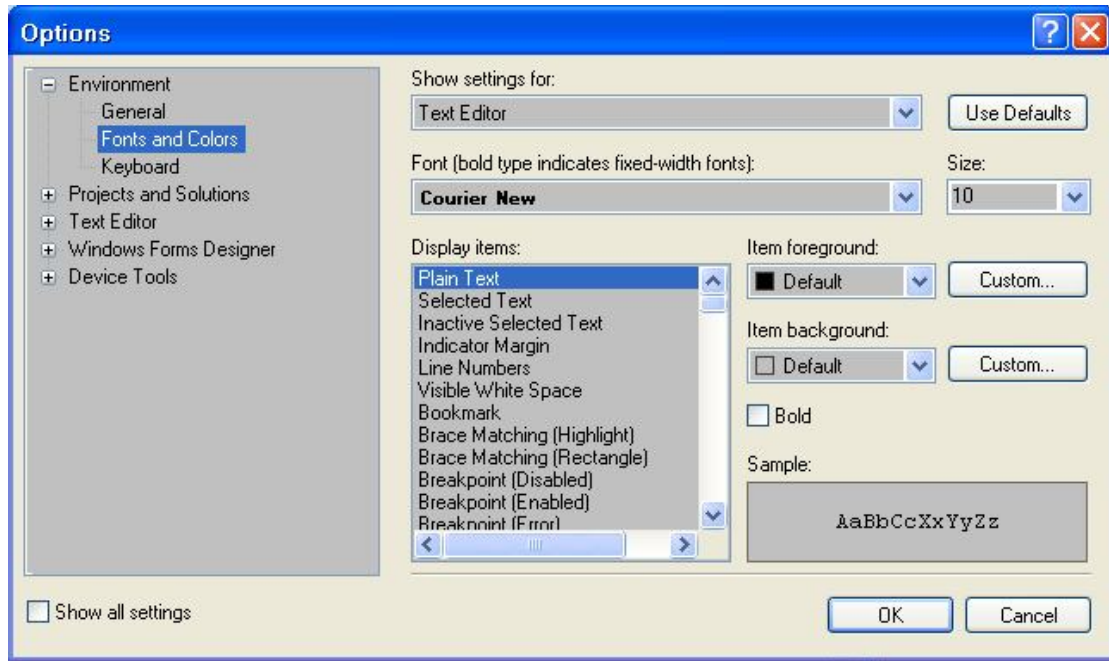
وهكذا يمنحك Snippet المرنة و معرفة الكثير من الصيغ التي تساعدك لكتابة الأكواد



تنسيق محرر الكود :

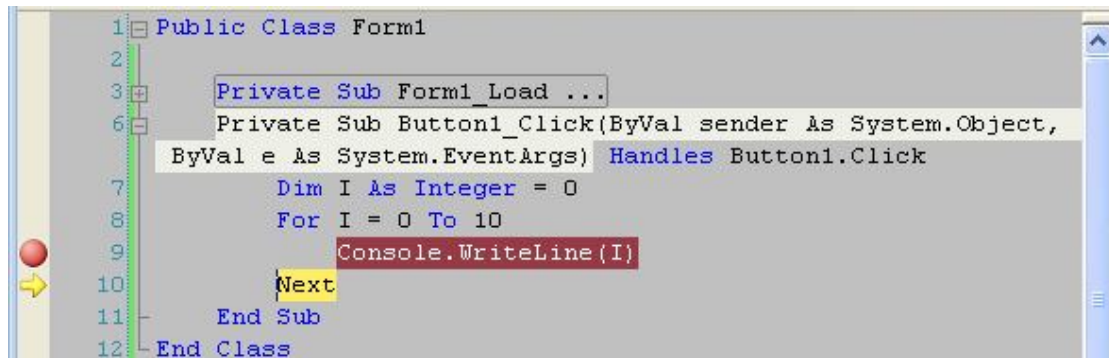
يمكنك تغيير لون خلفية و خط محرر الكود و ذلك بإتباع الخطوات التالية :

Tools - Option - Environment - Fonts and Colors

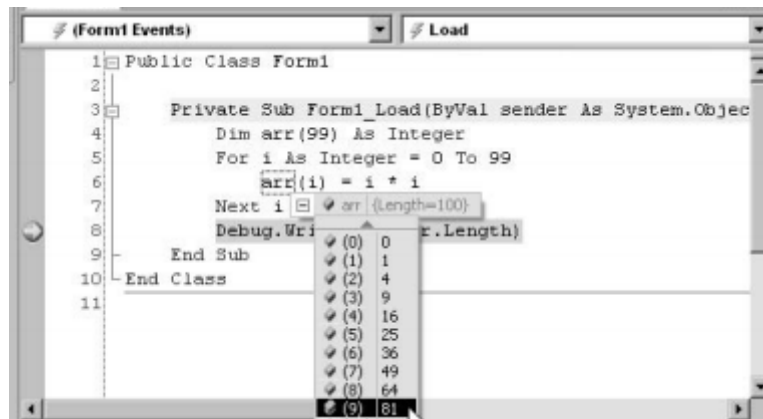


نقاط التوقف Break Points

تسمح لك بوقف البرنامج مؤقتاً لاستقراء و تجريب قيم المتغيرات أو الكائنات
 لإضافة نقطة التوقف انقر على هامش الإجراء أو حدد السطر ثم اضغط F9



اللون البني أي أنه تم وضع نقطة توقف في هذا السطر
 اللون الأصفر يدل على السطر الذي يتم تنفيذه . مرر المؤشر فوق المتغير I ستظهر قيمته.



سادسا : الشريط القياسي Standard Bar



الشكل 1/15 الشريط القياسي

و يمكن تقسيم الشريط إلى أجزاء حسب الوظائف

أدوات خاصة بالتعامل مع المشاريع :

نبدأ من اليسار إلى اليمين :



- إنشاء مشروع جديد (New Project)
- فتح مشروع موجود (Open Project)
- إضافة عنصر جديد (Add New Item)
- حفظ النافذة الحالية (Save Form)
- حفظ الكل (Save All)

أدوات خاصة بالتحريير :

نبدأ من اليسار إلى اليمين



قص ، نسخ ، لصق ، البحث السريع

أدوات خاصة بجعل التعليقات تعليقا أم لا

- الزر الأيسر : يجعل سطر التعليقات الحالي الذي يوجد به المؤشر تعليقا أو الأسطر المحددة



- الزر الأيمن: يلغي العملية

أدوات التراجع و الإعادة

أدوات التنقل بين النماذج و محرر الكود



أدوات التشغيل و تنقيح الأخطاء

نبدأ من اليسار إلى اليمين :

- تشغيل Run
- أسباب الكل
- إيقاف Stop

- الأزار الثلاثة الباقية خاصة بنقل نقطة التوقف



أدوات خاصة بعرض النوافذ الأساسية

- عرض مستكشف الحل Solution Explorer
- عرض نافذة الخصائص Properties
- عرض مستكشف المشروع Object Browser
- عرض مربع الأدوات ToolBox
- عرض نافذة قائمة الأخطاء Error List
- عرض نافذة Immediate

سابقا : شريط العنوان Title Bar

و يوجد أعلى الشاشة و يظهر كما في الشكل :



الشكل 1/16 شريط العنوان



WindowsApplication1 - Microsoft Visual Studio

Windows Application عنوان المشروع الافتراضي في حالة عدم تغيير اسم المشروع

القوائم Menus

قائمة ملف File

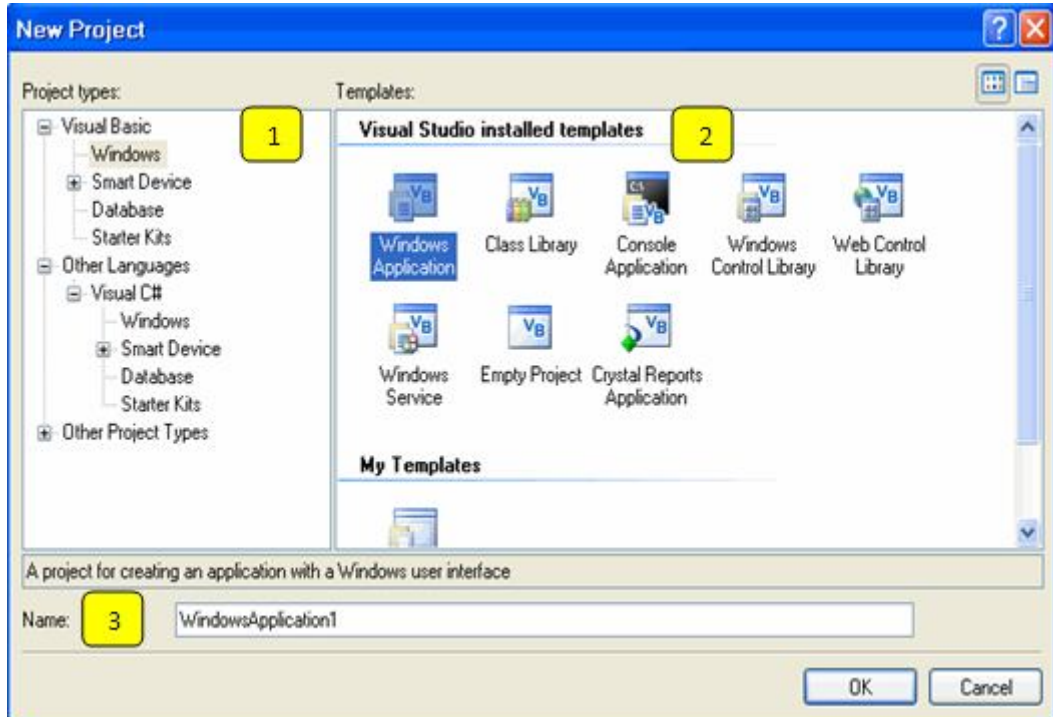
و تشمل على الأوامر التقليدية المعروفة من فتح وحفظ و غلق و إضافة المشاريع أو الملفات و الوصول إلى وظائف الطباعة



الشكل 1/17 محتويات قائمة ملف

إنشاء مشروع جديد : New Project

من هنا يمكنك البدء في إنشاء مشروع جديد، انقر على هذا الخيار لتطالعك النافذة التالية



و تحتوي على 3 أجزاء

1- Project Type : أنواع المشاريع مصنفة في مجموعات ليسهل عليك اختيار المشاريع

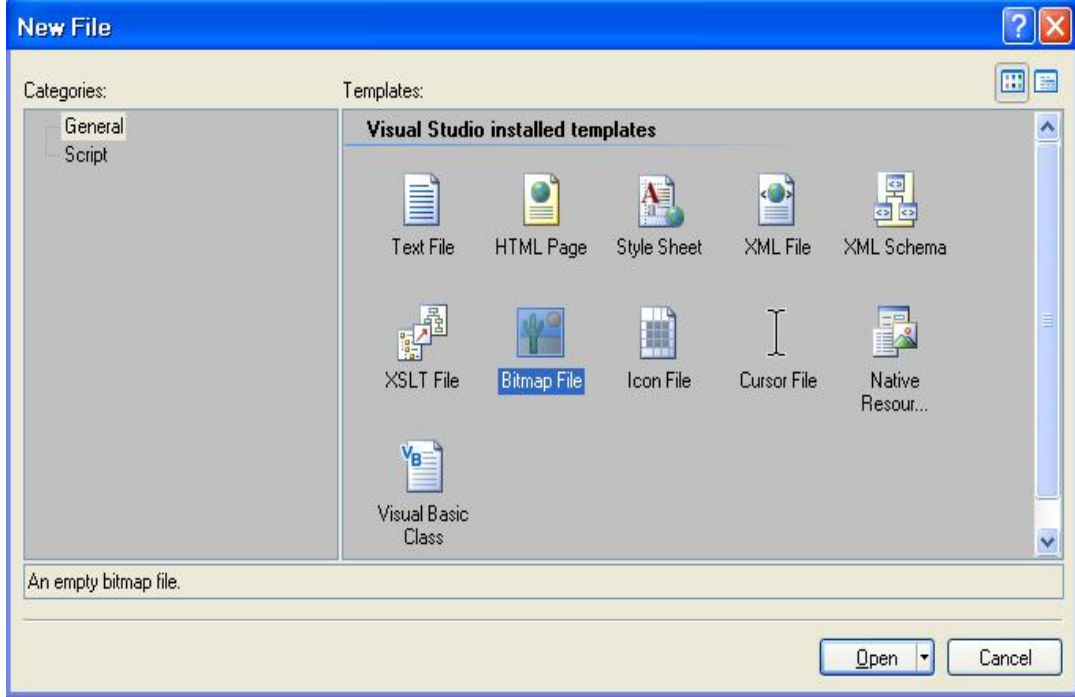
2- Templates : و يحتوي على أنواع التطبيقات

3- Name : اختيار اسم المشروع

حدد المشروع الذي تريده ثم التطبيق المناسب و اكتب اسما للمشروع و انقر Ok

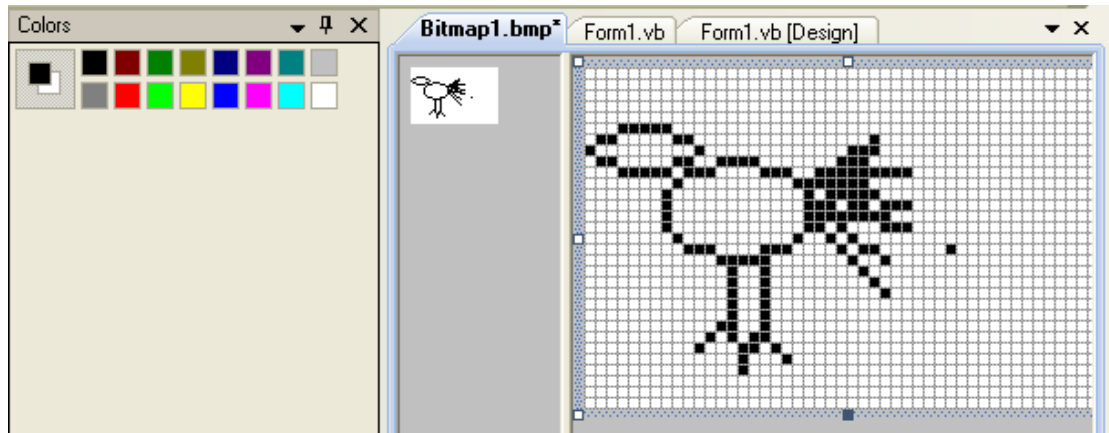
ملف جديد New File

لإضافة ملف جديد انقر هنا ستظهر أمامك النافذة التالية



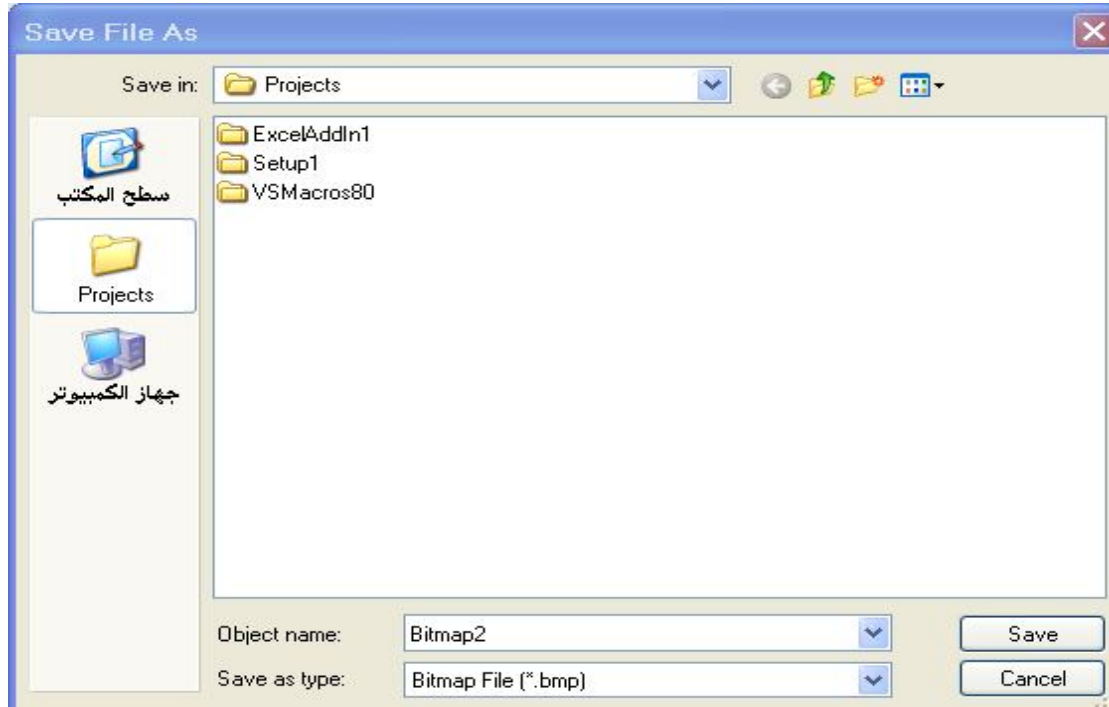
الشكل 1/18 : مربع حوار ملف جديد

فمثلا نختار ملف BitmapFile لرسم صورة ثم اضغط Open تظهر أمامك الشاشة التالية

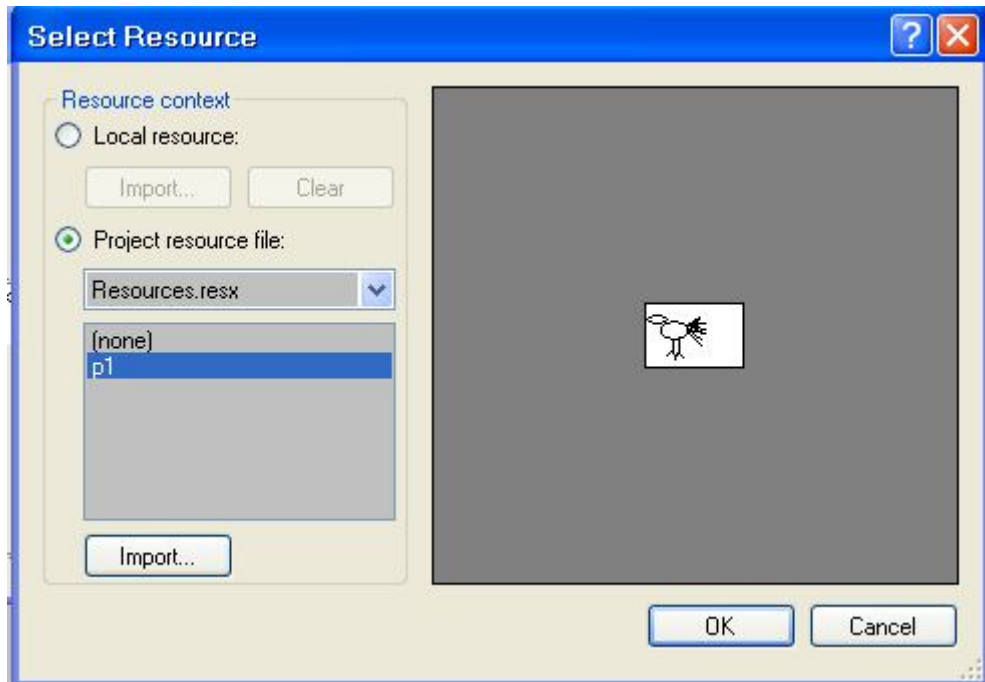


الشكل 1/19 برنامج الرسم

قم برسم الصورة كما في الشكل ثم اضغط زر إنهاء X للنافذة يظهر أمامك مربع حوار يخبرك ما إذا أردت حفظ الأيقونة أم لا اختر نعم يظهر مربع حوار حفظ باسم كما يلي



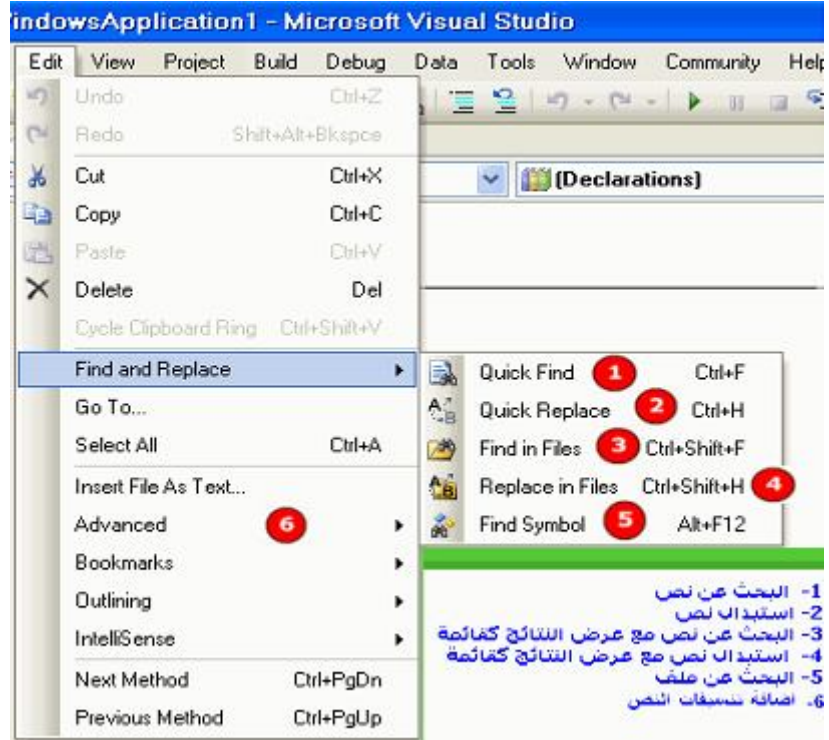
اختر مكانا لحفظ ملف الصورة ثم اضغط Save.
و لاستغلال هذا الملف (الصورة) قم بإدراج Picture Box اضغط السهم الصغير
الأبيض ثم اختر Choose Image تظهر النافذة التالية



اختر Import و ابحث عن المكان الذي قمت بحفظ الصورة فيه ثم اضغط فتح سيتم
إدراج هذه الصورة

قائمة التحرير Edit Menu

وتحتوي على الأوامر المعتادة، الخاصة بتحرير النصوص والعناصر، مثل النسخ والقص واللصق والحذف والبحث والاستبدال.. كما أنها تحتوي على أوامر متقدمة و تكون متاحة عند تحرير الكود



الشكل 1/20 قائمة العرض و محتوياتها

القائمة الفرعية "متطور" Advanced:

وهي لا تظهر إلا حينما تكون في محرر الكود.. وهي تحتوي على الأوامر التالية:

الأمر اجعل الحروف كبيرة Uppercase

الأمر اجعل الحروف صغيرة lowercase

الأمر "أظهر المسافات الفارغة" View White Space:

يؤدي هذا الأمر إلى استبدال المسافات برموز خاصة.. إن المسافات مهمة جدا في تنظيم سطور الكود ومقاطع الأوامر، حتى يسهل قراءة الكود فيما بعد.

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)
        Me.Text = "S"
    End Sub
End Class
```

الأمر "التفاف الأسطر" Word Wrap :

يقوم بالرجوع إلى السطر تلقائياً مما يسهل على المستخدم قراءة التعليمات دون استعمال

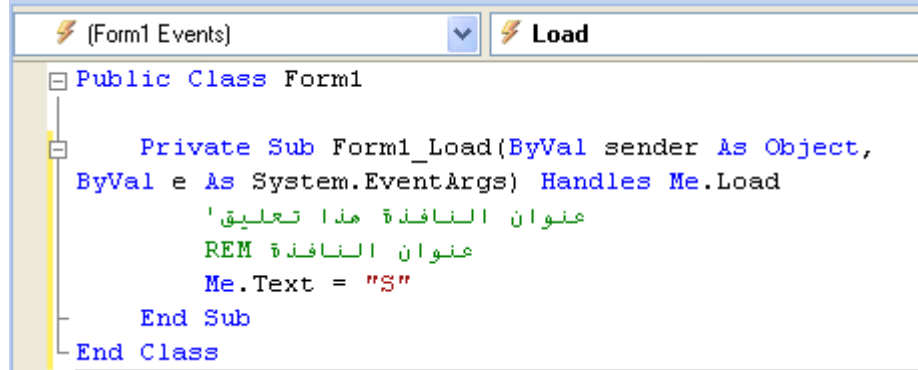
المنزلاقات و لا كسر الأسطر باستعمال العلامة (_) Underscore

الأمر "اعتبر / لا تعتبر التحديد تعليقا" Comment Selection / Uncomment Selection :

التعليقات جمل توضع لشرح الكود يتجاهلها المترجم عند تنفيذ البرنامج و لجعل جملة أو

سطرا تعليقا نسبقه بعلامة (') أو كلمة "Rem:" أو النقر على الأمر Comment

Selection



```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As Object,
        ByVal e As System.EventArgs) Handles Me.Load
        عنوان النافذة هذا تعليق
        REM عنوان النافذة
        Me.Text = "S"
    End Sub
End Class
```

أحيانا ما نحتاج إلى تهيمش بعض الأسطر من الكود حتى لا ينفذها البرنامج، ولكن دون

حذفها، وهذا بتحديدتها ثم استخدام أمر "اعتبر التحديد تعليقا" و حتى يمكن استعادتها عند

الحاجة استخدم الأمر لا تعتبر التحديد تعليقا".

القائمة الفرعية المساعدة الذكية "IntelliSense Submenu"

تمنحك المرونة و السهولة في كتابة التعليمات و تريحك من حفظ الخصائص و الوسائل و كذا

التقليل من الأخطاء الإملائية و هذه الإمكانيات من أبداع ما سيصادفك في بيئة التطوير، والتي

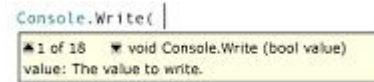
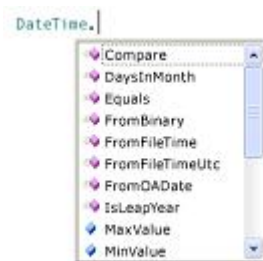
ستشعرك بالارتياح، و توفر لك الكثير من الوقت و الجهد.

ملاحظة : لقد تعرفنا عليها في الدرس الخاص بنافذة محرر الكود

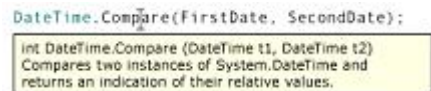
وهي تحتوي على خمسة أوامر :

الأمر "إظهار قائمة الأعضاء" List Members :

الأمر "معلومات عن المعاملات" Parameter Info :



الأمر "معلومات عاجلة" Quick Info :

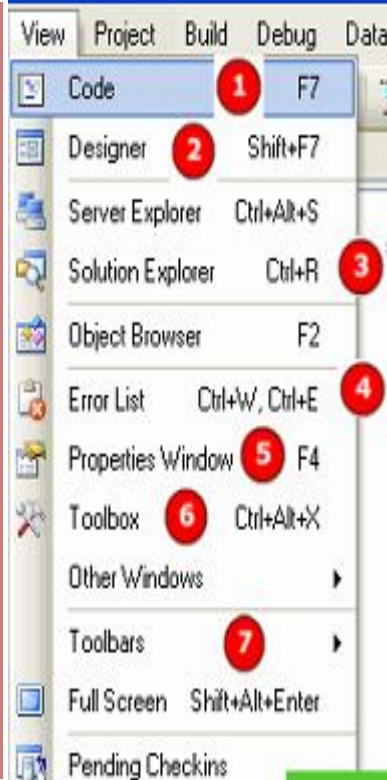


الأمر "أكمل الكلمة" Complete Word :

القائمة عرض View Menu

وتحتوى على الأوامر التي تعرض أيا من أشرطة الأدوات ونوافذ بيئة التطوير و تشمل الأوامر التالية

الوظيفة	الأمر
عرض نافذة الكود	Code
عرض نافذة النموذج	designer
عرض نافذة خاصة لقواعد البيانات	Server Explorer
عرض مستكشف الحل	Solution Explorer
مستعرض المشروع وعناصره	Object Browser
قائمة الأخطاء	Error List
نافذة الخصائص	Properties Windows
صندوق الأدوات	ToolBox
نوافذ أخرى	Other Windows
لعرض أشرطة الأدوات	ToolBar
ملء الشاشة	Full Screen

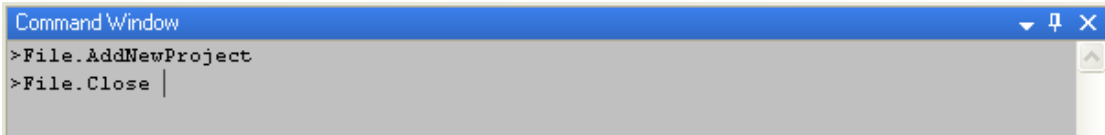


الشكل 1/22 قائمة العرض و محتوياتها

يمنحك Visual basic.Net الكثير من النوافذ الأخرى ستجدها في قائمة View نوافذ أخرى Other Windows . منها :

نافذة الأوامر **The Command Window**:

تقوم بتنفيذ الأوامر مباشرة دون استعمال القوائم . فبمجرد كتابة الكود و الضغط على Enter يتم تنفيذ الأمر

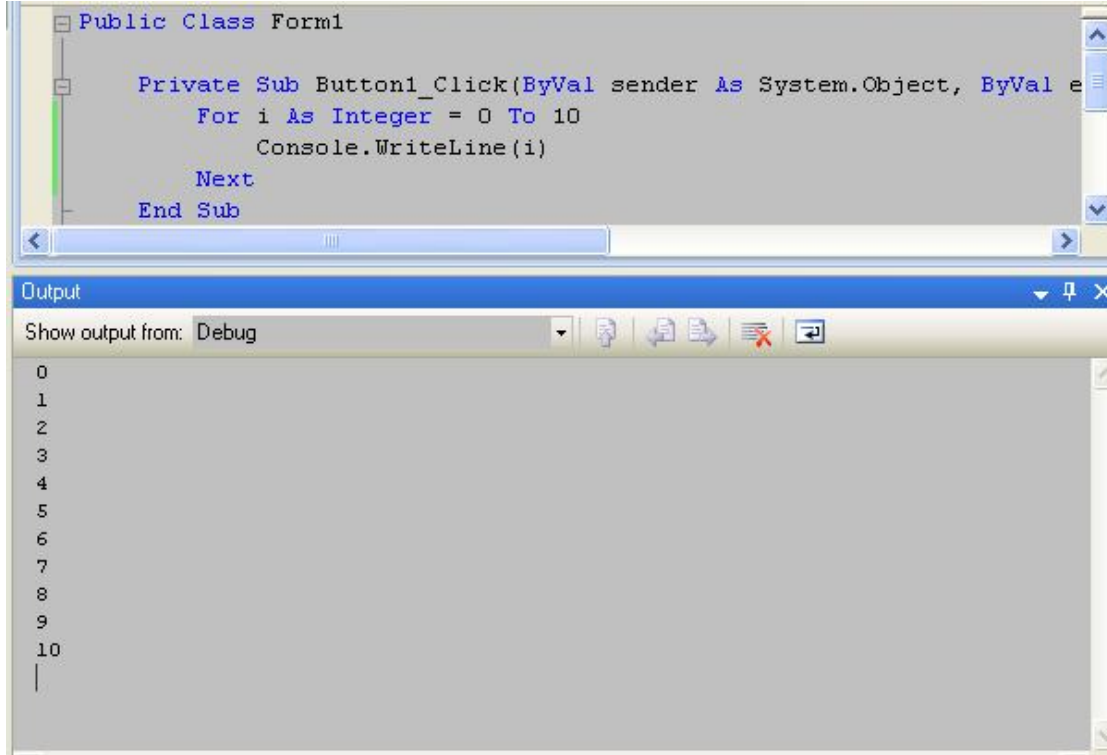


الشكل 1/23 النافذة الفورية

نافذة المخرجات **Output**

الفصل الأول : بيئة التطوير المتكاملة IDE

تقوم هذه النافذة بعرض النتائج التجريبية السريعة لاختبار التعليمات المكتوبة في الكود عن طريق الأمر Console.WriteLine انظر المثال :



```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        For i As Integer = 0 To 10
            Console.WriteLine(i)
        Next
    End Sub
End Class
```

Output

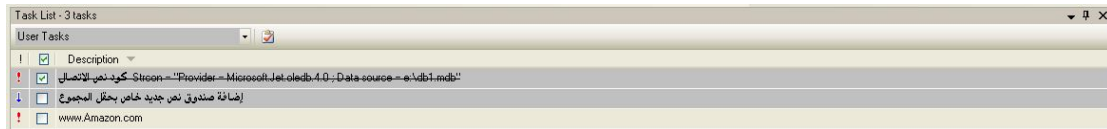
Show output from: Debug

0
1
2
3
4
5
6
7
8
9
10
|

الشكل 1/24 نافذة المخرجات

نافذة المهام Task List

تمكنتك هذه النافذة من تسجيل المهام المنتظرة و التي تود القيام بها فيما بعد أو كتابة مواقع الكترونية هامة أو أي شيء يهمك تريد تذكره في قائمة المهام . و لكن كيف نكتب هذه المهام اضغط السطر الفارغ في بداية قائمة المهام، واكتب ما تريد من تعليقات



الشكل 1/25 نافذة المهام

كلما تم القيام بمهمة تحدها ليتم تسطيرتها

قائمة الأخطاء Error List

تعرض كل الأخطاء الموجودة في البرنامج في قائمة . كلما ضغط على عنصر يتم الانتقال مباشرة إلى موضع الخطأ لتصحيحه.

جرب هذا المثال و لاحظ بمجرد الضغط على مفتاح ENTER يظهر خطأ متعرجاً أزرقاً يخبر بوجود خطأ

الفصل الأول : بيئة التطوير المتكاملة IDE

```
Button1 Click
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As EventArgs)
        Loadtxt()
        var1 = 5
    End Sub
End Class
```

و عند تنفيذ البرنامج تظهر رسالة بهذا النحو



تخبرك فيما كنت ترغب في تنفيذ آخر حالة ناجحة من البرنامج

طبعاً ستختار No لأن في كل مرة تريد تنفيذ البرنامج تظهر هذه النافذة في حالة اخترت No

ستظهر نافذة قائمة الأخطاء كما في الشكل

Error List					
2 Errors 0 Warnings 0 Messages					
	Description	File	Line	Column	Project
1	Name 'Loadtxt' is not declared.	Form1.vb	4	9	WindowsApplication1
2	Name 'var1' is not declared.	Form1.vb	5	9	WindowsApplication1

الشكل 1/26 قائمة الأخطاء

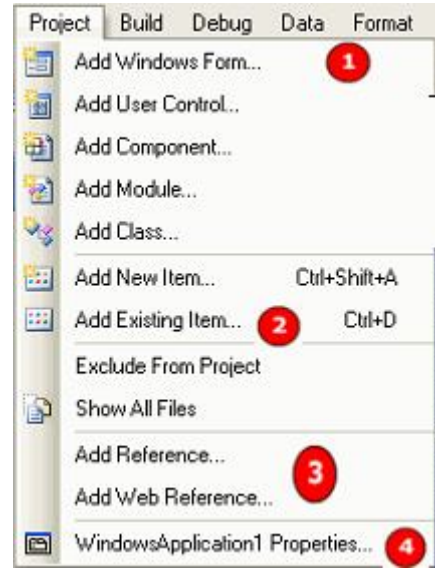
ويمكنك أن تنقر مرتين على العنصر من القائمة لتنتقل مباشرة إلى السطر الذي يحتوي الخطأ في الكود و تصلحه.

قائمة مشروع Project Menu

تمكنك هذه القائمة من إضافة كل ما تحتاجه لمشروعك الحالي من : نماذج و أدوات تحكم و قوالب و فئات و عناصر جديدة و المستعملة و المراجع

الفصل الأول : بيئة التطوير المتكاملة IDE

إدراج نافذة النموذج	Add Windows Form
إدراج أدوات خاصة	Add User Control
إدراج وحدة نمطية	Add Module
إدراج فئة	Add Class
إدراج عناصر أخرى	Add New Item
إدراج عناصر موجودة سابقا	Add Existing Item
إدراج مراجع	Add Reference
إدراج نافذة خصائص النموذج	Windows Application Properties

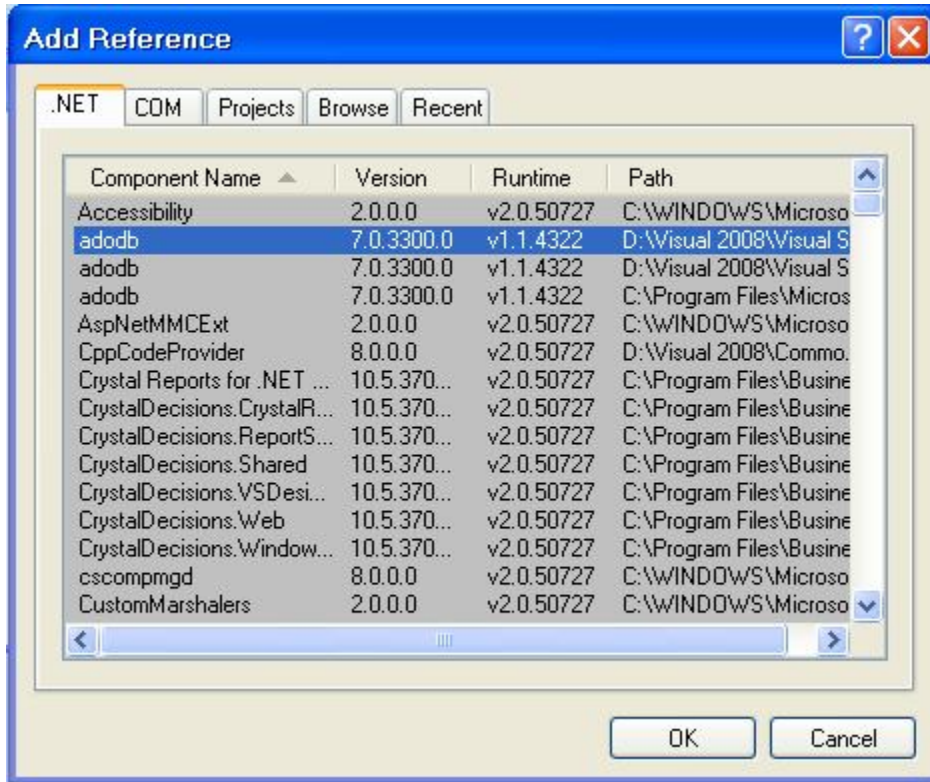


الشكل 1/27 قائمة المشروع

تدريب نموذجي : إضافة مراجع Reference

1. من قائمة عرض Project اختر الأمر Add Reference

تظهر نافذة إضافة المراجع

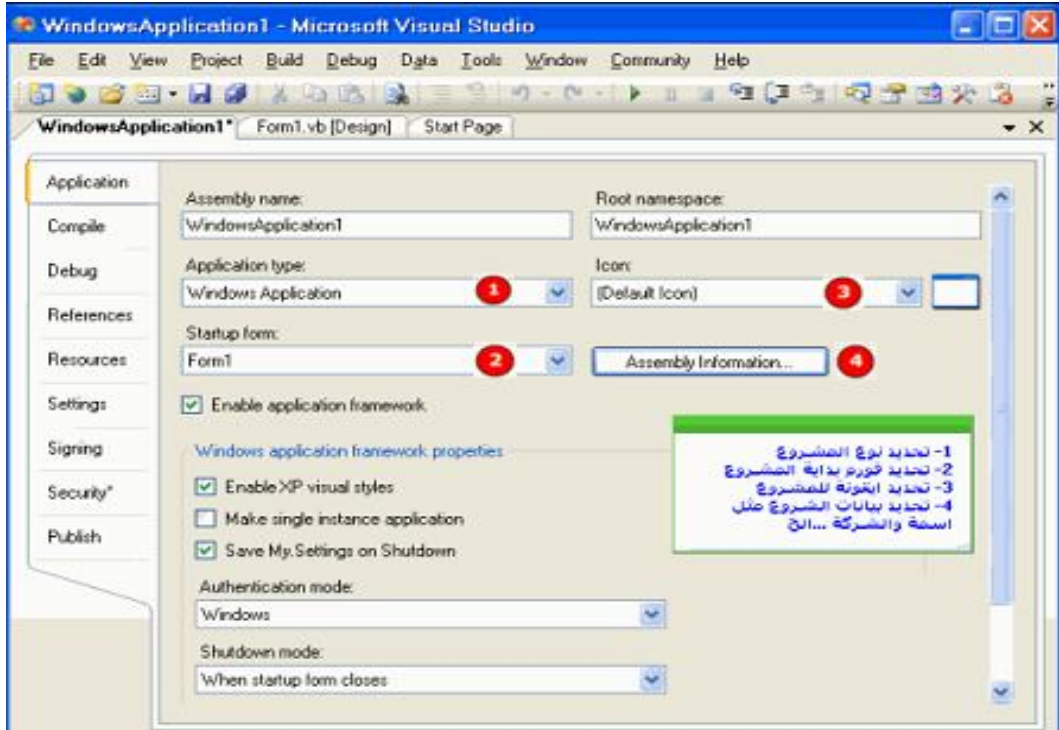


الشكل 1/28 نافذة إضافة مراجع

2. حدد الخيار كما في الشكل - وهو مرجع خاص بقواعد البيانات -

3. اضغط Ok سيتم إضافة المرجع و يمكنك استعماله في مشروعك

عند اختيار الخيار الأخير خصائص المشروع تظهر أمامنا نافذة مدير المشروع



الشكل 1/29 نافذة مدير المشروع

أولاً : تبويب Application

وعند ضغط زر Assembly Information تظهر نافذة معلومات عامة عن المشروع

Title	عنوان المشروع
Description	وصف بسيط عن المشروع
Company	البلد
Product	المنتج
Copyright	نسخة مرخصة لـ
Version	الإصدار

Assembly Information

Title: WindowsApplication1

Description:

Company:

Product: WindowsApplication1

Copyright: Copyright © 2012

Trademark:

Assembly Version: 1 0 0 0

File Version: 1 0 0 0

GUID: 498b655a-cae1-411d-9836-040c979e3ade

Neutral Language: (None)

Make assembly COM-Visible

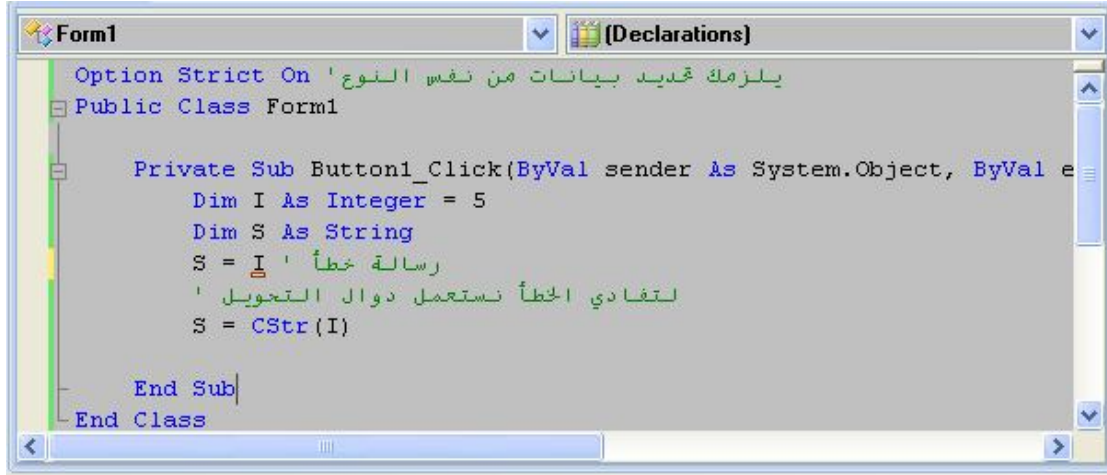
OK Cancel

الشكل 1/30 : نافذة معلومات المشروع

و يمكنك الاستعلام عن معلومات المشروع كما يلي :

الفصل الأول : بيئة التطوير المتكاملة IDE

ويفضل جعل هذا الخيار مفعل لجعل المعالجة أسرع وتحويل أنواع البيانات باستخدام الدوال أفضل



```
Option Strict On ' يلزمك تحديد بيانات من نفس النوع '
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim I As Integer = 5
        Dim S As String
        S = I ' رسالة خطأ
        ' لتفادي الخطأ نستعمل دوال التحويل
        S = CStr(I)
    End Sub
End Class
```

Option Compare -3

يحتوي على خيارين Binary / Text

1. **Binary**: يقوم بمعالجة النصوص اعتماداً على البيشري و هو المفضل و المفعل تلقائياً

2. **Text**: يقوم بمعالجة النصوص بتحسس حالة الأحرف

Option infer -4

يستعمل عند تعريف المتغيرات بدون تحديد نوعها . يقوم المعالج بتقريب نوع المتغير إلى

أقرب نوع ممكن . وهذا عندما يكون الخيار **Option Explicit** مفعل

فمثلاً : **Dim X = 10**

سيعتبر المتغير X من نوع عدد صحيح Integer وهذا لأن 10 عدد صحيح

أما عندما يكون الخيار غير مفعل OFF سيعتبر كل المتغيرات التي لم تحدد نوعية بياناتها من

نوع Object و هذا ما يجعل الجهاز أبطأ و يحجز مساحة أكبر

ثالثاً : تيويب Resources : ومنة يتم إنشاء ملفات تستخدم كمصادر

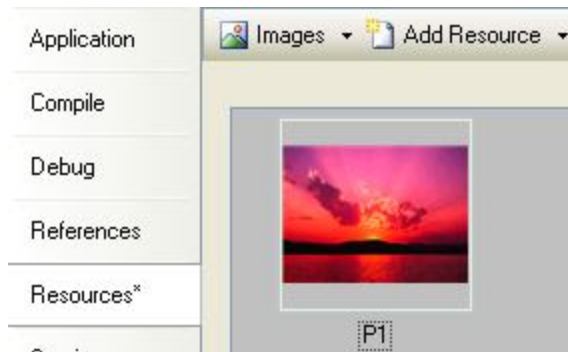


ومن ثم تستطيع التعامل مع الملفات التي أضفتها إلى مشروعك كمصادر بكتابة الكود المناسب
تدريب نموذجي :

1. أضف أداة زر Button و صندوق الصور Picture Box إلى النموذج Form
2. من قائمة Project اختر الأمر **Windows Application Properties**
تظهر نافذة خصائص المشروع
3. اختر التبويب Resource يظهر مربع حوار المصادر
4. انقر السهم الصغير Add Resources ثم اختر الأمر Add New String
تظهر نافذة على شكل جدول اكتب ما يلي

	Name	Value	Comment
▶	Title	إدراج صورة	
*			

5. انقر مرة أخرى على السهم الصغير Add Resources ثم اختر Add Existing File
يظهر مربع الحوار لإضافة مصادر موجودة و نختار صورة
6. تظهر أمامك الصورة و اختر لها اسما مثلا P1



6. قم بإغلاق نافذة المصادر و اختر Ok للحفظ

الفصل الأول : بيئة التطوير المتكاملة IDE

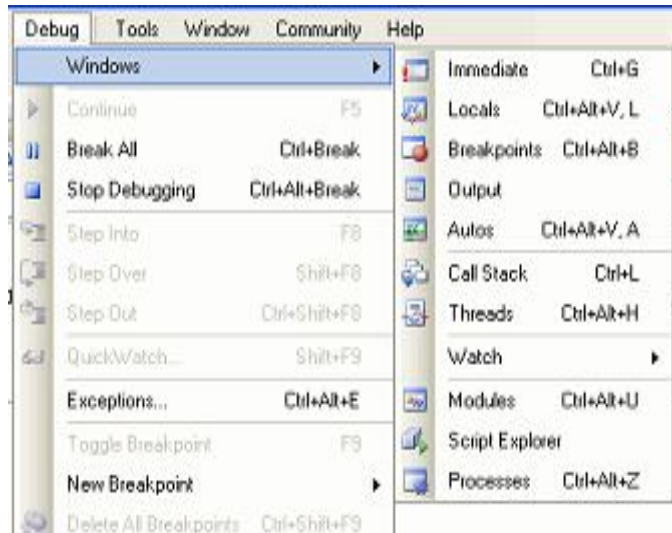
Executable File له الامتداد (.EXE)، حيث يمكنك تشغيله من خارج اللغة كأى برنامج عادي و في أي جهاز .



الشكل 1/31 قائمة البناء و محتوياتها

قائمة تنقيح الأخطاء Debug

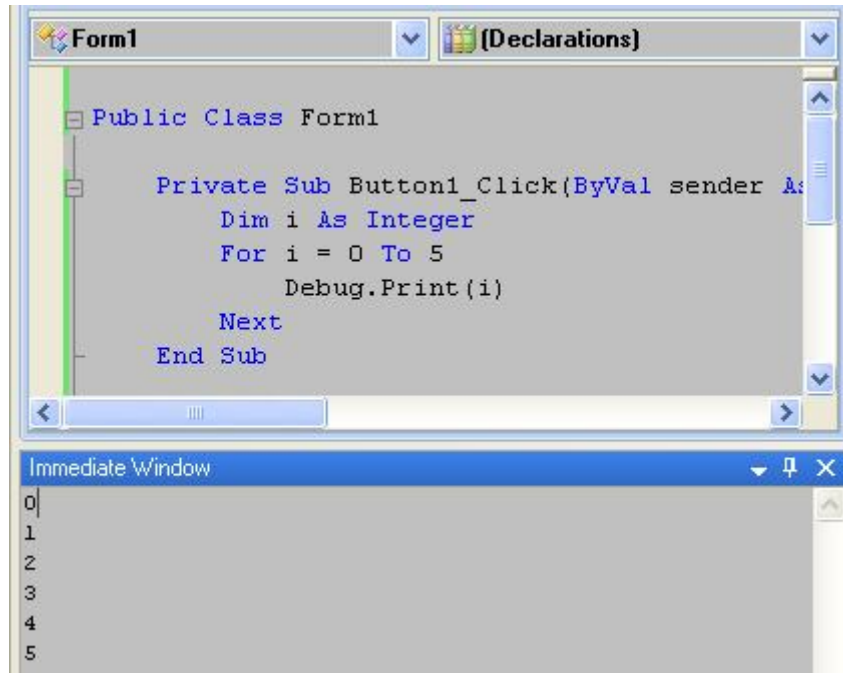
و تحتوي على أوامر خاصة بتتبع الأخطاء و تنقيحها بالإضافة إلى أوامر التشغيل و إيقاف تشغيل البرنامج و تشمل على القوائم التالية في حالة تنفيذ البرنامج :



1- **Windows** : و تحتوي على عدة قوائم فرعية أخرى أهمها :

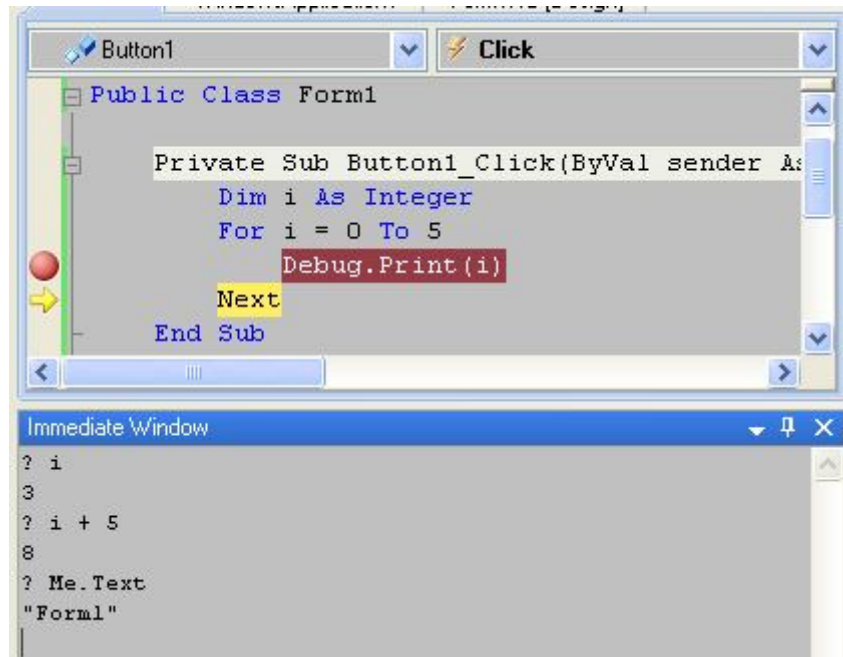
Immediate : وتستخدم لعرض النتائج الفورية للمتغيرات و الكائنات الموجودة في

مشروعك الحالي (وهذا في حالة التنفيذ) باستعمال الأمر `Debug.Print`



الشكل 1/32 النافذة الفورية

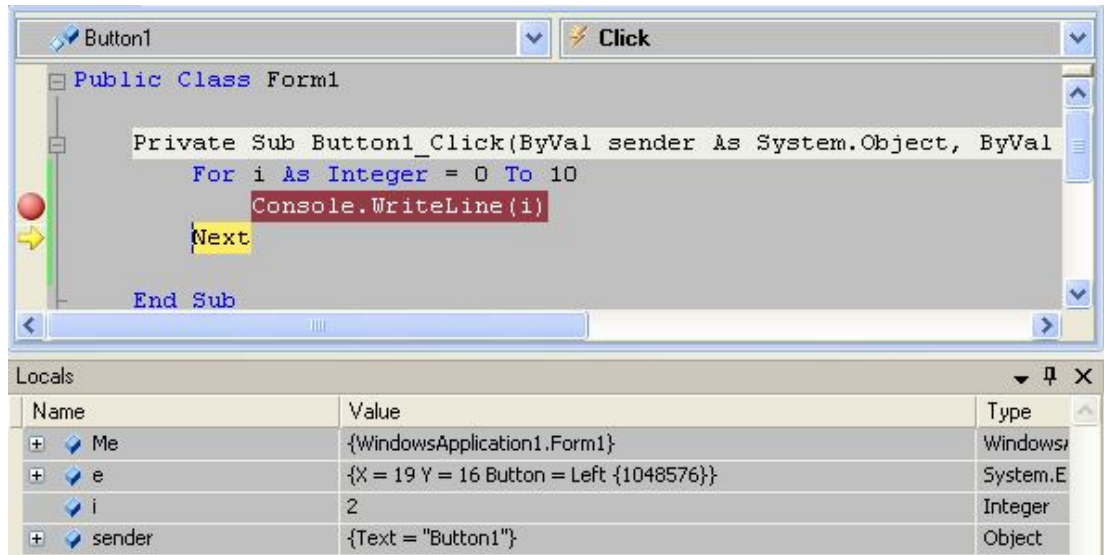
كما يمكنك الاستعلام عن قيمة المتغيرات و الكائنات في حالة التوقف كما في المثال



النافذة Locals

تعرض جميع قيم المتغيرات الخاصة بالإجراء الحالي وقت التنفيذ في حالة التوقف المؤقت وكلما نفذت خطوة من البرنامج (بضغط F8) سيتم تحديث قيم ما تغير من هذه المتغيرات

الفصل الأول : بيئة التطوير المتكاملة IDE

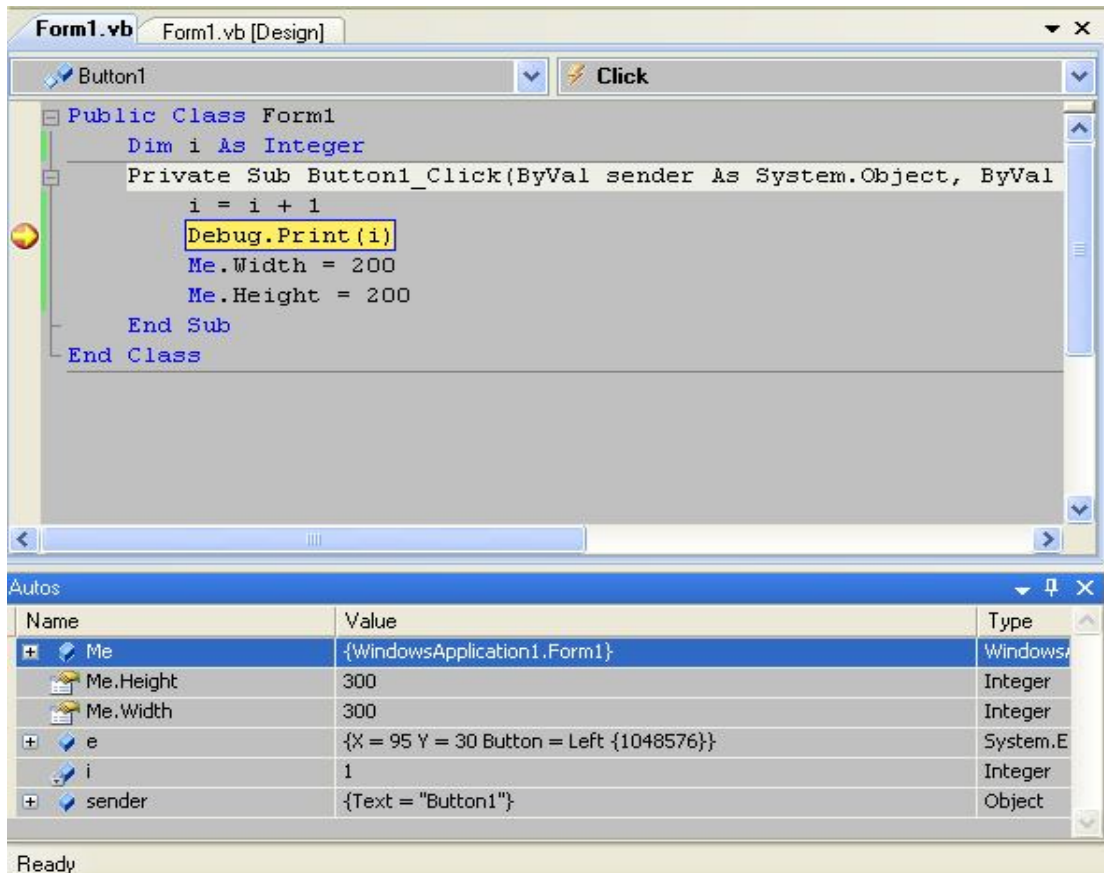


الشكل 1/33 النافذة المحلية

بإمكانك تعديل قيم المتغيرات بالنقر المزدوج على قيمة المتغير

نافذة Autos

تعرض قيم المتغيرات العامة و المحلية من سطر الإجراء الحالي و 3 سطور قبل و بعد هذا السطر و هذا وقت التنفيذ المؤقت



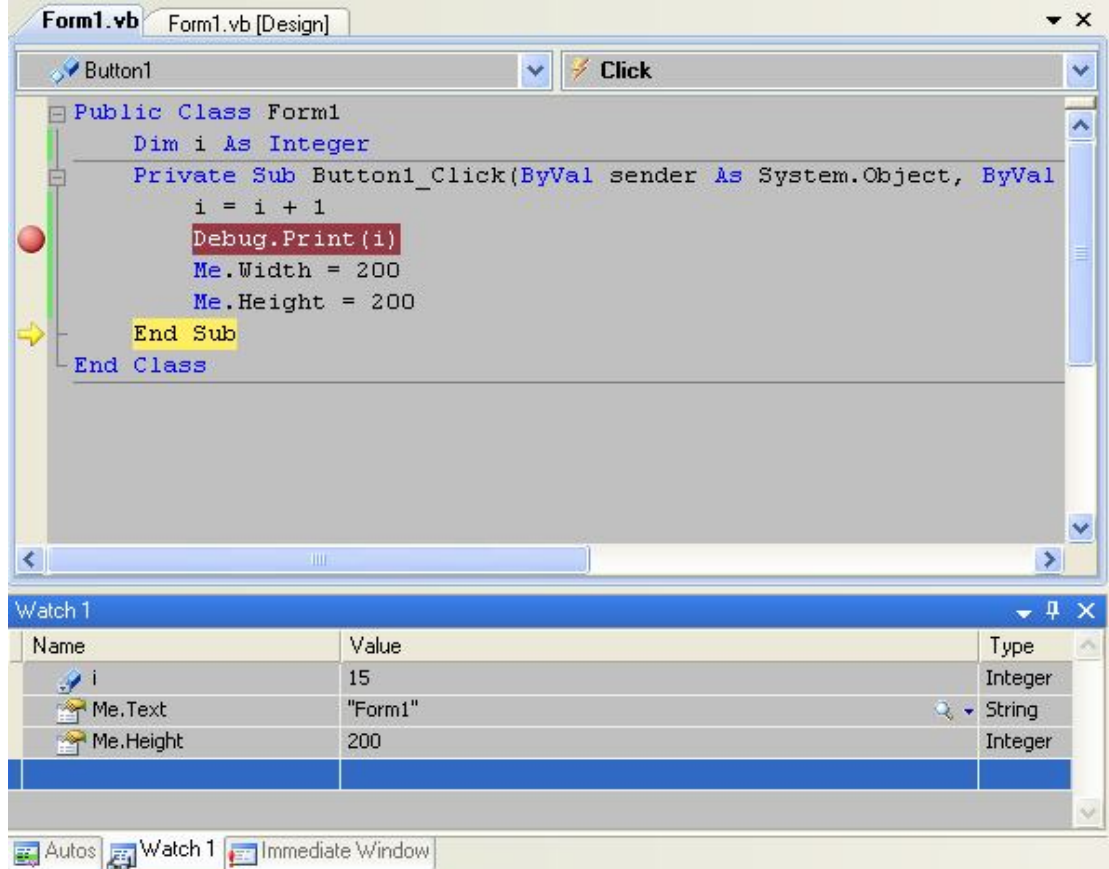
الشكل 1/34 النافذة Autos

الفصل الأول : بيئة التطوير المتكاملة IDE

بإمكانك تعديل قيم المتغيرات على سبيل الاختبار بالنقر المزدوج على قيمة المتغير و متابعة النتائج بالضغط F8

نافذة المراقبة Watch

تمكنك من معرفة قيمة المتغيرات أو الخصائص و ذلك بكتابة اسم المتغير أو الكائن ثم الخاصية فبمجرد الضغط على Enter تظهر قيمها في قسم Value



الشكل 1/35 نافذة المراقبة و المشاهدة

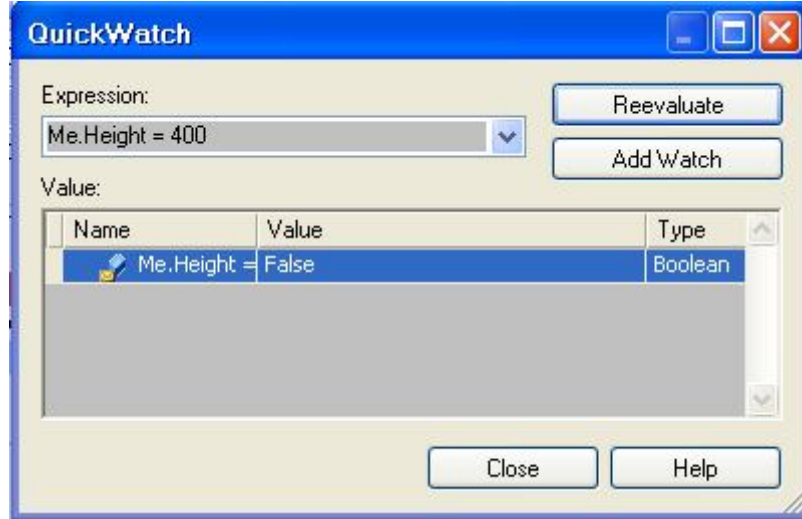
أوامر أخرى فرعية من قائمة Debug

- **Start Debugging** ▶ : يقوم بتشغيل أو تنفيذ البرنامج و في حالة وجود نقطة التوقف (وعند ظهور السهم الأصفر عند نقطة التوقف) يتغير دوره إلى متابعة (Continue)
- **Break All** ■■ : إيقاف الكل
- **Stop Debugging** ■ : إيقاف تنفيذ البرنامج
- **Step In to** ■■■ : انتقل إلى : يقوم بالقفز إلى السطر المناسب حسب التعليمات البرمجية

• نافذة المراقبة الفورية Quick Watch

الفصل الأول : بيئة التطوير المتكاملة IDE

تمكّنك من تتبّع قيم تعبيرات معقّدة بدلا من الاكتفاء بقيمة متغيّر منفرد وتعطيك نتائج منطقية True أو False ولإستخدام هذه النافذة، حدّد أيّ متغيّر أو تعبير (أيّ عمليّة تحتوى على متغيّر أو أكثر)، واضغط زرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط "مراقبة فوريّة" QuickWatch.. ستظهر لك نافذة "المراقبة الفوريّة" Quick Watch الموضّحة في الصورة:



الشكل 1/36 نافذة المراقبة و المشاهدة الفورية

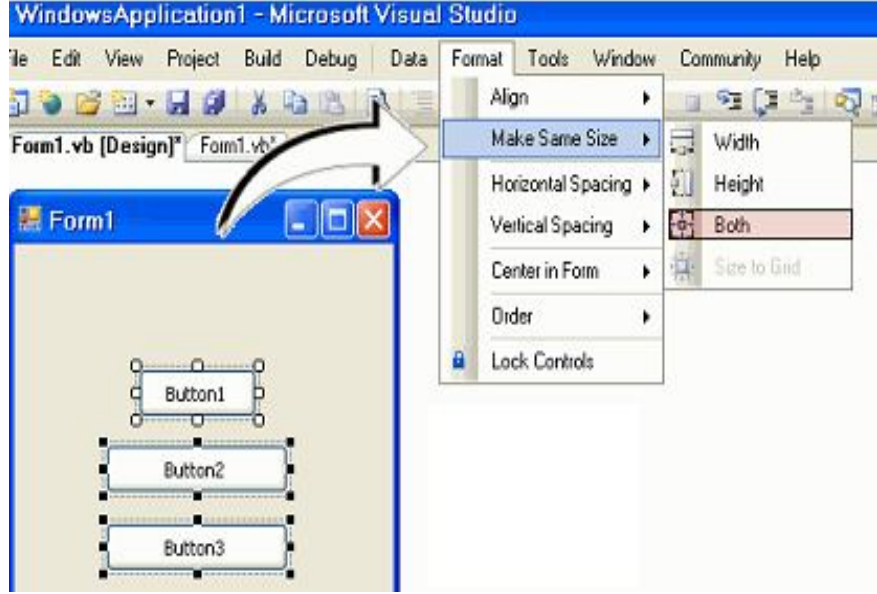
قائمة البيانات Data Menu

وتحتوي على الأوامر التي تستخدم في المشاريع التي تتعامل مع قواعد البيانات

الأمر	ينفذ
Show Data Sources	يعرض نافذة خاصة لإضافة قاعدة البيانات فوق نافذة مستكشف الحل
Add New Data Source	إضافة قاعدة بيانات جديدة
Add New Data Source	إضافة قاعدة بيانات جديدة

قائمة تنسيق Format Menu

تحتوي على الأوامر الخاصة بتنسيق الأدوات و ظهورها على النموذج.



الشكل 1/37 قائمة التنسيق و محتوياتها

Align : تحتوي على قوائم فرعية وتقوم بمحاذاة الأدوات

▪ **Lfets** : محاذاة الأدوات المحددة إلى اليسار

▪ **Centres** : توسط الأدوات المحددة على النموذج

▪ **Rights** : محاذاة الأدوات المحددة إلى اليمين

MaskeSameSize : لتحجيم الأدوات

▪ **Width** : تجعل الأدوات المحددة لها نفس العرض

▪ **Heigth** : تجعل الأدوات المحددة لها نفس الارتفاع

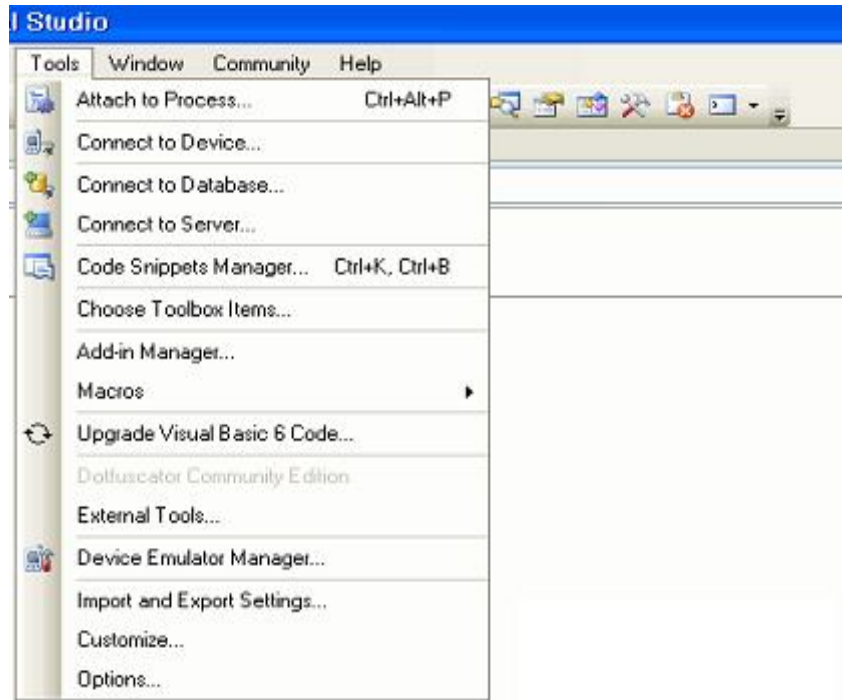
▪ **Both** : تجعل الأدوات المحددة لها نفس العرض و الارتفاع

Lock Controls

تقوم بوقف و عدم تحريك أو تغيير في الأدوات

قائمة الأدوات Tolls Menu

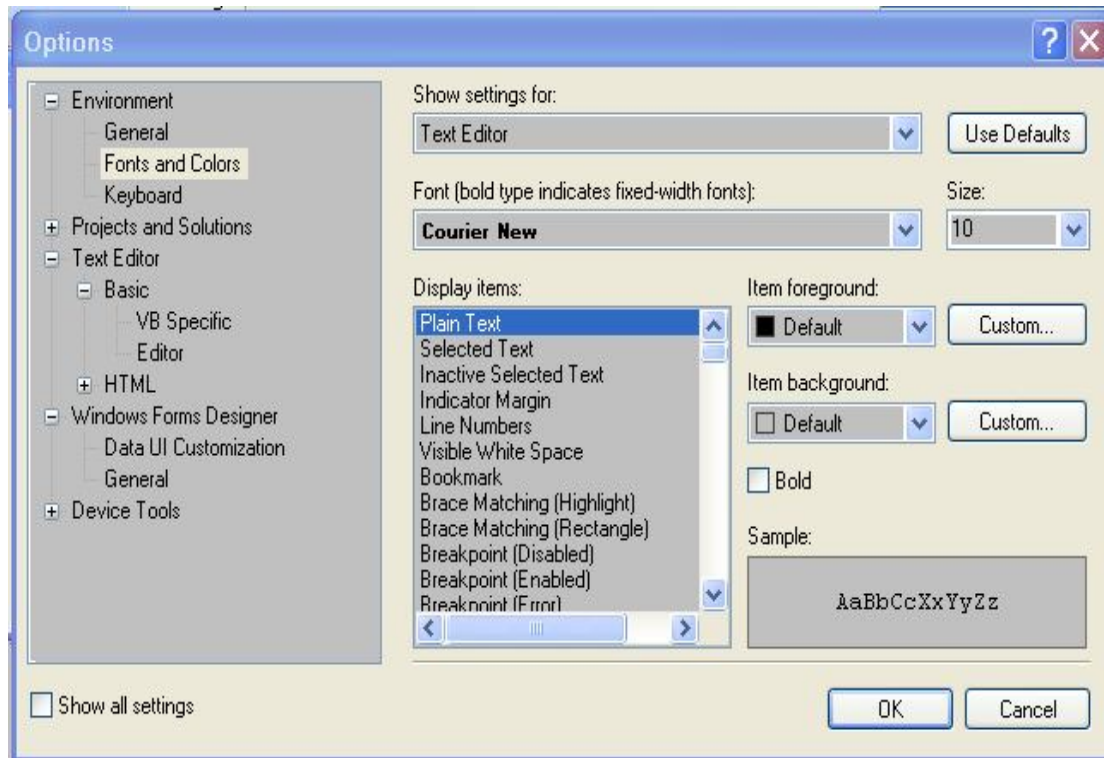
وتحتوي على العديد من الأدوات، التي تستخدم معظمها في لغة C++. ومن ضمن الأدوات المتاحة، الوحدات النمطية Macros، وهي تمنحك نفس القدرة المتاحة في تطبيقات Office، على أن تبسط وظائف بيئة التطوير، بوضع المهام كثيرة التكرار في وحدة نمطية تنفذها ألياً



الشكل 1/38 قائمة الأدوات و محتوياتها

ما يهمنا كثيرا الأمر الأخير Options خيارات

يقوم بفتح نافذة تستطيع من خلالها تحديد الخيارات الخاصة ببيئة التطوير



قائمة النوافذ Windows Menu

تحتوي هذه القائمة على الأوامر التي تتحكم في طبيعة ظهور نوافذ Visual Basic



الشكل 1/39 نافذة النوافذ و محتوياتها

Auto Hide : إخفاء تلقائي مثل نافذة Toll Box تظهر عند مرور الفأرة فوقها

Hide : يقوم هذا الخيار بإغلاق النافذة وتستطيع إظهارها مرة ثانية من القائمة view أو من خلال الضغط على مفتاح الاختصار لإظهارها

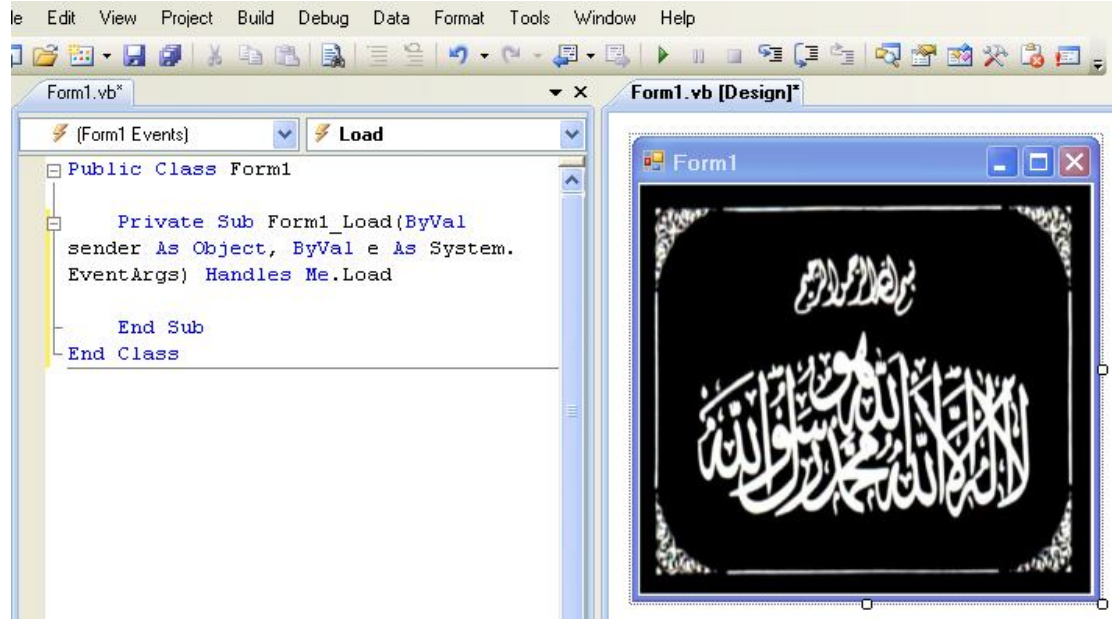
Auto Hide All يقوم بإخفاء تلقائي لكل النوافذ الثانوية الظاهرة

New Horizontal Tab Group : يقوم بعرض محرر الكود و نافذة التصميم معا متجاورين أفقيا انظر الصورة



New Vertical Tab Group

يقوم بعرض محرر الكود و نافذة التصميم متجاورين رأسياً كما في الصورة



Close All Documents: يقوم هذا الأمر بإغلاق كافة النوافذ المفتوحة

Reset Window Layout : يقوم بإعادة بيئة التطوير للوضع الافتراضي

Windows: يمكنك من اختيار النافذة التي تريدها أن تكون النشطة

قائمة المساعدة Help Menu

و تحتوي على الأوامر المختلفة للوصول إلى صفحات المساعدة Help

How Do : ماذا تريد أن تفعل

Search: الوصول إلى التعليمات عن طريق كلمة للبحث عنها

Contents : الوصول إلى التعليمات عن اختيار من المحتوى

الشجري

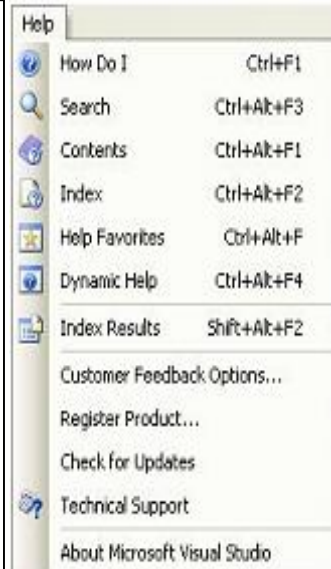
Index : الوصول إلى التعليمات عن طريق اختيار عنصر من

الفهرسة

Dynamic help : المساعدة الفعالة وهي تظهر لك

المواضيع المتعلقة بالأداة المحددة و تكون إلى جانب نافذة

الخصائص.



الشكل 1/40 نافذة المساعدة

الفصل الثاني : لغة البرمجة

📌 الخصائص و الطرق و الأحداث

📌 مفاهيم أساسية

📌 تعدد الأشكال

📌 قليل من الخصائص المشتركة

📌 قليل من الوسائل المشتركة

📌 قليل من الأحداث المشتركة

📌 مجيبات الأحداث

📌 إنشاء أدوات وقت التنفيذ

📌 إنشاء أداة جديدة بأحداثها

📌 المتغيرات و الثوابت

📌 تعريفها

📌 تسمية المتغيرات

📌 التصريح و الإعلان عن المتغيرات

📌 تخصيص قيم للمتغيرات

📌 أنواع المتغيرات

📌 المتغيرات الرقمية

📌 العمليات الحسابية على المتغيرات الرقمية

📌 تحديد نطاق الأعداد

📌 الدقة العشرية

📌 تنسيق الأرقام

📌 ما لا نهاية و القيم الشاذة

📌 المتغيرات الحرفية

📌 أهم وسائل المتغيرات الحرفية

📌 المتغيرات النصية

📌 تشبيك النصوص

📌 مقارنة النصوص

📌 المتغيرات المنطقية

📌 المتغيرات الزمنية

📌 المتغيرات الكائنية

📌 المتغيرات الخاصة بالمصفوفات

📌 اختبار أنواع المتغيرات

📌 التحويل بين الأنواع المختلفة للمتغيرات

📌 خيار التحويل الدقيق

الفصل الثاني : لغة البرمجة

- خيار التعريف الصريح
- مجال المتغير
- عمر المتغير
- دراسة الصيغة العامة للمتغيرات
- الثوابت
- المعاملات
- السجلات
- المرقامات
- مجالات الأسماء
- الوحدات النمطية
- جمل التحم في المسار
- If... Then الجملة الشرطية
- Select Case جملة اختيار الحالة
- For ... Next حلقة من .. إلى
- Do While حلقة نفذ بينما
- Do Until حلقة نفذ حتى
- While حلقة بينما
- For Each جملة مهما كان
- Go To اذهب إلى
- Choose اختر قيمة
- IIf الدالة
- الإجراءات و الدوال
- لماذا نعرف الإجراءات
- التصريح عن الإجراء و الدالة
- المعاملات
- تمرير المعاملات بالقيمة
- تمرير المعاملات بالمرجع
- مصفوفة المعاملات
- المعامل مصفوفة
- المعاملات الاختيارية
- الإجراء كمعامل
- أنواع أخرى للقيم المعادة
- إعادة تعريف الإجراء
- تطبيقات

شرح الكود :

قم بتنفيذ البرنامج بالضغط F5 و انقر (حدث) على الزر سيعرض رسالة (استجابة) مكتوبا فيها تمت الاستجابة و يتغير لون النموذج إلى لون أزرق

تعدد الأشكال Polymorphism

يمنحك Vb.Net هذه السمة الرائعة تمكنك من التعرف على خصائص و طرق و أحداث الكائنات بمجرد التعرف على واحدة منها مع أي أداة أو كائن فهي مشتركة بينها في الوظيفة مثلا : بمجرد التعرف على وظيفة الخاصية Text لأي كائن و التي تقوم بكتابة عنوان أو نص تريد عرضه فهي تقوم بنفس الدور في باقي الكائنات الأخرى . و هكذا بالنسبة للطرق و الأحداث . شي جميل حقا !

قليل من الخصائص المشتركة A Few Common Properties:

خاصية "الاسم" Name:

وهذه الخاصية تظهر في أعلى نافذة الخصائص دون مراعاة للترتيب الأبجدي، وذلك لأهميتها. وهي عبارة عن الاسم البرمجيّ للأداة، وهو الذي تكتبه في الكود.. مثال:
لو أسميت مربع نص Text1 نكتب في الكود مثلا :

```
Text1.Text = "Visual Basic.Net"
```

ملاحظة

نظراً لأن اسم الأداة سيظهر كثيراً في البرنامج، يجب عليك اختيار اسم يكون موحياً، حتى يمكنك فهم البرنامج عند قراءته فيما بعد.. والطريقة المتفق عليها في التسمية هي تكوين الاسم من مقطعين:

1- **المقطع الأول:** بادئة تدل على نوع الكائن، عبارة عن حرفين أو ثلاثة لاختصار اسمه..

وسنعطيك بعض الاقتراحات في الجدول التالي:

Button	Bt
CheckBox	Chk
ComboBox	Cmb
Form	Frm
Label	Lb
ListBox	Lst
Menu	Mnu
RadioButton	Rdo
RichTextBox	Rt

TextBox

Txt

2- المقطع الثاني: كلمة تدل على وظيفة الأداة، حتى يمكن فهمها بمجرد قراءة الاسم.. أمثلة:

▪ لو لديك مربع نص يكتب فيه المستخدم اسمه، فسمّه TxtUserName أو TxtName لو أردت الاختصار.

▪ ولو لديك لافتة تعرض مجموع عددين، فسمّها LbSum.

▪ ولو لديك قائمة مركبة تعرض أسماء الدول العربية، فسمّها CmbArabCountries.

▪ ولو لديك زر مكتوب عليه "موافق"، فسمّه BtOk.

خصائص الظهور :

Visible	ظهور أو عدم ظهور الأداة
Enabled	تمكين أو عدم تمكين الأداة
RightToLeft	الكتابة بالعربي أو بالأجنبي
TextAlign	محاذاة النص
Text	النص أو العنوان
Font	خاصية الخط و هي من الخصائص المجمعة تشمل على مجموعة كبيرة من الخصائص

خصائص الموقع و الحجم

الخاصية	نوع البيانات	الوظيفة
Bounds	المستطيل	تتطلب تحديد أربعة أبعاد أعلى اليسار و العرض و الارتفاع Button1.Bounds = New Rectangle(10, 20, 75, 25)
ClientRectangle	المستطيل	عرض المستطيل المماثل لمساحة النموذج Dim r As Rectangle = Me.ClientRectangle r.Inflate(-100, -100) Me.Bounds = Me.RectangleToScreen(r)
ClientSize	الحجم	يعرض و يثبت حجم الأداة مثل حجم النموذج Button1.Size = Me.ClientSize
DisplayRectangle	المستطيل	يعرض المستطيل المماثل للنطاق و المساحة الفعلية للأداة
Location	نقطة	موضع الأداة في الركن العلوي اليساري Button.Location = New Point(10, 15)
Size	النقطة	عرض و قراءة ارتفاع و عرض الأداة Button1.Size = New Size(150, 50)

الفصل الثاني : لغة البرمجة

رقم صحيح	، Top ، Left Height ، Width
Left : بعد الأداة عن الحافة اليسرى Top : بعد الأداة عن الحافة العلوية Width : عرض الأداة Height : ارتفاع الأداة	
<pre>Dim buttonCancel As New Button() buttonCancel.Top = 10 buttonCancel.Left = Button1.Right + 5 buttonCancel.Width = Button1.Width buttonCancel.Height = Button1.Height buttonCancel.Text = "&Cancel" Me.CancelButton = buttonCancel Me.Controls.Add(buttonCancel)</pre>	

خاصية تثبيت الهامش Anchor

تثبيت واحد أو أكثر من هوامش الأداة بالنسبة لحافة النموذج المقابلة لها، بحيث يظل هذا الهامش ثابتاً، مهما غير المستخدم من مساحة النموذج، حيث تتغير أبعاد الأداة تلقائياً، للمحافظة على هذا الهامش

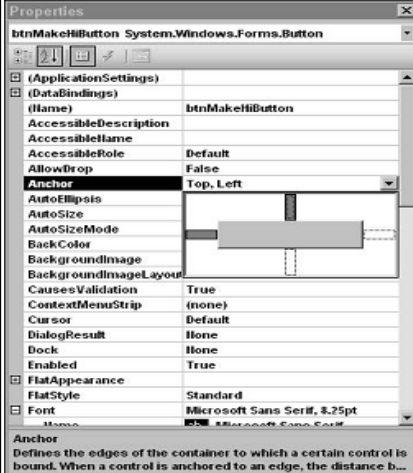
تدريب :

ضف أداة النص Text Box

غير خاصية Anchor بتحديد كل الجهات

ليتم تغيير شكل الأداة تبعاً لحجم النموذج كما في الشكل

شغل البرنامج ولاحظ تغيير حجم أداة النص وفق حجم النموذج



خاصية انطباق الحافة Dock

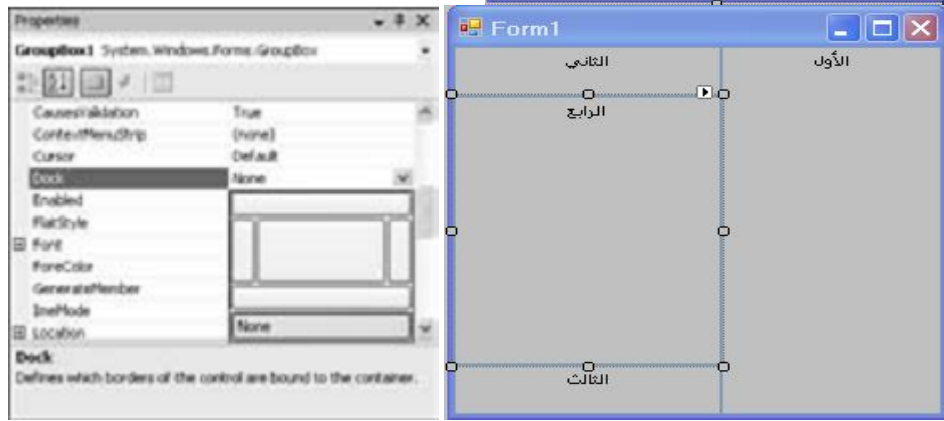
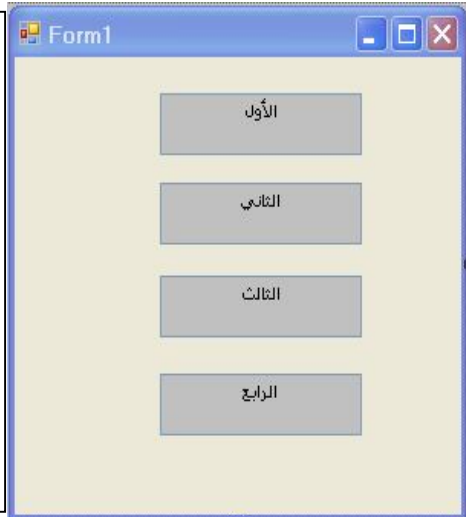
تمكّنك من جعل أيّ من حوافّ الأداة منطبقة على حافة النموذج المقابلة

تدريب :

الفصل الثاني : لغة البرمجة

قم بإضافة 4 صناديق نصوص Text Box غير خاصية تعدد الأسطر MultiLine إلى True لكل الصناديق

- ▶ غير خاصية Dock لكل أداة حسب ما يلي :
- ▶ الأول يأخذ القيمة Right
- ▶ الثاني يأخذ القيمة Top
- ▶ الثالث يأخذ القيمة Bottom
- ▶ الرابع يأخذ القيمة Fill



و هذا الجدول يوضح لنا قيم الخاصية Dock ووظائف كل قيمة

الزر	عملها	الخاصية
السفلي	انطباق الأداة من الحافة السفلية مع النموذج (النص الثالث من الشكل السابق)	Bottom
الأوسط	انطباق الأداة كلياً على النموذج	Fill
الأيسر	انطباق الأداة من الركن الأيسر مع النموذج	Left
None	بلا	None
الأيمن	انطباق الأداة من الحافة اليمنى مع النموذج	Right
الأعلى	انطباق الأداة من الحافة العلوية من الأداة	Top

خصائص الألوان :

الخاصية	عملها
ForeColor	لون الخط TextBox1.BackColor = Color.Green
BackColor	لون الخلفية TextBox1.ForeColor = Color.Red TextBox1.BackColor = Color.FromArgb (255, 0, 0)


خصائص التركيز :

الخاصية	عملها

الفصل الثاني : لغة البرمجة

Textbox3.TabIndex = 3	رقم يمثل ترتيب انتقال Tab عند النقر عليها	TabIndex
Textbox4.TabStop = False	True تسمح للمستخدم الوصول إلى الأداة باستعمال Tab False : لا يسمح لك الوصول إلى الأداة إلا باستعمال الفأرة	TabStop

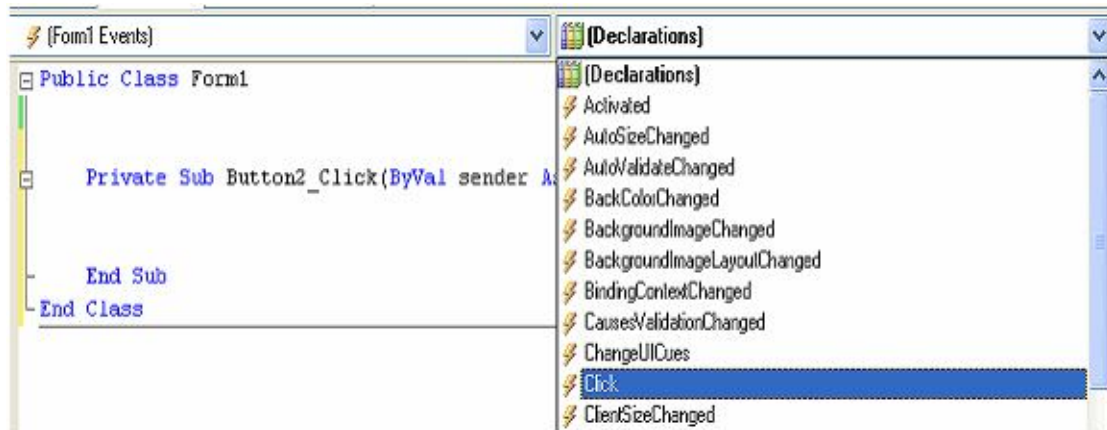
قليل من الوسائل المشتركة A Few Common Methods

الوسائل أو الطرق أو الوظائف هي إجراءات أو دوال تقوم بوظيفة محددة عند كتابة اسم الكائن متبوعة بنقطة تظهر قائمة أعضاء الكائن و نميز الوسائل بالرمز  اختر الوظيفة المناسبة ثم افتح قوس و ادخل المعاملات حسب التلميح الظاهر

الوسيلة	دورها	مثال
Focus	تنشيط و نقل التركيز إلى الأداة	Textbox1.Focus
Clear	مسح المحتوى	Textbox1.Clear
Show	عرض	Form2.Show
Hide	إخفاء	Form2.Hide
Scale	تستعمل لتكبير أو تصغير الأداة بنسبة	Textbox1.Scale(0.75)
Contains	تستعمل لمعرفة هل العنصر موجود أو لا	If Panel1.Contains(Button1) Then MsgBox("True") End If

قليل من الأحداث المشتركة A Few Common Events

عندما تتعرض الأداة إلى مؤثر خارجي كالنقر فتحدث الاستجابة مثلا عرض رسالة MsgBox هذه المؤثرات تعرف في لغة Visual Basic.Net بالأحداث ويمكن معرفة هذه الأحداث من خلال محرر الكود



الفصل الثاني : لغة البرمجة

عمله و حدوثه	الحدث
يحدث عند النقر على الأداة	Click
يحدث عند فقدان التنشيط	Deactivate
يحدث عند النقر المزدوج على الأداة	DoubleClick
يحدث عند انتقال التركيز إلى الأداة	GotFocus
يحدث طالما المفتاح مازال مضغوطة ويرجع بقيم تمثل حالة الضغط [Alt] ، [[Shift]] و [Ctrl] و يتعامل مع مفاتيح الوظائف F1,F2... و المفاتيح Alt، [[Ctrl ،Shift]]]	KeyDown
يحدث عند الضغط على مفتاح الحروف و الأرقام و [Enter] و [Tab] و [Esc]	KeyPress
يحدث عند فلت المفتاح	KeyUp
يحدث عند تنشيط و انتقال التركيز إلى الأداة	GotFocus
يحدث عند مغادرة تركيز الأداة	LostFocus
يحدث عند النقر بالماوس	MouseClicked
يحدث عند النقر المزدوج بالفأرة	MouseDown
يحدث طالما زر الفأرة ما زال مضغوطة	MouseDown
يحدث عند فقدان الأداة للتنشيط و غالبا ما يكتب هنا كود التحقق من الصحة عند المغادرة	MouseLeave
يحدث عند مرور الفأرة فوق الأداة	MouseMove
يحدث عند فلت زر الفأرة	MouseUp
يحدث عند استعمال عجلة الفأرة	MouseWheel
يحدث عند تحريك الأداة	Move
يحدث أثناء عملية التحقق من الصحة	Validating

أمثلة عن الأحداث :

VB.Net Code

```
Private Sub Form1_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Me.MouseDown
    Me.Text = "e = " & e.X & ",Y = " & e.Y
End Sub
```

هذه القطعة من الكود تعرض إحداثيات موقع مؤشر الفأرة على النموذج عند النقر بالماوس

VB.Net Code

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As _
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    MsgBox(Char.IsControl(e.KeyChar))
```

End Sub

هذه القطعة من الكود ترجع True إذا كان الحرف المضغوط حرف تحكم مثل Ctrl :

VB.Net Code

```
Private Sub TextBox1_KeyDown(ByVal sender As Object, _  
    ByVal e As System.Windows.Forms.KeyEventArgs) Handles _  
    TextBox1.KeyDown  
    If My.Computer.Keyboard.CtrlKeyDown Then  
        MsgBox("CTRL key down")  
    Else  
        MsgBox("CTRL key up")  
    End If  
End Sub
```

الكود السابق يبين لنا المفاتيح التي تخص الحدث KeyUp و KeyDown

VB.Net Code

```
Private Sub TextBox1_GotFocus(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles TextBox1.GotFocus  
    TextBox1.BackColor = Color.Yellow  
End Sub  
  
Private Sub TextBox1_LostFocus(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles TextBox1.LostFocus  
    TextBox1.BackColor = Color.Blue  
End Sub
```

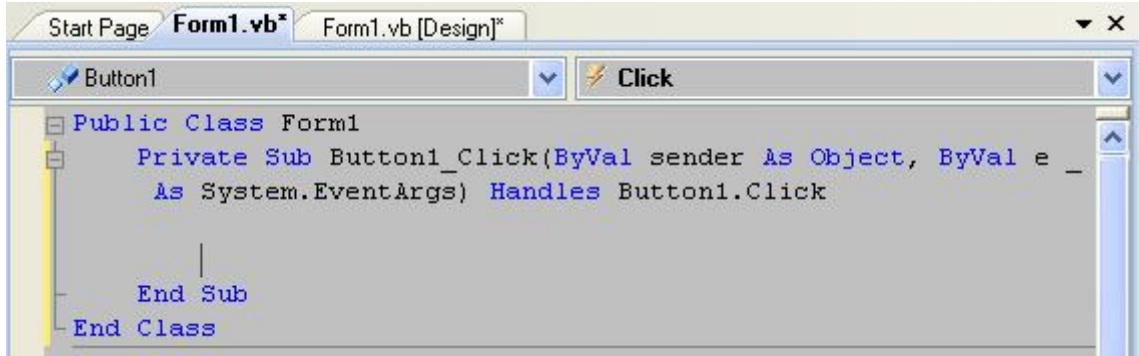
الكود السابق يقوم بتحديد لون خلفية النص عند انتقال التركيز إلى صندوق النص و عند مغادرة التركيز

VB.Net Code

```
Private Sub TextBox1_Validating(ByVal sender As Object, ByVal e As  
System.ComponentModel.CancelEventArgs) Handles TextBox1.Validating  
    If TextBox1.Text.Length > 4 Then  
        ErrorProvider1.SetError(TextBox1, "أرقام 4 أكثر من أدخلت لقد")  
    End If  
End Sub
```

الكود السابق يلزمنا بكتابة أربعة أحرف و إلا تظهر رسالة خطأ .
لو استبدلنا السطر المظلل بالعبارة : E.Cancle = True فلا يمكنك مغادرة صندوق النص و حتى لا يمكن الخروج و غلق النافذة .

مجيئات الأحداث "Event Handlers"



الكود السابق يتكون من حدث نقر الزر Button1_Click إن ما يجعل الإجراء مجيباً لحدث ما، هو الكلمة الأساسية "يجيب" Handles، والتي تأتي في نهاية تعريف الإجراء، متبوعاً باسم الحدث الذي سيستجيب للإجراء له، وهو في هذا المثال الحدث Button1.Click. يمكن لإجراء واحد أن يكون هو المسؤول عن الاستجابة لأكثر من حدثٍ ولأدواتٍ مختلفة، بحيث تؤدي كلها نفس الوظيفة

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click, TextBox1.DoubleClick
    TextBox1.Text = ""
End Sub
```

و في مثالنا السابق يتم تنفيذ هذا الإجراء عند النقر على الزر أو عند النقر المزدوج لصندوق النص

إنشاء أدوات وقت التنفيذ Creating Controls at Run Time

يمكنك إنشاء أدوات وقت التنفيذ و هذا بإنشاء متغير يمثل نسخة جديدة من الأداة ثم تحديد خصائصها و إضافتها إلى النموذج باستعمال الخاصية Controls انظر المثال :

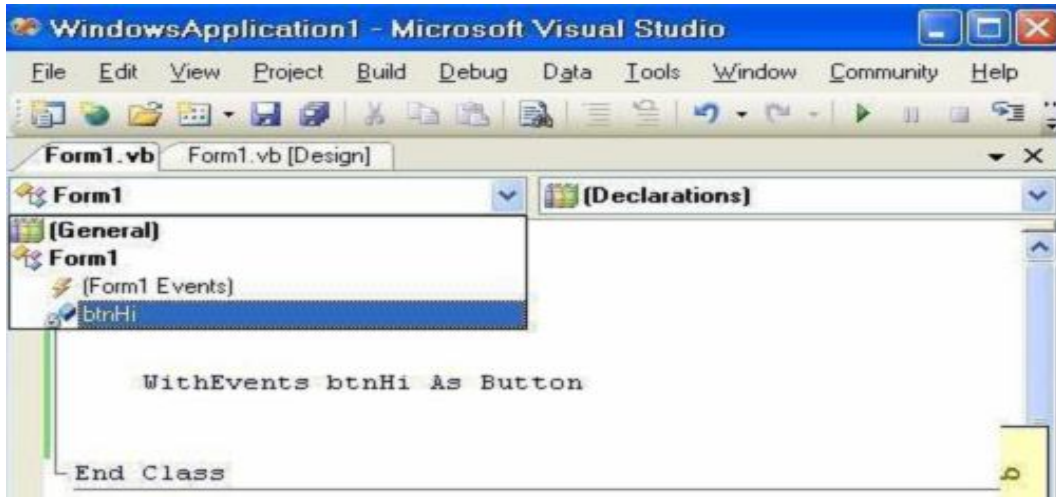
VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    Dim BtHi As New Button ' متغير يمثل نسخة عن أداة الزر
    BtHi.Text = "هنا انقر"
    BtHi.SetBounds(10, 10, 100, 50) ' تحديد موضع الأداة
    Me.Controls.Add(BtHi) ' إضافة الأداة إلى النموذج
End Sub
```

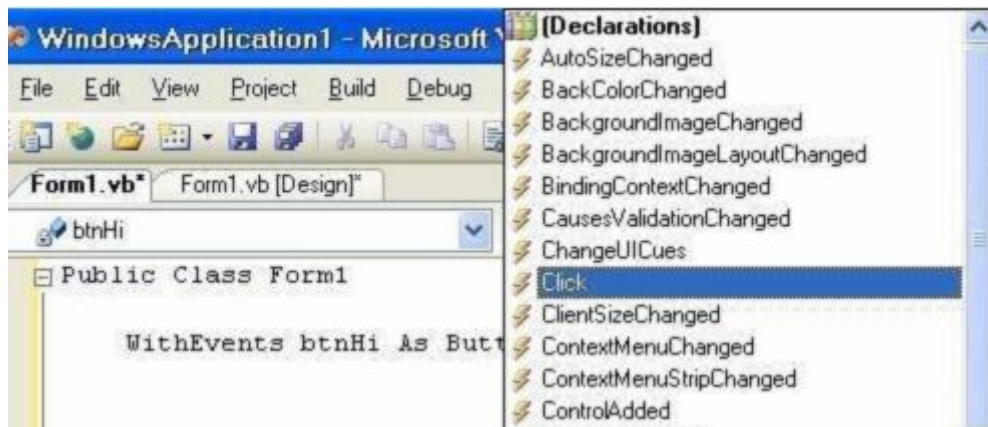
إنشاء أداة جديدة بأحداثها

نعرف متغيراً نصرح فيه عن أداة بأحداثها فنستعمل الكلمة المحجوزة WithEvents كما يلي :

الفصل الثاني : لغة البرمجة



- حدد الأداة btnHi من قائمة سرد الأدوات
- ثم من قائمة سرد الأحداث اختر الحدث Click



VB.Net Code

```
Private WithEvents BtnHi As New Button
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    BtnHi.Text = "هنا انقر"
    BtnHi.SetBounds(10, 10, 100, 50)
    Me.Controls.Add(BtnHi)
End Sub
' إضافة إجراء للزر الذي أنشئته
Private Sub BtnHi_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles BtnHi.Click
    MessageBox.Show("الرحيم الرحمن الله بسم")
End Sub
```

استدعاء الإجراءات بإضافة مستجيبا

يمكنك أن تربط الإجراء بالأداة عند إنشائها ، عن طريق جملة "أضف مستجيبا" AddHandler، والتي لها الصيغة التالية:

AddHandler Control . Event , **AddressOf** Control_Event

VB.Net Code

```
Private WithEvents BtnHi As New Button
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    BtnHi.Text = "هنا انقر"
    BtnHi.SetBounds(10, 10, 100, 50)
    Me.Controls.Add(BtnHi)
    AddHandler BtnHi.Click, AddressOf Button1_Click
End Sub
' إضافة إجراء للزر الذي أنشأته
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    MessageBox.Show("أكبر الله")
End Sub
```

المتغيرات و الثوابت Variables and Constants

ما هي المتغيرات ؟

المتغيرات هي حروف أو أسماء أو كلمات تخزن فيها البيانات في الذاكرة و سميت بالمتغيرات لأنك تستطيع تغيير قيمتها أثناء تشغيل البرنامج من تعريف المتغيرات يتضح لنا أن لكل متغير اسما و يحمل قيمة يختزنها إما أن تكون نصية أو رقمية أو تاريخ و وقت أي نوع آخر

 ملاحظة :


المتغير الاسم Name يمكن أن يأخذ قيمة نصية فمثلا : "الصالح" يجب أن توضع بين علامتي تنصيص

المتغير الرقم Number يمكن أن يخزن قيمة رقمية فمثلا : 5 فتكتب عادية لأن الكمبيوتر لا يفهم إلا لغة الأرقام

المتغير تاريخ الميلاد يمكن أن يخزن قيمة تاريخية مثلا : #1/5/2014# يجب أن توضع بين علامتين # #

المتغير متزوج يمكن أن يخزن قيمة منطقية نعم True أو False فنكتبها كما هي . لأن الكمبيوتر يترجم القيمة True إلى -1 و False إلى 0

الشروط الواجب توفرها لتسمية المتغيرات :

 ألا يزيد عن 255 حرفاً، وهو رقم كبير بالفعل بما يكفي.

الفصل الثاني : لغة البرمجة

النوع	حجمه	الأعداد التي يقبلها
Single	4	للأعداد ذات الدقة العشرية العادية.. وهو يمثل الأعداد السالبة: من -3.402823E38 إلى -1.401298E-45 ، والأعداد الموجبة من 1.401298E-45 إلى 3.402823E38. وفي هذا النوع، لا يمكن تمثيل الصفر بدقة، فهو يمثل دائما بقيمة صغيرة جدًا جدا، ولكنها أبدا ليست صفرا!
مزدوج Double	8	للأعداد ذات الدقة العشرية الفائقة، وهو يستخدم في الحسابات العلمية.
عشري Decimal	16	لتمثيل الأعداد الصحيحة والعشرية، حيث يمكنك من تحديد الدقة العشرية التي تريد التعامل معها، من 0 إلى 28 خانة عشرية.

التعامل مع المتغيرات الرقمية Numeric Variables

لن تكون للمتغيرات الرقمية فائدة ما إن لم تجري عليها العمليات الحسابية المختلفة

العمليات الحسابية على الأرقام

لن تكون هناك فائدة إذا كنت ستضع الأرقام في متغيرات، دون أن تستطيع أن تجريَ عليها بعض العمليات الحسابية.. وفي الجدول التالي علامات العمليات الحسابية الأساسية:

+	علامة الجمع.
-	علامة الطرح.
*	علامة الضرب.
/	علامة القسمة.. ويمكن أن يكون الناتج عددا صحيحا أو به أرقام عشرية.. فمثلا: $X = 7 / 2 ' 3.5$
\	علامة القسمة أيضا، ولكن الناتج هو عدد صحيح فقط.. فمثلا: $X = 7 \setminus 2 ' 3$
Mod	إحدى علامات القسمة أيضا، ولكنها تعطي الباقي من القسمة فحسب.. فمثلا: $X = 7 \text{ Mod } 2 ' 1$
^	الأس.. فمثلا $2 \times 2 \times 2$ تُكتب رياضيا بالصيغة 2^3 ، وتكتب في البرمجة كالتالي: $2 \wedge 3$

ويجب أن ألفت انتباهك إلى أهميّة وضع الأقواس في العمليات المتداخلة، وذلك حتى تضمن صحة إجراء العملية بالترتيب الذي تريده.. إن الترتيب الطبيعي الذي يجرى به VB العمليات الحسابية يسير تبعا للقواعد التالية:

الفصل الثاني : لغة البرمجة

يتم تنفيذ ما بين الأقواس أولاً.

إذا لم تكن هناك أقواس يتم تنفيذ الأسس أولاً.

ثم يتم تنفيذ الضرب والقسمة.

ثم بعد ذلك يتم تنفيذ الجمع والطرح.

لاحظ : عند كتابة $3+2*5$ بدون أقواس الناتج 13

أما عند وضع الأقواس هكذا $(3+2)*5$ يكون الناتج 25

إجراء العمليات على نفس المتغير

لو أردت زيادة 1 لمتغير X . فتقوم بتعريف متغير آخر y تكتب فيه $y = x+1$ ، ثم تسند

قيمة Y إلى المتغير X هذا مجهد . يقترح عليك Vb.Net هذه العمليات المختصرة حسب

الجدول التالي :

الطريقة التقليدية	الطريقة المختصرة المكافئة
$X = X + 1$	$X += 1$
$X = X + Y$	$X += Y$
$X = X - 4$	$X -= 4$
$X = X * 2$	$X *= 2$
$X = X / 9$	$X /= 9$
$X = X ^ 3$	$X ^= 3$

لا ترتبك ! كيف أقرأ التعبير $X = X + 1$ ؟

قيمة المتغير القديمة = قيمة المتغير الجديدة + 1

وهكذا بالنسبة لبقية التعبيرات .

تدريب نموذجي : آلة حاسبة بسيطة نستعمل فيها العمليات على المتغيرات .

1. من قائمة File اختر الأمر New Project

و اكتب في خانة الاسم Windows Application يظهر مربع حوار اختر التطبيق CalProject

2. قم بتصميم واجهة البرنامج على النحو التالي :

الفصل الثاني : لغة البرمجة

Fixed Or F Or f	يعرض أعدادا مقربة بثلاث علامات عشرية
Standard Or N Or n	تنسيق رقم عادي
Percent	النسبة المئوية
E Or e	أرقاما علمية
D Or d	أرقام عشرية. (base-10) system.
X Or x	أرقام عشرية. (base-16) system.
Yes/No	Displays "No" for zero and "Yes" for all other values.
true/False	Displays "False" for zero and "True" for all other values.
On/Off	Displays "Off" for zero and "On" for all other values.

أمثلة :

Code Vb.Net

```
Dim X1 As Double = 1500.5
Debug.Print(X1.ToString("G")) '1500,555
Debug.Print(X1.ToString("F")) '1500,56
Debug.Print(X1.ToString("c")) '1 500,56 €
Debug.Print(X1.ToString("N")) '1 500,56
Debug.Print(X1.ToString("E")) '1,500556
Debug.Print(250.ToString("X")) 'FA
```

ثانيا : باستعمال الدالة Format

الصيغة البسيطة



```
Dim result As String = Format(expression[, style])
Expression : .... البيانات رقمية أو نصية أو تاريخ
Style : الصيغة
```

نستعمل نفس الرموز الموضحة في جدول السابق

أمثلة :

Code Vb.Net

```
Dim X As Double = 1500.45
Console.WriteLine(Format(X, "G"))
Console.WriteLine(Format(X, "c"))
Console.WriteLine(Format(X, "Percent"))
Console.WriteLine(Format(X, "f"))
Console.WriteLine(Format(X, "N"))
```

يكون الناتج

```
1500,45
1 500,45 €
150045,00%
1500,45
1 500,45
```


المتغيرات الحرفية Character Variables

هي متغيرات طولها وحدتا ذاكرة 2 Bytes، يمكن أن تخزن فيها حرفا واحدا.. ونظرا لأن الكمبيوتر لا يعرف شيئا غير الأرقام، فهو يمثل الحروف بأرقام فمثلا الحرف "a" يمثل الرقم 65

و لتعريف متغيرات من هذا النوع نكتب

```
Dim MyChar As Char = "A"
```

حيث يخبر الحرف c، VB أن ما بين علامتي التنصيص حرف وليس نصا .. و يجب أن نكتب حرفا واحدا فقط .

أهم وسائل المتغيرات الحرفية : و هي مشتركة يمكن استعمالها مباشرة من الخلية Char

Method	الشرح
Is Control	إنه حرف تحكم
Is Digit	إنه رقم
IsPunctuation	إنه علامة ترقيم
IsLetter	إنه حرف
IsLetterOrDigit	إنه حرف أو رقم
IsLower	إنه حرف صغير
IsUpper	إنه حرف كبير
IsWhiteSpace	إنه مسافة بيضاء
IsSeparator	إنه فاصل
IsSymbol	إنه رمز
IsNumber	إنه رقم
GetNumericValue	اقرأ القيمة الرقمية
GetUnicodeCategory	اقرأ الطائفة

إنه حرف تحكم : IsControl

تستقل حرفا وتعيد True إذا كان الحرف المضغوط حرف تحكم مثل : Backspace، و Escape ... إلخ

Visual Basic.Net

```
Dim Value As Boolean = Char.IsControl("حرف")
```

```
Dim Value As Boolean = Char.IsControl ("نص", "نص")
```

Exp

```
Console.WriteLine(Char.IsControl(Chr(9))) 'True
```

```
Console.WriteLine(Char.IsControl("Test Char", 5)) 'False
```

إنه رقم IsDigit

تستقبل حرفا و تعيد True إذا كان هذا الحرف رقما من 0 إلى 9

Visual Basic.Net

```
Dim Value As Boolean = Char.IsDigit("حرف")
Dim Value As Boolean =Char .IsDigit ("نص", موضع,"نص")
Exp
Console.WriteLine(Char.IsDigit("5")) 'True
Console.WriteLine(Char.IsDigit("1 النص", 5)) ' True
```

تطبيق :السماح بكتابة الأرقام فقط في مربع نص

Visual Basic.Net

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles
    TextBox1.KeyPress
    If Not Char.IsDigit(e.KeyChar) Then e.Handled = True
End Sub
```

إنه حرف IsLetter

تستقبل هذه الوسيلة حرفا و تعيد True إذا كان الحرف المضغوط من حروف الكتابة و ليس رقما .

Visual Basic.Net

```
Dim Value As Boolean = Char.IsLetter("حرف")
Dim Value As Boolean =Char .IsLetter ("نص", موضع,"نص")
Exp
Console.WriteLine(Char.IsLetter("5")) 'False
Console.WriteLine(Char.IsLetter("1 نص", 1)) 'True
```

تطبيق : كتابة الحروف فقط في مربع النص

Visual Basic.Net

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles
    TextBox1.KeyPress
    If IsNumeric(e.KeyChar) Then e.Handled = True
End Sub
```

إنه حرف أو رقم IsLetterOrDigit

تستقبل حرفا أو رقما و تعيد True إذا كان الحرف المضغوط رقما أو رقما

Visual Basic.Net

```
Dim Value As Boolean = Char.IsLetterOrDigit ("حرف")
```

```
Dim Value As Boolean = Char.IsLetterOrDigit ("نص", 1)
```

Exp

```
Console.WriteLine(Char.IsLetterOrDigit("5")) 'True
```

```
Console.WriteLine(Char.IsLetterOrDigit("1", 1)) 'True
```

```
Console.WriteLine(Char.IsLetterOrDigit(Chr(9))) 'False
```

اقرأ القيمة الرقمية GetNumericValue

تستقبل رقما على شكل حرفا و تعيد لك هذا الرقم . و إذا كان هذا الحرف غير رقمي فتعيد -1

Visual Basic.Net

```
Dim Value As Double = Char.GetNumericValue("رقم يمثل حرف")
```

```
Dim Value As Double = Char.GetNumericValue("رقم يحتوي نص", 5)
```

Exp

```
Console.WriteLine(Char.GetNumericValue("5")) '5
```

```
Console.WriteLine(Char.GetNumericValue("1", 5)) '1
```

اقرأ الطائفة GetUnicodeCategory

تعيد لك هذه الدالة الطائفة التي ينتمي إليها الحرف.. هذه الطائفة هي تعبير رقمي من أعضاء المرقم UnicodeCategory،

Code Vb.Net

```
Console.WriteLine(Char.GetUnicodeCategory("a").ToString)
```

```
Console.WriteLine(Char.GetUnicodeCategory("(").ToString)
```

```
Console.WriteLine(Char.GetUnicodeCategory("_").ToString)
```

```
Console.WriteLine(Char.GetUnicodeCategory("").ToString)
```

```
Console.WriteLine(Char.GetUnicodeCategory("5").ToString)
```

تدريب نموذجي : معرفة نوع الحرف المدخل

Code Vb.Net

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, _  
    ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles
```

```
    TextBox1.KeyPress
```

```
        Me.Text = Char.GetUnicodeCategory(e.KeyChar).ToString
```

```
End Sub
```

إنه صغير IsLower

ترجع هذه الدالة True إذا كان الحرف المضغوط صغيرا Small.

لها صيغة مشابهة للصيغ السابقة

إنه كبير IsUpper

ترجع هذه الدالة True إذا كان الحرف المضغوط كبيراً Capital.

لها صيغة مشابهة للصيغ السابقة

إنه عدد IsNumber:

مماثلة للدالة IsDigit، إلا إنها تأخذ في اعتبارها الحروف الداخلة في تمثيل الأرقام في النظام السداسي عشري،

إنه علامة ترقيم IsPunctuation:

أعتقد أنّ وظيفتها واضحة!

إنه فاصل IsSeparator:

ترجع هذه الدالة True إذا كان الحرف المضغوط من الحروف الفاصلة، مثل المسافة وحرف الإدخال Enter ، والفاصلة والنقطتان المتعامدتان والأقواس ونقطة التوقف ... إلخ.

إنه مسافة بيضاء IsWhiteSpace:

ترجع هذه الدالة True إذا كان الحرف المضغوط مجموعة مسافات أو حرف الجدولة Tab أو مغذيات السطر Line feeds ... إلخ.

أما الوسائل الثلاث التالية فهي غنيّة عن التعريف:

حوّل إلى حرف صغير ToLower، حوّل إلى حرف كبير ToUpper، حوّل إلى نصّ ToString.

الدالة Chars

تستقبل نصاً وتعيد الحرف حسب الرقم الترتيبي له في النص و نبدأ الترقيم من 0

Code Vb.Net

```
Dim MyString As String = "a2d"  
MsgBox(MyString.Chars(1)) '2
```

بما أنّ الدالة Chars() تُرجع حرفاً، إذن فيمكن تطبيق وسائل الحروف عليها، كالتالي:

Code Vb.Net

```
Dim MyString As String = "a2d"  
MsgBox(Char.IsDigit(x.Chars(1))) 'true
```

المتغيرات النصية String Variables

هي متغيرات يمكنك أن تخزن أيّ نصّ، مهما كان نوعه (حروف، رموز، أرقام، علامات تنسيق ... إلخ)، ومهما كان طوله .

Code Vb.Net

```
Dim MyString As String  
MyString = "فارغ نص"
```

```
MyString = "حرف"  
MyString = "طويلا نصا اكتب"  
MyString = "15"  
MyString="05/25/2005"
```

تشبيك و دمج النصوص Concatenation

لدمج النصوص في نص واحد جديد استعمل العلامتين + أو &

Code Vb.Net

```
Dim S1 As String = "أشهد أن لا إله إلا الله "  
Dim S2 As String = "و أشهد أن محمدا رسول الله "  
Dim S As String = S1 + S2  
MsgBox(S) ' أشهد أن لا إله إلا الله و أن محمدا رسول الله
```

يمكنك استبدال علامة + بعلامة & ستحصل على نفس النتيجة
وبإمكانك استعمال الصيغة المختصرة كما يلي :

Code Vb.Net

```
Dim X As String = "Salah "  
X += "Meddour"  
Console.WriteLine(X) ' Salah Meddour
```

أو استعمال الدالة Concat لدمج النصوص

Code Vb.Net

```
Dim S1 As String = "الحمد"  
Dim s2 As String = "الله"  
MsgBox(String.Concat(S1, s2)) ' الحمد لله
```

مقارنة النصوص Comparison

لمقارنة النصوص نستعمل معاملات المقارنة (= ، > ، < ، <>)

1. في حالة تساوي نصين نستعمل =
2. في حالة اختلاف النصين نستعمل <>
3. أنك تستطيع مقارنة النصين باستخدام علامتي "<" و ">" وذلك لتعرف من منهما يسبق الآخر في الترتيب الأبجدي، حيث يكونُ الأسبقُ منهما في الترتيب الهجائي هو النص الأصغر
4. كما يمكن استعمال المعامل Like لمعرفة ما إذا كان نصا فيه صيغة Pattern محددة

تدريب نموذجي

الفصل الثاني : لغة البرمجة

1. قم بتصميم واجهة البرنامج كما يلي :



2. اكتب التعليمات التالية في حدث نقر الزر

Code Vb.Net

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim S1 As String = TextBox1.Text
    Dim S2 As String = TextBox2.Text
    If S1 = S2 Then MsgBox("متساويان النصان")
    If S1 <> S2 Then MsgBox("مختلفان النصان")
    If S1 < S2 Then MsgBox("الثاني يسبق الأول النص")
    If S1 > S2 Then MsgBox("1النص يسبق 2النص")
End Sub
```

استعمال المعامل يشبه Like

يمكنك استخدام المعامل "يشبه" Like لمعرفة ما إذا كان النص يحتوي على صيغة Pattern معينة أم لا..

الرموز المستعملة مع Like

الرمز	الدور	أمثلة
*	تحل محل مجموعة من	A* تعرض كل الأسماء التي تبدأ ب A
%	الحروف	A* تعرض كل الأسماء التي تنتهي ب A *Ali* تعرض كل الأسماء التي تشمل على الحروف في أي موضع كانت
?	تحل محل حرف واحد	A?i تعرض مثلاً Sai ، Ami ، Ali
[]	حرف واحد من مدى معين	AL[i,y] تعرض Ali ، Aly
#	تحل محل رقم واحد	F# تعرض F1 ، F2 ،
[!]	تستثني الحرف	*A[!q] تعرض كل الأسماء التي تنتهي بحرفين A و أي حرف آخر

أمثلة

Code Vb.Net

```
Dim S As String = "I Like Visual Basic 2008 .Net"  
If S Like "*Net*" Then MsgBox("موجودة الكلمة") ' true  
If S Like "*Net" Then MsgBox("Net تنتهي الجملة") ' true  
If S Like "Net*" Then MsgBox("Net ب تبدأ") ' False  
If S Like "*B?sic*" Then MsgBox("موجودة الكلمة") ' True  
If S Like "*200#*" Then MsgBox("موجودة الكلمة") ' True
```

المتغيرات المنطقية Boolean Variable

المتغيرات المنطقية تخزن قيمة صحيحة True أو قيمة خاطئة False وهي في الأساس أعداد صحيحة، فالقيمة "صواب" تعادل -1، والقيمة "خطأ" تعادل صفراً.. ويمكنك تعريف المتغيرات من هذا النوع

Dim Value As Boolean

لتخصيص قيمة للمتغيرات المنطقية نكتب :

Dim Value As Boolean ' False القيمة الافتراضية

Dim Value As Boolean = True

تستعمل المتغيرات المنطقية في المقارنة و التحقق من الصحة
فمثلا : اجعل إمكانية استعمال الزر Button موافقة لفرغ مربع النص من عدمه
فاكتب السطر التالي في حدث تغيير حروف مربع النص:

Code Vb.Net

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles TextBox1.TextChanged  
Button1.Enabled = (TextBox1.Text <> "")  
End Sub
```

إذا كان مربع النص فارغا فلا يمكنك استعمال الزر و العكس صحيح .
وتستخدم المتغيرات المنطقية أيضا في اختبار الشروط

Code Vb.Net

```
Dim Found As Boolean = True  
If Found = True Then MsgBox("بنجاح العملية تمت")
```

المتغيرات الزمنية Date Variables

تخزن البيانات التي تعبر عن التاريخ أو الوقت بتنسيق خاص كأعداد مزدوجة، بحيث يمثل العدد الصحيح التاريخ، ويمثل الجزء العشري الزمن.
ويتم تعريف هذا النوع من المتغيرات بكتابة

Code Vb.Net

```
Dim VarDate As Date
```


```
VarDate = #5/20/2000#
```

```
VarDate = #10/25/2000 12:30:15 PM#
```

```
VarDate = "mai 2,2000"
```

```
VarDate = Now ' التاريخ و الوقت الحالي
```

تلاحظ أننا نكتب التاريخ بطريقتين: إما كتاريخ رقمي مكتوب بين علامتي ##، أو كتاريخ نصي مكتوب بين علامتي تنصيص ""، حيث سيتم تحويله تلقائياً إلى تنسيق التاريخ و الوقت المناسب.

 ملاحظة : و هناك المزيد عن البيانات الزمنية مع خلية التاريخ و الوقت

المتغيرات الكائنية Object Variables

المتغيرات Object تستطيع تخزين أي نوع من البيانات . وينصح بعدم استعمالها لأنها تبطأ الجهاز حيث يقوم بعمليات معقدة لمعرفة نوع البيانات التي تخزنها ثم تحويلها إلى عمليات مناسبة .

و لتعريفها نكتب

```
Dim ObjVar As Object
```

```
ObjVar = "...أو تاريخ أو رقم أو نص تريد شيء أي"
```

المتغيرات الخاصة بالمصفوفات

تقوم بتخزين مصفوفة من أي نوع من البيانات نصية أو رقمية أو تاريخ أو كائنات ... عند تعريف مثل هذه المتغيرات نكتب الكلمة محجوزة Dim أو Private أو ... متبوعة باسم المتغير ثم قوسين بداخلها عدد يمثل عدد عناصر المصفوفة أو لا شيء في المصفوفة الديناميكية ثم كلمة AS متبوعة بنوع البيانات التي ستخزنها .

Code Vb.Net

```
Dim Id(3) As Integer
```

```
Dim Names(0 To 3) As String
```

```
Dim Grades() As Double = {12.5, 10, 14, 13.25}
```

```
Dim Students() As String = New Object() {"صالح", "محمد"}
```

اختبار أنواع المتغيرات :Examining Variable Types

استعمل الدالتين :

- "اقرأ النوع" GetType() وهي تُرجع نوعا Type عبارة عن كائن يمثّل نوع المتغير
- "اقرأ كود النوع" GetTypeCode() وهي تُرجع قيمةً ترمز للنوع

مثال :

Code Vb.Net

```
Dim Var As Double = 14.5
Console.WriteLine("& Var.GetType.ToString) ' System.Double
هو المتغير نوع"
```

لو أردت أن تتحقق من قيمة نوع المتغير و تستعملها في الجمل الشرطية استعمل :

1. Type كما في المثال التالي :

Code Vb.Net

```
Dim Var As Double = 14.5
If Var.GetType() Is GetType(Double) Then MsgBox("Yes")
```

2. استخدم التعبير IS TypeOf

Code Vb.Net

```
Dim Var As String = ""
If TypeOf Var Is String Then MsgBox("نصّ المتغير نوع")
```

ولو أردت أن تعرف اسم نوع المتغير، فيمكنك أن تستخدم الدالة TypeName، التي تُعيد نصّا يمثّل اسم النوع الذي ينتمي إليه المتغير

Code Vb.Net

```
Dim i As Integer
Dim a As Object
Console.WriteLine(TypeName(i)) ' Integer
Console.WriteLine(TypeName(a)) ' Nothing
a = "Ali"
Console.WriteLine(TypeName(a)) ' String
a = 15.12
Console.WriteLine(TypeName(a)) ' Double
```

كما أنّ لديك نوعيّة أخرى من الدوال، لمعرفة نوعيّة البيانات: هل هي رقميّة أم نصيّة ... إلخ.. انظر للجدول التالي:

تُرجع True، إذا كان معاملها رقما من أيّ نوع، حتّى لو كان نصّا يحتوي على رقم.	"إنّه رقمي" IsNumeric()
تُرجع True، إذا كان معاملها تاريخا أو وقتا.	"إنّه تاريخ" IsDate()

الفصل الثاني : لغة البرمجة

إنه لا شيء" IsNothing	تُرجع True، إذا كان معاملها غير مرتبط بأيّ كائن.
"إنه فارغ" IsDBNull()	تُرجع True، إذا كان معاملها غير مرتبط بأيّ كائن
"إنه مرجع" IsReference()	تُرجع True، إذا كان معاملها كائناً
IsArray	تُرجع True، إذا كان معاملها مصفوفة..

أمثلة :

Code Vb.Net

```
Dim x As Object = InputBox("أكتب شيئاً")
If IsNumeric(x) Then MsgBox("رقم إنه")
If IsDate(x) Then MsgBox("تاريخ إنه")
If IsDBNull(x) Then MsgBox("فارغ إنه")
If IsNothing(x) Then MsgBox("لاشيء إنه")
```

التحويل بين أنواع المتغيرات :Converting Variable Types

في الجدول التالي، ستجد كل الدوال التي تسمح لك بالتحويل بين الأنواع المختلفة للمتغيرات:

الدالة	تحوّل إلى
CBool	منطقي Boolean
CByte	وحدة ذاكرة Byte
CChar	حرف Unicode character
CDate	تاريخ Date
CDbl	عدد مزدوج Double
CDec	عدد عشري Decimal
CInt	عدد صحيح Integer (Int32)
CLng	عدد طويل Long (Int64)
CObj	كائن Object
CShort	عدد قصير Short (Int16)
CSng	عدد مفرد Single
CStr	نصّ String

Code Vb.Net

```
Dim A As Integer
```

```
Dim B As Double
```

نحول عدد صحيح إلى عدد مزدوج ' $B = CDb1(A)$

كما يمكنك التحويل بين أنواع المتغيرات المختلفة باستعمال الدالة العامة للتحويل CType



```
Dim result As typename = CType(expression, typename)
```

Expression : ... البيانات التي يتم تغييرها مهما كان نوعها نصية رقمية أو تاريخ ...

Typename : نوع البيانات التي سيتم التحويل إليها

مثال :

Code Vb.Net

```
Dim a As String = "1000"
```

```
Dim b As Single = CType(a, Single) ' حولنا المتغير من نصي إلى مفرد
```

كما يمكنك التحويل بين الأنواع المختلفة باستعمال الدالة Convert و التي تمتلك مرقم يحتوي على جميع التحويلات الممكنة . انظر المثال :

Code Vb.Net

```
Dim A As String = "10.5"
```

```
Dim B As Decimal = Convert.ToDecimal(A) ' تحويل من متغير نصي إلى رقم عشري
```

الخيار "التحويل الدقيق" Option Strict :

هذا الخيار مغلق في الوضع التلقائي، ليسمح للغة بنقل البيانات بين أنواع مختلفة دون استخدام دوال التحويل، ففي هذه الحالة، تقوم اللغة بإجراء التحويلات المناسبة تلقائيًا.

Code Vb.Net

```
Dim a As Integer = 5, b As String = "15"
```

```
Console.WriteLine(a + b) ' 20
```

قام Vb.Net بتحويل المتغير النصي b ألياً إلى متغير مكافئ ل a لتتم عملية الجمع و لو جعلنا الخيار Option Strict On ستظهر رسالة خطأ . انظر المثال


```

Form1.vb* Form1.vb [Design]*
Button1 Click
Option Strict On
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim a As Integer = 5, b As String = "15"
        Console.WriteLine(a + b) ' يظهر خط متعرج يعبر عن الخطأ
        Console.WriteLine(a & b) ' "515"
    End Sub
End Class
    
```

و لإصلاح الخطأ يلزمنا استعمال دوال التحويل. ولكن ما زال التحويل بين أنواع المتغيرات الرقمية يتم آليا . و في اتجاه واحد فقط حيث تضع المتغير الأصغر في المتغير الأكبر و يسمى بالتحويل الواسع Widening Conversion حيث يقوم بتحويل قيمة من نوع صغير Short إلى نوع كبير Double حفاظا على الدقة العشرية للعدد. و العكس غير ممكن

Code Vb.Net

```

Option Strict On
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim A As Single = 5.9999
        Dim B As Double = A
        MsgBox(A = B).ToString() 'true
    End Sub
    
```

الخيار "التعريف الصريح" Option Explicit

في الحالة الافتراضية يكون On وهنا يلزمك هذا الخيار تعريف كل المتغيرات في برنامجك . إما إذا جعلته مقفلا Off بكتابة الجملة في بداية الملف

```

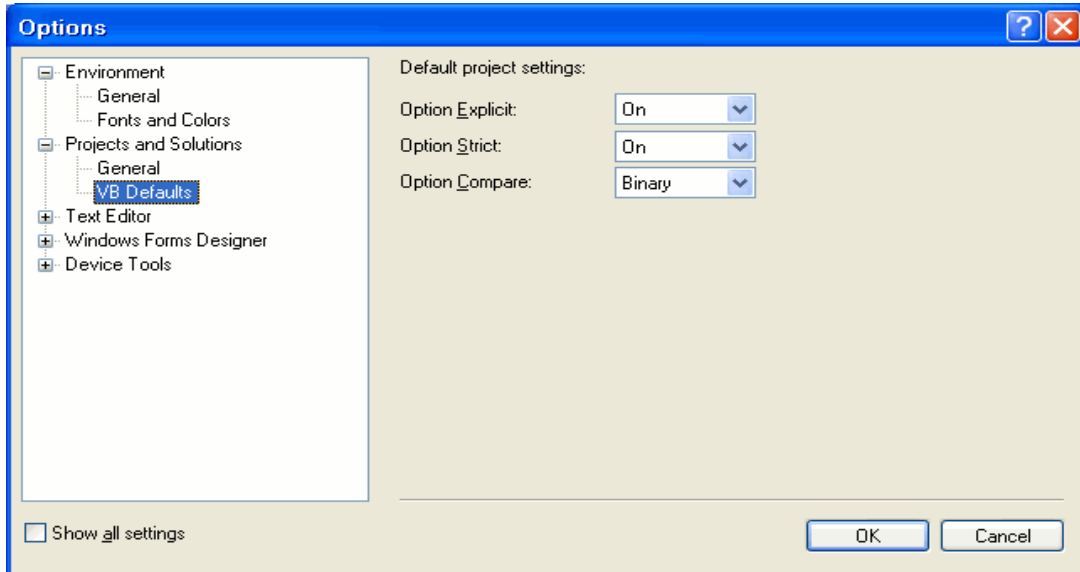
Option Explicit Off
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
    
```

في هذه الحالة نستخدم المتغيرات في برنامجنا هذا دون الحاجة لتعريفها . هنا يفهم المترجم أن هذه المتغيرات يمكنها أن تحوي أي نوع من البيانات لهذا سيعتبرها من نوع كائن Object حيث تحجز مساحة كبيرة في الذاكرة . ثم يقوم بالتحويلات المناسبة في العمليات وهذا ما يجعل الجهاز بطيئا .

```

Button1 Click
Option Explicit Off
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        MyName = "salah"
        i = 15
        قمنا بكتابة المتغيرات دون تعريفها
    End Sub
End Class
    
```

إن كتابة الخيارين Option Strict و Option Explicit في بداية كل ملف يؤثر في هذا المشروع فقط
 ماذا لو أردت أن يؤثر على كل ملفات المشروع ؟
 في هذه الحالة، يمكنك أن تختار هذين الخيارين من صفحات الخواص للمشروع الحالي، وذلك بضغط زر الفأرة الأيمن على اسم المشروع في متصفح المشاريع Solution Explorer، وضغط "خصائص" Properties من القائمة الموضعية.. وفي هذه الحالة، ستكون الإعدادات التي تختارها لهما عامّة لكل المشروع



مجال المتغير Variable Scope:

ما يعرف بقابلية الرؤية Visibility مجال المتغير هو مقطع الكود الذي يمكن من خلاله رؤية المتغير واستخدامه. وهذا المجال ببساطة، هو أصغر مقطع يوجد فيه المتغير، من المقاطع التالية:

- .Sub.. End Sub
- .Function.. End Function
- .If.. End If
- .For.. Next

.Do.. Loop -

.While.. End While -

هذا، ويُسمى المتغير المعرف على مستوى الإجراء أو الدالة متغيراً موضعياً Local Variable، بينما يسمى المتغير الذي نعرفه داخل أحد المقاطع متغيراً مقطعيًا Block Variable.. أما المتغيرات المعرفة على مستوى الخلية أو القالب، فتسمى متغيرات مستوى القالب .Module-level Variables

```
Public Class Form1
    Sub Mysub1()
        Dim x As Integer = 10
    End Sub
    Sub Mysub2()
        Console.WriteLine(x) ' لا يمكن الوصول إلى المتغير '
    End Sub
End Class
```

عمر المتغير The Lifetime of a Variable:

فيمثل الفترة التي يظل فيها المتغير محتفظاً بقيمته

- المتغيرات المعرفة على مستوى المشروع تحتفظ بقيمتها طوال تشغيل البرنامج.
- والمتغيرات المعرفة على مستوى النموذج، توجد عند تحميل النموذج Load،
- والمتغيرات المعرفة داخل الإجراءات، تحتفظ بقيمتها إلى أن ينتهي الإجراء،

دراسة الصيغة العامة لتعريف المتغيرات



```
[attribute_list] [accessibility] [Shared] [Shadows] [ReadOnly] _
Dim [ WithEvents] name [(bounds_list)] [As [New] type] [=
initialization_expression]
```

كل العبارات التي بين [] فهي اختيارية

من خلال الصيغة يتضح لنا ما هو مطلوب :

Dim Name As Type

عند تعريف المتغيرات نستعمل الكلمة المحجوزة Dim متبوعة باسم المتغير ثم كلمة As ثم نوع المتغير. أما باقي المعاملات اختيارية .

دراسة معاملات الصيغة

attribute_list

عبارة عن قائمة من الصفات و السمات تميز المتغير وتمنحه ميزة و سمة خاصة .

تحديد طول المتغير ' Dim Nom As String <VBFixedString(10)>

الفصل الثاني : لغة البرمجة

قابلية الوصول إلى المتغير و يأخذ القيم التالية :

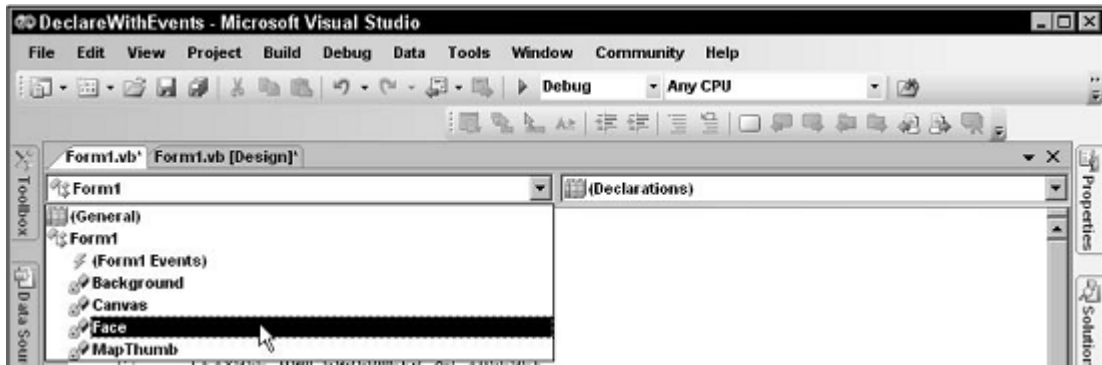
Public	قابلية الوصول للمتغيرات المعرفة ب public ممكنة من داخل المشروع الحالي أو من المشاريع الأخرى و يصرح بها في structure, namespace,module, class
Protected	
Private	قابلية الوصول إلى المتغيرات المعرفة ب Private ممكنة فقط من داخل القالب Module أو الخلية Class أو النموذج Form أو السجل Structure الذي ينتمي إليه.
Friend	قابلية الوصول للمتغير المعرف ب Friend ممكنة من داخل المشروع الحالي فقط و ليس من المشاريع الأخرى .
Static	قابلية الوصول للمتغيرات المعرفة ب Static ممكنة و دائمة إلى نهاية المشروع أو موت الكائن التابع له . و يعرف داخل إجراء
Protected Friend	المتغيرات المصرحة بكلمة Protected Friend قابلة للوصول إليها من خلال فئة أخرى من نفس المشروع

WithEvents

يستعمل لتعريف متغيرات خاصة لإنشاء أدوات بأحداثها. في هذا المثال نعرف متغيرا باسم Face على أنه كائن من نوع صندوق صورة كما يلي :

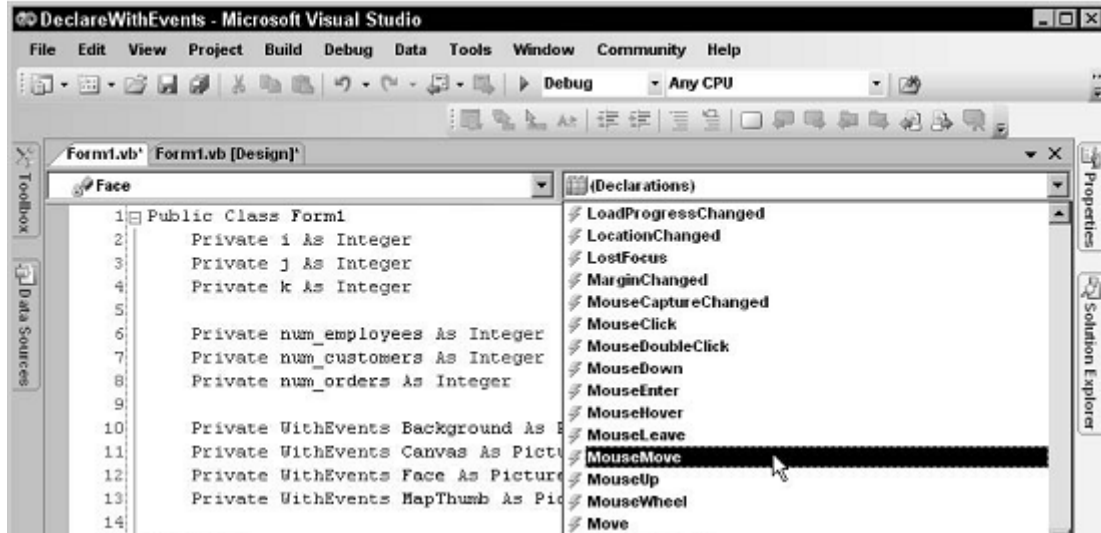
Dim WithEvents Face As New PictureBox

بمجرد تعريف المتغير بالكلمة المحجوزة WithEvents يتم إنشاء الكائن و تجده في مربع سرد الأدوات كما يظهر في الشكل :



و عند نقر الكائن من مربع الأدوات تظهر قائمة منسدلة على اليمين لأحداث هذا الكائن

الفصل الثاني : لغة البرمجة



يمكنك أن تختار الأحداث التي ترغب فيها و تلحقها بالأداة .

ReadOnly للقراءة فقط

يمكنك استعمال ReadOnly لتعريف المتغيرات في الوحدات النمطية modules و الفئات Classes و التراكيب Structures و مجالات الأسماء NameSpaces ليس داخل الإجراءات لا يمكنك تغيير قيمتها .

```
Public Class Form1
    Public ReadOnly X As Integer = 100
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        X = 50 ' تظهر رسالة خطأ لأننا حاولنا تغيير قيمة المتغير '
        MsgBox(X)
    End Sub
End Class
```

المتغيرات الخاصة بالمصفوفات و القوائم bounds_list

عند تعريف مثل هذه المتغيرات نكتب الكلمة المحجوزة Dim أو Private أو ثم اسم المتغير متبوعاً بقوسين داخلها حجم أو طول المصفوفة ثم الكلمة المحجوزة As ثم نوع المتغير

المعامل New

لتعريف متغير جديد من فئة أخرى أو متغير آخر أو كائن

Dim Txt As New TextBox ' إنشاء نسخة جديدة من كائن مربع النص

و هذا مثال يوضح لنا كيف ننشأ نسخة جديدة من سجل

Code Vb.Net

```
Public Class Form1
    Structure Persone
```

الفصل الثاني : لغة البرمجة

```

Dim Num As Integer
Dim Prénom As String
End Structure
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim P As New Persone
End Sub
End Class
    
```

المعامل Type :

يمثل نوع المتغير و إليك الجدول التالي يبين لنا كل أنواع المتغيرات

النوع	نوع المتغير	الحجم بالذاكرة بالبايت	القيمة التي يمكن تخزينها
متغيرات صحيحة	Byte	1	0 - 255
	SByte	1	-128 - 127
	Short OR Int16	2	-32768 - 32767
	UShort OR UInt16	2	0 - 65535
	Integer OR Int32	4	-2147483648 - 2147483647
	UInteger OR UInt32	4	0 - 4294967295
	Long OR Int64	8	-9223372036854775808 - 9223372036854775807
	ULong OR UInt64	8	0 - 18446744073709551615
متغيرات كسرية	Single	4	-3.40282e+038F - 3.40282e+038F
	Double	4	+1.79769e+308 - 1.79769e+308
	Decimal	8	-79228162514264337593543950335m - 79228162514264337593543950335m
متغير نصي	String	20 عند الحروف	أي سلسلة نصية من حروف وأرقام ورموز و ...
متغير حرفي	Char	2	أي حرف أو رقم أو رمز أو ...
متغير منطقي	Boolean	1	القيمة: true أو false
متغير التاريخ والوقت	DateTime	8	تاريخ أو وقت بكل التسميات المختلفة .
كائن	Object	8	أي قيمة من القيم السابقة .

تخصيص قيمة لمتغير initialization_expression

Code Vb.Net

```

Dim age As Short = 35
Dim Prénom As String = "Salah"
    
```

الفصل الثاني : لغة البرمجة

```
Dim Date_value As Date = #1/1/1995#
Dim Myarray() As String = {"a", "b", "c"}
```

```
Dim Nom As String
Nom = "Meddour"
```

و لتخصيص قيما للمتغيرات في سطر واحد نكتب

```
Dim Age As Integer = 35, Prénom As String = "Salah", Value_Date As
Date = #1/1/1995#
```

الثوابت Constants

هي كالمتغيرات إلا أنها تخزن قيما ثابتة و تبقى محافظة على هذه القيمة طيلة البرنامج و يفضل كتابتها في قسم التعريفات العامة أو في Module و لتعريفها نكتب الكلمة المحجوزة Const متبوعة باسم الثابت ثم كلمة As ثم نوع الثابت كما في المتغيرات الاختلاف البسيط هو كلمتي Dim و Const

```
Public Class Form1
    Const Pi As Double = 3.14
    Const Password As String = "Salah"
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
    End Sub
End Class
```

المعاملات Operators

يمنحك Vb.Net أنواع مختلفة من المعاملات .

المعاملات الرياضية Arithmetic Operators

المعامل	Operator	المثال	الناتج
^	Exponentiation	3^2	8
-	Negation	2-	2-
*	Multiplication	3*2	6
/	Division	2/3	1.5
\	Integer division	5 \ 17	3
Mod	Modulus	Mod 5 17	2
+	Addition	3+2	5
-	Subtraction	2-3	1
>>	Bit left shift	1 >> 10110111	01101110
<<	Bit right shift	1 << 10110111	01011011

معاملات المقارنة Comparison Operators

المعامل	Operator	الشرح	المثال
=	Equals	يساوي	5 = 5
<>	Not equals	يختلف	5 <> 2

الفصل الثاني : لغة البرمجة

5 > 2	أكبر من	Less than	>
5 >=2	أكبر أو يساوي	Less than or equal to	>=
2 < 5	أصغر من	Greater than	<
2 <=5	أصغر من أو يساوي	Greater than or equal to	<=
A is B (متغيرات)	إنه	Equality of two objects	Is
A is Not B	ليس	Inequality of two objects	IsNot
Typeof(obj) Is textbox	نوع من	Object is of a certain type	TypeOf...Is
A Like A	مثل	Matches a text pattern	Like

المعاملات الخاصة ب Like

الرمز	الدور	أمثلة
*	تحل محل مجموعة من الحروف	A* تعرض كل الأسماء التي تبدأ ب A A* تعرض كل الأسماء التي تنتهي ب A *Ali* تعرض كل الأسماء التي تشمل على الحروف في أي موضع كانت
?	تحل محل حرف واحد	A?i تعرض مثلا Sai , Ami , Ali
[]	حرف واحد من مدى معين	AL[i,y] تعرض ALy , Ali
#	تحل محل رقم واحد	F# تعرض F1 , F2 ,
[!]	تستثنى الحرف	A[!q] * تعرض كل الأسماء التي تنتهي بحرفين A و أي حرف آخر

المعاملات المنطقية

المعامل	الشرح	مثال
Not	النفى	Not true ' False
And	و	True and True ' True
Or	أو	True Or False ' True
Xor	عملية استثناء منطقية	False Xor False ' True
AndAlso	تتحقق من صحة القضية 1 فإذا كانت صحيحة تنتقل لاختبار القضية 2	True And True ' True
OrElse		

السجلات Structure

كثيرا ما تحتاج للتعامل مع مجموعات من المتغيرات الأنواع، لتخزين سجل واحد من البيانات، كأن تريد تخزين سجلات عن الطلبة، كل سجل يحتوي على اسم الطالب ورقمه وفصله... إلخ. ويعرف هذا النوع من التركيبات بالأنواع المعرفة من قبل المستخدم


(UDT) User Defined Types

ويمكنك أن تُعرّف مثل هذا السجل Structure كالتالي

الفصل الثاني : لغة البرمجة

VB.Net Code

```
Structure Student
    Dim Name As String
    Dim ID As Integer
    Dim StudentClass As Integer
End Structure
```

 ملاحظة :

هذا التعريف لا يمكن أن يكون داخل إيّ إجراء أو دالة، وإنما على مستوى النموذج أو القالب.

يمكنك استخدامه في برنامجك، كالتالي:

VB.Net Code

```
Dim Ahmad As Student
Ahmad.Name = "أحمد"
Ahmad.StudentClass = 5
Ahmad.ID = 40
```

التركيبات المتداخلة :

VB.Net Code

```
Structure Person
    Structure AddressStruct
        Dim City As String
        Dim Country As String
    End Structure
    Dim Name As String
    Dim Age As Integer
    Dim Address As AddressStruct
End Structure
```

للوصول إلى عناصر التراكيب المتداخلة

VB.Net Code

```
Dim Salah As Person
Salah.Name = "الصالح"
Salah.Age = 35
Salah.Address.City = "باتنة"
Salah.Address.Country = "الجزائر"
```

السجلات و الطرق

VB.Net Code

```
Structure Person
    Dim Name As String
```

الفصل الثاني : لغة البرمجة

```
Dim Age As Integer
' طريقة تعريف
Sub ShowData()
    Console.WriteLine(Name)
    Console.WriteLine(Age)
End Sub
End Structure
```

الوصول إلى السجل كما يلي :

VB.Net Code

```
Dim Salah As Person
    Salah.Name = "الصالح"
    Salah.Age = 35
    Salah.ShowData()
```

السجلات و المصفوفات

VB.Net Code

```
Structure Person
    Dim Name As String
    Dim Age As Integer
End Structure

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Students(10) As Person
    Students(0).Name = "Ali"
    Students(0).Age = 45

    Students(1).Name = "Ali"
    Students(1).Age = 45

    If Students(3).Equals(Students(4)) Then
        MsgBox("متماتلان الطالبان")
    End If
End Sub
```

المُرَقَمَات Enumerators

تعريف المُرَقَمَات :

وهي أنواع تحتوي على تعبيرات نصية مناظرة لمجموعة من الأرقام، لاستخدامها بدلا من هذه الأرقام عند كتابة الكود، ليصبح أكثر سهولة ووضوحا عند قراءته. وتُعرّف المُرَقَمَات في بداية النموذج أو القالب أو الخلية كالتالي

VB.Net Code

```
Private Enum EnumName
```

الفصل الثاني : لغة البرمجة

```
Exprition = Value1
.....
Exprition_N = ValueN

End Enum
```

مثال : عن مرقم أيام الأسبوع

VB.Net Code

Private Enum WeekDays

```
السبت = 0
Saturday = 0
الأحد = 1
Sunday = 1
الاثنين = 2
Monday = 2
الثلاثاء = 3
Tuesday = 3
الأربعاء = 4
Wednesday = 4
الخميس = 5
Thursday = 5
الجمعة = 6
Friday = 6
```

End Enum

تلاحظ أنه كتبنا نفس القيمة لأكثر من يوم لدلالة على نفس اليوم .
والآن تعال نستخدم هذا المرقم:

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim X, x1 As WeekDays
    X = WeekDays.السبت
    x1 = WeekDays.Saturday
    If X = x1 Then MsgBox("متساويتان")
End Sub
```

ويمكن تعريف المرقم على الصيغة المختصرة التالية:

VB.Net Code

الفصل الثاني : لغة البرمجة

<pre>Private Enum DigitNames Zero One Two Three Four Five Six Seven Eight Nine End Enum</pre>	حيث في هذه الحالة سيُعتبر أول عضو هو القيمة صفر، والثاني هو القيمة 1،.. وهكذا
--	--

المُرَقَمَات كموَشِّرَات **Flags**:

ماذا لو أردت أن تضع في متغيّر قيمتين أو أكثر معا؟
إنّ أهميّة ذلك لا تتّضح في مرَقَم أيّام الأسبوع، ولكنّها ستتّضح بالتأكيد في المرَقَم التالي

VB.Net Code

Enum FileAttribute

عادي
فقط_للقراءة
خفيّ
مضغوط

End Enum

هذا المرَقَم يعبّر عن سمات الملف.. المشكلة أنّ الملفّ يمكن أن يمتلك أكثر من سمة في نفس الوقت، إذ يمكن أن يكون خفيّا وللقراءة فقط.. أو مضغوطا وخفيّا.... إنّ أيّ متغيّر نعرّفه من هذا المرَقَم لن يحمل أكثر من قيمة، فكيف يمكن أن نعبر إذن عن مثل هذه الحالات؟
بسيطة.. أعد تعرّف هذا المرَقَم ليصبح كالتالي

VB.Net Code

<Flags()> Enum FileAttribute

عادي = 1
فقط_للقراءة = 2
خفيّ = 4
مضغوط = 8

End Enum

حيث قمنا بخطوتين أساسيتين:

- استخدمنا السمة <Flags()> في تعريف المرَقَم.

الفصل الثاني : لغة البرمجة

- كما رقمنا التعبيرات بأرقام ثنائية (2⁰، 2¹، 2²، 2³)، وذلك حتى نضمن أن جمع أي رقمين أو أكثر ينتج عنه عدد متفرّد غير قابل للتكرار.. بمعنى أن الرقم 6 مثلا، لا يمكن أن ينتج إلا من جمع 2 + 4، وبهذا فهو يدلّ على ملفّ للقراءة فقط وخفيّ. جرب ما يلي:

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Dim MyFileAttr As FileAttribute  
MyFileAttr = FileAttribute.حفيّ Or FileAttribute.فقط للقراءة  
MsgBox(MyFileAttr.ToString)  
End Sub
```

ستظهر الرسالة تحمل العبارة التالية: "للقراءة فقط، خفي". لاحظ أننا استخدمنا المعامل Or لدمج أكثر من سمة.. أمّا لو شئت التحقق من وجود سمة معيّنة، فعليك استخدام المعامل And كالتالي:

If MyFileAttr And FileAttribute.فقط للقراءة Then

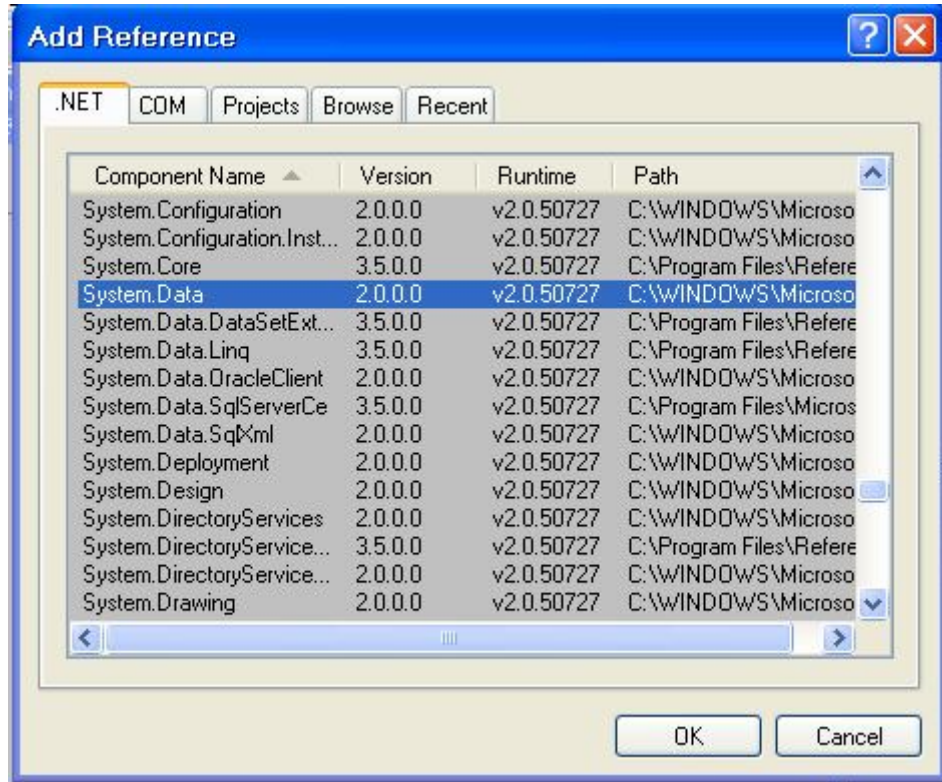
مجالات الأسماء Name Spaces

هو تجميع للملفات المتشابهة و التي تقوم بدور مشترك في مجموعات يضمها مجال واحدة يسمى مجال الأسماء NameSpaces فمثلا : مجال الأسماء الخاص بقواعد البيانات و الملفات ... ولاستدعاء مجال الأسماء الخاص بقواعد البيانات نكتب أعلى الفئة ما يلي

VB.Net Code

```
Imports System.Data.OleDb  
Public Class Form1  
End Class
```

كما يمكنك إضافة مجال أسماء عن طريق القائمة Project ثم اختر Reference



كما يمكنك القيام بعمل مجالات أسماء خاصة بك تحتوي العديد من الإجراءات أو Class الخاصة بك، ويمكنك إضافتها في نفس النموذج وان كنت أحبذ أن تضعها في Modules لوحدها.

لإضافة Modules إلى المشروع أختار Project تم Add Modules بعد أن يتم إضافته اكتب فيه التالي:

```

VB.Net Code
Namespace Test
    Class FileFolder

    End Class
    Class Devices

    End Class
    Class Database

    End Class
End Namespace
    
```

لقد أضفنا ثلاث فئات Classes فئة خاصة بالمجلدات و أخرى بالمحركات و الأخيرة بقواعد البيانات .

و الآن نقوم باستدعاء و اختبار مجال الأسماء الذي تم إنشاؤه من النموذج كما يلي :

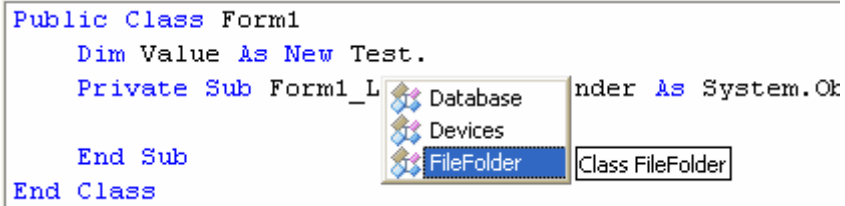
الفصل الثاني : لغة البرمجة

```
Public Class Form1
    Dim Value As New te
    Private Sub Fo
End Sub
End Class
```



و بمجرد كتابة نقطة بعد مجال الأسماء Test الخاص بنا تظهر قائمة لأعضاء لهذا المجال

```
Public Class Form1
    Dim Value As New Test.
    Private Sub Form1_Load(sender As System.Ok
End Sub
End Class
```

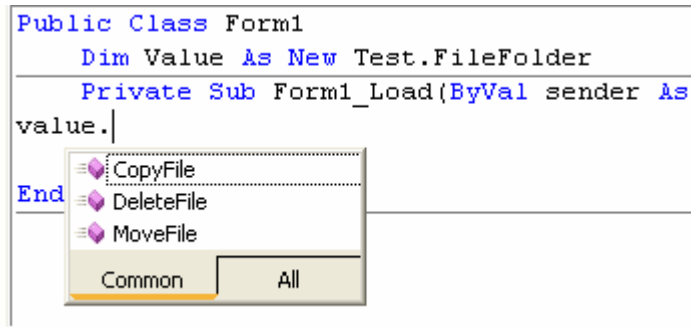


مجال الأسماء الخاص بنا فارغ علينا أن نكتب فيه بعض التعليمات كما يلي :

```
VB.Net Code
Namespace Test
    Class FileFolder ' فئة خاصة بالمجلدات
        Public Sub CopyFile() ' إجراء خاص بنسخ الملفات
        End Sub
        Public Sub DeleteFile() ' إجراء خاص بحذف الملفات
        End Sub
        Public Sub MoveFile() ' إجراء خاص بنقل الملفات
        End Sub
    End Class
    Class Devices
    End Class
    Class Database
    End Class
End Namespace
```

ومن نموذجنا يمكننا أن نرى التالي

```
Public Class Form1
    Dim Value As New Test.FileFolder
    Private Sub Form1_Load(ByVal sender As
value.|
End
```



الفصل الثاني : لغة البرمجة

و لكن مازلت هذه الإجراءات لم تقم بوظائفها تحتاج إلى إضافة كود و تعليمات لكل منها.

VB.Net Code

```
Namespace Test
Class FileFolder
Public Sub CopyFile(ByVal FileFrom As String, ByVal FileTo As String)
My.Computer.FileSystem.CopyFile(FileFrom, FileTo, True)
End Sub
Public Sub DeleteFile(ByVal FileName As String)
My.Computer.FileSystem.DeleteFile(FileName)
End Sub
Public Sub MoveFile()

End Sub
End Class
Class Devices

End Class
Class Database

End Class
End Class
End Namespace
```

الآن صار بإمكاننا أن نقوم بنسخ ملف و حذفه

VB.Net Code

```
Dim Value As New Test.FileFolder
Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click
Value.CopyFile("C:\Mytxt.txt", "d:\mytxt.txt")
End Sub
```

الوحدات النمطية Modules

مجرد قالب يحتوي على مجموعة من المتغيرات والإجراءات الفرعية والدوال بحيث يمكن وضع الإجراءات التي تؤدي وظائف متشابهة أو مرتبطة، في قالب له اسم واضح

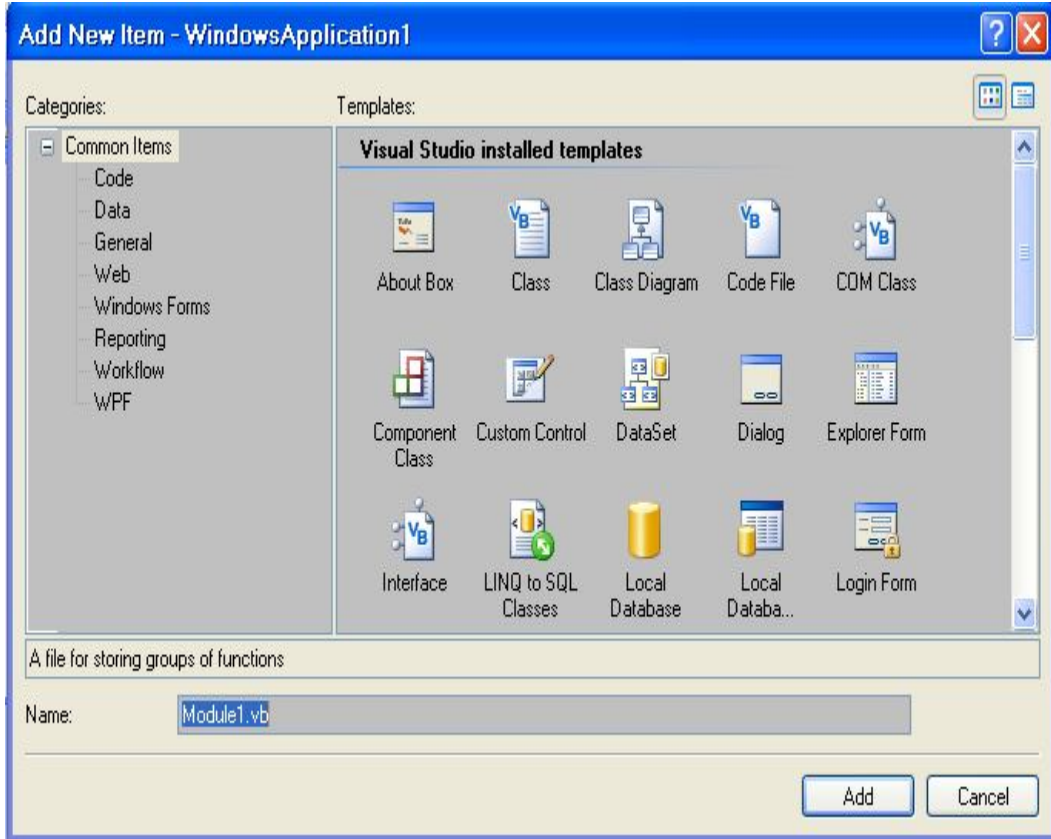
إنشاء وحدة نمطية Creat Module

لإضافة وحدة نمطية إلى البرنامج اتبع الخطوات التالية :

- اختر القائمة مشروع Project
- اختر الأمر Add Modules

الفصل الثاني : لغة البرمجة

- يظهر أمامك مربع حوار إضافة عنصر Add Items حيث يكون العنصر المحدد هو Module
- غير اسم الوحدة من خانة Name إلى FirstModule أو اترك الاسم الافتراضي Module1.vb
- ثم اضغط Ok



و هكذا تكون قد أضفت Module جديد إلى مشروعك و يظهر كما في الشكل :

```
VB.Net Code
Module firstModule
End Module
```

و الآن نكتب دالة الجمع كما يلي :

```
VB.Net Code
Module firstModule
Function Sum(ByVal x As Integer, ByVal Y As Integer) As Integer
    Sum = x + Y
End Function
End Module
```

يمكن أن تحتوى الوحدة عدة وحدات برمجية فقد ترغب تقسيمها إلى وحدات حسب وظائفها

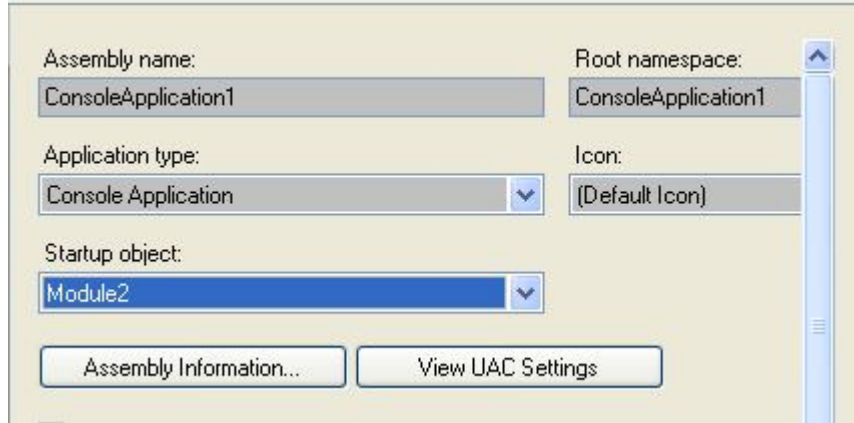
```
VB.Net Code
Module Settxt
    Sub Main()
        .....
    End Sub
End Module
Module Gettxt
    Sub Main()
        .....
    End Sub
End Module
```

الإجراء Sub Main

له طابع خاص يتم تنفيذه مع بداية تنفيذ البرنامج
كما يمكنك كتابة أكثر من إجراء Sub Main

```
VB.Net Code
Module Module1
    Sub Main()
        Console.WriteLine("2 Programme")
    End Sub
End Module
Module Module2
    Sub Main()
        Console.WriteLine("1 Programme")
    End Sub
End Module
```

ولكن كيف يتم اختيار الإجراء الذي يتم تنفيذه أولاً ؟
الإجراء الذي ستحدده أنت من خانة Startup Object كما يلي :



الإجراء الأول هو الموجود في الوحدة البرمجية 2

الإجراء Sub New

هو إجراء يتم تنفيذه بمجرد إنشاء نسخة من الكائن التابع له

ما هو الإجراء الأسبق New أو Main ؟

طبعا الإجراء New

```

VB.Net Code
Module Module1
Sub Main()
    Console.WriteLine("Sub Main Play Second")
    Console.ReadLine()
End Sub

Sub New()
    Console.WriteLine("Sum New Play First")
    Console.ReadLine()
End Sub
End Module
    
```

ملاحظة : لتجريب الإجراء Sub Main و Sub New استعمل التطبيق Console

جمل التحكم في المسار Flow Control Statments

أولا : الجملة الشرطية If ... Then

تقوم هذه الجملة من التحقق من الشرط فإن كان صحيحا تنفذ المطلوب و إن كان خاطئ تفقز

إلى End if و لا تنفذ شيئا. و هذه الصيغة العامة لها

الصيغة العامة	
If condition [Then]	' إذا تحقق الشرط فإنه
[statements]	' يتم تنفيذ هذه التعليمات
End If	' نهاية الشرط
Or:	
If condition Then [statements]	' جملة الشرط في سطر واحد

الفصل الثاني : لغة البرمجة

ما هو الشرط هو تعبير منطقي يحتمل الصح أو الخطأ فيعطي نتيجة تكون True أو False تحتوي الشروط على معاملات مختلفة
معاملات المقارنة :

=	يساوي
>	أكبر
<	أصغر
<>	يختلف
>=	أكبر أو يساوي
<=	أصغر أو يساوي

المعاملات المنطقية

و	And	يكون صحيحا إذا تحقق الشرطان معا
أو	Or	يكون صحيحا إذا تحقق شرط واحد يكفي
لا	Not	ينفي
	Xor	يكون صحيحا إذ كان الشرطان غير محققين
وكذا	Andalso	إذا تحقق الشرط الأول يكمل العملية و إلا يغادر الحلقة
	OrElse	

و معاملات آخر

إنه	IS
إنه ليس	IsNot
نوع إنه	Typeof Is
مثل	Like

مثال

VB.Net Code

```
Dim Moyen As Double = Val(InputBox("الطالب معدل ادخل"))
If Moyen >= 10 Then
    MsgBox("ناجح الطالب")
End If
-----Or
If Moyen >= 10 Then MsgBox("ناجح الطالب")
```

جملة الشرط و الاستثناء If Then ... Else

تقوم هذه الجملة بالتحقق من الشرط فإذا كان صحيحا تنفذ التعليمات المطلوبة و إلا تنفذ التعليمات المطلوبة الخاصة بعدم تحقق الشرط

If condition [Then]	' إذا تحقق الشرط فإنه
[statements]	' يتم تنفيذ هذه التعليمات
Else	' و إلا
[statements]	' نفذ هذه التعليمات
End If	' نهاية الشرط
Or:	
If condition Then [statements] Else [statements]	' جملة الشرط في سطر واحد

مثال : إذا تحصل الطالب على المعدل فيكون ناجحاً و إلا فهو راسب

 VB.Net Code

```
Dim Moyen As Double = Val(InputBox("الطالب معدل ادخل"))
If Moyen >= 10 Then
    MsgBox("ناجح الطالب")
Else
    MsgBox("راسب الطالب")
End If
----- Or
If Moyen >= 10 Then MsgBox("ناجح الطالب") Else MsgBox("راسب الطالب")
```

يمكنك تنفيذ أكثر من جملة أو تعليمة في حالة تحقق الشرط أو عدمه كما في المثال

 VB.Net Code

```
Dim Moyen As Double = Val(InputBox("الطالب معدل ادخل"))
If Moyen >= 10 Then
    MsgBox("ناجح الطالب")
    MsgBox("مبروك ألف تهانينا")
Else
    MsgBox("راسب الطالب")
    MsgBox("النجاح لعدم نتأسف")
End If
----- Or
If Moyen >= 10 Then MsgBox("ناجح الطالب") : MsgBox("مبروك ألف تهانينا") Else
MsgBox("راسب الطالب") : MsgBox("النجاح لعدم نتأسف")
```

جملة الشرط المتفرعة ...else ...elseif ...

إذا تحقق الشرط الأول تنفذ التعليمات الخاصة به . و إلا تتحقق من الشرط الثاني إذا كان

محققاً تنفذ التعليمات الخاص بالشرط الثاني ... و هكذا و إذا لم يتحقق أي شرط تنفذ

التعليمات الموافقة

If condition1 Then ' إذا تحقق الشرط الأول
statements1 ... ' نفذ التعليمة الأولى
Elseif condition2 ' و إلا إذا تحقق الشرط الثاني
statements2... ' نفذ التعليمة الثانية
Else ' و إذا لم يتحقق أي شرط
statements3 ... ' نفذ التعليمة 3
End If

ملاحظة : يستحسن الترتيب المنطقي للشروط حتى تحصل على النتائج الصحيحة

مثال :كتابة الأرقام بالحروف حسب الرقم المدخل

VB.Net Code

```
Dim Number As Integer = Val(InputBox("10 من أقل رقما ادخل"))
If Number = 0 Then
    MsgBox("صفر")
Elseif Number = 1 Then
    MsgBox("واحد")
Elseif Number = 2 Then
    MsgBox("اثنان")
Elseif Number = 3 Then
    MsgBox("ثلاثة")

Elseif Number = 9 Then
    MsgBox("تسعة")
Else
    MsgBox("معروف غير رقم")
End If
```

جمل الشرط المتداخلة

يتحقق المترجم ما إذا كان الشرط المحدد في الجملة الخارجية محققا سيتم بذلك الدخول إلى جملة الشرط الداخلية و يتحقق من الشرط الثاني الموجود في الجملة الداخلية فإن كان محققا نفذ الشرطين معا و إلا سيخرج من البرنامج . انظر المثال

VB.Net Code

```
Dim IntX As Integer = 1
If IntX = 1 Then ' الخارجية الحلقة شرط تحقق
    MsgBox.Show("One") ' مايلي ينفذ
    IntX = 2 ' المتغير قيمة أصبحت 2
    If IntX = 2 Then ' الداخلية الجملة شرط تحقق
        MsgBox.Show("Two") ' مايلي ينفذ
    End If
End If
```

استخدام المعاملات :

1. المعامل يختلف (<> Using Not Equal To)

VB.Net Code

```
Dim Password As String = "Salah"  
If TextBox1.Text <> Password Then  
    MsgBox("خاطئة مرور كلمة")  
End If
```

2 المعامل و And

VB.Net Code

```
Dim UserName As String = "Salah", Password As Integer = 1975  
If TextBox1.Text = UserName And TextBox2.Text = Password Then  
    MsgBox("صحيحة مرور كلمة")  
End If
```

4. المعامل أو Or

VB.Net Code

```
Dim UserName As String = "Salah", Password As Integer = 1975  
If TextBox1.Text = UserName or TextBox2.Text = Password Then  
    MsgBox("صحيحة مرور كلمة")  
End If
```

في المثال الأول يجب أن يكون الشرطين محققين معا . وفي المثال الثاني يكفي تحقق أحد الشرطين.

5. و كذا AndAlso

يستعمل لاختصار التحقق من الشروط عديمة الفائدة.. فلو كان الشرط الأول خاطئا، فلن يتم التحقق من باقي الشروط.. فمثلا لو جرّبت

VB.Net Code

```
Dim UserName As String = "Salah", Password As Integer = 1975  
If TextBox1.Text = UserName AndAlso TextBox2.Text = Password Then  
    MsgBox("صحيحة مرور كلمة")  
End If
```

إذا لم يتحقق الشرط الأول فلا داعي لاختبار بقية الشروط.

جملة اختيار الحالة Select Case

تستعمل بديلا عن جملة الشرط المتفرعة وذلك تسهيلا في كتابة الكود .
وهذا عندما تكون كل المقارنات بين متغير واحد، ومجموعة من القيم من نفس نوع المتغير..
ويوضع اسم المتغير بعد جملة Select Case، بينما توضع القيم المختلفة واحدة بواحدة بعد
كلمة Case، متبوعة بالمقطع الذي سيتم تنفيذه في حالة تحقق التساوي:

الصيغة العامة

```
Select [ Case ] testexpression  
[ Case expressionlist  
[ statements ] ]  
[ Case Else  
[ elstatements ] ]  
End Select
```

تعبير Select Case

قيمة 1 Case

مقطع 1 (يتم تنفيذه حينما يساوي التعبير القيمة 1)

قيمة 2 Case

مقطع 2 (يتم تنفيذه حينما يساوي التعبير القيمة 2)

.....
.....

Case Else

مقطع_أخير (يتم تنفيذه حينما لا يساوي التعبير أي قيمة مما سبق)

End Select

مثال : كتابة الأرقام بالحروف

نحتاج إلى أداتين Textbox و Label

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
Dim Number As Integer = Val(TextBox1.Text) ' 10 من أقل رقما ادخل  
Select Case Number  
Case 0  
Label1.Text = "صفر"  
Case 1  
Label1.Text = "واحد"  
Case 2  
Label1.Text = "اثنان"  
.. ....
```



```
Case Else
    Label1.Text = "رقم ليس"
End Select
End Sub
```

استعمال معاملات المقارنة

نضيف أداة الزر Button و صندوق نص TextBox و لافتة عنوان Label
ثم نكتب في حدث نقر الزر

VB.Net Code

```
Dim Moyen As Double = TextBox1.Text
Select Case Moyen
    Case Is > 15
        Label1.Text = "تهاني"
    Case Is >= 13
        Label1.Text = "تشجيع"
    Case Is >= 11
        Label1.Text = "شرف لوحة"
    Case Is >= 9
        Label1.Text = "لاشيء"
    Case Else
        Label1.Text = "مرضية غير نتائج"
End Select
```

تحديد مجموعة قيم

VB.Net Code

```
Select Case TextBox1.Text
    Case "A", "B", "C"
        Label1.Text = "حرف هذا"
    Case "1", "2", "3"
        Label1.Text = "رقم هذا"
    Case Else
        Label1.Text = "معرف غير"
End Select
```

الحلقات التكرارية Loop Structures

أولا : الحلقة من ... إلى For ... Next

تستعمل لتكرار تعابير أو عمليات عدد محدد من المرات . وهذا ما يجعل العمل سهلا و
مريحا للمستخدم و تتطلب تحديد قيمة البداية و قيمة النهاية و قيمة القفزة Step

```
For counter [ As datatype ] = start To end [ Step step ]  
  [ statements ]  
  [ Exit For ]  
  [ statements ]  
Next [ counter ]
```

For [قيمة_الزيادة Step] نهاية To بداية = اسم_متغير
 {المقطع الذي سيتم تكراره}
Next [اسم_المتغير]

مثال : كتابة الأعداد من 0 إلى 5 مرة واحدة

VB.Net Code

```
For i As Integer = 0 To 5  
  Console.WriteLine("{0}", i)  
Next
```

استعمال القفز Step

VB.Net Code

```
For i As Integer = 0 To 10 Step 2 ' طول القفزة 2  
  Console.WriteLine("{0}", i)  
Next
```

الرقم 0
الرقم 2
الرقم 4
الرقم 6
الرقم 8
الرقم 10

الخروج من الحلقة Exit For

نفس المثال السابق و لكن عند الوصول إلى الرقم 4 نخرج من الحلقة

VB.Net Code

```
For i As Integer = 0 To 10 Step 2  
  Console.WriteLine("{0}", i)  
  If i = 4 Then Exit For  
Next
```

الرقم 0
الرقم 2
الرقم 4

الحلقات و المصفوفات

الفصل الثاني : لغة البرمجة

VB.Net Code

```
Dim MyArray() As String = {"A", "b", "c"}
For i = 0 To MyArray.Count - 1
    Console.WriteLine(MyArray(i))
Next
```

A
b
c

الحلقات المتداخلة

تكون جدول الضرب من 1 إلى 9 في هذا المثال :

VB.Net Code

```
For i As Integer = 1 To 9
    For j As Integer = 1 To 9
        Console.WriteLine(i & "x" & j & " = " & i * j)
    Next
Next
```

ثانيا : الجملة نفذ مرارا DoLoop

تستعمل لتكرار التعابير أو العمليات عدة مرات حتى تحقق الشرط .

الصيغة العامة 

```
Do { While | Until } condition
    [ statements ]
    [ Exit Do ]
    [ statements ]
Loop
-or-
Do
    [ statements ]
    [ Exit Do ]
    [ statements ]
Loop { While | Until } condition
```

أ- جملة نفذ بينما Do While

تتحقق ما إذا كان الشرط محققا فيقوم بتنفيذ التعليمات التالية . و عندما يصل إلى نهاية الحلقة Loop يعود إلى البداية و كلما تحقق الشرط تكرر العملية . و في حالة عدم تحقق الشرط يتم الخروج من الحلقة . ولها صيغتان

الصيغة الأولى 

```
Do
    statements()
    [Exit Do]
    statements()
Loop While Condition
```

الفصل الثاني : لغة البرمجة

Do

نفذ التعليمات

اخرج من الحلقة [Exit Do]

تعليمات

اخرج إذا تحقق الشرط Loop While

تنفذ الجملة مرة واحدة قبل التحقق من الشرط ثم تنفذ الشرط فإن تحقق تواصل و إلا سيتم الخروج و بذلك تضمن تنفيذ الجملة على الأقل مرة

VB.Net Code

```
Dim i As Integer = 0
Do
    i = i + 1
    Console.WriteLine(i)
Loop While i < 5
```

الصيغة الثانية 

Do

statements()

[Exit Do]

statements()

Loop While Condition

نفس المثال السابق و لكن نضع الشرط الآن في الأول

VB.Net Code

```
Dim i As Integer = 0
Do While i < 5
    i = i + 1
    Console.WriteLine(i)
Loop
```

سنحصل على نفس الناتج في كلا الحالتين

1
2
3
4
5

الخروج من الحلقة :

VB.Net Code

```
Dim i As Integer = 0
Do While i < 10
    i = i + 1
    Console.WriteLine(i)
    If i = 5 Then
        Exit Do
```

End If
Loop

ب- جملة نفذ حتى Do until

تنفذ هذه الجملة التعليمات طالما لم يتحقق الشرط و عندما تصل إلى الشرط المحدد تتوقف
و لها صيغتان

 الصيغة الأولى

Do

التعليمات نفذ ' Statements()

[Exit Do]

statements()

Loop Until Condition ' الشرط تحقق غاية إلى

تنفذ الجملة مرة واحدة قبل التحقق من الشرط ثم تنفذ الشرط فإن تحقق تواصل و إلا سيتم
الخروج و بذلك تضمن تنفيذ الجملة على الأقل مرة

 VB.Net Code

```
Dim I As Integer
```

```
Do
```

```
I = I + 1
```

```
Console.WriteLine(I)
```

```
Loop Until I = 5
```

 الصيغة الثانية

Do Until Condition ' الشرط تحقق غاية إلى

التعليمات نفذ ' Statements()

[Exit Do]

statements()

Loop

نفس المثال السابق و لكن نضع الشرط الآن في الأول

 VB.Net Code

```
Dim I As Integer
```

```
Do Until I = 5
```

```
I = I + 1
```

```
Console.WriteLine(I)
```

```
Loop
```

و في كلا الحالتين سيتم طبع الأعداد من 1 إلى 5.

الصيغة اللانهائية

Do	' طالما تحقق الشرط '
[statements]	' نفذ التعليمات '
If Statement Then Exit Do	' إذا تحقق الشرط اخرج من الحلقة '
[statements]	' تعليمات أخرى '
Loop	' نهاية الحلقة '

نفس المثال السابق

 VB.Net Code

```
Dim I As Integer
Do
    I = I + 1
    Console.WriteLine(I)
    If I = 5 then Exit Do
Loop
```

ثالثا : حلقة التكرار بينما While

وهي مثل جملة التكرار Do while مع اختلاف في المصطلحات .

While condition	' طالما تحقق الشرط '
[statements]	' نفذ التعليمات '
[Exit While]	' اخرج من الحلقة '
[statements]	
End While	' نهاية الحلقة '

نفس المثال السابق

 VB.Net Code

```
Dim I As Integer = 0
While I < 5
    I = I + 1
    Console.WriteLine(I)
End While
```

رابعا : جملة مهما كان For Each

تستعمل لتكرار تعليمات أو عمليات عن عناصر مصفوفة أو مجموعة أو كائنات ...
تتطلب منك تحدد اسم العنصر ثم نوعه و المجموعة التي ينتمي إليه

```
For Each element [ As datatype ] In group
    [ statements ]
    [ Exit For ]
    [ statements ]
Next [ element ]
```

الشرح :

المجموعة التي تنتمي إليها In نوع العنصر As العنصر (مهما كان)

مقطع من الكود

اخرج من الحلقة

مقطع آخر من الكود

Next

أ- مع الأدوات

VB.Net Code

```
For Each c As Control In Me. Controls
    If TypeOf c Is TextBox Then c.Text = ""
    If TypeOf c Is ComboBox Then c.Text = ""
Next
```

مهما كانت الأداة c من أدوات النموذج إذا كان نوعها صندوق نص او صندوق سرد سيتم تفرغ محتواها عند نقر الزر

ب- مع المصفوفات Arrays

VB.Net Code

```
Dim Names() As String = {"Salah", "Ali", "Said"}
For Each N As String In Names
    Console.WriteLine(N)
Next
```

مهما كان الاسم N من مصفوفة الأسماء سيتم طباعته.

ج- مع المجموعات Collection

VB.Net Code

```
Dim col As New Collection()
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    col.Add("Salah", "S")
    col.Add("Ali", "A")
    col.Add("Med", "M")
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    For Each obj As Object In col
        Dim str As String = CType(obj, String)
        Console.WriteLine(str)
    Next
End Sub
```

خامسا : اذهب إلى GoTo

تتطلب هذه الجملة تحديد اللافتة أو مكان داخل الإجراء لتنتقل إليه .

```
VB.Net Code

If TextBox1.Text = "" Then
    GoTo Line1
line1:
    MsgBox("النص صندوق في البيانات ادخل")

End If
```

إذا كان صندوق النص فارغا فاذهب إلى GoTo السطر Line1 ليعرض رسالة تخبرك بأن النص فارغا لتدخل فيه البيانات.

مثال آخر


```
VB.Net Code

Dim I As Byte = 0
Looping:
    I = I + 1
    Console.WriteLine(I)
    If I < 5 Then GoTo Looping
```

إذا كان I أصغر من 5 يطبع الأعداد من 1 إلى 5 إما إذا كان غير ذلك ينتقل إلى سطر الإعادة Looping ليكرر التعليمات و هكذا ...

سادسا : اختر قيمة Choose

تقوم باختيار قيمة وحيدة من مجموعة قيم وهذا حسب الرقم الترتيبي Index المحدد في الدالة

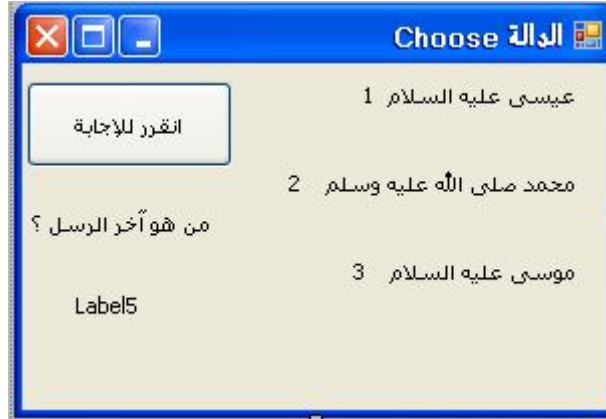
 الصيغة العامة

```
Dim result As Object = Choose(index, item_1[, item_2[... item_n]])
Choose(..... القيمة 2 , القيمة 1 , الرقم الترتيبي)
```

Index : رقم يمثل القيمة التي تريد اختيارها فمثلا : Index يمثل 2 أي يعرض القيمة 2 إذا اخترت الإجابة رقم 2 التي كتبت في خانة اختيار radiobutton2 من بين القيم الأخرى ستحصل على الإجابة الصحيحة .

مثال :

صمم النموذج على النحو التالي و اكتب



VB.Net Code

```
Dim x As String = InputBox("الصحيحة الاجابة يمثل الرقم اختر")
Dim Result As Object = Choose(x, Label1.Text, Label2.Text, Label3.Text)
If x = 2 Then
    Label5.Text = Result
Else
    MsgBox("خاطئة اجابتك")
End If
```

سابعاً: الدالة IIF

تستعمل اختصاراً الجملة الشرط IfThen....Else

الصيغة العامة 

```
Dim result As Object = Iif(expression, truePart, falsePart)
IIF ( ما يعرضه في حالة الخطأ , ما يعرضه في حالة الصواب , تعبير منطقي )
```

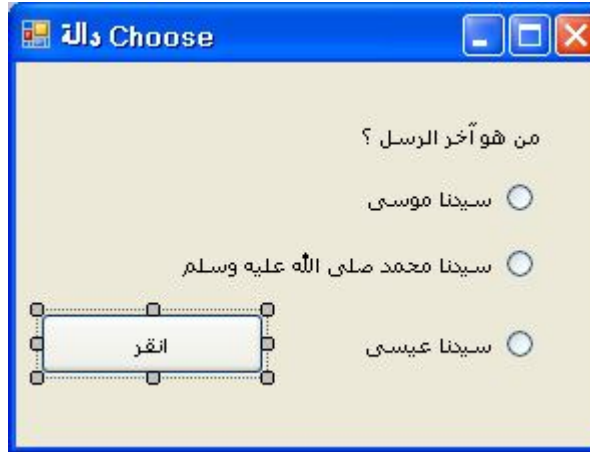
مثال :

تدريب نموذجي :

1. قم بإنشاء مشروع جديد من قائمة File اختر New Project

يظهر مربع حوار مشروع جديدة انقر Ok

2. قم بتصميم نافذة التصميم كما يبدو في الشكل



3. ثم في حدث نقر الزر اكتب التعليمات التالية

VB.Net Code

```
Dim x As String  
x = If(RadioButton2.Checked, "صحيحة إجابة", "خاطئة إجابة")  
MsgBox(x)
```

إذا تم اختيار الخانة الثانية تعرض رسالة إجابة صحيحة و إلا تعرض رسالة إجابة خاطئة

الإجراءات و الدوال Subs & Functions

الإجراء : هو عبارة عن مجموعة من الأوامر و التعليمات البرمجية ذات علاقة فيما بينها التي تقوم بمهمة معينة

ويتم وضع تعريف الإجراء داخل مقطع أيّ خلية End Class أو قالب Module.. End Module، ولكن ليس داخل أي إجراء فرعيّ أو دالة.

لماذا نعرف الإجراءات ؟

كثيرا ما نكرر كتابة مقطع من الكود في مواضع مختلفة، حتّى إنك تتسائل في ملل: ألا توجد وسيلة تريحني من تكرار كتابة هذه الأجزاء كل مرّة ؟

الحل هو الإجراءات: كل ما ستفعله هو أنّك ببساطة ستجمع الجزء الذي يتكرّر معك كثيرا في برامجك، وتضعه في إجراء بالاسم الذي يعجبك، ثم تناديه بعد ذلك، من أيّ موضع من البرنامج ولو اكتشفت خطأ أو أردت تعديلا في الأداء، فما عليك سوى تعديل هذا الإجراء مرّة واحدة!

التصريح بالإجراء و الدالة

صيغة الإجراء الفرعي Sub procedures



```
Sub SubName(ByVal [arguments])  
    Sub statements  
    [Return value]  
End Sub
```

[معاملات الإجراء] اسم الإجراء Sub [مجال الإجراء]

مقطع من الكود

End Sub

صيغة الدالة Function:



```
Function FunctionName(ByVal [arguments]) As Type  
    function statements  
    [Return value]  
End Function
```

[القيمة_المعادَة As] [معاملات الدالة] اسم الدالة Function [مجال الدالة]

مقطع من الكود

End Function

حيث مجال الإجراء أو الدالة هو واحدة من هذه الكلمات:

1- Public: ليصبح الإجراء متاحا للاستخدام من أي موضع في مشروعك، أو من أي

مشروع آخر يضيف مرجعا لمشروعك.

2- Friend: ليصبح الإجراء متاحا للاستخدام من أي موضع في مشروعك، لكن ليس

من أي مشروع آخر.

3- Private: ليصبح الإجراء متاحا للاستخدام، فقط من داخل القالب Module أو

الخلية Class أو النموذج Form أو السجل Structure الذي ينتمي إليه.

4- Protected.

5- Protected Friend.

أمّا معاملات الإجراء أو الدالة، فهي مجموعة من المتغيرات تعرّفها بين قوسين (بدون كلمة

Dim)، لاستقبال البيانات التي تحتاجها لتنفيذ وظيفة الإجراء أو الدالة.. ويمكن أن تكون هذه

المعاملات من أي نوع: رقمية أو نصية أو زمنية أو كائنات أو حتى مصفوفات.

مثال:

الفصل الثاني : لغة البرمجة

الإجراء الحالي يعرض تاريخ اليوم في رسالة..

```
VB.Net Code
Public Class Form1
    Sub ShowDate() ' إجراء يعرض تاريخ اليوم
        MsgBox(Now())
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        ShowDate() ' استدعاء الاجراء
    End Sub
End Class
```

تعريف الدوال :

مجموعة من الأوامر و التعليمات البرمجية مثل الإجراءات تقوم بعمليات محددة حسابية أو منطقية أو غيرها و تشمل على معاملات و تعود لنا بمجموعة من المخرجات ، يمكن استدعاؤها بواسطة كتابة اسمها
مثال : عن دالة تعود لنا بنتائج جمع عددين

```
VB.Net Code
Function Sum(ByVal x As Integer, ByVal Y As Integer) As Integer
    Sum = x + Y
    Return Sum
End Function
```

و لاستدعاء هذه الدالة نكتب

```
VB.Net Code
Private Sub Button1_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) _
    Handles Button1.Click
    MsgBox(Sum(3, 2)) ' 5
End Sub
```

ملاحظات على استدعاء الدوال والإجراءات:

- يجب دائما أن توضع المعاملات بين قوسين.
- يجب إرسال العدد الذي تنتظره الدالة من المعاملات، لا أكثر ولا أقل.
- يجب إرسال المعاملات بالترتيب الذي تريده الدالة أو الإجراء (كما سيظهر لك في تلميح الشاشة بعد كتابة اسم الدالة أو الإجراء).

- يجب أن تتوافق القيم التي ترسلها للمعاملات، مع القيم التي تقبلها هذه المعاملات، فمثلاً: لن يقبل منك VB أن ترسل تاريخاً أو نصاً لمعامل تم تعريفه على أنه عدد صحيح.

المعاملات Parameters

هي متغيرات توضع بين قوسين بعد اسم الإجراء أو الدالة

```

Public Class Form1
    Sub MySub(ByVal x As Integer, ByRef y As String, ByVal z As Byte)
    End Sub

```

الصيغة العامة لكتابة المعامل

VB.Net Code

```
[ <AttributeList> ] ParameterName [ ( ) ] [ As ParameterType ] [= DefaultValue ]
```

AttributeList : قائمة الصفات

و هي تمثل قائمة من الصفات التي تميز المعامل و هي تحدد عموماً الطريقة التي يتم تمرير بها القيم عند استدعاء الإجراء أو الدوال .

```

Public Class Form1
    Sub MySub(

```

ParameterName : اسم المتغير الحالي (المعامل)

ParameterType : نوع بيانات المتغير المحلي (المعامل) قد تكون عدد صحيح

Integer أو نص ...

DefaultValue : القيمة الافتراضية

تكون مطلوبة بالنسبة للمعاملات الاختيارية Optional Parameters. تمثل بثابت في

المتغيرات أو حتى

الفصل الثاني : لغة البرمجة

لاشي Nothing مع الفئات أو المصفوفات ... انظر المثال التالي :

VB.Net Code

```
Sub Mysub(Optional ByVal x As Integer = 1)
```

```
End Sub
```

ما هي فائدة المعاملات؟:

إنّ لها فائدة جمّة، فهي تمكّنك من التحكم في أداء الإجراء.. فمثلا، الإجراء الفرعيّ "عرض التاريخ" يعرض تاريخ اليوم فقط.. لكن افترض أنّك تريد أن تعرض تاريخ يوم آخر، وليكن يوم ميلاد شخص ما، فماذا تفعل؟ في هذه الحالة أنت تحتاج لإرسال معامِل لهذا الإجراء لتحديد التاريخ الذي تريده.. انظر لهذا التعديل:

VB.Net Code

```
Sub ShowDate(ByVal BirthDate As Date)
```

```
MsgBox(BirthDate)
```

```
End Sub
```

والآن لاستخدام هذا الإجراء يمكنك كتابة التالي:

```
ShowDate(#2/9/1960#)
```

كما يمكنك إرسال القيمة من خلال متغيّر كالتالي:

VB.Net Code

```
Dim MyBirthDate As Date = #2/9/1960#
```

```
ShowDate(MyBirthDate)
```

تمرير المعاملات

إليك الجدول التالي يبين لنا طرق تمرير المعاملات

المعامل	الدور
ByVal	عند إرسال متغيّرًا بالقيمة تنسخ قيمة فإنّ أيّ تغيير يحدث لهذا المعامل، لا يؤثر أبداً على المتغيّر الأصليّ
ByRef	عند إرسالك متغيّرًا بالمرجع ليكون معاملاً إجراء يمكن تغيير و تعديل قيمة المتغيرات المرسلّة .
Optional	عند إرسال متغيّرًا اختياريًا يمكنك أن تختار أيّ قيمة له و يشترط أن تخصص له قيمة افتراضية
ParamArray	عند إرسال متغيّرًا بمصفوفة لإجراء يستقبل مصفوفة من القيم

المعامل بالقيمة ByVal

في المثال التالي نسند القيمة 12 للمتغير A ثم نستدعي الأجراء DisplayDouble لتضعيف المتغير يعطي الناتج 24 ثم نعود لقيمة المتغير الأصلية الابتدائية هل يبقى محافظة على قيمته الأصلية 12 أم يكتسب القيمة الجديدة 24؟
لاحظ أنه بقي محافظا على قيمته الأصلية 12 و هذا هو دور المعامل المعرف ب ByVal .

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A As Integer = 12
    Debug.WriteLine("Main: " & A) '12
    DisplayDouble(A) '24
    Debug.WriteLine("Main: " & A) '12
End Sub

Private Sub DisplayDouble(ByVal X As Integer)
    X *= 2
    Debug.WriteLine("DisplayDouble: " & X)
End Sub
```

المعامل بالمرجع ByRef

على عكس المعامل بالقيمة فإن المعاملات بالمرجع ByRef لا تبقى محافظة على قيم المتغيرات الأصلية بل تتغير وتكتسب قيما جديدة بعد التعديل انظر نفس المثال السابق و لكن تم تعريف المعامل بالمرجع.

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A As Integer = 12
    Debug.WriteLine("Main: " & A) ' 12
    DisplayDouble(A) '24
    Debug.WriteLine("Main: " & A) ' 24
End Sub

Private Sub DisplayDouble(ByRef X As Integer)
    X *= 2
    Debug.WriteLine("DisplayDouble: " & X)
End Sub
```

مصفوفة المعاملات:

إذا أردت أن تكتب دالة المتوسط الحسابي ستواجهك مشكلة : كم عدد العوامل ؟
و لكن نريدها أن تكون عامة بحيث يمكنني أن أرسل لها أي عدد من المعاملات، لحساب
متوسطها الحسابي. فما هو الحل ؟
الحل : استعمال معامل المصفوفات ParamArray بحيث يمكنك إرسال أي عدد من
المعاملات في مصفوفة مرنة بطول غير محدود انظر المثال :

VB.Net Code

```
Function Avg(ByVal ParamArray Numbers() As Integer) As Double
    Dim I As Integer, Sum As Integer
    For I = 0 To Numbers.GetUpperBound(0)
        Sum += Numbers(I)
    Next
    Return Sum / Numbers.Length
End Function
```

والآن جرّب ما يأتي في حدث ضغط أي زر

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    MsgBox(Avg(2, 3, 4, 5, 7, 3))'4
End Sub
```

وهناك طريقة أخرى لاستدعاء هذه الدالة، بإرسال مصفوفة لها كمعامل

VB.Net Code

```
Dim No() As Integer = {2, 3, 4, 5, 7}
MsgBox(Avg(No)) '4.2
```

المعامل المصفوفة:

هناك حل آخر لكتابة دالة المتوسط الحسابي، وهو إرسال الأعداد التي تريدها في مصفوفة
عادية هذه المرة (بدلاً من مصفوفة المعاملات).. جرّب ما يلي

VB.Net Code

```
Function Avg(ByVal Numbers() As Integer) As Double
    Dim I As Integer, Sum As Integer
    For I = 0 To Numbers.GetUpperBound(0)
        Sum += Numbers(I)
    Next
    Return Sum / Numbers.Length
End Function
```


كل ما حدث هو أن أزلنا كلمة ParamArray من تعريف المعامل Numbers()، وإن كنا تبعنا اسم المعامل بقوسين فارغين، لنعبّر عن كونه مصفوفة مرنة، تستطيع استقبال أيّ مصفوفة بأيّ حجم. ولكنّ هذا التعديل يستدعي حدوث اختلاف في طريقة استدعاء الدالة، فستصير كالتالي:

VB.Net Code

```
Dim X() As Integer = {2, 3, 4, 5, 7}
MsgBox(Avg(X))
```

المعاملات الاختيارية Optional Arguments

نقوم بتعريف دالة لحساب الأسس، بمنحها العدد الذي يمثل الأساس، والعدد الذي يمثل الأسّ، وسنجعل هذا الأخير اختيارياً، بحيث لو لم نرسله عند الاستدعاء، تكون قيمته الافتراضية 2، مما يعني حساب مربع العدد

VB.Net Code

```
Function Exps(ByVal No As Double, Optional ByVal Ex As Double = 2) As Double
Return (No ^ Ex)
End Function
```

والآن يمكنك تجربة كل التعبيرات التالية، فكلّها ممكنة

VB.Net Code

```
Dim X As Double = Exps(2, 3) '8
X = Exps(3, 3) '27
X = Exps(2) '4
```

ملاحظة :

ويمكن تعريف أكثر من معامل اختياري، ولكنها يجب أن تكون الأخيرة في ترتيب المعاملات، حيث لا يمكن أن يكون هناك معامل اختياري، ثم يتبعه معامل مطلوب Required استخدام إجراء كمعامل لإجراء آخر:

يمكن إرسال إجراء (دالة أو إجراء فرعي) كمعامل لإجراء آخر وهذا باستخدام المندوب Delegate

ولتعريف مندوب عن إجراء فرعي، استخدم الصيغة التالية:

Delegate Sub (معاملات الإجراء) اسم المندوب

ولتعريف مندوب عن دالة استخدم الصيغة التالية:

Delegate Function القيمة المعادة As (معاملات الدالة) اسم المندوب

بعد هذا يمكنك استخدام اسم هذا المندوب كنوع لمعاملات الإجراءات.. تعال نرى ذلك ابدأ بتعريف المندوب التالي:

Delegate Sub MsgBox (Msg As String)

والآن سنستخدمه في تعريف الإجراء التالي:

```
Sub ShowMsg(M As MsgBox)
```

```
    Call M("رسالة إيضاحية")
```

```
End Sub
```

الآن بقيَ أن نعرّف بعض الإجراءات التي يمكن أن تصلح كمعاملات للإجراء ShowMsg:

```
Sub Show1(S As String)
```

```
    MsgBox (s)
```

```
End Sub
```

```
Sub Show2(MessageText As String)
```

```
    MsgBox (MessageText & " أخرى")
```

```
End Sub
```

```
Sub Show3(T As String)
```

```
    MsgBox (T & " ثلاثة")
```

```
End Sub
```

والآن يمكنك استدعاء الإجراء ShowMsg بأيّ من الطرق التالية:

```
ShowMsg(AddressOf Show1)
```

ستظهر الرسالة "رسالة إيضاحية"

```
ShowMsg(AddressOf Show2)
```

ستظهر الرسالة "رسالة إيضاحية أخرى"

```
ShowMsg(AddressOf Show3)
```

ستظهر الرسالة "رسالة إيضاحية ثلاثة"

لاحظ استخدام كلمة AddressOf قبل اسم الإجراء، وذلك حتى يستطيع VB التفريق بين

الدالة المرسله كمعامل، والدالة التي ترسل قيمتها المعادة Return Value كمعامل.

استدعاء الإجراء باستخدام متغير:

هناك وسيلة أخرى للقيام بعمل المندوب Delegate، وإن كانت تختلف عنه بعض الشيء،

وذلك باستخدام الدالة CallByName، التي ظهرت لأول مرة مع صدور نسخة VB6..

ولهذه الدالة الصيغة التالية:

_ , اسم الخلية التي تحتوي على الإجراء) CallByName = القيمة_ المعادة

(مصفوفة معاملات نوع الإجراء المستدعى، نصّ أو متغير نصّي يمثل اسم الإجراء

ويأخذ المعامل الثاني واحدة من قيم المرقم CallType التالية:

Method	استخدم هذا النوع مع الإجراءات والدوال.
Get	استخدم هذا النوع عندما تريد قراءة قيمة خاصية.
Set	استخدم هذا النوع عندما تريد تغيير قيمة خاصية.
Let	لم يعد هذا النوع متاحا في VB.Net.

ولا بدّ أنّك استنتجت السبب في أنّ المعامل الأخير لهذه الوسيلة مصفوفة معاملات.. طبعا حتى ترسل معاملات الإجراء الذي تستدعيه.

وطبعا القيمة المعادة من هذه الدالة، هي القيمة المعادة من المعادة من الدالة التي تستدعيها. ما رأيك أن نعيد كتابة الدالة ShowMsg التي كتبناها في الفقرة السابقة، ولكن باستخدام الاستدعاء بالاسم هذه المرة؟.. هذه هي:

Sub ShowMsg(MsgSub As String)

CallByName (Me, MsgSub, CallType.Method, "رسالة إيضاحية")

End Sub

الآن يمكنك استدعاء هذه الدالة، مرسلا لها أسماء الإجراءات الثلاثة التي عرفناها سابقا، كالتالي:

ShowMsg("Show1") ' ستظهر الرسالة "رسالة إيضاحية"

ShowMsg("Show2") ' ستظهر الرسالة "رسالة إيضاحية أخرى"

ShowMsg("Show3") ' ستظهر الرسالة "رسالة إيضاحية ثالثة"

أعتقد أنّك لاحظت الفارق في طريقة الاستدعاء بين هذه الوسيلة وبين استخدام المندوب Delegate.. وأنت حرّ في اختيار الطريقة التي تناسبك أكثر.

أنواع أخرى للقيم المعادة:

يمكن للدوال أن تعيد مصفوفات Arrays أو سجلات Structures أو كائنات من أي نوع ... غير الأعداد و النصوص

وفي مثالنا هذا نريد أن نكتب متتالية حسابية نعين فيها قيمة البداية و مقدار الزيادة ثم عدد الحدود ونحصل على مصفوفة من الأعداد

VB.Net Code

```
Function Series(ByVal Start As Integer, ByVal B As Integer, ByVal N As Integer)
As Integer()
    Dim X(N) As Integer, I As Integer
    X(0) = Start ' الأولى المصفوفة خانة في الأول الحد وضع '
    For I = 1 To N - 1
        ' الزيادة قيمة + السابقة الخانة قيمة = المصفوفة في الحالية الخانة قيمة '
        X(I) = X(I - 1) + B
    Next
    Return X ' مصفوفة المعادة القيمة '
End Function
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim S() As Integer
    S = Series(2, 3, 10)
    For i As Integer = 0 To UBound(S)
        Console.WriteLine(S(i))
    Next
End Sub
```

تعرض متتالية حسابية أساسها 3 و حدها الأول 2 و عدد حدودها 10

إعادة تعريف الإجراء Overloading

في بعض الأحيان تستعمل إجراء لنفس الغرض لكن معاملاته تختلف . فمثلا تريد أن تكتب إجراء البحث في قاعدة البيانات عن شخص فإما أن تعرف رقمه أو اسمه . فماذا تفعل في مثل هذه الحالات ؟

نكتب إجراءات تبدأ بكلمة **Overloads** متبوعة بنفس الاسم بمعاملات مختلفة .

VB.Net Code

```
Overloads Function Add(ByVal x As Integer, ByVal y As Integer) As Integer
    Return x + y
End Function
Overloads Function Add(ByVal x As Double, ByVal y As Double) As Double
    Return x + y
End Function
Overloads Function Add(ByVal x As Long, ByVal y As Long) As Long
    Return x + y
End Function
```

قمنا بإعادة تعريف دالة الجمع بحيث يمكننا جمع الأعداد من أنواع مختلفة ثم نكتب في حدث نقر الزر

الفصل الثاني : لغة البرمجة

VB.Net Code

```
Console.WriteLine("***** Fun with Method Overloading *****")
' جمع أعداد صحيحة
Console.WriteLine(Add(10, 10))
' جمع أعداد من نوع Long
Console.WriteLine(Add(900000000000, 900000000000))
' جمع أعداد من نوع مزدوج
Console.WriteLine(Add(4.3, 4.4))
```

تطبيقات عن الدوال :

دالة حساب مجموع مصفوفة :

VB.Net Code

```
Function sum(ByVal No() As Integer) As Integer
    For i As Integer = 0 To UBound(No)
        sum = sum + No(i)
    Next
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim x() As Integer = {1, 2, 3, 5}
    MsgBox(sum(x))
End Sub

Function sum(ByVal ParamArray No() As Integer) As Integer
    For i As Integer = 0 To UBound(No)
        sum = sum + No(i)
    Next
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    MsgBox(sum(1, 2, 3))
End Sub
```

حساب مضروب عدد

مضروب العدد (ن) = ن × مضروب (ن - 1)

VB.Net Code

```
Function Factorial(ByVal n As Integer) As Double
    If n = 0 Then التوقف شرط
        Factorial = 1
    Else
        Factorial = n * Factorial(n - 1)
    End If
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
MsgBox(Factorial(3)) '6  
End Sub
```

حساب باقي القسمة

VB.Net Code

```
Function ModEx(ByVal dN As Double, ByVal dD As Double) As Double  
    ModEx = dN - Int(dN / dD) * dD  
    'القاسم في ( الحاصل - المقسوم ) = الباقي  
End Function
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    MsgBox(ModEx(4, 3))  
End Sub
```


الفصل الثالث : الأدوات Controls

- أداة صندوق النص Textbox
- أداة العنوان Label
- زر التحكم Button
- أداة القائمة List Box
- صندوق السرد Combo Box
- قائمة مربع الاختيار Check List box
- قائمة الصور Image List
- قائمة العرض List View
- شجرة العرض Tree View
- خانة الاختيار radio button
- مربع الاختيار Check Box
- أداة التلميح Tool Tip Text
- أداة Notif Icon
- أداة Error Provider
- أداة التبويب Tab Control
- القوائم Menu Stript
- القائمة المختصرة Context Menu Strip
- مربعات الحوار القياسية
- الخواص المشتركة
- مربع حوار فتح Open File Dialog
- مربع حوار حفظ Save File Dialog
- مربع حوار الألوان Color Dialog
- مربع حوار خط Font Dialog
- مربع حوار مستعرض المجلدات Folder Browser Dialog
- أدوات الطباعة
- النص المنسق Rich Text box
- محرر النصوص
- المؤقت Timer

- شريط التقدم Progress Bar
- أداة التاريخ و الوقت Date Time Picker
- أداة الترقيم Numeric Up Down
- أدوات الملفات
- صندوق الصور Picture Box
- جدول عرض البيانات Data Grid View
- شريط الحالة Status Bar
- شريط الأدوات Tool Strip
- أداة التنسيق Masked Text Box

أداة صندوق النص TextBox

استعمالها

تستعمل لإدخال وعرض النصوص بإمكانك جعلها للقراءة فقط فلا يمكنك تغيير محتواها ، فهي تستعمل لعرض البيانات في عدة سطور ، ملتفة حسب حجم الأداة و يمكنك أيضا إضافة تنسيقات متنوعة تشمل النص كله (عند جعل سمك الخط غليظ و لونه أحمر فهذا ينطبق على النص كله فلا يمكنك جعل جزء منه غير ذلك) و إذا أردت تنويع التنسيقات فاختر الأداة الغنية (RichTextBox).

خصائصها :

أولا : من خلال نافذة الخصائص

الخاصية	القيمة	الدور
Name	اختر اسم برمجي تفهمه	يستخدم للتعامل مع الأداة برمجيا للوصول إلى خصائصه
Text	أي نص	لإدخال النصوص
TextAlign	Left : إلى اليسار Right : إلى اليمين Center : توسيط	محاذاة النص
MultiLine	False : الكتابة في سطر واحد True : الكتابة في عدة سطور	تعدد الأسطر
ScrollBars	None : بلا Horizontal : الأفقي Vertical : العمودي Both : كليهما	عرض المنزلاقات في حالة عدم ظهور النص
WordWrap	True : للقبول False : للرفض	التفاف النص أي جعل النص يتناسب مع حجم الأداة فيعود إلى السطر بدون الحاجة إلى المنزلاقات
AcceptsReturn	True : للقبول False : للرفض	يستقبل زر الإدخال
AcceptsTab	True : للقبول False : للرفض	في حالة القبول يستقبل زر الانتقال و يكتب حرف جدولة يعادل 4 مسافات في مربع النص و الانتقال مسموح ب Ctr+Tab أما في حالة الرفض تجعل المستخدم ينتقل للأداة التالية عند ضغط زر الانتقال Tab.
MaxLength	عدد	أقصى عدد من الحروف

الفصل الثالث الأدوات

للقراءة فقط	True: لا يمكن تغيير النص False: يمكن	ReadOnly
حرف كلمة مرور	رمز مناسب مثلا (*)	PasswordChar
لون خلفية الصندوق	اختر اللون من مربع الحوار	BackColor
نمط الحدود	None : بلا Fixed3D : ثلاثية الأبعاد FixedSingle : لإظهار حد خطي	Border Style
إمكانية التعامل مع الأداة	True : ممكن False : غير ممكن	Enabled
تنسيق الخط (نوعه ، حجمه ، سمكه ، تسطيره ...)	حدد ما تريد من مربع الحوار	Font
لون الخط	حدد اللون من مربع الحوار	ForeColor
تحديد اتجاه النص	No : لا Yes : نعم من اليمين إلى اليسار	Right To Left
الرقم الفهرسي الترتيبي للانتقال بالمجدولة	رقم	Tab Index
إمكانية استقبال الانتقال بالمجدولة	True : نعم False : لا	Tab Stop
إمكانية الظهور	True : نعم False : لا	Visible
خاصية الأسطر	اكتب نصا طويلا	Lines
الموقع	X : البعد عن الحافة اليسرى للنموذج Y : البعد عن الحافة العلوية اليسرى للنموذج	Location

ثانيا من خلال الكود

التعليمة	الخاصية
TextBox1.Text = "Visual Basic" ' يجب أن يكون بين علامتي تنصيص	كتابة النص
TextBox1.AppendText(".Net")	إضافة سطر
TextBox1.BackColor = Color.Blue	لون الخلفية
TextBox1.ForeColor = Color.Black	لون الخط

الفصل الثالث الأدوات

<code>TextBox1.BorderStyle = BorderStyle.FixedSingle</code>	نمط الحد
<code>TextBox1.Clear()</code>	مسح المحتوى
<code>TextBox1.MaxLength = 4</code>	أقصى حد من الحروف
<code>TextBox1.ScrollBars = ScrollBars.Vertical</code>	إدراج شريط التمرير
<code>TextBox1.TextAlign = HorizontalAlignment.Center</code>	محاذاة
<code>MsgBox(TextBox1.TextLength)</code>	طول النص
<code>TextBox1.SelectAll()</code>	تحديد الكل
<code>TextBox1.Focus()</code>	نقل التركيز
<code>TextBox1.SelectionStart = 0</code>	بداية التحديد
<code>TextBox1.Size = New Size(300, 400)</code>	تغيير الحجم
<code>TextBox1.Undo()</code>	تراجع
<code>MsgBox(Me.TextBox1.CanUndo)</code>	إمكانية التراجع
<code>TextBox1.Copy()</code> ' نسخ	نسخ
<code>TextBox1.Paste()</code>	لصق
<code>TextBox1.Cut()</code>	قص
<code>TextBox1.Visible = True</code>	إمكانية الظهور
<code>TextBox1.Tag = "salah"</code>	القيمة المرجعية
<code>TextBox1.Width = 400</code>	العرض
<code>TextBox1.Height = 200</code>	الارتفاع
<code>TextBox1.Top = 100</code>	البعد عن حافة النموذج العلوية
<code>TextBox1.Multiline = True</code>	متعدد الأسطر
<code>TextBox1.PasswordChar = "*"</code>	رمز كلمة المرور
<code>Me.TextBox1.ClearUndo()</code>	مسح كل عمليات التراجع المخزنة
<code>Me.TextBox1.WordWrap = False</code>	إلتفاف النص
<code>TextBox1.ReadOnly = True</code>	للقراءة فقط
<code>TextBox1.CharacterCasing = CharacterCasing.Upper</code> 'ALI <code>TextBox1.CharacterCasing = CharacterCasing.Lower</code> 'ali <code>TextBox1.CharacterCasing = CharacterCasing.Normal</code>	حالة الأحرف

<pre>For L As Integer = 0 To TextBox1.Lines.Length - 1 MsgBox(TextBox1.Lines(L)) Next If TextBox1.Modified = True Then MsgBox("النص تعديل تم") End If Dim FindString As String = InputBox("للبحث كلمة ادخل", "بحث", "عنها") If FindString = "" Then Exit Sub Dim TextStart As Integer TextStart = TextBox1.Text.IndexOf(FindString) If TextStart > 0 Then TextBox1.SelectionStart = TextStart TextBox1.SelectionLength = FindString.Length End If TextBox1.Focus()</pre>	<p>قراءة الأسطر</p> <p>تم التعديل</p> <p>البحث بطريقة IndexOf</p>
--	---

ملاحظة

كل الخصائص متشابه من أداة لأخرى فكل خاصية تطرقنا إليها فلا نعيدها

أهم أحداث صندوق النص **TextBox Events**

استقبال التركيز **GotFocus**

VB.Net Code

```
Private Sub TextBox1_GotFocus(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TextBox1.GotFocus
    TextBox1.BackColor = Color.Blue
End Sub
```

مغادرة التركيز **lostFocus**

VB.Net Code

```
Private Sub TextBox1_LostFocus(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TextBox1.LostFocus
    TextBox1.BackColor = Color.Yellow
End Sub
```

في المثالين السابقين عند انتقال التركيز إلى أداة النص يصبح لونها أزرق و عند مغادرة التركيز يصبح لونها أصفر .

الحديث **KeyDown**

VB.Net Code

```
Private Sub TextBox1_KeyDown(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.KeyEventArgs) _
    Handles TextBox1.KeyDown
    If e.Alt Then
        MsgBox("Alt المفتاح ضغط تم")
    ElseIf e.Shift Then
        MsgBox("Shif المفتاح ضغط تم")
    ElseIf e.Control Then
        MsgBox("Conrol المفتاح ضغط تم")
    End If
End Sub
```

الحدث KeyPress

VB.Net Code

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.KeyPressEventArgs) _
    Handles TextBox1.KeyPress
    If Char.IsLetter(e.KeyChar) Then e.Handled = True
End Sub
```

إن هذين الحدثين يستقبلان الرقم الدال على الحرف المضغوط وليس الحرف نفسه، وذلك من خلال التعبير .e.KeyCode

ويجب أن تلاحظ أن رقم الزرّ KeyCode، مختلف عن ذلك المسمى ASCII، ففي الشفرة الأخيرة، يكون لكل حرف رقم مختلف، فمثلاً الحرف الكبير A يختلف رقمه عن الحرف الصغير a.. ولكن في رقم الزرّ KeyCode، فإن لكل زر على لوحة المفاتيح رقماً خاصاً به، وبهذا تجد أن حرفي A و a لهما نفس الرقم لأنهما على نفس الزرّ.. إذن كيف نفرّق بينهما؟

إن هذين الحدثين يمنحانك القدرة على معرفة حالة الأزرار Shift و Ctrl و Alt، عن طريق اختبار إذا ما كانت الخصائص التالية صواباً أم خطأ: e.Shift و e.Control و e.Alt.. وبهذا تستطيع أن تعرف بالضبط أن حرف A الكبير هو المقصود، ما دام زر Shift مضغوطاً.

حدث المغادرة Leave

VB.Net Code

```
Private Sub TextBox1_Leave(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TextBox1.Leave
    If Not IsNumeric(TextBox1.Text) Then TextBox1.Focus()
End Sub
```

في المثال السابق لا يمكن مغادرة أداة النص إلا بعد كتابة رقم فيها .

حدث التغيير TextChanged

```

VB.Net Code

Private Sub TextBox1_TextChanged(ByVal sender _
    As System.Object, ByVal e As System.EventArgs) _
    Handles TextBox1.TextChanged
    Me.Text = ""
    Me.Text = TextBox1.Text
End Sub
    
```

في هذا المثال يتغير نص عنوان النافذة تبعاً لتغييرات أداة النص

حدث التحقق من الصحة Validating

```

VB.Net Code

Private Sub TextBox1_Validating(ByVal sender As Object, _
    ByVal e As System.ComponentModel.CancelEventArgs) _
    Handles TextBox1.Validating
    If TextBox1.Text = "" Then e.Cancel = True
End Sub
    
```

في هذا المثال تتحقق أداة النص ما إذا كان النص فارغاً فلا يمكنك فعل أي شيء و هذا بتفعيل الخاصية

Cancel إلى True

التعامل مع النص المحدد

<pre> TextBox1.SelectedText = _ TextBox1.SelectedText.ToUpper() TextBox1.SelectedText = "" Dim FindString As String = InputBox("للبحث كلمة ادخل") </pre>	<p style="text-align: right;">تحويل النص المحدد إلى حروف كبيرة مسح النص المحدد البحث و تحديد الكلمة</p>
<pre> "بحث", "عنها") If FindString = "" Then Exit Sub Dim TextStart As Integer TextStart = TextBox1.Text.IndexOf(FindString) If TextStart > 0 Then TextBox1.SelectionStart = TextStart TextBox1.SelectionLength = FindString.Length End If </pre>	<p style="text-align: right;">تحديد كلمة من خلال Select</p>
<pre> TextBox1.Focus() TextBox1.Select(الطول , البداية) TextBox1.Focus() TextBox1.Select(2, 5) </pre>	

السماح و المنع Validation

<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If Char.IsNumber(e.KeyChar) Then e.Handled = True End Sub</pre>	<p>كتابة الحروف فقط</p>
<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If Char.IsNumber(e.KeyChar) = False Then e.Handled = True End Sub</pre>	<p>كتابة الأرقام فقط</p>
<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If e.KeyChar = CChar("A") = False Then e.Handled = True End Sub</pre>	<p>كتابة حرف معين فقط</p>
<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If Char.IsNumber(e.KeyChar) = False Then If e.KeyChar = CChar(".") Then e.Handled = False Else e.Handled = True End If End If End Sub</pre>	<p>كتابة الأرقام أو الأعداد العشرية</p>
<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If Char.IsLower(e.KeyChar) = False Then e.Handled = True End Sub</pre>	<p>كتابة الحروف الصغيرة فقط</p>
<pre>Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress If Char.IsLower(e.KeyChar) = true Then e.Handled = True End Sub</pre>	<p>كتابة الحروف الكبيرة فقط</p>
<pre>Private Sub TextBox1_Leave(ByVal sender As Object, _ ByVal e As System.EventArgs) Handles TextBox1.Leave If Not IsNumeric(TextBox1.Text) Then TextBox1.Focus() End Sub</pre>	<p>مغادرة التركيز إلا إذا كان رقماً فقط</p>

مشروع محرر النصوص TextPad

1. إنشاء القوائم

الفصل الثالث الأدوات

قائمة ملف : تتكون من القوائم الفرعية التالية :

جديد ، فتح ، حفظ ، حفظ باسم ، إغلاق

قائمة تحرير : تتكون من القوائم الفرعية التالية :

نسخ ، قص ، لصق ، تراجع ، إعادة ، البحث ، استبدال ، ترقيم

بحيث يبدو الشكل كما يلي :



كتابة التعليمات

متغير عام خاص باسم الملف ' **Public CFile As String**

If Editor.Modified **Then**

If MsgBox("التغيرات حفظ في ترغب هل", MsgBoxStyle.Question +
MsgBoxStyle.YesNo) = MsgBoxResult.Yes **Then**

حفظToolStripMenuItem_Click(Me, e)

Else

Editor.Clear()

End If

End If

Editor.Clear()

If MsgBox("التغيرات حفظ في ترغب هل", MsgBoxStyle.Question +

جديد

فتح

<pre> MsgBoxStyle.YesNo) = MsgBoxResult.No Then OpenFile1.Filter = "All File txt (*.tx) *.txt" If OpenFile1.ShowDialog = Windows.Forms.DialogResult.OK Then CFile = OpenFile1.FileName Editor.Text = My.Computer.FileSystem.ReadAllText(CFile) End If End If </pre>	
<pre> If Editor.Modified Then If CFile = "" Then حفظToolStripMenuItem_Click(Me, e) Else My.Computer.FileSystem.WriteAllText(CFile, Editor.Text, False) End If End If </pre>	حفظ
<pre> If Editor.Text = "" Then Exit Sub Else SaveFile1.Filter = "All text (*.txt) .txt" If SaveFile1.ShowDialog = Windows.Forms.DialogResult.OK Then CFile = SaveFile1.FileName My.Computer.FileSystem.WriteAllText(CFile, Editor.Text, False) End If End If </pre>	حفظ باسم
<pre> If Editor.Modified Then If MsgBox("التغيرات حفظ في ترغب هل", MsgBoxStyle.Question + MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then حفظToolStripMenuItem_Click(Me, e) End If End If If MsgBox("الخروج في حقا ترغب هل", MsgBoxStyle.Question + MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then End End If </pre>	خروج

<pre> If Editor.Modified Then Editor.Undo() End If </pre>	تراجع
<pre> If Editor.Modified Then Editor.Copy() End If </pre>	نسخ
<pre> ----- Clipboard.SetDataObject(Editor.SelectedText) </pre>	
<pre> If Editor.Modified Then </pre>	قص

<pre> Editor.cut() End If ----- Clipboard.SetDataObject(Editor.SelectedText) Editor.SelectedText = "" </pre>	
<pre> If Editor.Modified Then Editor.Past() End If ----- </pre>	لصق
<pre> With Clipboard.GetDataObject If .GetDataPresent(DataFormats.Text) Then Editor.SelectedText = .GetData(DataFormats.Text) End If End With </pre>	
<pre> Dim I As Integer Dim NewText As New System.Text.StringBuilder() For I = 0 To Editor.Lines.Length - 1 NewText.Append((I + 1).ToString & vbTab & _ Editor.Lines(I) & vbCrLf) Next Editor.Text = NewText.ToString </pre>	ترقيم
<pre> Dim SelStart As Integer Dim SrchMode As CompareMethod Form1.Editor.Focus() If ChkCase.Checked = True Then SrchMode = CompareMethod.Binary Else الحرف حالة يراعي لا أنه يعني مما نصّي، البحث ' SrchMode = CompareMethod.Text End If With Form1.Editor SelStart = InStr(1, .Text, TextBox1.Text, SrchMode) If SelStart = 0 Then MsgBox("الكلمة على العثور يتم لم") Exit Sub End If .Select(SelStart - 1, TextBox1.Text.Length) .ScrollToCaret() ' المستخدم أمام النص ظهور من تأكد ' End With </pre>	البحث
<p>نفس التعليمات نغير العبارة الملون ب</p> <pre> SelStart = InStr(.SelectionStart + 2, .Text, TextBox1.Text, SrchMode) </pre>	البحث عن التالي
<pre> If Form1.Editor.SelectedText = TextBox1.Text Then Form1.Editor.SelectedText = TextBox2.Text </pre>	استبدال

<pre> End If Button1_Click(sender, e) ' Dim CurPos, CurSel As Integer Dim SrchMode As CompareMethod ' الاختيار مربع قيمة على بناء البحث، نوع متغير قيمة تحديد ' If ChkCase.Checked = True Then ' (الحرف حالة مراعاة) هي كما الحرف حالة يطابق أنه يعني مما ثنائي، البحث ' SrchMode = CompareMethod.Binary Else ' الحرف حالة يراعي لا أنه يعني مما نصّي، البحث ' SrchMode = CompareMethod.Text End If With Form1.Editor CurPos = .SelectionStart CurSel = .SelectionLength .Text = Replace(.Text, TextBox1.Text, _ TextBox2.Text, , , SrchMode) .SelectionStart = CurPos .SelectionLength = CurSel End With </pre>	<p>استبدال الكل</p>
--	-------------------------

<pre> Dim x As String FontDialog1.ShowDialog() x = FontDialog1.Font.Name Dim textFont As New Font(x, FontDialog1.Font.Size, FontDialog1.Font.Style) Editor.Font = textFont Private Sub FontDialog1_Apply(ByVal sender As Object, _ ByVal e As System.EventArgs) Handles FontDialog1.Apply Editor.Font = FontDialog1.Font Editor.ForeColor = FontDialog1.Color End Sub Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripMenuItem.Click FontDialog1.ShowDialog() FontDialog1_Apply(Me, e) End Sub </pre>	<p>تنسيق الخط حسب مربع الحوار</p> <p>بكيفية أخرى</p>
---	--

أداة العنوان Label

الفصل الثالث الأدوات

تستخدم لعرض النصوص و لا يمكن للمستخدم تغيير محتواها . و تستعمل عنوانا لأدوات أخرى يوضح للمستخدم الغرض أو الحدث الذي تقوم به .
مثال نكتب عنوانا (اسم الطالب) للأداة Label ليبين للمستخدم ماذا يفعل على الأداة TextBox.



خصائص أداة العنوان Properties

الخاصية	القيمة	الشرح
AutoSize	: True : False	يتغير طول الأداة على حسب طول و قصر النص لا يتغير طوله يبقى ثابتا كما كان في وضع التصميم
Anchor	اختر الجهة	تستخدم للحفاظ على موقع الأداة على النافذة عند تغيير حجمها
ContextMenu	اسم القائمة	تسمح بإظهار القائمة المختصرة عند الضغط على الزر الأيمن للفأرة على الأداة Label
Cursor	اختر نوع المؤشر	اختيار نوع المؤشر الذي يظهر عند المرور فوق الأداة
Dock	Right Left Top Bottom	تملأ الأداة الجانب الأيمن للنافذة تملأ الأداة الجانب الأيسر للنافذة تملأ الأداة الجانب العلوي للنافذة تملأ الجانب السفلي للنافذة

الفصل الثالث الأدوات

تملاً كامل النافذة بلا	Fill None	Locked
تستخدم لتثبيت حجم و مكان الأداة في وقت التصميم	لا : True يمكن التغيير ممك : False	
يحدد طول الأداة يحدد عرض الأداة	Height Width	Size
تستخدم ما إذا كانت الأداة تستجيب للعلامة & وتكتب في الخاصية Text فبذلك يمكنك الوصول إلى الأداة بواسطة Alt و الحرف	True False	UseMnemonic

ملاحظة

كل الخصائص متشابهة من أداة لأخرى فكل خاصية تطرقنا إليها فلا نعيدها

زر التحكم Button

يستخدم لتنفيذ أوامر و تعليمات برمجية بعد النقر عليه .

مثال : عندما نقوم بإدخال كلمة مرور في صندوق النص انقر زر التحكم Button للدخول إلى البرنامج.



أهم خصائص زر التحكم Properties

الشرح	القيمة	الخاصية
تستخدم لعرض صورة كخلفية للزر	اختر الصورة	BackgroundImage
تستخدم للحفاظ على قيمة نتيجة نافذة الحوار التي فتحت بالطريقة ShowDialog	بلا : None قيمة الخاصية : Ok	DialogResult



ملاحظة

كل الخصائص متشابهة من أداة لأخرى فكل خاصية تطرقنا إليها فلا نعيدها

خصائص زر التحكم من خلال الكود

VB.Net Code

```

Dim button1 As New Button()
    button1.Name = "bt1" ' الاسم البرمجي للزر
    button1.Text = "Clik hier" ' عنوان الزر
    button1.Anchor = AnchorStyles.Right ' تثبيت بعد الزر عن الحافة اليسرى عند تغيير حجم النافذة
    button1.AutoSize = True ' السماح بتغيير حجم الزر وفق نص العنوان
    button1.BackColor = Color.Blue ' لون خلفية الزر أزرق
    button1.BackgroundImage = ImageList1.Images(0) ' اسناد صورة لخلفية الزر
    button1.BackgroundImageLayout = ImageLayout.Center ' توسيط صورة الخلفية
    Button1.Size = New Size(120, 70) ' تحديد حجم الزر بتحديد الارتفاع والعرض
    button1.TabIndex = 1 ' ترتيب الفهرسي للانتقال بواسطة الجدولة
    button1.TabStop = True ' السماح للمجدولة بزيارة الزر

    button1.Font = New Font("Arial", Me.Font.Size,
    FontStyle.Bold) ' تحديد نوع الخط و حجمه و سمكه
    button1.ForeColor = Color.Yellow ' لون الخط أصفر
    button1.Enabled = True ' إمكانية التعامل مع الزر
    button1.Location = New Point(70, 25) ' تموقع الزر بتحديد بعده عن الحافة اليسرى للنافذة و العلوية
    button1.SetBounds(70, 50, 200, 100) ' تحديد موضع الزر
    button1.TextAlign = ContentAlignment.BottomCenter ' محاذاة النص
    button1.Visible = True ' إمكانية الرؤية
    
```

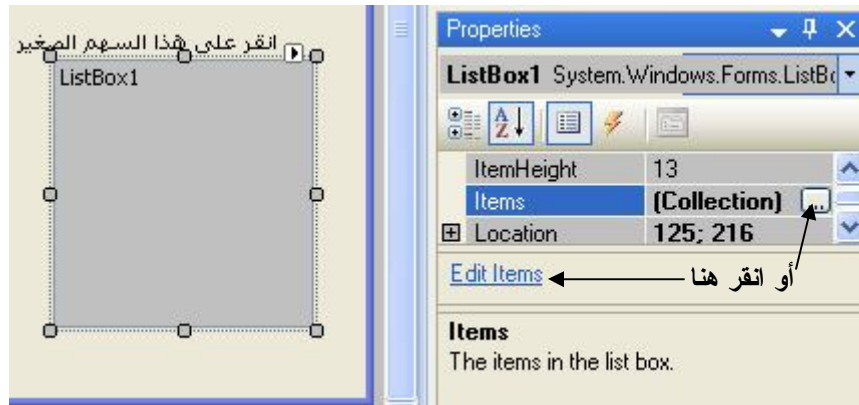
جعل الزر يملأ جهة من النافذة ' `button1.Dock = DockStyle.Left` ' جعل

إضافة الزر الجدي ' `Me.Controls.Add(button1)` ' Me.Controls.Add(button1)

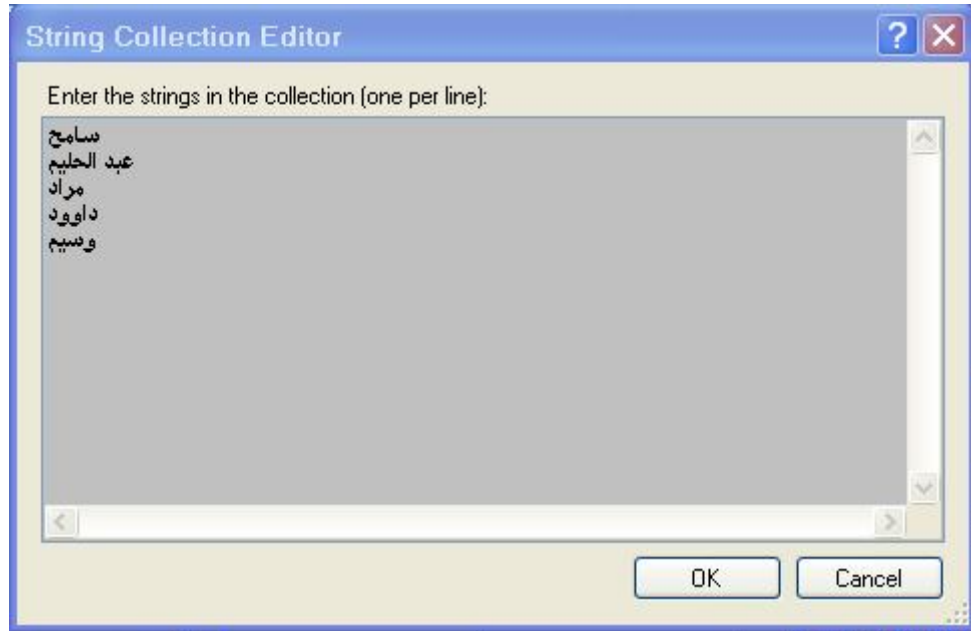
أداة القائمة List Box

تستخدم لعرض مجموعة من العناصر (البيانات) على شكل قائمة بحيث يمكن للمستخدم اختيار عنصر أو أكثر . مثال : عرض أسماء الطلاب

إضافة عناصر إلى القائمة

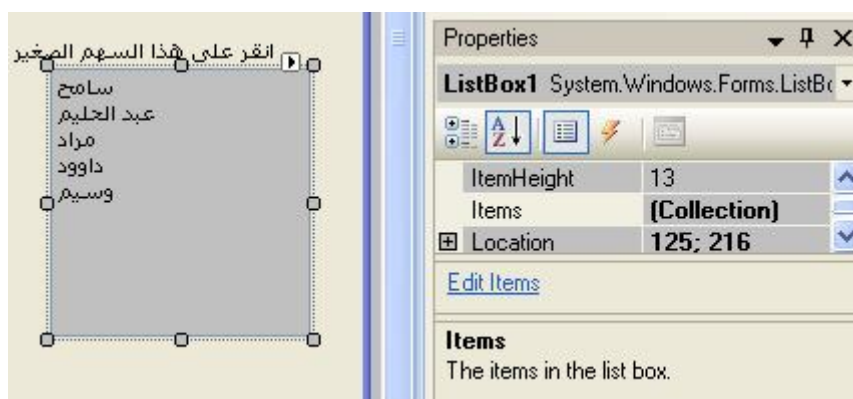


تظهر أمامك النافذة التالية منها نضيف العناصر



عند إضافة العناصر اضغط OK و عد إلى القائمة ستجدها بالشكل

الفصل الثالث الأدوات



خصائص أداة القائمة وقت التصميم

الخاصية	القيمة	الشرح
ColumnWidth	صفر عدد صحيح	النظام يحدد عرض الأداة يمثل عرض العمود في القائمة متعددة الأعمدة يجب أن تكون الخاصية <code>MultiColumn = True</code>
Data Source	اختر اسم الجدول	تستخدم لتحديد مصدر (الجدول) البيانات التي يتم عرضها في القائمة
DisplayMeber DrowMode	اختر اسم الحقل Normal OwnerDrawFixed OwnerDrawVariable	تستخدم لتحديد اسم الحقل لعرض بياناته في القائمة عادي تحديد ارتفاع يمثل ارتفاع كل الأعمدة وذلك من الخاصية <code>Height</code> تحديد شكل عرض العناصر يمكنك تحديد ارتفاعات مختلفة للعناصر
MultiColumn	True False	إمكانية عرض عمود عرض عمود واحد
ScrollAlwaysVisible	True False	تستخدم لعرض أشرطة التمرير دائما
SelectionMode	None One MultiSimple MultiExtended	لا يمكن تحديد أي عنصر من القائمة يستطيع تحديد عنصر واحد فقط يستطيع تحديد أكثر من عنصر يستطيع تحديد أكثر من عنصر مع إمكانية استخدام مفاتيح التحكم <code>Shift</code> ، <code>Alt</code> ، <code>Ctrl</code>
Sorted	True False	إمكانية الفرز و الترتيب الترتيب يكون حسب تحميل المستخدم للقائمة

خصائص القائمة وقت التنفيذ

الفصل الثالث الأدوات

ولكي نتعامل مع عناصر القائمة من الكود، يمكنك استخدام مجموعة العناصر Items Collection، وطبعاً صار مألوفاً لديك شكل المجموعات Collections، فهي تشابه المصفوفات، فمثلاً أول عنصر في القائمة هو Items(0)، ويليه Items(1)، وهكذا.

وكلّ المجموعات، تحتوي مجموعة عناصر القائمة على الخواص التقليديّة: "العدد" Count، و"إضافة" Add، و"إزالة" Remove، و"إفراغ" Clear، وهي مماثلة لنظيراتها التي تعاملنا معها في القوائم وأدوات النماذج.

ويمكنك التعامل مع العناصر المحدّدة في القائمة بهذه الوسائل:

- رقم العنصر المحدّد SelectedIndex:
- مجموعة أرقام العناصر المحدّدة SelectedIndices Collection:
- العنصر المحدّد SelectedItem:
- مجموعة العناصر المحدّدة SelectedItems Collection:

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ListBox1.Items.Add("Test") ' إضافة عنصر
    MsgBox(ListBox1.Items.Count) ' عدد العناصر
    ListBox1.Items.Insert(1, "Test1") ' إدراج عنصر في موضع محدد
    ListBox1.Items.Remove("Test1") ' حذف عنصر
    ListBox1.Items.RemoveAt(0) ' حذف عنصر من موضع محدد
    ListBox1.Items.Clear() ' مسم محتوي القائمة
End Sub
```

إضافة مجموعة من العناصر عن طريق الحلقة

VB.Net Code

```
Dim value As Integer
For value = 0 To 100
    ListBox1.Items.Add("Test " & value)
Next
```

إضافة عناصر مصفوفة إلى القائمة

VB.Net Code

```
Dim Values() As String = {"A", "b", "c"}
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    For i As Integer = 0 To Values.Count - 1
        ListBox1.Items.Add(Values(i))
    Next
End Sub
```

```
Next
End Sub
```

إدراج مجموعة عناصر بطريقة

VB.Net Code

```
ListBox1.Items.AddRange(New Object() _
    {"داود", "مراد", "وسيم", "سامح"})
ListBox1.Sorted = True
```

عرض كل عناصر القائمة

VB.Net Code

```
Dim Itm As Object
For Each Itm In ListBox1.SelectedItems
    Console.WriteLine(Itm)
Next
-----
For i As Integer = 0 To ListBox1.Items.Count - 1
    Console.WriteLine(ListBox1.Items(i).ToString)
Next
```

معرفة العنصر المحدد

VB.Net Code

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    MsgBox(ListBox1.Text)
End Sub
-----
ListBox1.SelectedIndex = 1 ' تحديد العنصر عن طريق الرقم الفهرسي
Me.Text = ListBox1.SelectedItem
```

البحث التقريبي بطريقة FindString

فهي تبحث عن نص و لو كان جزءا من كلمة فمثلا عند عن كلمة "عبد" فهي تقبل كلمة " عبد الله" أو "عبد الرحمن" ...

الصيغة العامة 

```
(رقم بداية البحث, "الكلمة التي تبحث عنها") = FindString
MsgBox(ListBox1.FindString("عبد", 1))
```

البحث عن النص المطابق FindStringExact

الصيغة العامة 

```
رقم بداية البحث, "الكلمة التي تبحث عنها" = FindString(
العنصر
MsgBox(ListBox1.FindString("عبد", 1))
```

البحث باستخدام IndexOf

الصيغة العامة 

```
Dim I As Integer = ListBox1.Items.IndexOf("عنصر البحث")
MsgBox(ListBox1.Items.IndexOf("عبد"))
```

وهذا مثال شامل لطريقتي البحث المطابق و التقريبي

VB.Net Code 

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim SrchWord As String
    Dim ID As Integer
    With ListBox1
        SrchWord = InputBox("عنه البحث تريد النص اكتب")
        ID = .FindStringExact(SrchWord)
        If ID >= 0 Then
            MsgBox("العنصر رقم " & ID.ToString & " = " & _
                (.Items(ID)).ToString, , "تام تطابق")
            .TopIndex = ID
            .SelectedIndex = ID
        Else
            ID = .FindString(SrchWord)
            If ID >= 0 Then
                MsgBox("العنصر رقم " & ID.ToString & _
                    " = " & (.Items(ID)).ToString, , "تقريبي عثور")
                .TopIndex = ID
                .SelectedIndex = ID
            Else
                MsgBox("العنصر " & SrchWord & _
                    "بالقائمة موجود غير")
            End If
        End If
    End With
```

End Sub

إدراج أعمدة إلى قائمة

VB.Net Code

```
Dim fmtStr As String = "{0,-10}{1,12}{2,14}{3,12}"
With ListBox1.Items
    .Clear()
    .Add(String.Format(fmtStr, "College", "Enrollment", _
        "Endowment", "Public SS"))
    .Add(String.Format(fmtStr, "Harvard", 6660, 19.2, 0.659))
    .Add(String.Format(fmtStr, "Yale", 5278, 10.1, 0.532))
End With
```

أداة صندوق السرد ComboBox

تستعمل لغرض عرض مجموعة من العناصر على شكل قائمة يمكن للمستخدم اختيار عنصر أو مجموعة من العناصر . كأداة القائمة بالإضافة لغرض الكتابة كأداة نص .

ملاحظة

كل ما قلناه عن أداة القائمة فهو ينطبق عن أداة صندوق السرد مع زيادة هذه الخصائص

MaxDropDownItem : أكبر عدد ممكن من العناصر يمكن عرضه عند اتسداد القائمة

ملاحظة

1. يمكنك الوصول إلى باقي العناصر بواسطة أشرطة التمرير
2. يجب أن تكون الخاصية `Style = DropDown` أو `Style = DropDownList`

الخاصية	القيمة	الشرح
MaxDropDownItem	عدد صحيح	لعرض عدد العناصر المنسدلة

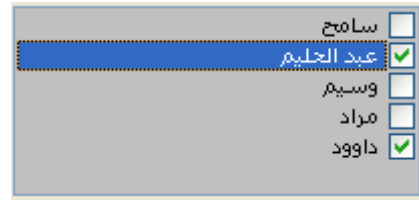
النمط Style: تستخدم للتحكم في شكل ونمط أداة صندوق السرد

Style	Simple	على شكل أداة نص
	DropDown	قائمة منسدلة يمكن الكتابة فيها
	DropDownList	قائمة منسدلة لا يمكن الكتابة فيها



قائمة مربع الخيار CheckListBox

تستخدم كما تستخدم أداة القائمة List لكنها تعرض بجوار كل عنصر مربع اختيار



ملاحظة

خصائص أداة قائمة مربع الخيار هي نفسها خصائص أداة ListBox أو ComboBox

أداة قائمة الصور ImageList

تخزن و تحفظ مجموعة من الصور وتحملها في الذاكرة لتسريع استخدامها في البرنامج وهي من الأدوات الهامة تتعامل مع كل الأدوات . فكل أداة تمتلك الخاصية ImageList فبمجرد تحميل قائمة الصور يمكنك إسناد أي صورة أو أيقونة لأي أداة عن طريق هذه الخاصية.

كيفية الاستعمال :

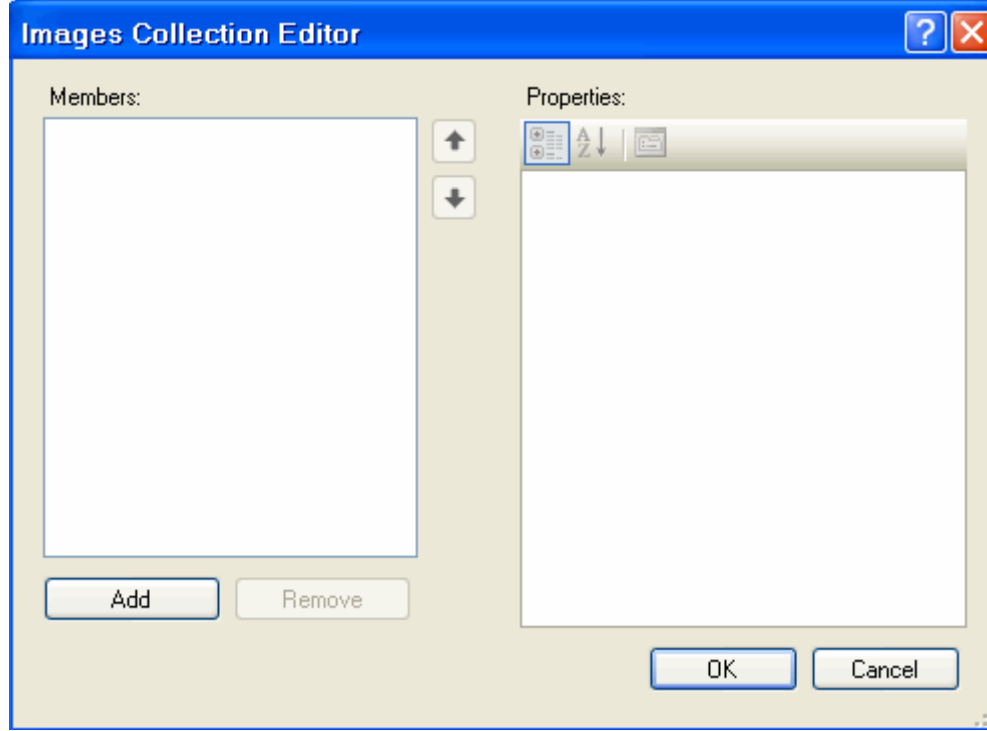
1. قم بإضافة الأداة ImageList من صندوق الأدوات تظهر النافذة التالية



الفصل الثالث الأدوات

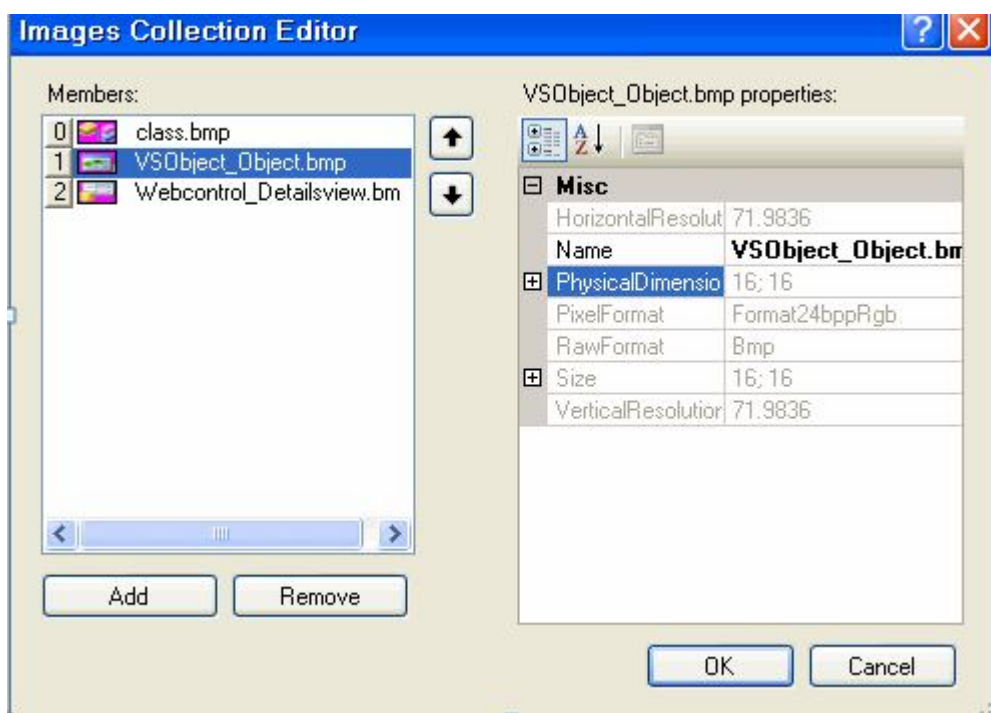
الشرح	الخاصية
تحدد حجم الصور و الأيقونات (كل الصور تأخذ نفس الحجم)	Image Size
تحديد درجة وضوح وجودة الصورة أو الأيقونة	Image Bit Depth
لتحميل و استيراد الصور أو الأيقونات	Choose image Images

عند اختيارك الخيار الأخير لإضافة الصور أو الأيقونات تظهر النافذة التالية



الشرح	الزر
إضافة الصور أو الأيقونات	Add
نقل مكان الصور نحو الأعلى	↑
نقل مكان الصورة نحو الأسفل	↓
لحذف الصورة	Remove

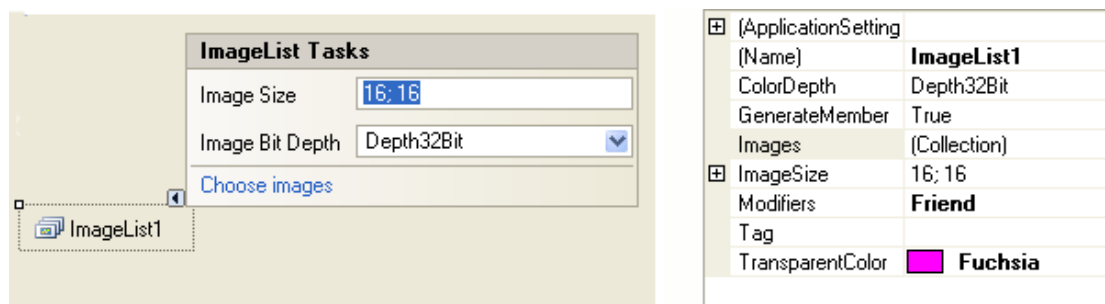
بعد إضافة الصور عن طريق الزر Add يصبح الشكل كما يلي



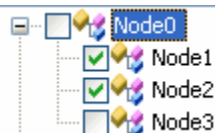
تلاحظ بجوار كل صورة رقم يمثل الرقم الفهرسي Index و هو ما نستعمله لتحميل و استيراد الصورة

ملاحظة

لجعل الصورة شفافة نختار الخاصية **TransparentColor** و نختار لون الخلفية المناسب وفي مثالنا نختار اللون الوردي لأن لون خلفية الصور وردي



ولذلك وبعد استخدام هذه الأيقونات مع أداة ما لا تظهر الخلفية أنظر للشكل التالي:



إضافة صورة عن طريق الكود

الصيغة العامة


```
ImageList1.Images.Add(Image.FromFile(مسار الصورة))
```

```
ImageList1.Image.Add(Image.FromFile("C:\Image\ToolBar\save.bmp"))
```

الوصول إلى الصورة

الصيغة العامة 

```
Control.Image = imagelist1.images(الرقم الفهرسي)
```

```
Button1.Image = ImageList1.Images(0)
```

أداة قائمة العرض ListView

تستخدم لعرض مجموعة من العناصر بأيقونات . و تستعمل كواجهة مثل التي توجد في مستكشف ويندوز



أهم خصائص قائمة العرض:

هذه هي الخصائص التي تتحكم في مظهر قائمة العرض:

عرض View:

استخدم هذه الخاصية لتختار كيف سيتم عرض العناصر في القائمة.. ولهذه الخاصية القيم التالية:

أيقونات كبيرة، تحت كل منها اسم العنصر.. وهذه هي القيمة الافتراضية.	LargeIcon
أيقونات صغيرة، بجوار كل منها اسم العنصر.	SmallIcon
قائمة بأسماء العناصر، مرتصة عمودياً من أعلى لأسفل.	List
يتم عرض اسم العنصر في عمود، مع وجود عدة أعمدة أخرى تمثل بيانات العنصر.	Report

تنظيم العناصر Arrange:

استخدم هذه الخاصية لتختار كيفية ترتيب العناصر في قائمة العرض.. وهي تأخذ القيم التالية:

الفصل الثالث الأدوات

افتراضي Default	عندما يتم تحريك العنصر (بسحبه بالفأرة) فإنه يبقى في مكانه الجديد.
يسار Left	يتم محاذاة العناصر إلى جهة اليسار.
محاذاة الشبكة SnapToGrid	عندما يحرك المستخدم العنصر، فإنه يستقر في أقرب نقطة لخطوط شبكة غير مرئية على الأداة.
أعلى Top	يتم محاذاة العناصر إلى الجهة العليا من الأداة.

تنظيم تلقائي AutoArrange:

إذا جعلت هذه الخاصية True، فسيتم إعادة ترتيب العناصر تلقائيًا لو قام المستخدم بحذف أو تحريك أحدها.

طراز عناوين الأعمدة HeaderStyle:

عند عرض العناصر بتفاصيلها (خاصية View بالقيمة Report)، فإن هذه الخاصية تستخدم لتحديد طراز عناوين الأعمدة التي يتم فيها عرض العناصر وتفاصيلها.. وقيم هذه الخاصية هي:

Clickable	يتم عرض عناوين الأعمدة مع قدرة المستخدم على ضغطها بالفأرة، ليتم ترتيب عناصرها تبعاً لمحتويات العمود المضغوط.
Nonclickable	يتم عرض عناوين الأعمدة لكن المستخدم لا يستطيع ضغطها بالفأرة.
None	لا يتم عرض عناوين الأعمدة.

السماح بترتيب الأعمدة AllowColumnReorder:

إذا جعلت هذه الخاصية True، فسيكون باستطاعة المستخدم أن يسحب العمود (بضغط عنوانه) ليحركه إلى أي موضع، بحيث يغير ترتيبه بين الأعمدة.

التشيط Activation:

استخدم هذه الخاصية لتحديد الفعل الذي ينشط عناصر القائمة، من بين الأفعال التالية:

OneClick	يتم تشيط العنصر بضغط واحدة.
Standard	يتم تشيط العنصر بنقره مرتين بالفأرة.
TwoClick	يتم تشيط العنصر مع تغيير لون نصه بنقره مرتين بالفأرة.

تحديد الصف بالكامل FullRowSelect:

إذا جعلت هذه الخاصية True، فسيتمكن المستخدم من تحديد العنصر بكل تفاصيله (أعمدته) عند ضغط أي خانة في أي عمود.

خطوط الشبكة GridLines:

إذا جعلت هذه الخاصية True، فسيتم عرض خطوط فاصلة بين العناصر وتفاصيلها.

تحرير اللافتة LabelEdit:

إذا جعلت هذه الخاصية True، فسيتمكن المستخدم من تغيير اسم العنصر.

التفاف الالفةة LabelWrap:

إذا جعلت هذه الخاصية True، فسيتمّ التفاف العناوين الطويلة للعناصر إلى السطر التالي، وذلك أثناء عرض العناصر كأيقونات.

تحديد بالإشارة HoverSelection:

إذا جعلت هذه الخاصية True، فسيتمّ استخدام من تحديد أيّ عنصر بمجرد الإشارة إليه بالفأرة دون ضغطه.

متعدّدة التحديد MultiSelect:

إذا جعلت هذه الخاصية True، فسيتمّ استخدام من تحديد أكثر من عنصر في نفس الوقت، بنفس الطرق التي يحدّد بها العناصر في متصفحّ الويندوز Explorer (أعتقد أنّك تعرفها).

مربّعات اختيار CheckBoxes:

إذا جعلت هذه الخاصية True، فسيتمّ عرض مربّع اختيار بجوار كلّ عنصر.. وللتعامل مع العناصر المختارة Selected، يمكنك استخدام الخاصيتين: "أرقام العناصر المختارة" CheckedIndices و"العناصر المختارة" CheckedItems.. الطريف أنّ بإمكانك اختيار أيّ شكل تريده بدلا من مربّع الاختيار التقليديّ، وللفعل ذلك، استخدم خاصية قائمة صور الحالة "StateImageList"، حيث يمكنك أن تضع بها اسم نسخة من الأداة قائمة الصور "ImageList".. قائمة الصور هذه يجب أن تحتوى على صورتين على الأقلّ: الصورة الأولى (رقم 0) سيتمّ عرضها بديلا لمربّع الاختيار في وضع عدم الاختيار Unchecked، والصورة الثانية (رقم 1) سيتمّ عرضها بديلا لمربّع الاختيار في وضع الاختيار Checked.. وفي كلّ الأحوال لن يؤثر ذلك على أيقونة العنصر.

العنصر الفعّال FocusedItem:

استخدم هذه الخاصية لتعرف العنصر الفعّال حاليا في القائمة، ذلك الذي يستقبل ضربات أزرار لوحة المفاتيح.

أعلى عنصر TopItem:

استخدم هذه الخاصية لمعرفة أعلى عنصر مرئيّ في القائمة.

ترتيب Sorting:

تحدّد كيفية ترتيب عناصر القائمة، وهي تأخذ القيم التالية:

بدون ترتيب.	None
ترتيب العناصر تصاعدياً.	Ascending
ترتيب العناصر تنازلياً.	Descending

التعامل مع عناصر قائمة العرض من خلال نافذة التصميم

للتعامل مع قائمة العرض نقوم بإدراج الأداةين ListView و ImageList إلى النموذج نضيف أداة ImageList و نغير خصائصها كما في الشكل حيث سنستخدم أيقونة بحجم 32، عمق ألوان 32 Bit وهو عالي الوضوح.

الفصل الثالث الأدوات

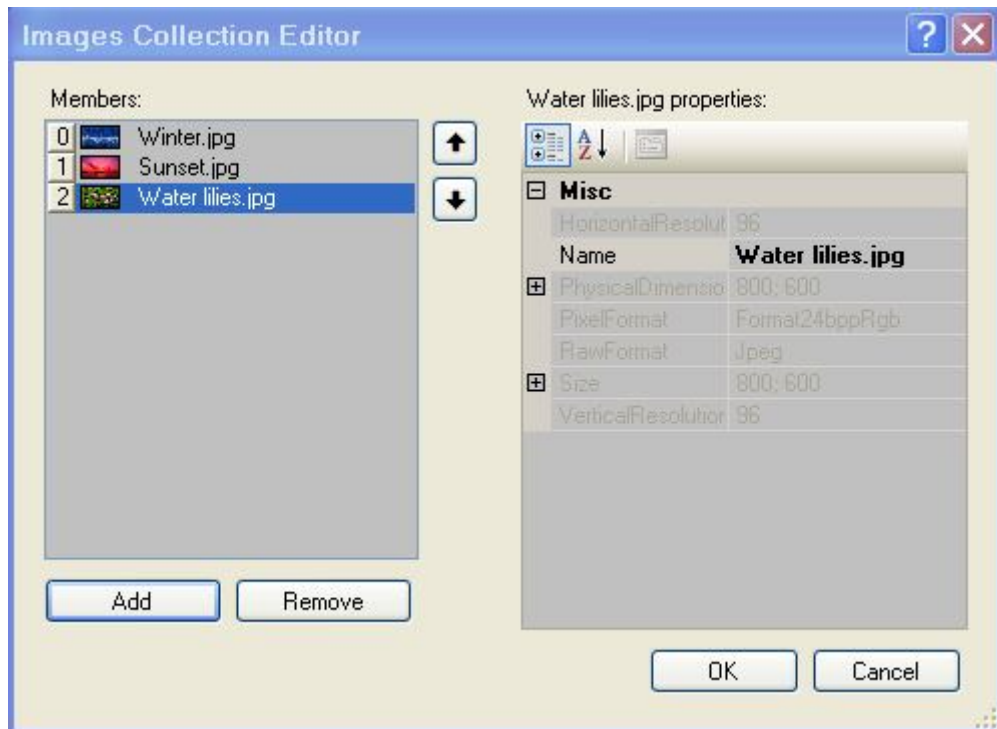
ImageList Tasks

Image Size: 32, 32

Image Bit Depth: Depth32Bit

[Choose images](#)

الآن نضيف بعض الأيقونات من خلال الضغط على ChooseImage تظهر النافذة التالية



كما تعلمنا سابقا . ثم اضغط موافق Ok

نعود إلى الأداة ListView انقر على السهم الصغير بجانب الأداة تظهر النافذة التالية

ListView Tasks

[Edit Items](#)

[Edit Columns](#)

[Edit Groups](#)

View: LargeIcon

Small ImageList: (none)

Large ImageList: ImageList1

[Dock in parent container](#)

الشرح	الخيار
إضافة عناصر إلى الأداة بشكل يدوي	Edit items
إضافة وتعديل الأعمدة التي نريد استخدامها	Edit Columns
إضافة وتعديل المجموعات التي تستخدم لتجميع العناصر على أساسها	Edit Groups
نوعية العرض (انظر درس الخصائص)	View

الفصل الثالث الأدوات

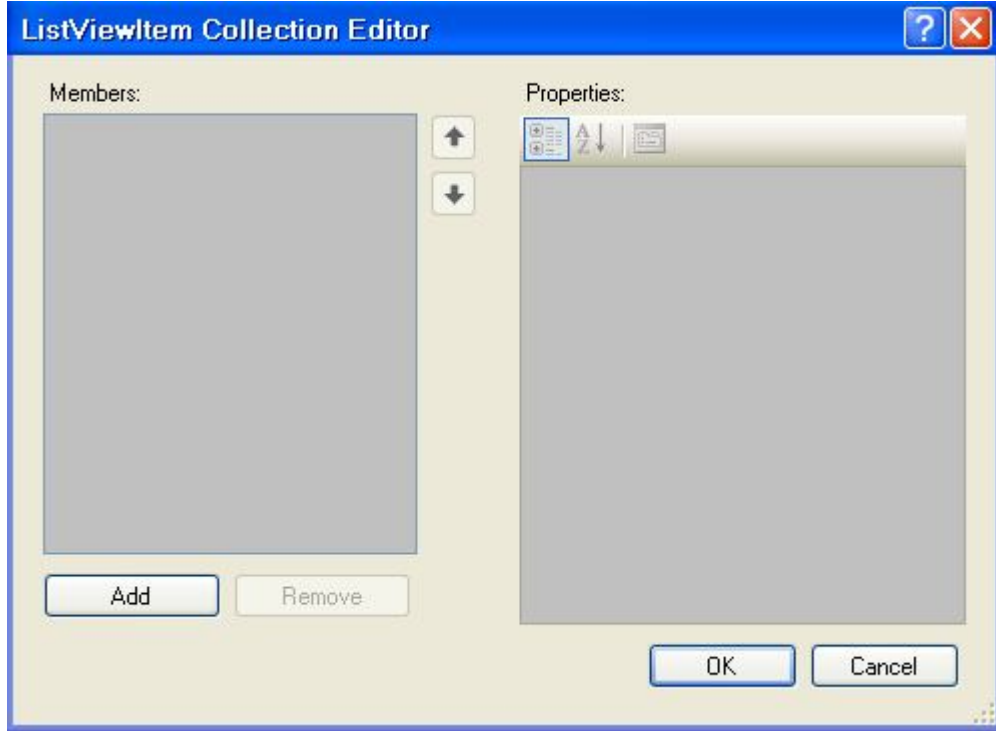
اختيار قائمة الصور التي تحتوي على الأيقونات المصغرة

Small ImageList

اختيار قائمة الصور التي تحتوي على الأيقونات الكبيرة

Large ImageList

اختر الخيار الأول Edit Items لإضافة عناصر جديدة أو اختر الخاصية Items ثم انقر زر الاستعراض بجانب Collection تظهر النافذة التالية



الشرح	الزر
إضافة الصور أو الأيقونات	Add
نقل مكان الصور نحو الأعلى	↑
نقل مكان الصورة نحو الأسفل	↓
لحذف الصورة	Remove

اضغط الزر Add لإضافة بعض العناصر تظهر النافذة بشكل التالي



من خلال الخاصية : Text : نكتب عناوين العناصر

ImageIndex : نختار رقم الأيقونة المرافقة للعنوان

حدد الخصائص كما في الشكل السابق و بنفس الطريقة نضيف عنصر آخر الخزائن حيث يبدو الشكل في الأخير كما يلي

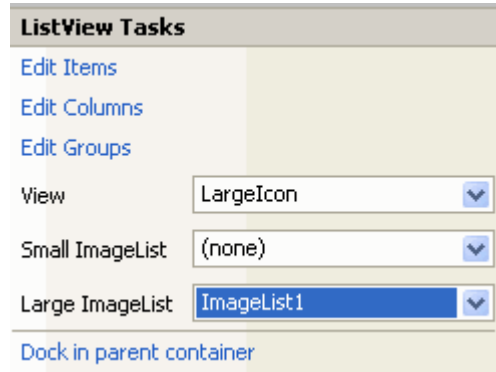


ثم اضغط موافق Ok ستحصل على الشكل



إضافة أعمدة

نعود إلى الأداة ListView ثم ننقر السهم الصغير تظهر النافذة



تغير الخاصية View إلى Details

ثم نختار الخيار EditColumns لإضافة أعمدة تظهر نافذة مشابهة للنافذة التي أضفنا منها العناصر بمجرد الضغط على الزر Add تظهر النافذة بالشكل



من خلال الخاصية Text نكتب عنوان العمود مثلا : مكتبة
 نضيف عمود آخر بنفس الطريقة وليكن عنوانه العدد فيصبح الشكل



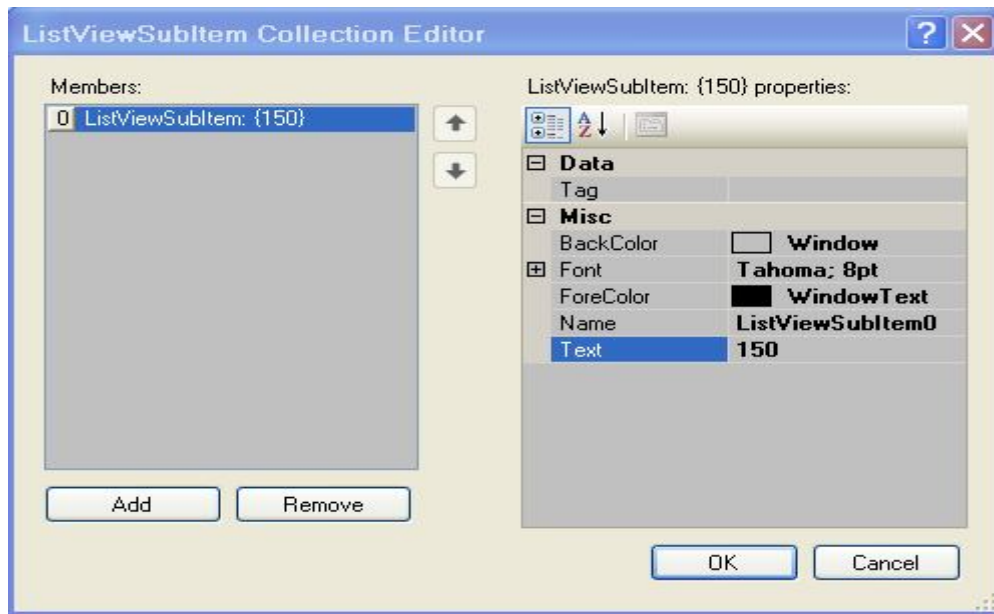
ثم اضغط Ok و لاحظ الناتج



لاحظ أن عمود العدد فارغ للإضافة عناصر فرعية SubItems إليه نتبع ما يلي
عد إلى الأداة ListView انقر السهم الصغير ثم اختر Edit Items تظهر النافذة



تأكد من العنصر المحدد هو الكتب ثم اختر الخاصية SubItems من اليمين تظهر نافذة أخرى مشابهة



من خلال الخاصية Text نكتب عدد الكتب و ليكن 150 تم اضغط Ok
حدد الخيار الثاني الخزان ثم بنفس الكيفية السابقة نضيف العنصر الفرعي الذي يمثل عدد الخزائن

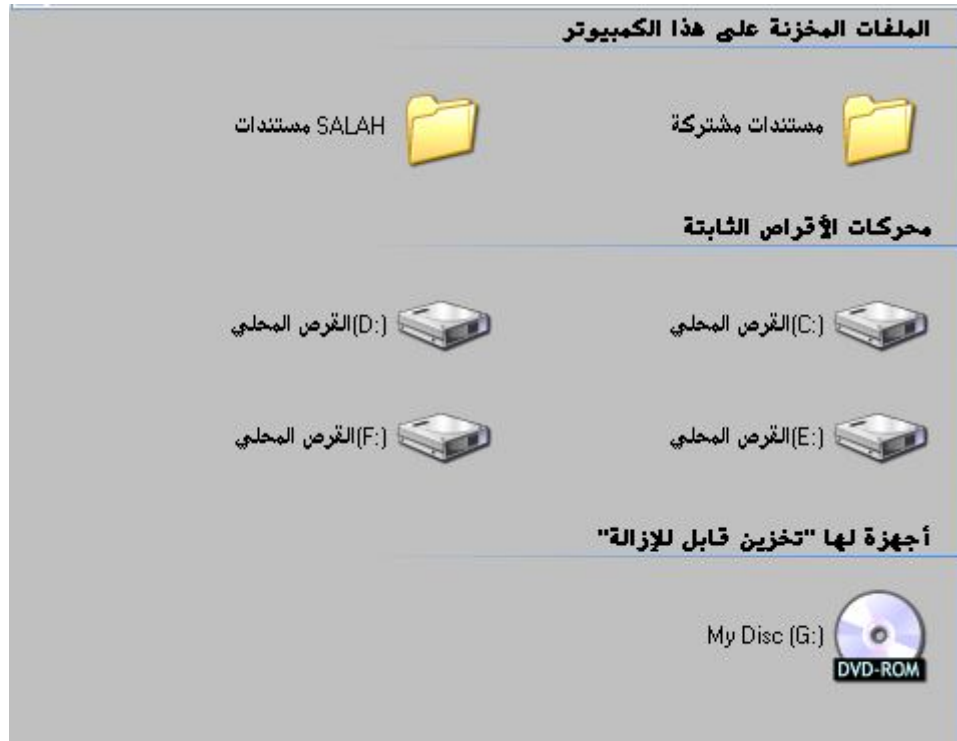


ثم اضغط Ok و لاحظ الناتج



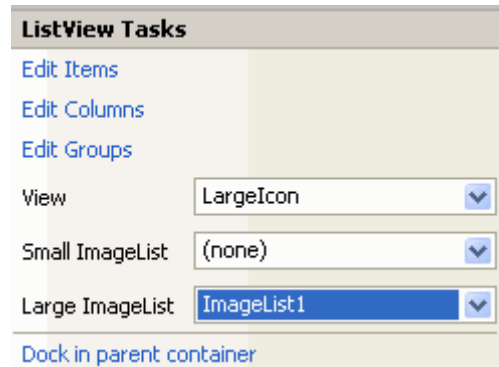
تكوين مجموعات

الآن نريد إضافة مجموعات تجمع فيها و تفرز العناصر حسب نمط معين كما يفعل نظام التشغيل وفي جهاز الكمبيوتر تحديداً .

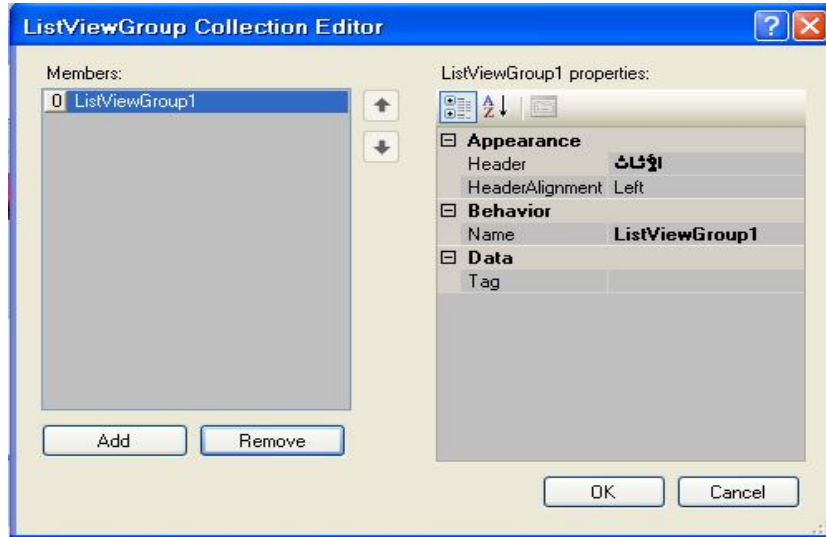


كيف نفعل ذلك ؟

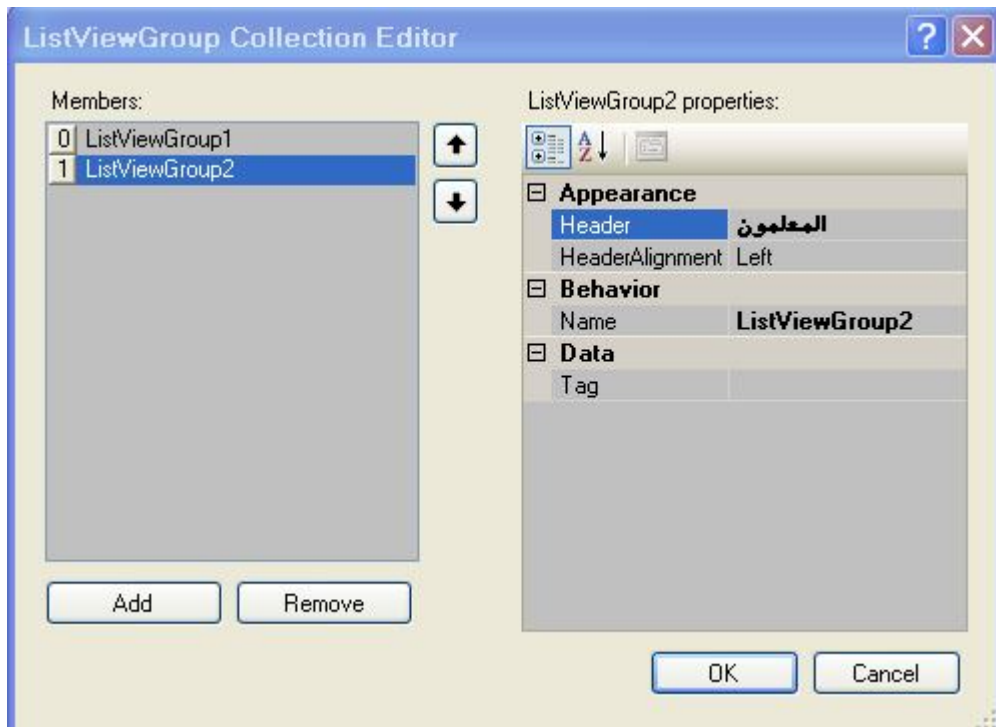
نعود إلى نافذة التالية



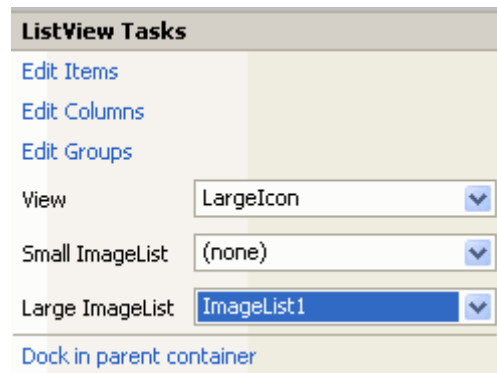
نحدد الخيارات حسب ما ورد في الشكل السابق ثم نختار Edit Groups تظهر النافذة التالية



نضغط على الزر Add و نكتب اسم المجموعة و ليكن (الأثاث) في الخاصية Header و بنفس الكيفية نضيف المجموعة الثانية (المعلمون) حتى يبدو الشكل



ثم اضغط Ok عد إلى النافذة السابق و اختر الأمر Edit Items لتوزيع العناصر على شكل مجموعات



نعود إلى النافذة التالية :



نضيف العنصرين الصالح و علي كما تعلمنا سابقاً



نحدد العنصر الأول ثم من خلال الخاصية Group نحدد أثاث و نفس الشيء بالنسبة للعنصر الثاني ثم نحدد العنصر الثالث (الصالح) و نضعه في مجموعة المعلمين من خلال الخاصية Group ونفس الشيء بالنسبة للعنصر (علي) حتى يبدو الشكل.



ثم اضغط Ok يكون الناتج كما يلي: مبروك



قائمة العرض من خلال الكود

تمتلك قائمة العرض الخاصية Items، التي تمكنك من التعامل مع جميع عناصرها.. هذه الخاصية هي مجموعة Collection، ولا بد أنك قد صرت خبيراً في التعامل مع المجموعات.. ولإضافة عنصر لهذه المجموعة، استخدم الوسيلة الشهيرة إضافة Add، بالصيغة التالية:

إضافة العناصر

 الصيغة العامة

```
ListView1.Items.Add("رقم الصورة", "اسم العنصر")
```

ندرج أداة قائمة الصور ImageList ونضيف لها بعض الأيقونات

نربط أداة ListView بأداة ImageList من خلال الخاصية Large ImageList ونختار ImageList1 ثم نكتب الكود:

 VB.Net Code

```
ListView1.Items.Add("الحيوانات", 0)
```

```
ListView1.Items.Add("النباتات", 1)
```

قراءة العنصر

 VB.Net Code

```
Me.Text = ListView1.Items(1).Text ' يعرض النباتات '
```

```
-----
Private Sub ListView1_ItemSelectionChanged(ByVal sender _
    As Object, ByVal e As System.Windows.Forms._
    ListViewItemSelectionChangedEventArgs) Handles ListView1.ItemSelectionChanged
    Me.Text = e.Item.Text
End Sub
```

تكوين مجموعات

 VB.Net Code

الفصل الثالث الأدوات

```
' تكوين المجموعات
ListView1.Groups.Add("", "الحيوانات")
ListView1.Groups.Add("", "النباتات")
' تحميل المجموعة الأولى
ListView1.Items.Add("أسد", 0)
ListView1.Items(0).Group = ListView1.Groups(0)
ListView1.Items.Add("غزال", 0)
ListView1.Items(1).Group = ListView1.Groups(0)
' تحميل المجموعة الثانية
ListView1.Items.Add("تفاح", 1)
ListView1.Items(2).Group = ListView1.Groups(1)
ListView1.Items.Add("برتقال", 1)
ListView1.Items(3).Group = ListView1.Groups(1)
```



إضافة الأعمدة Columns



الصيغة العامة

```
CHeader = ListView1.Columns.Add(محاذاة العنوان, عرضه, عنوان العمود)
```

VB.Net Code

```
' يجب اختيار نوع العرض Details
ListView1.View = View.Details
' تكوين العناصر الأساسية
ListView1.Items.Add("الحيوانات", 0)
ListView1.Items.Add("النباتات", 1)
' تكوين الأعمدة
ListView1.Columns.Add("الفئة")
ListView1.Columns.Add("مثل")
ListView1.Columns.Add("التكاثر")
```

الفئة	مثل	التكاثر
الحيوانات		
النباتات		

إضافة العناصر الفرعية



الصيغة العامة

```
ListView1.Items(رقم العنصر الأساسي).SubItems.Add("عنوان العنصر الفرعي")
```

VB.Net Code

```
' يجب اختيار نوع العرض Details
ListView1.View = View.Details
' تكوين العناصر الأساسية
ListView1.Items.Add("الحيوانات", 0)
```

الفئة	مثل	التكاثر
الحيوانات	أسد	ولود
النباتات	تفاح	زهري


```

ListView1.Items.Add("النباتات", 1)
' تكوين الأعمدة الثابتة
ListView1.Columns.Add("الفئة")
ListView1.Columns.Add("مثل")
ListView1.Columns.Add("التكاثر")
' تحميل الصف الأول
ListView1.Items(0).SubItems.Add("أسد")
ListView1.Items(0).SubItems.Add("ولود")
' تحميل الصف الثاني
ListView1.Items(1).SubItems.Add("تفاح")
ListView1.Items(1).SubItems.Add("زهري")
    
```

استعمال ListViewItem

VB.Net Code

```

' Details نوع العرض
ListView1.View = View.Details

' تكوين الأعمدة
ListView1.Columns.Add("الرقم", 50, HorizontalAlignment.Right)
ListView1.Columns.Add("الاسم", 90, HorizontalAlignment.Right)
ListView1.Columns.Add("اللقب", 50, _
HorizontalAlignment.Right)
' (الخانة 1 من الصف 1) العنوان الرئيسي
Dim item1 As New ListViewItem("1", 0)
' (بقية خانة الصف 1) العناوين الفرعية
item1.SubItems.Add("سامح")
item1.SubItems.Add("12")

Dim item2 As New ListViewItem("2", 1)
item2.SubItems.Add("وسيم")
item2.SubItems.Add("14")
ListView1.Items.AddRange(New ListViewItem() {item1, item2})
    
```

الرقم	الاسم	اللقب
1	سامح	12
2	وسيم	14

أداة شجرة العرض TreeView

تستخدم لعرض البيانات على شكل شجري أي أن كل عنصر يحتوي على مجموعة عناصر فرعية وهذه العناصر الفرعية يمكن أن تحتوي على مجموعة عناصر أخرى وهكذا . مثل مستكشف Windows التعامل مع الأداة من خلال نافذة الخصائص

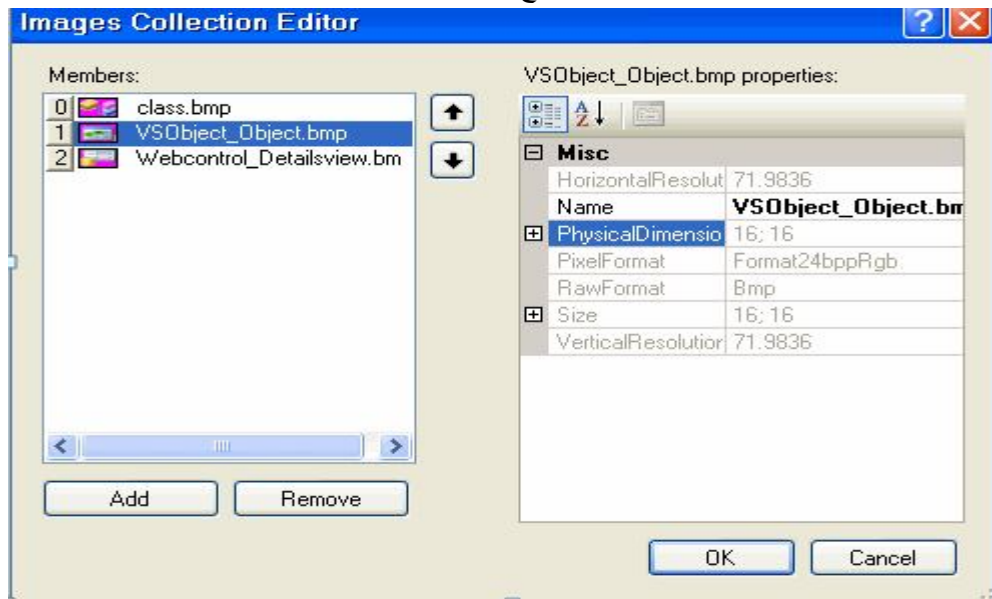
الفصل الثالث الأدوات



أولا نقوم بإدراج أداة قائمة الصور ImageList لأدراج أيقونات نضيفها أمام عناصر شجرة العرض



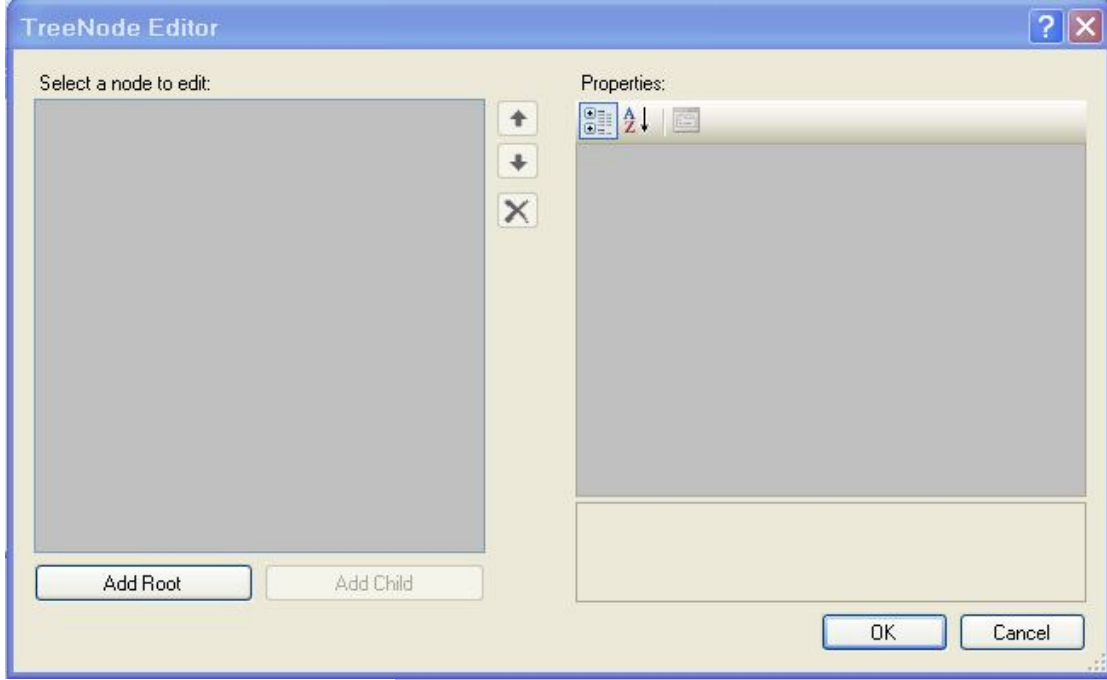
بعد إضافة الصور عن طريق الزر Add يصبح الشكل



1. قم بإدراج الأداة TreeView من صندوق الأدوات

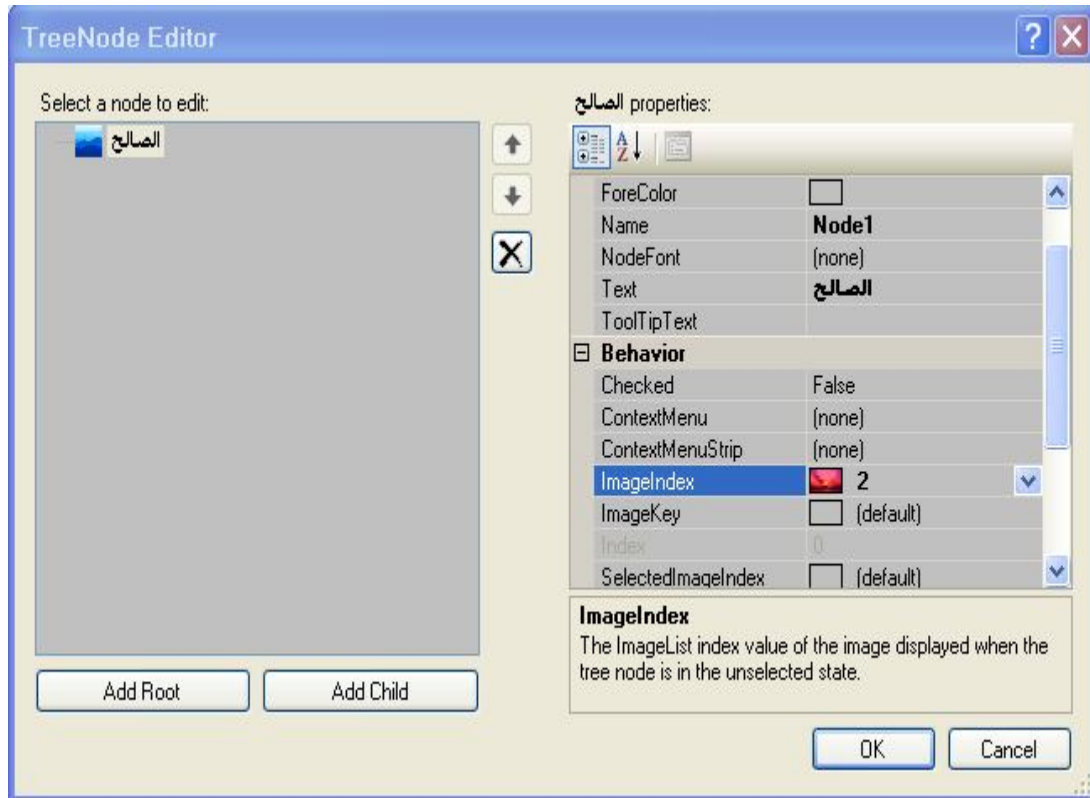
2. من نافذة الخصائص اختر الخاصية ImageList حدد الأداة ListImage1

3. من نافذة الخصائص اختر الخاصية Nodes
4. ثم انقر زر الاستعراض أمام Collection
5. تظهر النافذة التالية منها نضيف العناصر

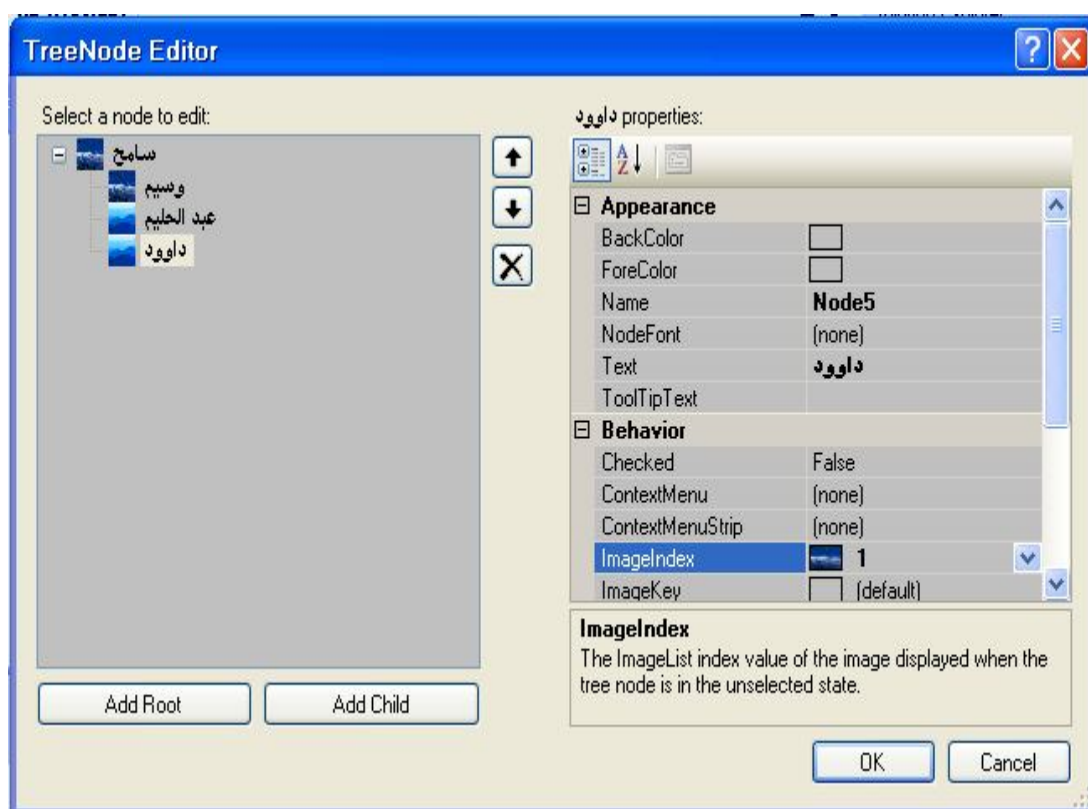


الشرح	الزر
إضافة العنصر الرئيسي (الأب)	Add Root
إضافة العنصر الفرعي (الابن)	Add Child
لنقل مكان العنصر نحو الأعلى	↑
لنقل مكان العنصر نحو الأسفل	↓
لحذف عنصر	✕

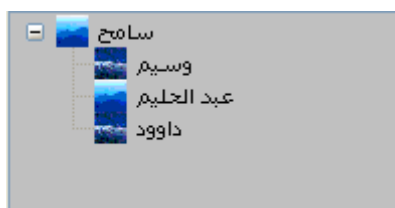
بعد الضغط على الزر Add Root تظهر النافذة التالية :



قم بإضافة العنصر الأب (الرئيسي) وسمه الصالح و ذلك من الخاصية Text و حدد الأيقونة المرافقة له من خلال الخاصية ImageInex باختيار الرقم الفهرسي للأيقونة ثم قم بإضافة الأبناء (العناصر الفرعية) بالضغط على الزر Add Child ثم اكتب أسماء الأبناء من خلال الخاصية Text ، ثم حدد الأيقونة من خلال الخاصية ImageIndex كما فعلت سابقا ، و هذا بالنسبة لباقي الأبناء و أخيرا تحصل على الشكل.



أخيرا اضغط زر موافق Ok تظهر شجرة العرض كما يلي :



التعامل مع الأداة من خلال الكود

إضافة عناصر Add Nodes

من خلال معرفتنا بالمجموعات Collections، فإننا نعرف الطريقة Add لأضافه العناصر

 الصيغة العامة

رقم الأيقونة التي تظهر عند النقر، الرقم الفهرسي للأيقونة، اسم العنصر، مفتاح)
TreeView1.Nodes.Add(

وإضافة العناصر الرئيسية نكتب

 VB.Net Code

```
TreeView1.Nodes.Add("الصالح")
```

```
TreeView1.Nodes.Add("علي")
```

الفصل الثالث الأدوات

إما لإضافة عناصر فرعية للعنصر الرئيسي الأب نكتب

VB.Net Code

```
TreeView1.Nodes.Add("الصالح") ' الجذر الرئيسي
' الجذور الفرعية الأبناء
TreeView1.Nodes(0).Nodes.Add("سامح")
TreeView1.Nodes(0).Nodes.Add("وسيم")
TreeView1.Nodes(0).Nodes.Add("داوود")

TreeView1.Nodes.Add("علي")
TreeView1.Nodes(1).Nodes.Add("محمد")
TreeView1.Nodes(1).Nodes.Add("نوال")
```

أما إذا رغبت فإدراج أيقونات للعناصر

أضف ImageList وهذه الأداة إلى النموذج ، قم بإضافة بعض الأيقونات 16-16 إلى الأداة ImageList

VB.Net Code

```
TreeView1.ImageList = ImageList1
TreeView1.Nodes.Add("", "الصالح", 0)
TreeView1.Nodes(0).Nodes.Add("", "سامح", 1, 1)
TreeView1.Nodes(0).Nodes.Add("", "وسيم", 1, 1)
TreeView1.Nodes(0).Nodes.Add("", "داوود", 1, 1)

TreeView1.Nodes.Add("", "علي", 0)
TreeView1.Nodes(1).Nodes.Add("", "محمد", 1, 1)
TreeView1.Nodes(1).Nodes.Add("", "نوال", 1, 1)
```

قراءة عنصر محدد

VB.Net Code

```
Me.Text = TreeView1.Nodes.Item(1).Text ' جذر ريسي
Me.Text = TreeView1.Nodes(1).Nodes(0).Text ' جذر ثانوي
```

معرفة اسم العنصر المحدد

VB.Net Code

```
Private Sub TreeView1_AfterSelect(ByVal sender As System. _
    Object, ByVal e As System.Windows.Forms. _
    TreeViewEventArgs) Handles TreeView1.AfterSelect
    MsgBox(e.Node.Text)
End Sub

-----
Dim node As TreeNode
For Each node In TreeView1.Nodes
```

```
If node.Checked = True Then MsgBox(node.Text)
Next
```

```
-----
Dim node As TreeNode
  For Each node In TreeView1.Nodes(0).Nodes
    If node.Checked = True Then MsgBox(node.Text)
  Next
```

عرض الجذور الرئيسية

 VB.Net Code

```
Dim node As TreeNode
  For Each node In TreeView1.Nodes
    ListBox1.Items.Add(node.Text)
  Next
```

عرض فروع جذر رئيسي

 VB.Net Code

```
Dim node As TreeNode
  For Each node In TreeView1.Nodes(0).Nodes
    ListBox1.Items.Add(node.Text)
  Next
```

```
-----
Dim node As TreeNode
  For Each node In TreeView1.Nodes(e.Node.Index).Nodes
    ListBox1.Items.Add(node.Text)
  Next
```

حذف العنصر المحدد

 VB.Net Code

```
TreeView1.SelectedNode.Remove()
```

حذف عنصر بجميع فروع

 VB.Net Code

```
Dim Node As TreeNode
Node = TreeView1.Nodes(0).Nodes(7)
Node.Remove
```

الشجرة و قواعد البيانات

 VB.Net Code

```
Public Class Form1
  Dim Con As New ADODB.Connection
  Dim Rst1, Rst2 As New ADODB.Recordset
```

```
Dim strCon As String = "Provider = microsoft.jet.oledb.4.0;" & _
```

```
"data source = h:/dbvb6.mdb"
```

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Try
        Con.Open(strCon)

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        Rst1.CursorLocation = ADODB.CursorLocationEnum.adUseClient
        Rst1.Open("select * from box", Con,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)

        Do Until Rst1.EOF
            TreeView1.Nodes.Add("", Rst1.Fields("العنوان").Value)
            Rst1.MoveNext()
        Loop
        If Rst2.State Then Rst2.Close()
        Rst2.Open("select * from code ", Con)
        Do Until Rst2.EOF
            TreeView1.Nodes(Rst2.Fields("الرقم").Value -
1).Nodes.Add(Rst2.Fields("التعليق").Value)
            Rst2.MoveNext()
        Loop
    End Sub
```

```
Private Sub TreeView1_AfterSelect(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs) Handles TreeView1.AfterSelect
    On Error Resume Next
    Dim rst As New ADODB.Recordset
    If Con.State Then Con.Close()
    Con.Open(strCon)
    rst.Open("select * from code where التعليق =" &
TreeView1.SelectedNode.Text & "", Con)
    TextBox1.Text = rst.Fields("التعليقات").Value
    rst.MoveNext()
End Sub
```

```
Private Sub TreeView1_NodeMouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.TreeNodeMouseClickEventArgs) Handles
TreeView1.NodeMouseClick
```


End Sub
End Class

أداة خانة الاختيار RadioButton

تستعمل لتحديد ما إذا كان الخيار مفعّل On و يعود بقيمة True أو غير مفعّل Off و يعود بقيمة False
يمكن استعمال مجموعة من خانات الاختيار لاختيار حالة واحدة فقط وهذا هو الاختلاف بين أداة خانة الاختيار و مربع الاختيار انظر المثال :



VB.Net Code

الاستعلام عن حالة خانة الاختيار

```
If RadioButton1.Checked = True Then  
    MsgBox("نعم : الإجابة")  
Else  
    MsgBox("لا : الإجابة")  
End If
```

أداة مربع الاختيار CheckBox

تستعمل لتحديد ما إذا كان الخيار مفعّل On و يعود بقيمة True أو غير مفعّل Off و يعود بقيمة False
يمكن استعمال مجموعة من مربعات الاختيار لعرض مختلف الخيارات حسب التحديد الذي يختاره المستخدم
(يمكن تحديد خيار واحد أو كل الخيارات) انظر المثال :

VB.Net Code

الاستعلام عن حالة مربع الاختيار

```
If CheckBox1.Checked = True Then
    MsgBox("نعم : الإجابة")
Else
    MsgBox("لا : الإجابة")
End If
```

أداة التلميح ToolTipText

تستخدم لعرض تلميح أو معلومات سريعة عن الأداة عند مرور المؤشر عليها. و الآن قم بإضافة أداة نص إلى النموذج و اكتب ما يلي :

Vb.Net Code

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim Value As New ToolTip
    Value.SetToolTip(TextBox1, "كاملاً الاسم إدخال أرجوا")
End Sub
```

شغل البرنامج ولاحظ الناتج

كما يمكنك إضافة خصائص أخرى


الشرح	الخاصية
تحدد شكل ونمط الرسالة للأيقونة	ToolTipIcon
تحدد الشكل الدائري للتلميح	IsBalloon
تحدد عنوان رسالة التلميح	ToolTipTitle
تحدد النص الذي نضيفه و يظهر كتلميح للأداة	SetToolTip

وهذا مثال يوضح ما قلناه

Vb.Net Code

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim Value As New ToolTip
    Value.ToolTipIcon = ToolTipIcon.Info
    Value.IsBalloon = True
    Value.ToolTipTitle = "تلميح"
    Value.SetToolTip(TextBox1, "كاملاً الاسم إدخال أرجوا")
End Sub
    
```



شغل البرنامج و لاحظ الناتج

أداة NotifyIcon

تستخدم لعرض أيقونة بشريط المهام لنظام التشغيل تعرف المستخدم ماذا يفعل ؟



أو تخبر المستخدم بأن البرنامج في طور التشغيل مثل برامج الحماية (مثل برنامج Avira) كيف تعمل ذلك ؟

نضيف أداة NotifyIcon إلى النموذج ثم نكتب :

VB.Net Code

```

Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    NotifyIcon1.Icon = Me.Icon
    NotifyIcon1.Visible = True
End Sub
    
```

ولاحظ ظهور أيقونة جديدة بشريط المهام لنظام التشغيل تحمل صورة أيقونة النموذج ماذا لو أردت أن تظهر بها رسالة ؟

أولا نتعرف على بعض الخصائص اللازمة

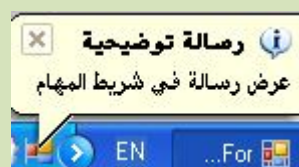
الفصل الثالث الأدوات

الخاصية	الشرح
BalloonTipIcon	تحديد نوعية الأيقونة (أيقونة معلومات Info)
Icon	تحديد الأيقونة
BalloonTipText	تحديد نص الرسالة
BalloonTipTitle	تحديد عنوان الرسالة
ShowBalloonTip	عرض الأيقونة

ثم نكتب هذا الكود :

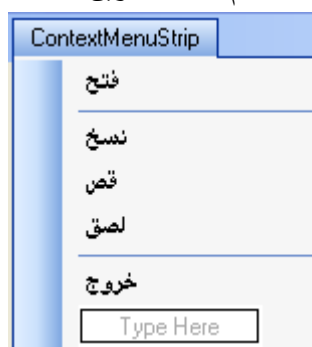
VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As _
System.EventArgs) Handles Me.Load
    NotifyIcon1.Icon = Me.Icon
    NotifyIcon1.BalloonTipIcon = ToolTipIcon.Info
    NotifyIcon1.BalloonTipText = " عرض رسالة في شريط المهام "
    NotifyIcon1.BalloonTipTitle = "العنوان"
    NotifyIcon1.ShowBalloonTip(12)
    NotifyIcon1.Visible = True
End Sub
```



و لاحظ الناتج في شريط المهام

كما يمكن عرض قائمة لمختصرة تظهر عند النقر على الأيقونة كما يلي :
نقوم بإضافة الأداة ContextMenuStrip1 ثم نكتب عناوين القائمة كما يلي :

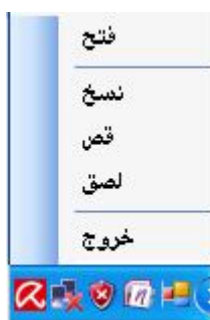


ثم نكتب الكود التالي أسفل الكود السابق

VB.Net Code

```
NotifyIcon1.ContextMenuStrip = ContextMenuStrip1
```

نفذ البرنامج وانقر بزر الفأرة الأيمن و لاحظ الناتج



أداة ErrorProvider

تستخدم للتحقق من البيانات المدخلة في أداة أخرى و تعرض رسالة عند حدوث خطأ تخبر المستخدم عن سبب الخطأ

كيف نعمل ذلك ؟

1. أضف أداة ErrorProvider و مربع النص TextBox إلى النموذج
2. ثم نكتب الكود التالي في حدث TextChanged

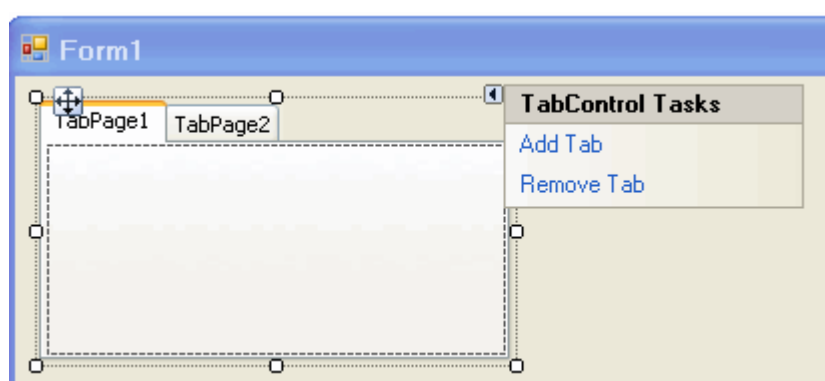
```
VB.Net Code
Private Sub TextBox1_TextChanged(ByVal sender As System.
Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged
    If Char.IsNumber(TextBox1.Text) Then
        ErrorProvider1.Clear()
        ErrorProvider1.Icon = Me.Icon
    Else
        ErrorProvider1.SetError(TextBox1, "رقمية قيمة ادخل")
    End If
End Sub
```

في المثال السابق يتحقق المترجم من القيمة المدخلة في مربع النص إذا كانت غير رقمية تظهر أيقونة مشعة تومض معلنة عن حدوث خطأ وعند تمرير المؤشر فوق الأيقونة تظهر رسالة تخبر المستخدم عن سبب الخطأ .

أداة التبويب TabControl

تستعمل لعرض الأدوات في شكل صفحات و تبويات متعددة . فهي تسهل للمستخدم تجميع الأدوات في مجموعات و تصنيفها حسب خاصية أو وظيفة معينة . وبذلك يمكنه إضافة الكثير من الأدوات في نموذج واحد . كيف نعمل ذلك ؟

1. أضف الأداة TabControl إلى النموذج
2. انقر السهم الصغير تظهر النافذة التالية



3. انقر الخيار Add Tab سيتم إضافة تبويب جديد أما إذا أردت حذفه فانقر الخيار Remove Tab

يمكنك التنقل من تبويب إلى آخر وعرض محتوى كل تبويب بالنقر على عناوين التبويبات .

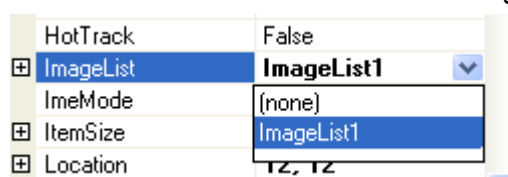
التعامل مع التبويبات من خلال الخاصية TabPages

أضف الأداة قائمة الصور ImageList و أضف إليها بعض الصور المعبرة للدلالة عن عناوين التبويبات

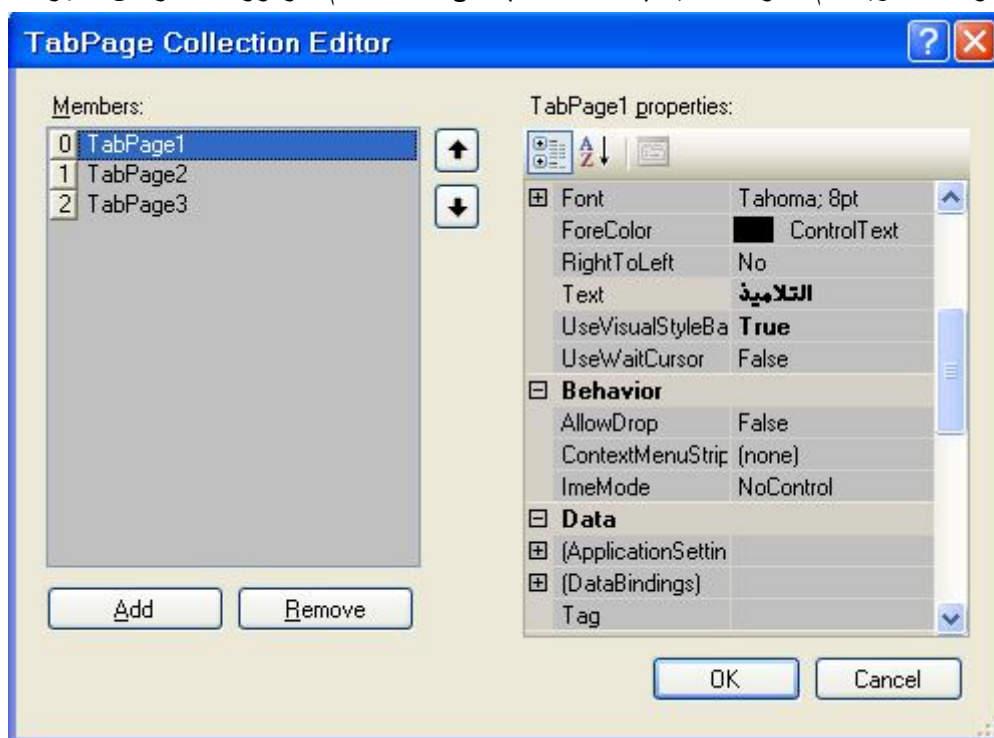
كما تعلمنا سابقا مع الأداة ImageList

بعدها انقر على الأداة من الناحية العلوية وليس على الصفحات ومن نافذة الخصائص ابحث عن الخاصية

ImageList كما بالشكل التالي :



انقر أداة التبويب ثم اختر الخاصية TabPages (Collection) ثم انقر زر الاستعراض تظهر النافذة

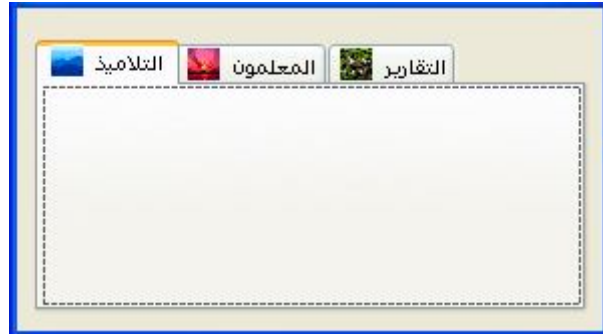


الفصل الثالث الأدوات

من الخاصية Text نحدد اسم التبويب بعد أن نختاره من قائمة Mebers ومن خلال الخاصية ImageIndex نحدد رقم الصورة التي توافق التبويب المختار.

ImageList	Text	Tab
0	التلاميذ	TabPage1
1	المعلمون	TabPage2
2	التقارير	TabPage3

و أخيرا اضغط موافق Ok لتحصل على :



أهم خصائص التبويب

الخاصية	الشرح
Multi Line	True : إمكانية عرض التبويبات بشكل متوازي في حالة امتلاء الصف
Appearance	False : عرض التبويبات بشكل متتالي و لا يمكن عرضها بشكل متوازي
Alignment	تغيير المظهر تعرض رؤوس الصفحات من عدة اتجاهات حسب الطلب

من الأشياء التي قد تحتاجها برمجياً

VB.Net Code

إضافة الصفحات

```
TabControl1.TabPages.Add("", "3 الصفحة", 1)
```

VB.Net Code

تحديد التبويب

```
TabControl1.SelectedIndex = 1
```

الفصل الثالث الأدوات

TabControl1.SelectTab(1)

VB.Net Code

مسح كل الصفحات

TabControl1.TabPages.Clear()

VB.Net Code

حذف تبويب

TabControl1.TabPages.RemoveAt(2)

VB.Net Code

عدد الصفحات

MsgBox(TabControl1.TabPages.Count)

VB.Net Code

إدراج تبويب في موضع

TabControl1.TabPages.Insert(2, "3 التبويب")

حتى الآن لازلنا نتعامل مع رؤوس الصفحات

VB.Net Code

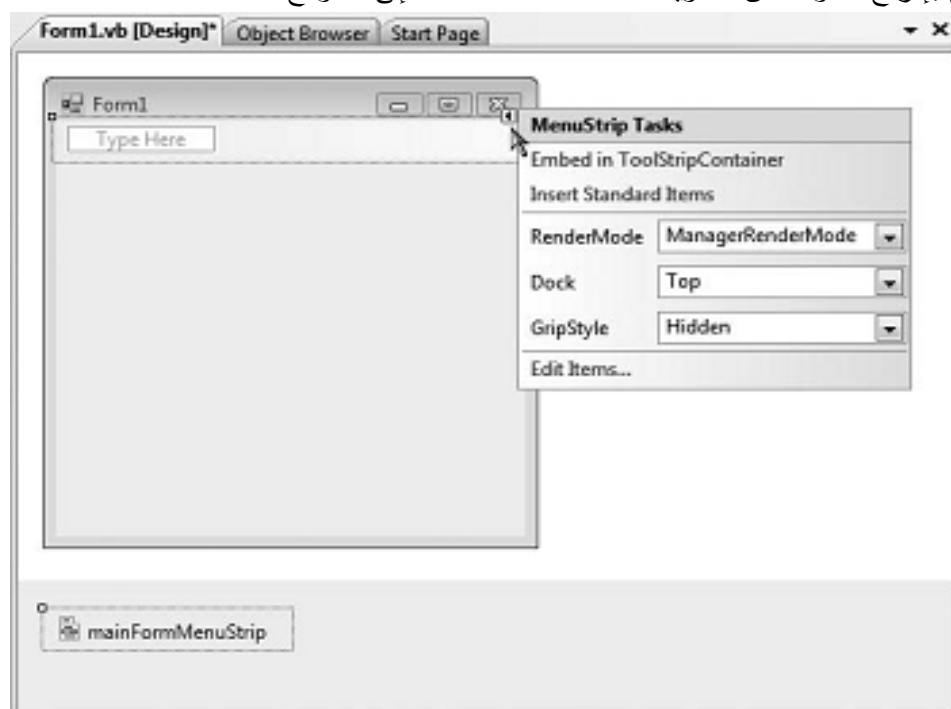
تلوين خلفية الصفحة

TabControl1.TabPages.Item(1).BackColor = Color.BlueViolet

أداة القوائم Menu Strip

تستعمل لإنشاء القوائم ثم القوائم الفرعية.

قم بإدراج الأدوات من الحاوية Menu & Toolbars إلى النموذج



انقر السهم الصغير تظهر نافذة صغيرة بها الخيارات التالية

الفصل الثالث الأدوات

	Imbed in toolStripContainer
إدراج القوائم الافتراضية المعروفة	Insert Standerd items
يقدم لك أنواع القوائم	RenderMode
انطباق القائمة على إحدى حواف النموذج حسب الاختيار	Dock
إدراج مساك إلى القائمة أو إخفائه	GripStyle
محرر عناصر القوائم (وهو الخيار الأهم)	Edit Itmes

إدراج القوائم الافتراضية Insert Standerd items

اختر من النافذة الصغيرة الخيار Insert Standerd items لتحصل على القائم الافتراضية كما بالشكل التالي



يمكنك التعديل في هذه القائمة لتحصل قائمة تناسب برنامجك

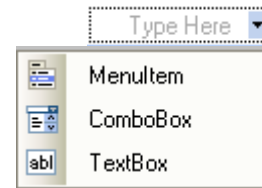
إنشاء القوائم

عند النقر على السهم الصغير نحو الأسفل في القائمة تتسدل أمامك الخيارات التالية

MenuItem : لإضافة قائمة

ComboBox : لإضافة صندوق سرد

TextBox : لإضافة مربع نص



نختار الخيار الأول لإضافة قوائم أو نكتب مباشرة في الخانة Type Here اسم القائمة و ليكن (ملف)
يظهر الشكل كما يلي :



لاحظ أن Vs.Net يتيح لك خيارين الأول على اليمين لإضافة القوائم الرئيسية أما الخيار الثاني في الأسفل

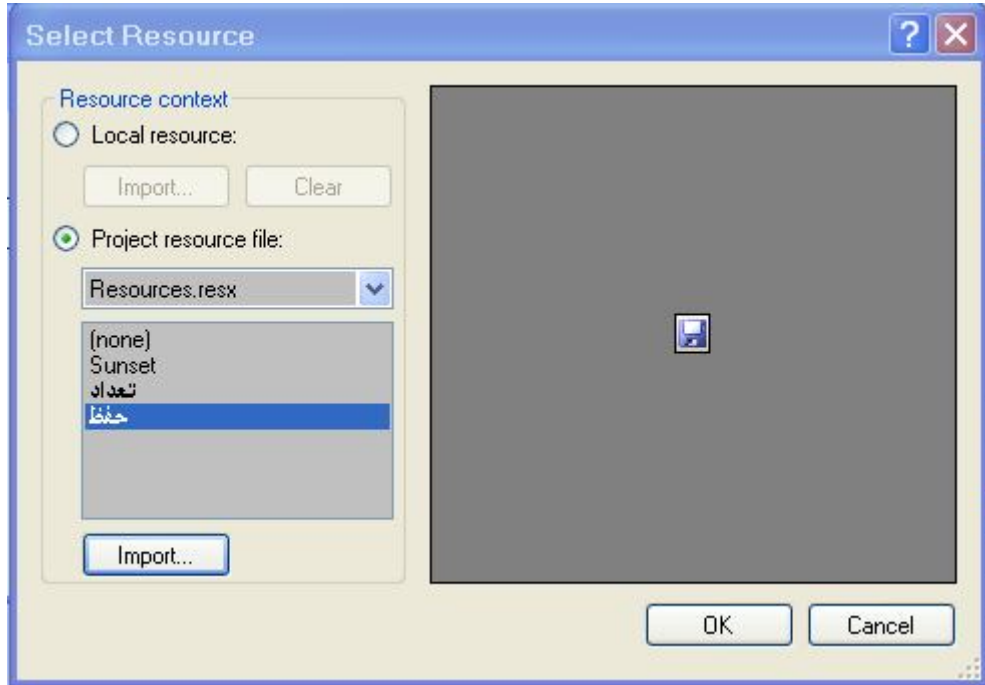


لإضافة القوائم الفرعية.
قم بإضافة قوائم فرعية للقائمة ملف في
خانة Type Here بهذا الشكل .

لإدراج خطوط فاصلة كما في القائمة السابقة نكتب في خانة Type Here الرمز (-)

إدراج أيقونة للقائمة

انقر بجوار عنوان القائمة التي تريد أن تضيف لها أيقونة بزر الفأرة الأيمن ثم اختر من القائمة المنسدلة Set Image تظهر النافذة التالية اختر Import لاستيراد صور . اختر الصورة أو الأيقونة من مربع حوار فتح



ثم اضغط Ok ستجد الأيقونة مضافة بجوار القائمة المحددة.
وهكذا نواصل بالنسبة لباقي القوائم.

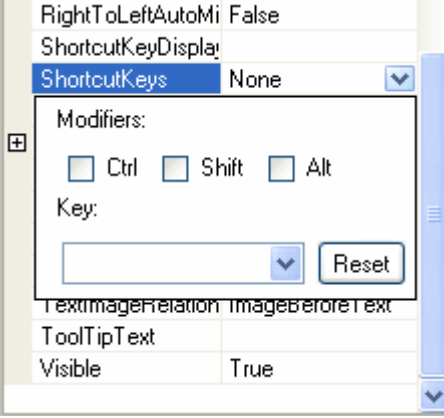
إدراج مفاتيح الاختصار

الفصل الثالث الأدوات

1. حدد العنصر من القائمة الذي ترغب أن تضيف له مفتاح الاختصار (جديد)
2. من نافذة الخصائص اختر الخاصية ShortcutKeys

حدد خانة الاختيار Ctrl
ثم اضغط السهم نحو الأسفل تظهر قائمة طويلة من الأرقام و الحروف اختر منها N
عد إلى القائمة تجد مفتاح الاختصار بجانب العنصر فتح

← Ctrl+N جديد



أهم الخصائص

الخاصية	الشرح
Check on click	وضع علامة صح دلالة على تحديد العنصر من القائمة
Checked	توضع علامة صح لتحديد العنصر وقت التصميم
Display Style	None بلا Text عرض العنوان فقط Image عرض الصورة فقط Image and Text عرض الصورة و العنوان معا
Image	لاستيراد صورة أو أيقونة للعنصر من القائمة
Text	تحديد عنوان عنصر القائمة.

أداة القائمة المختصرة ContextMenuStrip

تستخدم لعرض قائمة عند النقر بزر الفارة الأيمن على أي أداة
ضف الأداة إلى النموذج ثم انقر عليها فتظهر لك بأعلى النافذة قم بتعبئتها بالأوامر المناسبة كما بالشكل

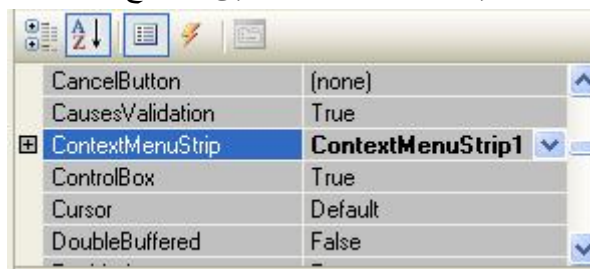


ثم نختار أداة أخرى و لتكن نافذة النموذج نسند هذه القائمة إليها فبمجرد النقر بزر الفأرة الأيمن على النموذج تظهر القائمة المنسدلة و هذا بكتابة الكود

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
    Me.ContextMenuStrip = ContextMenuStrip1
End Sub
```

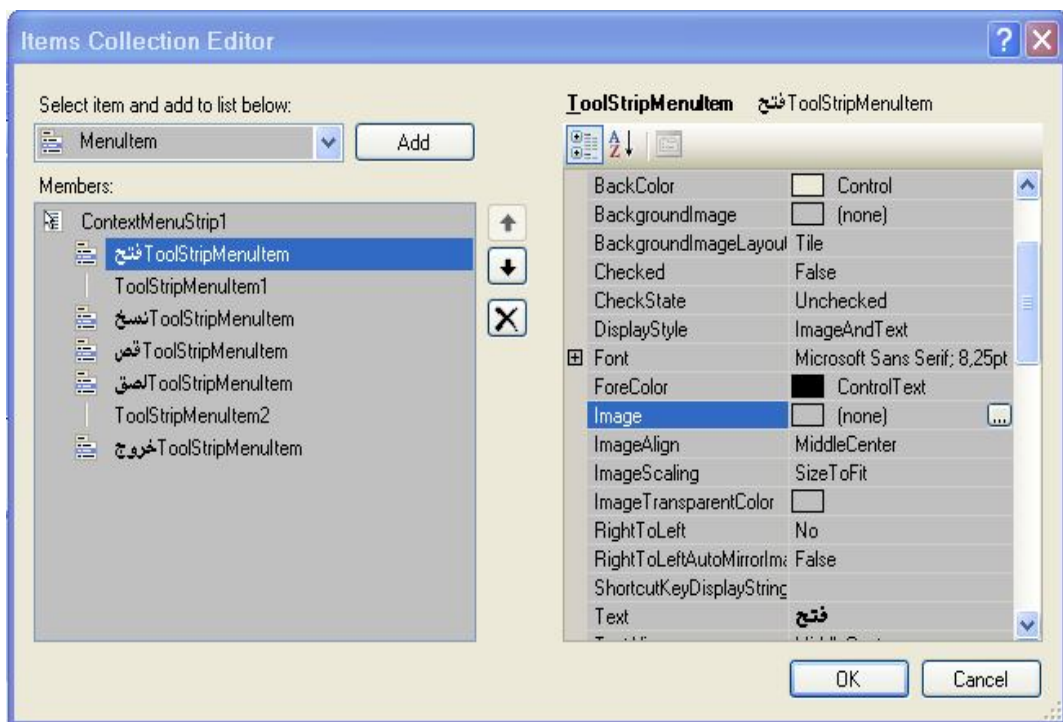
كما يمكنك إسناد القائمة المختصرة إلى النموذج من نافذة الخصائص كما يلي :



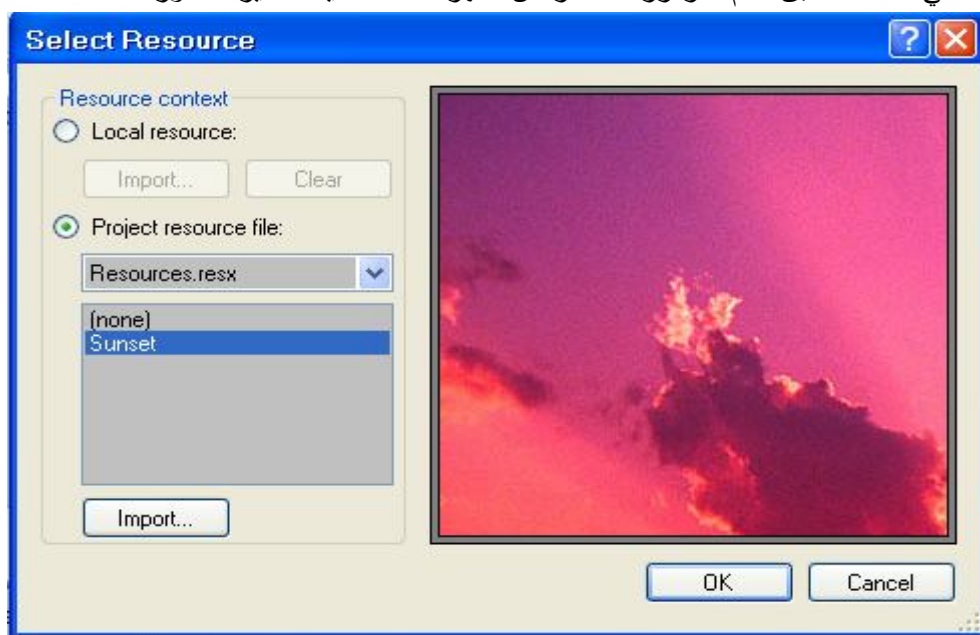
والآن شغل البرنامج وانقر بزر الفأرة الأيمن على النموذج لتجربة القائمة المختصرة كما يمكنك إدراج صور لعناصر (عناوين) القائمة المختصرة و ذلك بإتباع الخطوات

1. حدد الأداة ContextMenuStrip1

2. اختر الخاصية Items (Collection) تظهر النافذة التالية (الشكل 101)



حدد العنصر الذي تريد أن تضيف له صورة من قسم Members وليكن فتح ثم اختر الخاصية Image كما في الشكل السابق . ثم انقر زر الاستعراض تظهر النافذة التالية لاستيراد صورة



اختر زر Import لإضافة صورة من مكان تختاره ثم اضغط موافق Ok و لاحظ الناتج

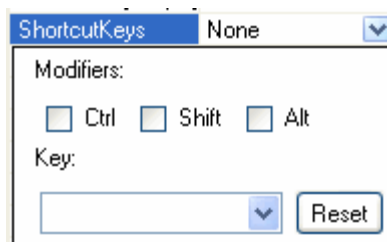


ومن الخصائص الجيدة والتي توفر عليك الوقت الخاصية ShortcutKeys ومن هنا يمكنك تحديد مفتاح مختصر لاي عنصر بالقائمة.

حيث يتم تنفيذه بحال نقر هذا الاختصار . عد إلى النافذة التي أضفنا منها الصورة الشكل 101

1. حدد العنصر الذي تريد أن تضيف له مفتاح الاختصار وليكن نسخ

2. اختر الخاصية ShortcutKeys يظهر الشكل



3. من صندوق الاختيار اختر مفتاح الاختصار و ليكن C ثم حدد مربع الاختيار Ctrl

4. اضغط موافق Ok و لاحظ الناتج



مربعات الحوار القياسية Dialog Boxes

الخواص المشتركة

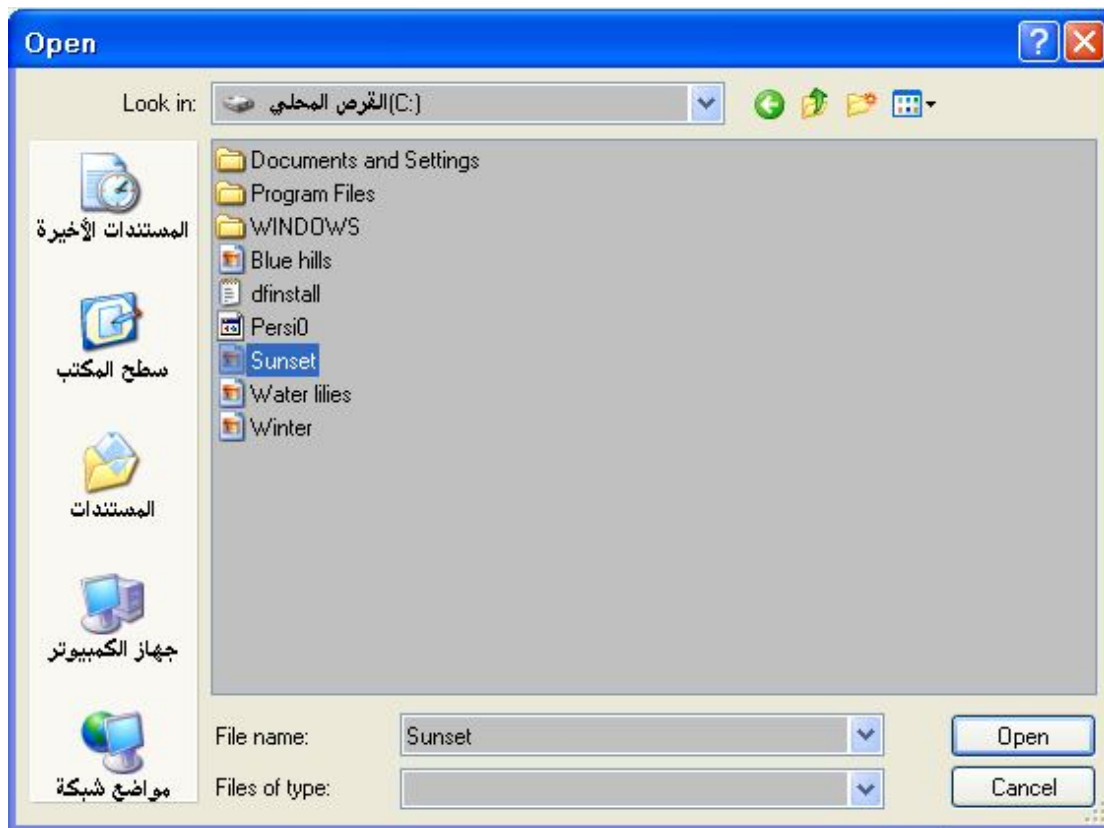
OpenFileDialog1.Title = "ملف فتح"	عنوان صندوق الحوار
OpenFileDialog1.DefaultExt = ".rtf"	نوع الملف الافتراضي
OpenFileDialog1.Filter = "Rich text file (*.rtf) *.rtf text File (*.txt) *.txt Html	تصفية

الفصل الثالث الأدوات

<code>file (*.htm) *.htm all file (*.*) *.*"</code>	
<code>OpenFileDialog1.FilterIndex = 1</code>	رقم المرشح
<code>OpenFileDialog1.ShowDialog()</code>	إظهار صندوق الحوار
<code>If OpenFileDialog1.FileName = "" Then Exit Sub</code>	في حالة عدم تحديد الملف
<code>Dim StrEXT As String</code>	استكشاف أنواع الملف
<code>StrEXT = System.IO.Path.GetExtension(OpenFileDialog1.FileName)</code>	وتحديد نوعه
<code>AddExtension = true ' حفظ الملف تلقائيا بالامتداد الافتراضي</code>	إضافة الامتداد
<code>CheckFileExists = true</code>	التحقق من وجود ملف
<code>CheckPathExists = true</code>	التحقق من وجود مسار
<code>DereferenceLinks = False ' يعيد اسم اختصار الملف</code>	مرجعية الارتباطات
<code>DereferenceLinks = True ' يعيد مسار اسم و مسار الملف</code>	
<code>FileName</code>	اسم الملف
<code>InitialDirectory</code>	المجلد الابتدائي
<code>OpenFileDialog1.InitialDirectory = Application.ExecutablePath</code>	هو المسار الحالي
<code>RestoreDirectory = true ' عرض آخر مجلد ما لم تحدد المجلد المبدئي</code>	احتفظ بالمجلد
<code>ValidateNames =true</code>	التحقق من الاسماء
<code>MultiSelect = true ' يسمح بتحديد عدة ملفات من مجلد واحد</code>	تعدد الاختيارات
<code>FileNames =true ' تحدد مصفوفة من أسماء الملفات المختارة</code>	أسماء الملفات
<code>ShowReadOnly = true ' عرض مربع حوار للقراءة فقط</code>	للقراءة فقط
<code>ReadOnlyChecked = true ' وضع علامة على مربع حوار للقراءة فقط</code>	اختيار للقراءة فقط

مربع الحوار فتح OpenFileDialog

يستعمل لفتح ملف عن طريق مربع الحوار Open التابع لويندوز



لفتح مربع الحوار Open استعمل الطريقة ShowDialog

VB.Net Code

```
If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then  
    Me.Text = OpenFileDialog1.FileName  
End If
```

تصفية الملفات

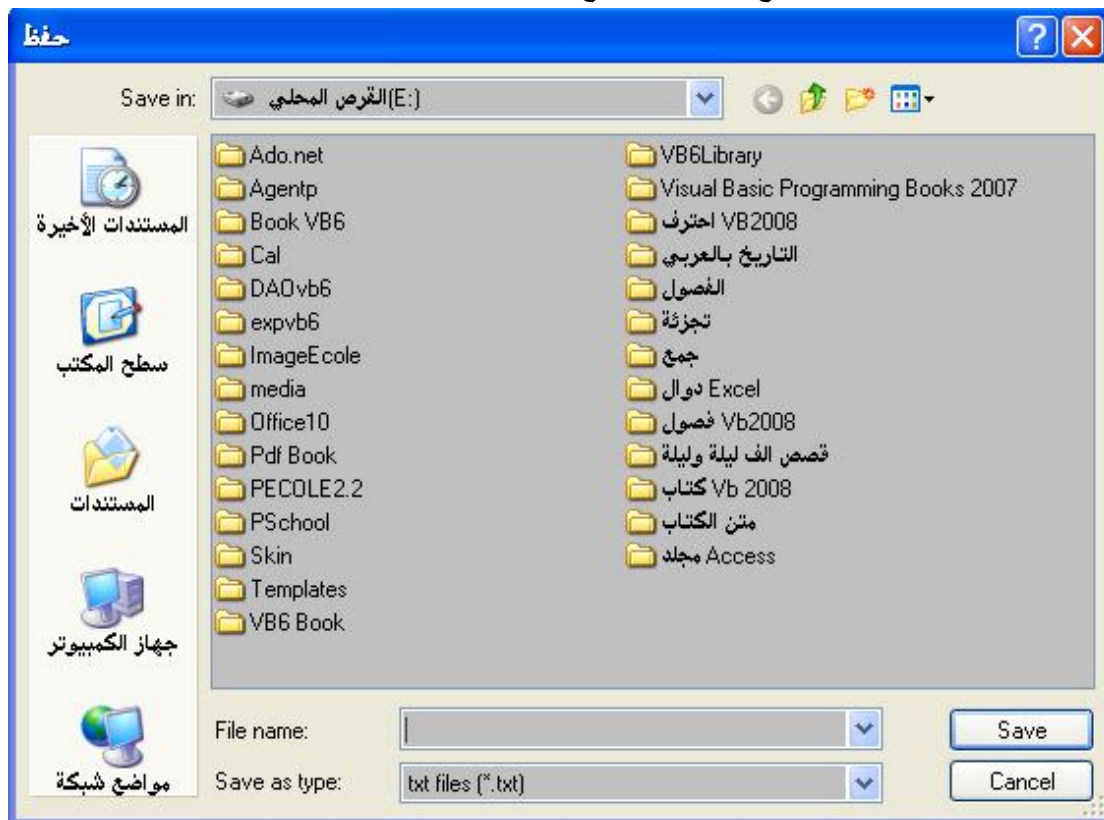
VB.Net Code

```
OpenFileDialog1.Title = "فتح"  
OpenFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"  
  
If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then  
    Me.Text = OpenFileDialog1.FileName
```


End If

مربع الحوار حفظ Save File Dialog

يستعمل لحفظ الملفات عن طريق مربع الحوار حفظ التابع لويندوز



لعرض مربع الحوار حفظ دائما استعمل الطريقة ShowDialog

VB.Net Code

```
SaveFileDialog1.Title = "حفظ"
```

```
SaveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
```

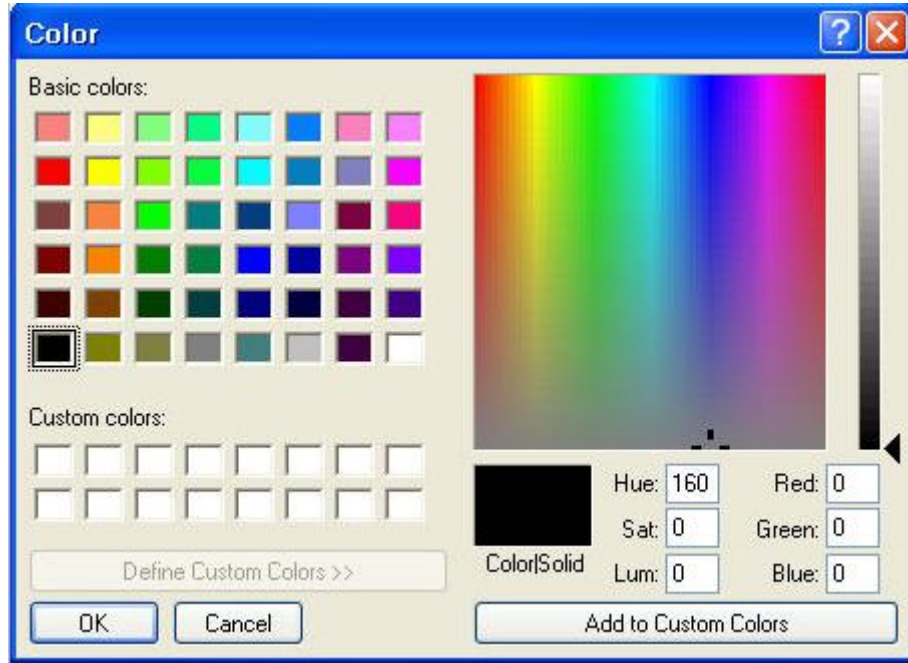
```
If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
```

```
    Me.Text = SaveFileDialog1.FileName
```

```
End If
```

مربع حوار الألوان ColorDialog

يستعمل لعرض نافذة لاختيار الألوان



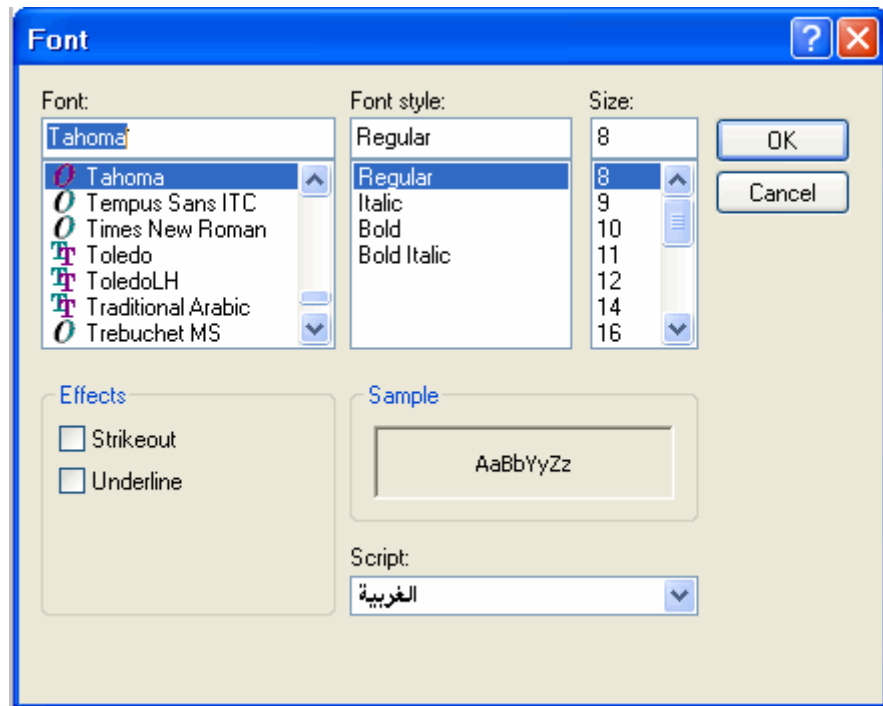
لعرض مربع حوار الألوان نكتب

VB.Net Code

```
If ColorDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then  
    Me.BackColor = ColorDialog1.Color  
End If
```

مربع الحوار الخط FontDialog

يستعمل لفتح نافذة نختار منها نوع الخط وحجمه وسمكه



فتح مربع الحوار الخط

VB.Net Code

```
If ColorDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
    Me.BackColor = ColorDialog1.Color
End If
```

عرض قائمة الخطوط

VB.Net Code

```
Dim f As System.Drawing.Text.InstalledFontCollection = New _
System.Drawing.Text.InstalledFontCollection
    Dim fFamily As FontFamily
    For Each fFamily In f.Families
        ListBox1.Items.Add(fFamily.Name)
    Next
```

مربع الحوار FolderBrowserDialog

يستعمل لتحديد مسار مجلد



استعمال مربع الحوار FolderBrowserDialog لمعرفة مسار المجلد المحدد

VB.Net Code

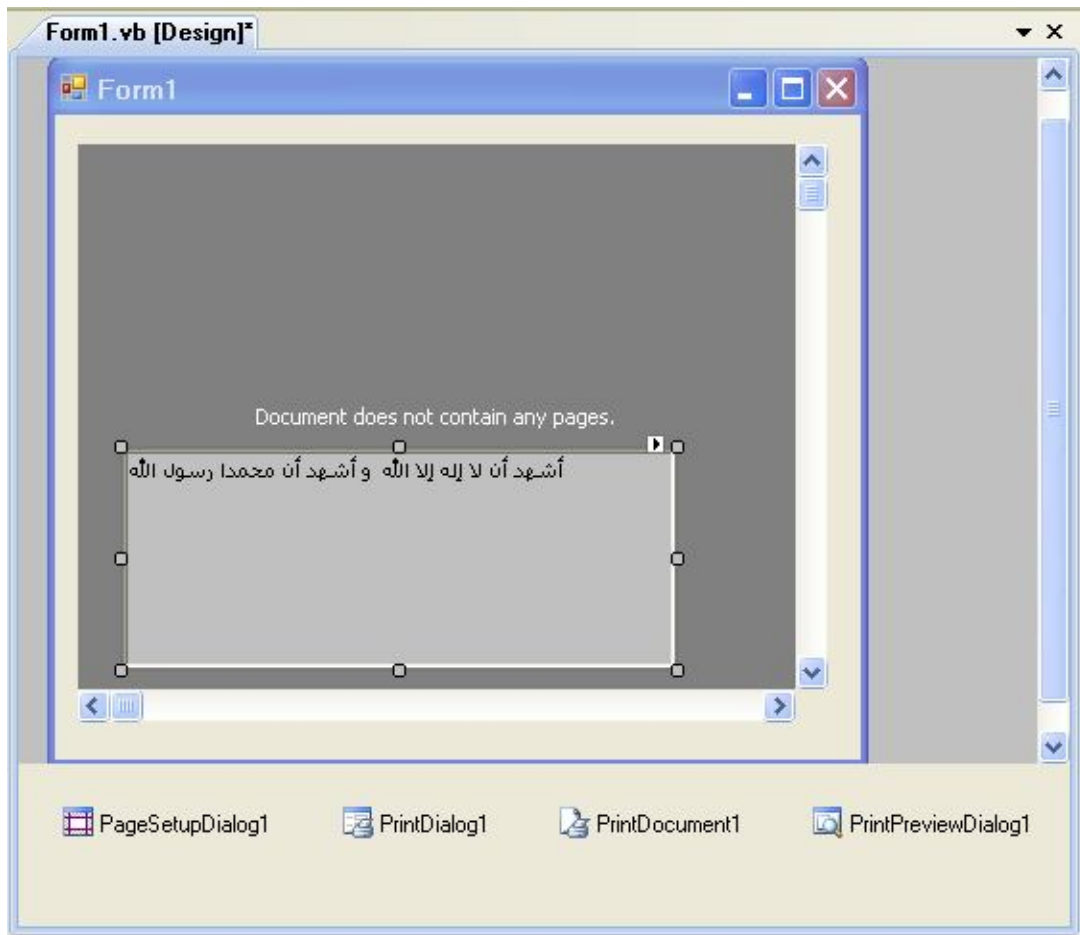
```
If FolderBrowserDialog1.ShowDialog = Windows.Forms.DialogResult.OK
Then
    MsgBox(FolderBrowserDialog1.SelectedPath)
End If
```

أدوات الطباعة

1. أنشئ مشروعاً جديداً
2. قم بإضافة الأدوات التالية على النموذج

الأداة	الشرح
PrintPreviewDialog	تظهر المعاينة في نافذة مستقلة بها
PageSetupDialog	إعداد الصفحة
PrintDialog	طباعة
PrintDocument	المستند
PrintPreviewControl	تظهر المعاينة على النموذج كما تشاهده
RichTextBox	النص العني

اكتب شيئاً في النص . يبدو الشكل كما يلي :



نربط الأدوات «PrintPreviewControl» ، «PrintDialog» ، «PageSetupDialog»

PrintPreviewDialog

مع المستند من خلال الخاصية Document

+	(ApplicationSetting	
	(Name)	PageSetupDialog1
	AllowMargins	True
	AllowOrientation	True
	AllowPaper	True
	AllowPrinter	True
+	Document	PrintDocument1 ▼
	EnableMetric	(none)
	GenerateMember	PrintDocument1
+	MinMargins	0, 0, 0, 0

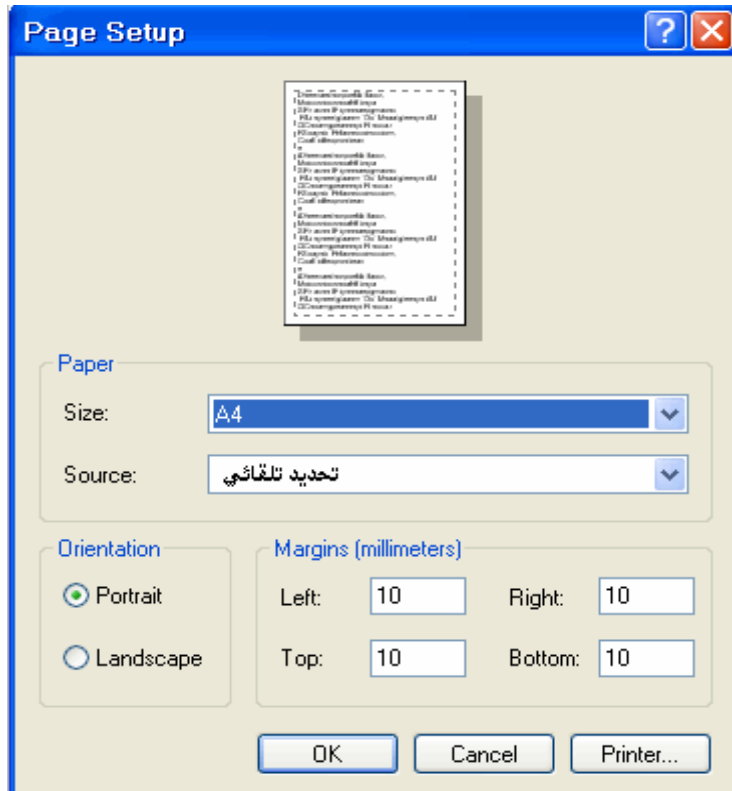
ثم نكتب في زر إعداد الصفحة

Vb.Net Code

```
Private Sub إعدادالصفحة!ToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
إعدادالصفحة!ToolStripMenuItem.Click
PageSetupDialog1.Document = PrintDocument1
```

```
PageSetupDialog1.ShowDialog()
End Sub
```

فبعدها ستحصل على هذه النافذة أنظر للشكل التالي

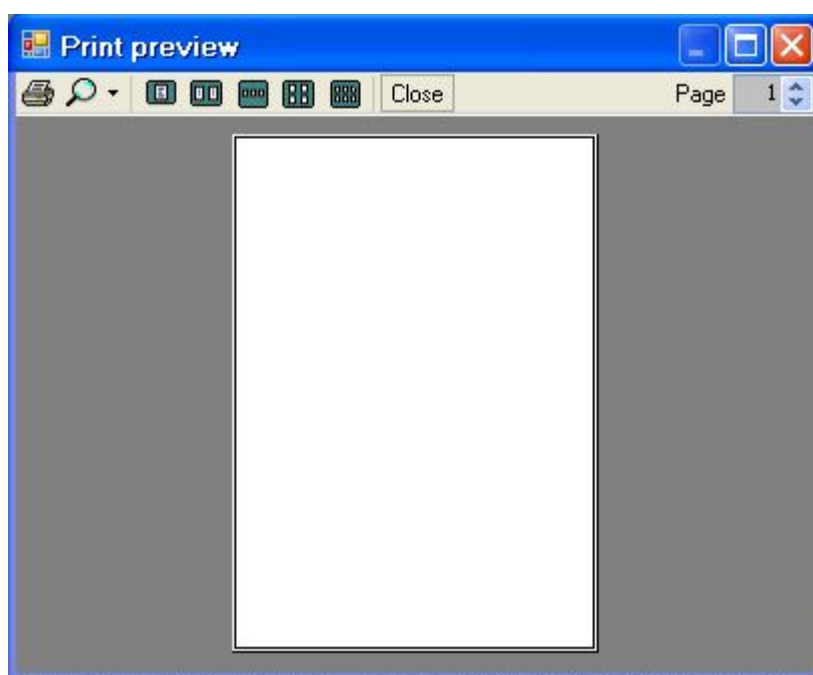


ثم نكتب في زر معاينة قبل الطباعة

Vb.Net Code

```
Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripMenuItem.Click
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()
End Sub
```

فستحصل على النافذة التالية



طبعاً المستند خالي تماماً لأننا وحتى الآن لم نقم بالتعامل مع المستند PrintDocument وكل ما فعلناه هو ربطه بهذه الأدوات فقط.

الآن جاء الدور للتعامل مع المستند .

أنقر عليه نقرتين لتنتقل إلى الحدث التالي وكما بالشكل واكتب

Vb.Net Code

```
Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim rectDraw As New RectangleF(e.MarginBounds.Left, e.MarginBounds.Top, e.MarginBounds.Width, e.MarginBounds.Height)
    Dim PrintFont As New Font("Arial", 10)
    Dim strFormat As New StringFormat
    strFormat.Trimming = StringTrimming.Word
    e.Graphics.DrawString(Me.Rtbdoc.Text, PrintFont, Brushes.Black, rectDraw, strFormat)
End Sub
```

بعد ضغط زر معاينة قبل الطباعة ستحصل على

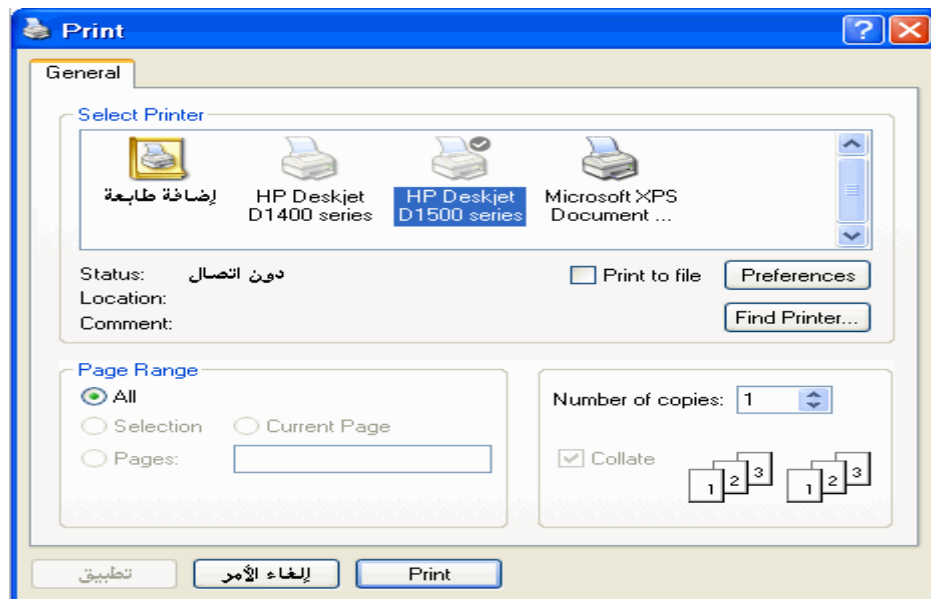


ثم نكتب في زر طباعة

Vb.Net Code

```
Private Sub طباعةToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles طباعةToolStripMenuItem.Click
    PrintDialog1.Document = PrintDocument1
    If PrintDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        PrintDocument1.Print()
    End If
End Sub
```

يمكننا رؤية هذه النافذة



خصائص Rich Text Box

استعمالها :

نسمي هذه الأداة "مربع النصّ الثري"، وهو فعلا ثريّ بالإمكانيات التي تتيح لك تنسيق النصّ بطريقة تقارب ما تراه في وثائق Microsoft Word.

إنّ هذه الأداة تمتلك جميع قدرات مربع النصّ العاديّ، بالإضافة لقدرتك على اختيار خطّ وحجم ولون وهامش أيّ جزء من النصّ على حدة، كما يمكنك إضافة الصور داخل النصّ.

خصائصها

كثير من خصائص هذه الأداة مماثلة لخصائص مربع النصّ التقليديّ، لذلك فلن نتحدّث إلا عن الخصائص الجديدة و إليك الجدول التالي يلخص لنا أهم خصائص النصّ المحدد

الخاصية	الشرح	مثال
SelectedText	النصّ المحدد.. وهنا نتعامل مع الحروف فقط،	<code>RichTextBox1.SelectedText = UCASE (RichTextBox1.SelectedText)</code> تحويل حروف النصّ المحدد إلى حروف كبيرة
SelectedRTF	تنسيق النصّ المحدّد	تستفيد من هذه الخاصية في نقل نصّ بتنسيقه من مربع نصّ منسق إلى آخر
SelectionStart	موضع بداية التحديد	<code>RichTextBox1.SelectionStart = 4 RichTextBox1.Focus ()</code>
SelectionLength	طول النصّ المحدد	<code>MsgBox (RichTextBox1.SelectionLength)</code>
SelectionFont	خط النصّ المحدد	<code>FontDialog1.ShowDialog () RichTextBox1.SelectionFont = FontDialog1.Font</code>
SelectionAlignment	محاذاة النصّ المحدد	<code>RichTextBox1.SelectionAlignment=HorizontalAlignment.Center</code>
SelectionColor	لون النصّ المحدد	<code>ColorDialog1.ShowDialog () RichTextBox1.SelectionColor = ColorDialog1.Color</code>
SelectionProtected	حماية النصّ المحدد	<code>RichTextBox1.SelectionProtected = True</code> لا يمكن للمستخدم التغيير في النصّ المحدد حتى ولو بعد حذف التحديد
SelectionBullet	علامة تعداد نقطي	<code>RichTextBox1.SelectionBullet = True</code>
SelectionIndent	هامش بداية السطر	<code>RichTextBox1.SelectionIndent = True</code>

الفصل الثالث الأدوات

مثال	الشرح	الخاصية
<code>RichTextBox1.SelectionColor = Color.Blue</code> <code>RichTextBox2.Text = RichTextBox1.Text</code>	النص غير منسق	Text
<code>RichTextBox1.SelectionColor = Color.Blue</code> <code>RichTextBox2.Rtf = RichTextBox1.Rtf</code>	النص منسق	RTF
<code>RichTextBox1.SelectionBullet = True</code> <code>RichTextBox1.BulletIndent = 30</code>	هامش العلامة النقطية	BulletIndent
لو جعلت قيمة هذه الخاصية True، فسيتمكن المستخدم من تحديد كل الكلمة بنقرها مرتين بالفأرة.	تحديد الكلمة تلقائياً	AutoSelectWord
<code>Me.Text = RichTextBox1.UndoActionName</code> Delete إذا كانت تراجعاً عن كتابة Typing أو تراجع عن حذف Unknown أو أي شيء آخر غير معروف	اسم عملية التراجع	UndoActionName
<code>Me.Text = RichTextBox1.RedoActionName</code>	اسم عملية الإعادة	RedoActionName
<code>Sub RichTextBox1_LinkClicked(ByVal sender As Object, ByVal e As System.Windows.Forms.LinkClickedEventArgs) _ Handles RichTextBox1.LinkClicked</code> <code>System.Diagnostics.Process.Start(e.LinkText)</code> <code>End Sub</code>	تنسيق عنوان انترنت	DetectURLs

وسائل أداة Rich Text Box

الشرح	الوسيلة
تستعمل حفظ محتويات مربع النص المنسق في ملف	SaveFile
لتحميل محتويات ملف و عرضه في مربع النص المنسق	LoadFile
نسخ محتوى مربع النص المنسق في الحافظة	Copy
قص محتوى مربع النص في الحافظة	Cut
لصق محتوى نص من الحافظة إلى مربع النص المنسق	Past
تقوم بالبحث في نص مربع النص المنسق	Find
لتعرف الحرف الموجود في مربع النص المنسق في إحداثيات معينة أو قريباً منها.	GetCharFromPosition
معرفة موضع الحرف من الإحداثيات	GetCharIndexFromPosition

حفظ لملف SaveFile :



RichTextBox1.SaveFile(نوع الملف، مسار الملف)

ويمكن ألا ترسل المعامل الثاني، حيث سيتم حفظ الملف بامتداد .rtf. للمحافظة على تنسيقه.. كما يمكن أن ترسل المعامل الثاني بإحدى قيم المرقم RichTextBoxStreamType، وهي:

تحفظ حروف النصّ بغير تنسيق.	PlainText
تحفظ النصّ بدون تنسيقه، ودون أن تحفظ أيّ كائنات OLE موجودة بداخل مربع النصّ كالصور مثلاً.	RichNoOLEObjs
تحفظ النصّ بتنسيقه.	RichText
تحفظ النصّ بدون تنسيق، ولكن تحفظ كائنات OLE.	TextTextOLEObjs
تحفظ النصّ بتنسيق Unicode.	UnicodePlainText

VB.Net Code

```
RichTextBox1.SaveFile("c:\test.rtf", RichTextBoxStreamType.RichText)
```

تحميل ملف LoadFile



RichTextBox1.LoadFile(نوع الملف، مسار الملف)

حيث أن نوع الملف هو معامل اختياري. و تعرف على قيمه في وسيلة الحفظ

VB.Net Code

```
RichTextBox1. LoadFile ("c:\test.rtf", RichTextBoxStreamType.RichText)
```

أوامر القص و النسخ و اللصق

VB.Net Code

```
RichTextBox1.Copy()
RichTextBox1.Cut()
RichTextBox1.Paste()
```

```
Clipboard.SetDataObject(RichTextBox1.SelectedText)' نسخ
```

ألصق

```
With Clipboard.GetDataObject
If .GetDataPresent(DataFormats.Text) Then
RichTextBox1.SelectedText = .GetData(DataFormats.Text)
End If
End With
```

بحث Find



RichTextBox1.Find(طراز البحث, نص)

Pos = RichTextBox1.Find(طراز البحث, موضع نهاية البحث, موضع بداية البحث, نص)

متغير من نوع عددي يعيد بموضع كلمة البحث : Pos

تعيد هذه الوسيلة موضع أول حرف منه مع تحديد نصّ البحث في مربع النصّ، وإن لم يكن موجوداً فإنّها تعيد -1.

حيث يأخذ طراز البحث واحدة من قيم المرقّم RichTextBoxFinds التالية:

البحث مع مراعاة حالة الأحرف.	MatchCase
عند العثور على النصّ لا يتمّ تحديده في مربع النصّ.	NoHighlight
يبدأ البحث من نهاية مربع النصّ لبدايته.	Reverse
البحث عن النصّ بحيث يكون كلمة كاملة مستقلة، وليس جزءاً من كلمة أخرى في مربع النصّ.	WholeWord
لا يوجد طراز بحث خاصّ.	None

ويمكنك أن تدمج أكثر من قيمة من هذه القيم معاً باستخدام عمليّة Or .. انظر المثال التالي:

VB.Net Code

```
Dim Pos As Integer
```

```
Pos = RichTextBox1.Find("كلمة", RichTextBoxFinds.WholeWord _
```

```
Or RichTextBoxFinds.Reverse)
```

```
MsgBox(Pos)
```

البحث عن مصفوفة من الحروف حيث تعيد موضع أول حرف تجده من هذه الحروف.

VB.Net Code

```
Dim Pos As Integer
```

```
Dim Ch() As Char = {"a", "b", "c"}
```

```
Pos = RichTextBox1.Find(Ch)
```

```
MsgBox(Pos)
```

معرفة الحرف من الإحداثيات GetCharFromPosition:

استخدم هذه الوسيلة لتعرف الحرف الموجود في مربع النصّ المنسق في إحداثيات معيّنة أو قريباً منها.. ويمكنك استخدام هذه الخاصية لمعرفة أقرب حرف للموضع الذي يمرّ فوقه مؤشر الفأرة، وذلك بكتابة الجملة التالية في حدث حركة الفأرة MouseEventArgs:

VB.Net Code

```
Private Sub RichTextBox1_MouseMove(ByVal sender As Object, ByVal e _  
As System.Windows.Forms.MouseEventArgs) _
```

```
Handles RichTextBox1.MouseMove
Dim P As New Point(e.X, e.Y)
' للنموذج كعنوان الفأرة لمؤشر حرف أقرب اعرض '
Me.Text = RichTextBox1.GetCharFromPosition(P)
End Sub
```

معرفة موضع الحرف من الإحداثيات `GetCharIndexFromPosition`

هذه الوسيلة التي تعيد لك رقم الحرف القريب من مؤشر الفأرة في مربع النصّ

VB.Net Code

```
Private Sub RichTextBox1_MouseMove(ByVal sender As Object, ByVal e _
As System.Windows.Forms.MouseEventArgs) _
Handles RichTextBox1.MouseMove
Dim P As New Point(e.X, e.Y)
Me.Text = RichTextBox1.GetCharIndexFromPosition(P)
End Sub
```

مشروع برنامج الكتابة

1. أنشئ مشروعاً جديداً واختر اسماً له
2. قم بإدراج الأدوات التالية :

RichTextBox ، StatusStrip ، ToolStrip ، MenuStrip

أدوات الطباعة ، مربعات الحوار القياسية

3. إنشاء القوائم (راجع موضوع إنشاء قوائم)

القائمة	الاسم البرمجي
ملف	FileToolStripMenuItem
تحرير	EditToolStripMenuItem
خط	FontToolStripMenuItem
إدراج	InsertToolStripMenuItem

القوائم الفرعية التابعة للقائمة ملف

القائمة الفرعية	الاسم البرمجي
جديد	NewToolStripMenuItem
فتح	OpenToolStripMenuItem
حفظ	SaveToolStripMenuItem
حفظ باسم	SaveAsToolStripMenuItem
طباعة	PrintToolStripMenuItem
إعداد الصفحة	MiseMenuItem
معاينة	PrintPreviewToolStripMenuItem

الفصل الثالث الأدوات

ExitToolStripMenuItem	خروج
-----------------------	------

القوائم الفرعية الخاصة بقائمة تحرير

القائمة الفرعية	الاسم البرمجي
تراجع	UndoToolStripMenuItem
إعادة	RedoToolStripMenuItem
قص	CutToolStripMenuItem
نسخ	CopyToolStripMenuItem
ألصق	PasteToolStripMenuItem
تحديد الكل	SelectAllToolStripMenuItem
بحث	FindToolStripMenuItem
استبدال	ReplaceToolStripMenuItem

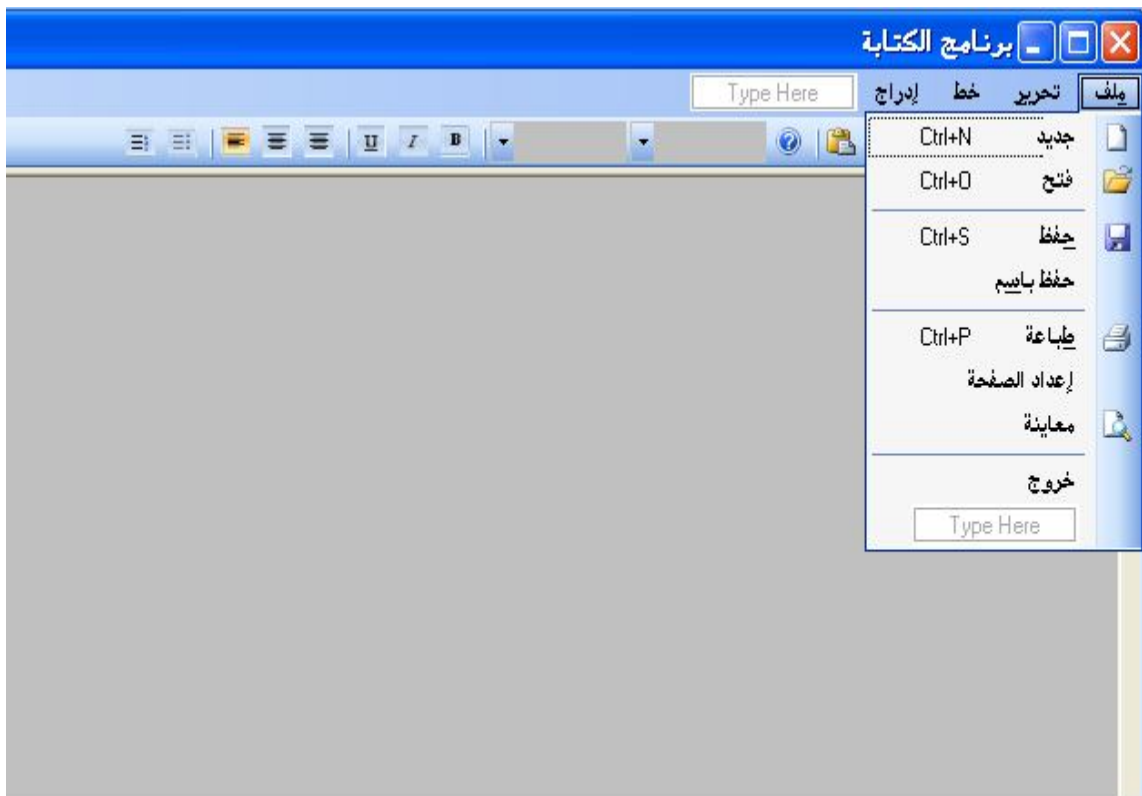
القوائم الفرعية الخاصة بالخط

القائمة الفرعية	الاسم البرمجي
تنسيق الخط	CustomFont

القوائم الفرعية الخاصة بالقائمة إدراج

القائمة الفرعية	الاسم البرمجي
إدراج صورة	InsertPicture

قم بالإضافات اللازمة إلى الشريط ToolStrip حتى تحصل الشكل التالي :



4. كتابة التعليمات

نقوم باستيراد مجال الأسماء الخاص برسم الصور

`Imports System.Drawing.Image`

ثم نعرف المتغيرات العامة التالية

`Dim CurrentFile As String` 'تعريف متغير لحفظ اسم الملف'

`Dim CheckPrint As Integer` 'متغير للطباعة'

```

VB.Net Code
جديد
Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NewToolStripMenuItem.Click
If RtbDoc.Modified = True Then
If MessageBox.Show("حفظ", "التغيرات حفظ في ترغب هل",
MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then
Exit Sub
Else
RtbDoc.Clear()
End If
Else
RtbDoc.Clear()
End If
CurrentFile = ""
Me.Text = "جديد ملف"
End Sub
    
```

VB.Net Code

فتح

```
Private Sub OpenToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OpenToolStripMenuItem.Click
    If RtbDoc.Modified = True Then
        If MessageBox.Show("حفظ", "التغيرات حفظ في ترغب هل",
        MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then
            Exit Sub
        Else
            OpenFile()
        End If
    Else
        OpenFile()
    End If
End Sub
```

VB.Net Code

إجراء الفتح

```
Sub OpenFile()
    OpenFileDialog1.Title = "ملف فتح"
    OpenFileDialog1.DefaultExt = ".rtf"
    OpenFileDialog1.Filter = "Rich text file (*.rtf)|*.rtf| text File (*.txt)|*.txt| Html file (*.htm)|*.htm| all file (*.*)|*.*"
    OpenFileDialog1.FilterIndex = 1
    OpenFileDialog1.ShowDialog()
    If OpenFileDialog1.FileName = "" Then Exit Sub
    Dim StrEXT As String
    StrEXT = System.IO.Path.GetExtension(OpenFileDialog1.FileName)
    StrEXT = StrEXT.ToUpper
    Select Case StrEXT
        Case Is = ".RTF"
            RtbDoc.LoadFile(OpenFileDialog1.FileName, RichTextBoxStreamType.RichText)
        Case Else
            Dim Txtreader As System.IO.StreamReader
            Txtreader = New System.IO.StreamReader(OpenFileDialog1.FileName)
            RtbDoc.Text = Txtreader.ReadToEnd
            Txtreader.Close()
            Txtreader = Nothing
            RtbDoc.SelectionStart = 0
            RtbDoc.SelectionLength = 0
        End Select
    CurrentFile = OpenFileDialog1.FileName
    RtbDoc.Modified = False
    Me.Text = CurrentFile.ToString
End Sub
```

VB.Net Code

حفظ باسم


```

Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles SaveAsToolStripMenuItem.Click
    SaveFileDialog1.Title = "ملف فتح"
    SaveFileDialog1.DefaultExt = ".rtf"
    SaveFileDialog1.Filter = "Rich text file (*.rtf)|*.rtf| text File (*.txt)|*.txt| Html file
(*.htm)|*.htm| all file (*.*)|*.*"
    SaveFileDialog1.FilterIndex = 1
    SaveFileDialog1.ShowDialog()
    If SaveFileDialog1.FileName = "" Then Exit Sub
    Dim StrEXT As String
    StrEXT = System.IO.Path.GetExtension(SaveFileDialog1.FileName)
    StrEXT = StrEXT.ToUpper
    Select Case StrEXT
        Case Is = ".RTF"
            RtbDoc.SaveFile(SaveFileDialog1.FileName, RichTextBoxStreamType.RichText)
        Case Else
            Dim txtWriter As System.IO.StreamWriter
            txtWriter = New System.IO.StreamWriter(SaveFileDialog1.FileName)
            txtWriter.Write(RtbDoc.Text)
            txtWriter.Close()
            txtWriter = Nothing
            RtbDoc.SelectionStart = 0
            RtbDoc.SelectionLength = 0
        End Select
    CurrentFile = SaveFileDialog1.FileName
    RtbDoc.Modified = False
    Me.Text = CurrentFile.ToString
End Sub

```

 VB.Net Code

حفظ

```

Private Sub SaveToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SaveToolStripMenuItem.Click
    If CurrentFile = "" Then
        SaveAsToolStripMenuItem_Click(Me, e)
        Exit Sub
    End If
    Dim StrEXT As String
    StrEXT = System.IO.Path.GetExtension(OpenFileDialog1.FileName)
    StrEXT = StrEXT.ToUpper
    Select Case StrEXT
        Case Is = ".RTF"
            RtbDoc.SaveFile(CurrentFile)
        Case Else
            Dim txtWriter As System.IO.StreamWriter
            txtWriter = New System.IO.StreamWriter(CurrentFile)
            txtWriter.Write(RtbDoc.Text)

```

```

txtWriter.Close()
txtWriter = Nothing
RtbDoc.SelectionStart = 0
RtbDoc.SelectionLength = 0
RtbDoc.Modified = False
End Select
Me.Text = CurrentFile.ToString

End Sub

```

VB.Net Code إعداد الصفحة

```

PageSetupDialog1.Document = PrintDocument1
PageSetupDialog1.ShowDialog()

```

VB.Net Code معاينة

```

Private Sub PrintPreviewToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
PrintPreviewToolStripMenuItem.Click
PrintPreviewDialog1.Document = PrintDocument1
PrintPreviewDialog1.ShowDialog()
End Sub

```

VB.Net Code طباعة

```

Private Sub PrintToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PrintToolStripMenuItem.Click
PrintDialog1.Document = PrintDocument1
If PrintDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
PrintDocument1.Print()
End If
End Sub

```

VB.Net Code معالجة المستند

```

Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object,
ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles
PrintDocument1.PrintPage
Dim rectDraw As New RectangleF(e.MarginBounds.Left,
e.MarginBounds.Top, _
e.MarginBounds.Width, e.MarginBounds.Height)

Dim PrintFont As New Font("Arial", 10)
Dim strFormat As New StringFormat

```

```
strFormat.Trimming = StringTrimming.Word

e.Graphics.DrawString(Me.RtbDoc.Text, PrintFont, _
    Brushes.Black, rectDraw, strFormat)
End Sub
```

الأوامر الخاصة بقائمة تحرير

VB.Net Code

تراجع

```
Private Sub RedoToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles RedoToolStripMenuItem.Click
    If RtbDoc.CanUndo Then RtbDoc.Undo()
End Sub
```

VB.Net Code

إعادة

```
Private Sub RedoToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles RedoToolStripMenuItem.Click
    If RtbDoc.CanRedo Then RtbDoc.Redo()
End Sub
```

VB.Net Code

قص

```
Private Sub CutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles CutToolStripMenuItem.Click
    Try
        RtbDoc.Cut()
    Catch ex As Exception
        MsgBox("القص يمكن لا", MsgBoxStyle.Critical)
    End Try
End Sub
```

```
-----
Clipboard.SetDataObject(Editor.SelectedText)
Editor.SelectedText = ""
```

VB.Net Code

نسخ

```
Private Sub CopyToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles CopyToolStripMenuItem.Click
    Try
        RtbDoc.Copy()
    Catch ex As Exception
        MsgBox("فارغ النص يمكن لا", MsgBoxStyle.Critical)
    End Try
```

End Sub

Clipboard.SetDataObject(Editor.SelectedText)

VB.Net Code

لصق

```
Private Sub PasteToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PasteToolStripMenuItem.Click
    Try
        RtbDoc.Paste()
    Catch ex As Exception
        MsgBox("اللتصق يمكن لا", MsgBoxStyle.Critical)
    End Try
End Sub
```

```
With Clipboard.GetDataObject
If .GetDataPresent(DataFormats.Text) Then
Editor.SelectedText = .GetData(DataFormats.Text)
End If
End With
```

VB.Net Code

تحديد الكل

RtbDoc.SelectAll()

VB.Net Code

البحث الأول

```
Dim StartPosition As Integer
Dim SearchType As CompareMethod
If RadioButton1.Checked = True Then
    SearchType = CompareMethod.Binary
Else
    SearchType = CompareMethod.Text
End If
StartPosition = InStr(1, Form1.RtbDoc.Text, TextBox1.Text, SearchType)
If StartPosition = 0 Then
    MsgBox("الكلمة على العثور يتم لم" & " " & TextBox1.Text,
MsgBoxStyle.Critical)
Exit Sub
End If
Form1.RtbDoc.Select(StartPosition - 1, TextBox1.Text.Length)
Form1.RtbDoc.ScrollToCaret()
Form1.Focus()
```

VB.Net Code

البحث عن التالي

```
Dim StartPosition As Integer = Form1.RtbDoc.SelectionStart + 2
Dim SearchType As CompareMethod
If RadioButton1.Checked = True Then
    SearchType = CompareMethod.Binary
Else
    SearchType = CompareMethod.Text
End If
StartPosition = InStr(StartPosition, Form1.RtbDoc.Text, TextBox1.Text,
SearchType)
If StartPosition = 0 Then
    MsgBox("الكلمة على العثور يتم لم" & " " & TextBox1.Text,
MsgBoxStyle.Critical)
Exit Sub
End If
Form1.RtbDoc.Select(StartPosition - 1, TextBox1.Text.Length)
Form1.RtbDoc.ScrollToCaret()
Form1.RtbDoc.Focus()
```

VB.Net Code

استبدال

```
If Form1.RtbDoc.SelectedText = TextBox1.Text Then
    Form1.RtbDoc.SelectedText = TextBox2.Text
End If
Button1_Click(sender, e) ' التالي عن البحث إجراء استدعاء '
```

VB.Net Code

استبدال الكل

```
Dim CurPos, CurSel As Integer
Dim SrchMode As CompareMethod
' الاختيار مربع قيمة على بناء البحث، نوع متغير قيمة تحديد '
If RadioButton1.Checked = True Then
    ' (الحرف حالة مراعاة) هي كما الحرف حالة يطابق أنه يعني مما ثنائي، البحث '
    SrchMode = CompareMethod.Binary
Else
    ' الحرف حالة يراعي لا أنه يعني مما نصي، البحث '
    SrchMode = CompareMethod.Text
End If
With Form1.RtbDoc
```

```

CurPos = .SelectionStart
CurSel = .SelectionLength
.Text = Replace(.Text, TextBox1.Text, _
TextBox2.Text, , , SrchMode)
.SelectionStart = CurPos
.SelectionLength = CurSel
End With

```

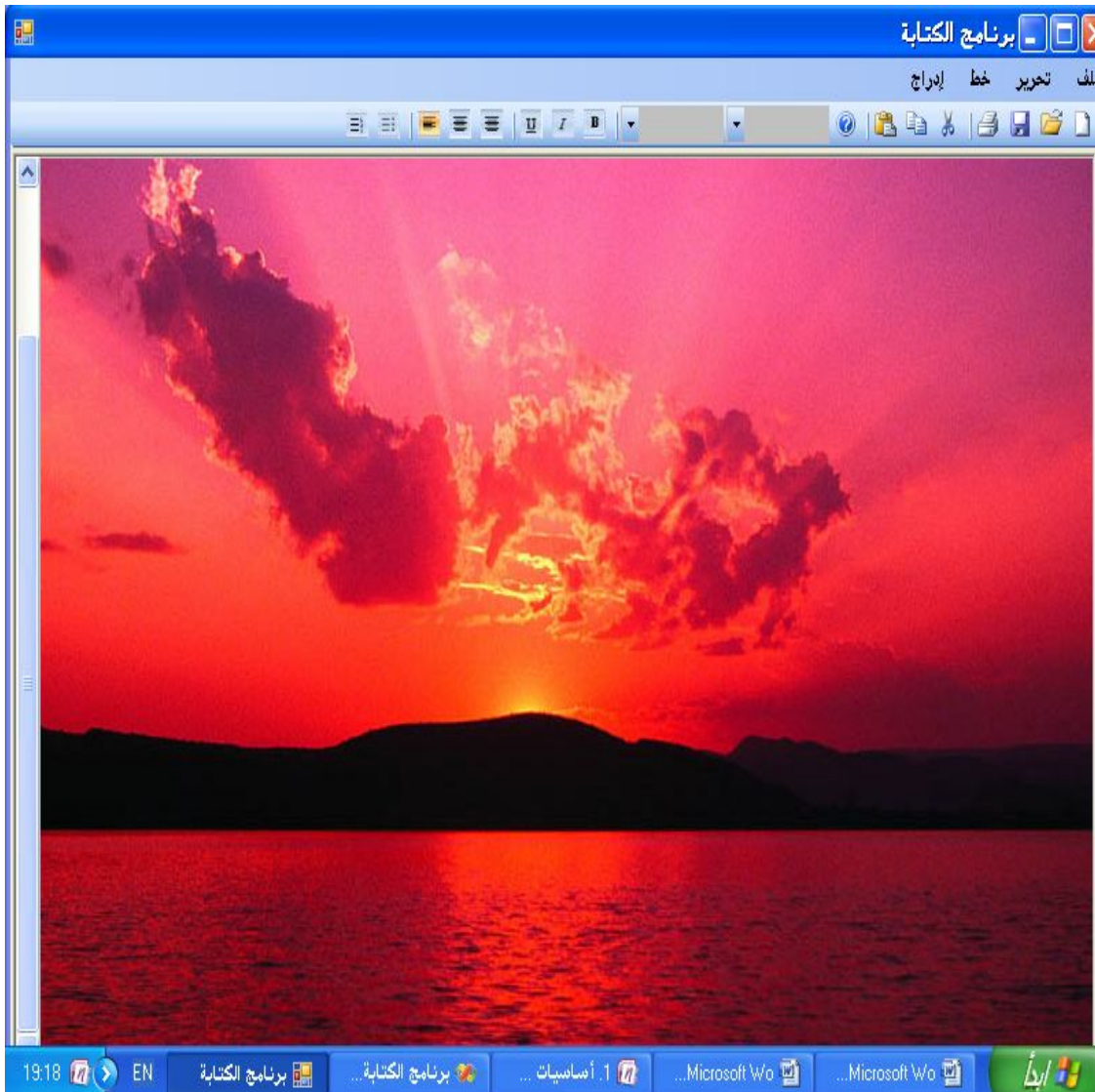
VB.Net Code

إدراج صورة

```

Private Sub InsertPicture_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles InsertPicture.Click
    OpenFileDialog1.Title = "صورة إدراج"
    OpenFileDialog1.DefaultExt = ".bmp"
    OpenFileDialog1.Filter = "bitmap file (*.bmp)|*.bmp| jepG File
(*.jpg)|*.jpg| gif file (*.gif)|*.gif| all file (*.*)|*.*"
    OpenFileDialog1.FilterIndex = 1
    OpenFileDialog1.ShowDialog()
    If OpenFileDialog1.FileName = "" Then Exit Sub
    Try
        Dim X As String = OpenFileDialog1.FileName
        Dim Img As Image
        Img = Image.FromFile(X)
        Clipboard.SetDataObject(Img)
        Dim Df As DataFormats.Format
        Df = DataFormats.GetFormat(DataFormats.Bitmap)
        If Me.RtbDoc.CanPaste(Df) Then
            Me.RtbDoc.Paste(Df)
        End If
    Catch ex As Exception
        MsgBox("الصورة إدراج يمكن لا", MsgBoxStyle.Critical)
    End Try
End Sub

```



VB.Net Code تنسيق الخط حسب مربع الحوار

```
If FontDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then  
    RtbDoc.Font = FontDialog1.Font  
End If
```

VB.Net Code تلوين الخط حسب مربع الحوار

```
If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then  
    RtbDoc.SelectionColor = ColorDialog1.Color  
End If
```

VB.Net Code غامق / عادي

```
If Not (RtbDoc.SelectionFont.Bold) Then
    RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont,
FontStyle.Bold)
Else
    RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont,
FontStyle.Regular)
End If
```

 VB.Net Code

مائل/عادي

```
If Not (RtbDoc.SelectionFont.Italic) Then
RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont, FontStyle.Italic)
Else
RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont, FontStyle.Regular)
End If
```

 VB.Net Code

تسطير/عادي

```
If Not (RtbDoc.SelectionFont.Underline) Then
RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont,
FontStyle.Underline)
Else
RtbDoc.SelectionFont = New Font(RtbDoc.SelectionFont, FontStyle.Regular)
End If
```

 VB.Net Code

تسطير/عادي

```
RtbDoc.SelectionAlignment = HorizontalAlignment.Right
RtbDoc.SelectionAlignment = HorizontalAlignment.Center
RtbDoc.SelectionAlignment = HorizontalAlignment.Left
```

 VB.Net Code

تعداد نقطي/عادي

```
tbDoc.BulletIndent = 10
RtbDoc.SelectionBullet = True
-----
tbDoc.BulletIndent = 0
RtbDoc.SelectionBullet = False
```

 VB.Net Code

المسافة البادئة

```
RtbDoc.SelectionIndent = 5
```


VB.Net Code ترقيم الأسطر

```
Dim I As Integer
Dim NewText As New System.Text.StringBuilder
For I = 0 To RtbDoc.Lines.Length - 1
    NewText.Append((I + 1).ToString & vbTab & _
        RtbDoc.Lines(I) & vbCrLf)
Next
RtbDoc.Text = NewText.ToString
```

VB.Net Code استعمال المفاتيح

```
Private Sub RtbDoc_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles RtbDoc.KeyPress
    Dim ch As Char
    If System.Char.IsControl(e.KeyChar) Then Exit Sub
    ch = Char.ToUpper(e.KeyChar)
    Select Case ch.ToString
        Case "@"
            RtbDoc.SelectedText = "AT"
        Case "#"
            RtbDoc.SelectedText = "BPT"
        Case "$"
            RtbDoc.SelectedText = "DLR"
        Case "%"
            RtbDoc.SelectedText = "O/O"
        Case "&"
            RtbDoc.SelectedText = "AND"
        Case Else
            RtbDoc.SelectedText = ch
    End Select
    e.Handled = True

End Sub
```

VB.Net Code استعمال المفاتيح الوظيفية

```
Private Sub RtbDoc_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles RtbDoc.KeyPress
    Dim ch As Char
    If System.Char.IsControl(e.KeyChar) Then Exit Sub
    ch = Char.ToUpper(e.KeyChar)
    Select Case ch.ToString
        Case "@"
            RtbDoc.SelectedText = "AT"
        Case "#"
            RtbDoc.SelectedText = "BPT"
        Case "$"
            RtbDoc.SelectedText = "DLR"
    End Select
    e.Handled = True

End Sub
```

```

RtbDoc.SelectedText = "DLR"
Case "%"
RtbDoc.SelectedText = "O/O"
Case "&"
RtbDoc.SelectedText = "AND"
Case Else
RtbDoc.SelectedText = ch
End Select
e.Handled = True

End Sub
    
```

المؤقت Timer

يستخدم للقيام بتعليمات برمجية أو عمليات محددة في وقت معين .

الخاصية	الشرح
Interval	تحديد الوقت حيث كل ثانية تمثل ب 1000
Enabled	إمكانية عمل المؤقت : True المؤقت موقوف : False .

كيف نعمل ذلك ؟

1. أضف أداة المؤقت Timer و أداة العنوان Label

2. نكتب الكود التالي :

VB.Net Code

```

Private Sub Timer1_Tick(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Timer1.Tick
    Label1.Text = Now
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
    Timer1.Interval = 1000
    Timer1.Enabled = True
End Sub
    
```

20/03/2012 15:04:28

أداة شريط التقدم ProgressBar

يستعمل لعرض حالة تقدم عملية ما عند التنفيذ . فمثلا عند نسخ ملفات إلى موضع آخر يظهر شريط تقدم

يوضح لنا إلى المدى الذي وصلت إليه العملية



أهم خصائصه

الفصل الثالث الأدوات

الخاصية	الشرح
Value	القيمة التي يشير إليها العداد تصاعديا أو تنازليا
Maximum	أعلى قيمة
Minimum	أدنى قيمة

و إليك هذا المثال :

قم بإضافة الأدوات التالية : Button ، ListBox ، ProgressBar

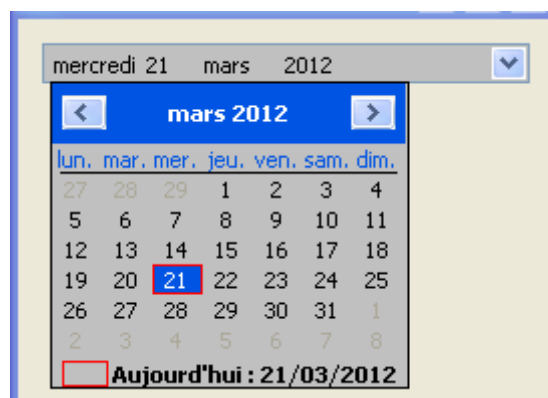
ثم اكتب الكود التالي في الزر

VB.Net Code

```
ProgressBar1.Minimum = 0
ProgressBar1.Maximum = 10000
For i As Integer = 0 To 10000
    ListBox1.Items.Add(i)
    ProgressBar1.Value = i
Next
```

الأداة DateTimePicker

تستعمل لعرض التاريخ أو التاريخ الحالي أو رزنامة الشهر الحالي .
عند تشغيل البرنامج و النقر على الأداة تبدو بهذا الشكل.



من أهم خصائصها

الخاصية	القيمة	الشرح
Value		تعرض قيمة التاريخ أو الوقت الحالي
maximumsize	عدد	أعلى قيمة محددة للتاريخ
mindate	عدد	أقل قيمة محددة للتاريخ
ShowUpDaown	True False	تعرض التاريخ على شكل أداة NumericUpDown عادي
Format	long	عرض التاريخ طويل
	Short	عرض التاريخ قصير
	Time	عرض الوقت

الفصل الثالث الأدوات

مخصص	Custom	Format Custom
تحدد طريقة إظهار التاريخ حسب الرموز المحددة	اكتب الرموز	
لون خط الرزنامة	اختر اللون	CalendarForeColor
لون خلفية رزنامة الشهر	اختر اللون	CalendarMonthBackground
لون خط عنوان الرزنامة	اختر اللون	CalendarTitleForeColor

والرموز التي يمكننا التعامل معها هي كما بالشكل التالي :

الرمز	المعنى	النتيجة
yyyy	السنة ومن الممكن اختصاره إلى yy	2009 او 09
MM	الشهر ولاحظ يجب أن تكون الحروف كبيرة	08
dd	اليوم	04
yyyy-MM-dd	تاريخ	04-08-2009
ss	إظهار الثواني	15
hh	السعة	8
mm	الدقائق	30
Hh:mm:ss	الوقت	8:30:15

ملاحظة

يجب أن نحدد القيمة Custom للخاصية Format حتى يمكننا التعامل مع الخاصية Format Custom

عرض التاريخ الحالي

VB.Net Code

```
Me.Text = DateTimePicker1.Value
```

التاريخ الهجري

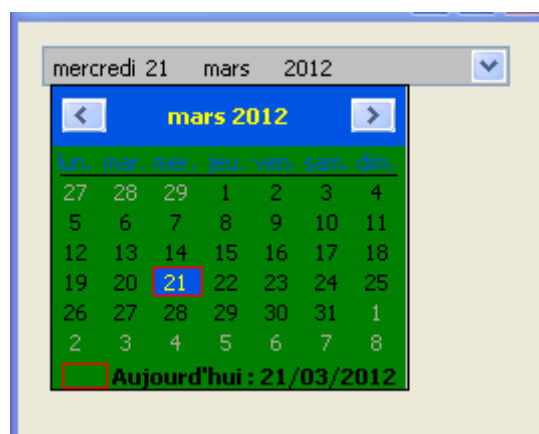
```
Dim DateHj As Date = Me.DateTimePicker1.Value
MsgBox(DateHj.ToString("dd/MM/yyyy", _
System.Globalization.CultureInfo.CurrentCulture))
```

تغيير ألوان الرزنامة

VB.Net Code

```
DateTimePicker1.CalendarForeColor = Color.Black
DateTimePicker1.CalendarMonthBackground = Color.Green
DateTimePicker1.CalendarTitleForeColor = Color.Yellow
```

شغل البرنامج و لاحظ الناتج بعد النقر على سهم الأداة.



أداة NumericUpDown

تستعمل لعرض قائمة منسدلة من الأرقام تمثل عدادا نستطيع من خلال الأسهم أن نزيد أو ننقص من قيمته .

أهم خصائصه

الخاصية	الشرح
Value	القيمة التي يشير إليها العداد تصاعديا أو تنازليا
Increment	تحديد قيمة التزايد أو تناقص (الفقرة).
Maximum	أعلى قيمة
Minimum	أدنى قيمة
Dechmal Place	جعل الأعداد عشرية وذلك بتحديد عدد أرقام الجزء العشري

و هذا مثال يلخص ما قلناه

```

VB.Net Code
Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NumericUpDown1.ValueChanged
    Me.Text = NumericUpDown1.Value
End Sub

Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    NumericUpDown1.Maximum = 100
    NumericUpDown1.Minimum = 5
    NumericUpDown1.Increment = 5
End Sub
    
```

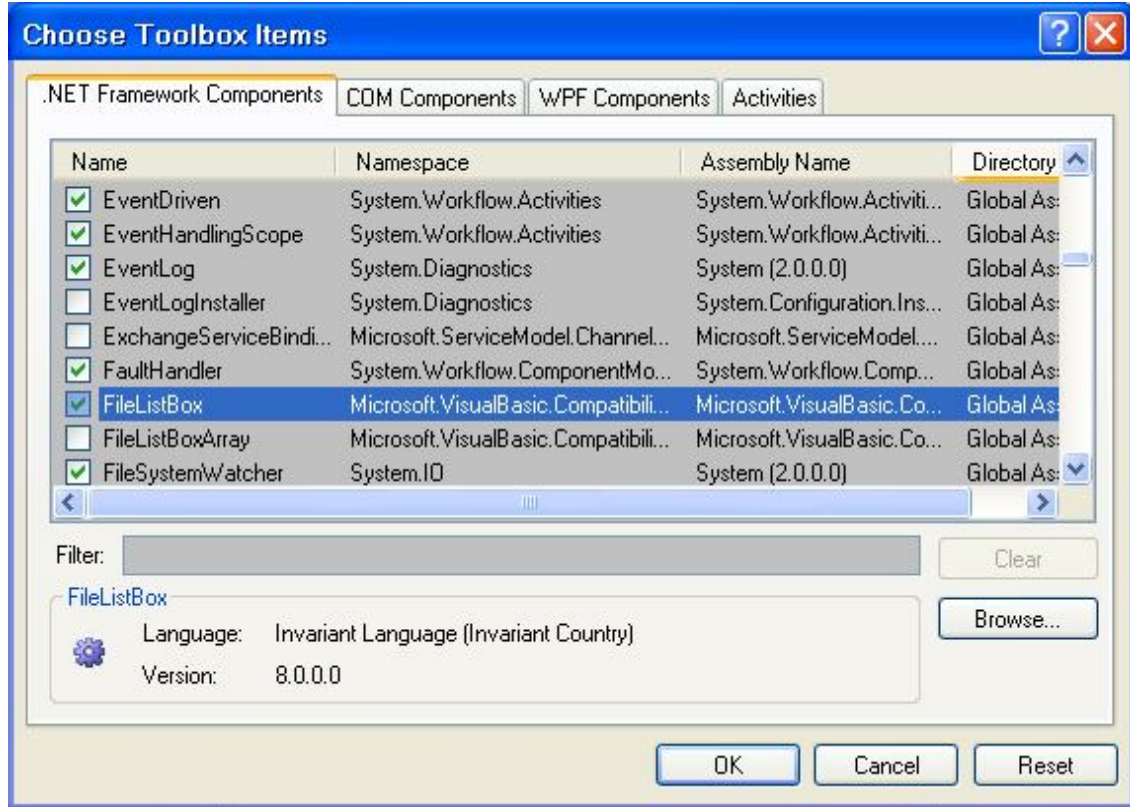
أدوات الملفات Drivelist - Dirlist-Filelist

تستخدم لعرض محتوى الملفات و المجلدات .

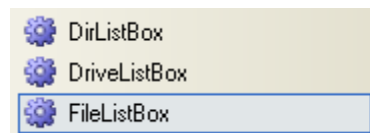
كيف نعمل ذلك ؟

الفصل الثالث الأدوات

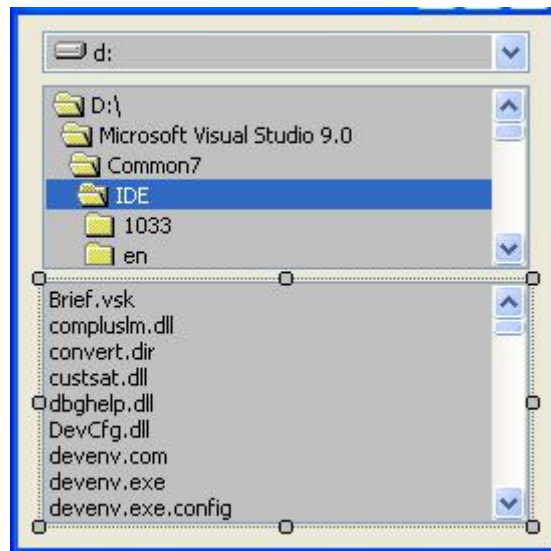
قم بإضافة الأدوات التالية DirList ، Drivelist ، Filelist و ذلك بنقر على صندوق الأدوات بالزر الأيمن للفأرة ثم اختر Choose Items تظهر النافذة التالية :



حدد الأدوات الثلاثة من النافذة السابقة ثم اضغط موافق Ok . بعدها ستجد هذه الأدوات كما بالشكل التالي:



الأداة الأولى كما بالصورة تستعرض المجلدات.
والثانية تحدد القرص النشط.
والثالثة تستعرض الملفات.
أولا ضع الأدوات على النموذج كما بالشكل التالي:



ثم اكتب الكود التالي

VB.Net Code

```
Private Sub DriveListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DriveListBox1.SelectedIndexChanged
    Me.DirListBox1.Path = Me.DriveListBox1.Drive
End Sub
```

```
Private Sub DirListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DirListBox1.SelectedIndexChanged
    Me.FileListBox1.Path = Me.DriveListBox1.Drive
    Me.FileListBox1.Path = Me.DirListBox1.Path
End Sub
```

نختار الحدث SelectedIndexChanged حتى إذا تم تغيير محرك الأقراص تم تغيير المجلدات والملفات.
و ربطنا أداة المجلدات بمحرك الأقراص ثم ربطنا أداة الملفات بأداة المجلدات .

أداة صندوق الصور PictureBox

تستخدم لعرض الصور و التعامل معها .

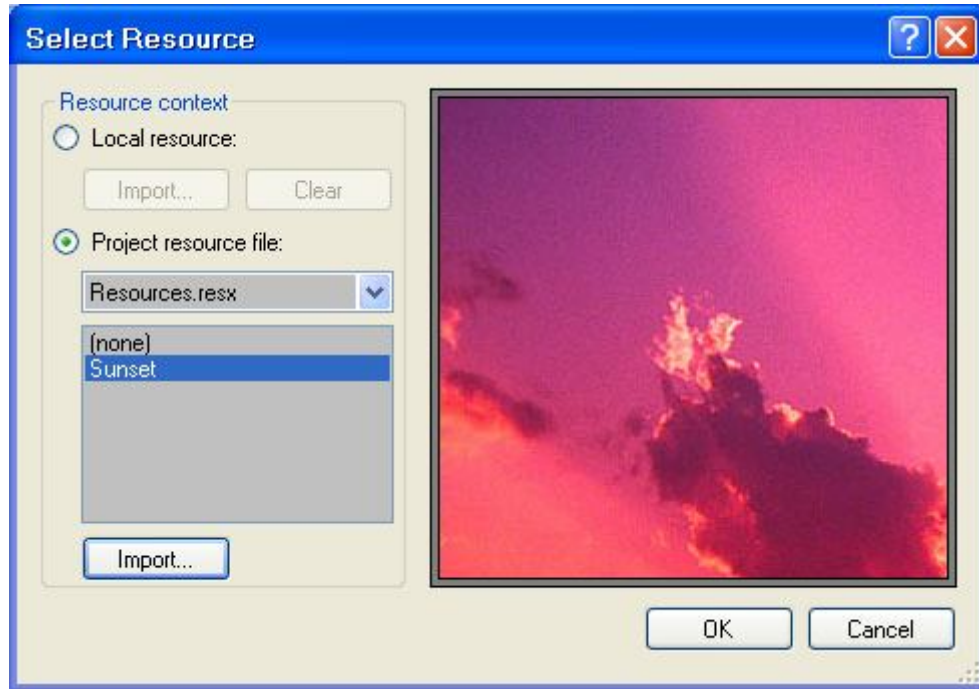
الخاصية	القيمة	الشرح
SizeMode	اختر قيمة (الجدول 1)	اختيار كيفية ظهور الصورة ومحاذاتها في مربع الصورة
BorderStyle	Fixed3D FixedSingle None	ثابتة مجسمة ثابتة مفردة بدون حافة
Image	مربع حوار	اختيار الصورة التي تعرضها في مربع الصورة

الفصل الثالث الأدوات

بتحديد الصورة التي تعرضها في خلفيّة مربع الصورة	مربع حوار	BackgroundImage
الجدول 1		
تظهر الصورة طبيعية كما هي بدون تعديل.. فإذا كانت أكبر من مربع الصورة، فسيختفي جزء منها، وإذا كانت أصغر، فلن تملأ كل مساحة مربع الصورة.		Normal
سيتمّ عرض الصورة بحيث تتوسّط مربع الصورة.		CenterImage
سيتمّ جعل حجم الصورة يناسب مساحة مربع الصورة بالضبط.. ورغم أنّ هذه الطريقة هي أفضل طريقة لعرض الصورة، إلا إنّ عليك أن تراعي تناسب أبعاد مربع الصورة مع أبعاد الصورة، حتّى لا تنتشوه الصورة.		StretchImage
سيتمّ تغيير أبعاد مربع الصورة ليُطابق أبعاد الصورة.. إنّ هذا الاختيار غير مفضل، لأنّ وضع صورة كبيرة في مربع الصورة سيجعله يغطّي باقي الأدوات الموجودة على النموذج.		AutoSize

إضافة صورة

انقر السهم الصغير ثم اختر الخيار Choose Image تظهر النافذة التالية :



انقر زر Import لاستيراد صورة . ثم اضغط Ok سيتمّ حفظها في الملف الذي يحمل نفس اسم النموذج

إضافة صورة إلى كائن

الصيغة العامة 

`Control.BackgroundImage = New Bitmap("مسار الصورة كاملاً")`

Control.Image = New Bitmap("مسار الصورة كاملا")

Control.Image = Image.FromFile("مسار الصورة كاملا")

ملاحظة

الخاصية Image مفيدة مع الصور المتحركة من نوع Gif يمكنك رؤية هذه الحركة

خصائص الموضوع المرافقة للخاصية Image و التي تعرفنا عليها سابقا

VB.Net Code

```
PictureBox1.BackgroundImageLayout = ImageLayout.None
PictureBox1.BackgroundImageLayout = ImageLayout.Center
PictureBox1.BackgroundImageLayout = ImageLayout.Stretch
PictureBox1.BackgroundImageLayout = ImageLayout.Tile
```

وسائل الكائن Image

حفظ صورة :

الصيغة العامة

PictureBox1.Image.Save("c:\tmpImage.bmp", نوع الصورة)

للتعامل مع نوع الصورة يستحسن استيراد مجال الأسماء **Imports System.Drawing.Imaging** وبذلك تستطيع استخدام اسم المرقم مختصرا. ولهذا المرقم القيم التالية:

Bmp	Bitmap، وهو أبسط أنواع الصور
Emf	.Enhanced Windows metafile
Exif	.Exchangeable Image Format
Gif	.Graphics Interchange Format
Icon	أيقونة.
Jpeg	.Joint Photographic Experts Group
MemoryBmp	لحفظ الصورة كمصفوفة ذاكرة.
Png	.W3C Portable Network Graphics
Tiff	.Tagged Image File Format
Wmf	.Windows metafile

واضح طبعا مدى ثراء أنواع الصور التي يتعامل معها VB.Net،

تدوير وعكس RotateFlip:

تمكّنك هذه الخاصية من تدوير الصورة أو عكسها أو كلا الأمرين معا



```
PictureBox1.Image.RotateFlip(RotateFlipType.التدوير)
PictureBox1.Refresh()
```

حيث النوع هو إحدى قيم المرقم RotateFlipType التالية:

لا تدوير ولا عكس.. واضح طبعاً أنك بهذا لن تفعل شيئاً في الصورة (يفيد هذا في إلغاء عمليات التدوير والعكس السابقة، لإعادة الصورة إلى وضعها الأصلي).	RotateNoneFlipNone
تدوير الصورة بزاوية 90 بدون عكسها.	Rotate90FlipNone
تدوير الصورة بزاوية 90 وعكسها أفقيًا.	Rotate90FlipX
تدوير الصورة بزاوية 90 وعكسها أفقيًا ورأسيًا.	Rotate90FlipXY
تدوير الصورة بزاوية 90 وعكسها رأسيًا.	Rotate90FlipY

واعتقد أنك تستطيع فهم باقي هذه التعبيرات:

Rotate180FlipNone – Rotate180FlipX – Rotate180FlipXY – Rotate180FlipY –
 Rotate270FlipNone – Rotate270FlipX – Rotate270FlipXY – Rotate270FlipY –
 RotateNoneFlipX – RotateNoneFlipXY – RotateNoneFlipY.

VB.Net Code

```
Dim img As Image = Image.FromFile("D:\P1.bmp")
PictureBox1.Image = img
img.RotateFlip(RotateFlipType.RotateNoneFlipY)
-----
PictureBox1.Image.RotateFlip(RotateFlipType.RotateNoneFlipY)
PictureBox1.Refresh()
```

كُون صورة مصغرة **GetThumbnailImage**:



Image.GetThumbnailImage(بيانات, إلغاء, الارتفاع, العرض)

VB.Net Code

```
Dim img As Image = Image.FromFile("D:\P1.bmp")
PictureBox1.Image = img.GetThumbnailImage(200, 200, Nothing, Nothing)
```

الصور و الكائن Clipboard:

استخدم الجملة التالية لنسخ الصورة المعروضة في مربع الصور إلى الحاة:

الصيغة العامة 

`Clipboard.SetDataObject(PictureBox1.Image, True)`

حتى تبقى الصورة في الحافظة: `True`

ويمكنك استخدام الجملة التالية لعرض الصورة من الحافة في مربع الصورة:

الصيغة العامة 

`PictureBox1.Image =`

`Clipboard.GetDataObject.GetData(DataFormats.Bitmap)`

أداة جدول عرض البيانات DataGridView

تستعمل عرض البيانات خاصة مع قواعد البيانات.

الخاصية	الشرح
DataSource	تحديد الجدول الذي يتم عرض البيانات منه
Allow user to Resiz Rows	إمكانية تغيير أبعاد الصفوف غير ممكن
Cell Border Style	تحديد نمط حدود الخلايا
Column header border style	تحديد نمط حدود العنوان
ColumnheaderHeight	تحديد ارتفاع شريط العنوان
ColumnheaderVisable	إخفاء أو إظهار شريط العنوان
MultiSelect	متعدد التحديد يمكن تحديد أكثر من صف أو عمود
RowheaderVisable	إخفاء / إظهار شريط الصفوف
RowheaderWith	تحديد ارتفاع شريط الصفوف

عند النقر على السهم الصغير تظهر نافذة منها نتعامل مع الخصائص المهمة للجدول .

Choose Data Source : يمكنك من تحديد مصدر البيانات وتحديد الجدول الذي ترغب في استعراضه

Edit Columns : تحرير الأعمدة

Add Colume : إضافة أعمدة

Enabeld Adding : إمكانية الإضافة

Enabled Editing : إمكانية التحرير

DataGridView Tasks

Choose Data Source (none)

Edit Columns...

Add Column...

Enable Adding

Enable Editing

Enable Deleting

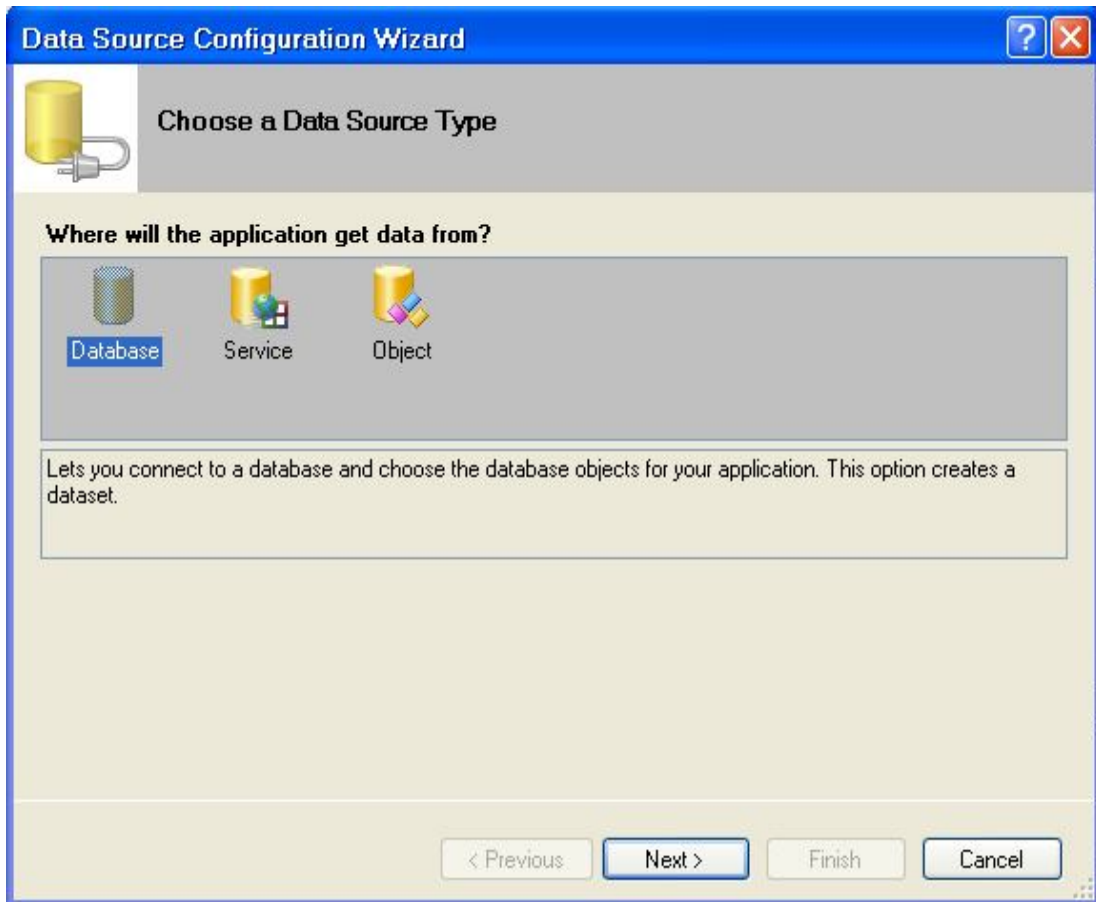
Enable Column Reordering

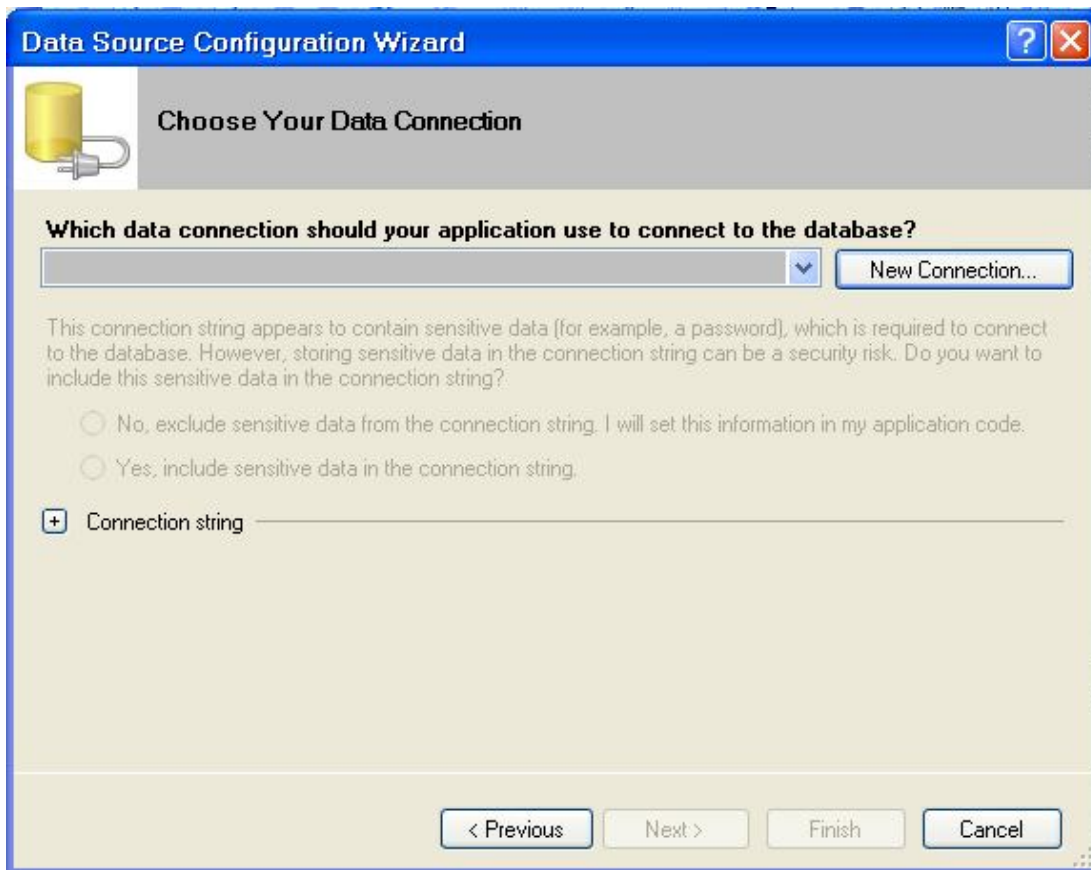
Dock in parent container

اختيار جدول من قاعدة البيانات :

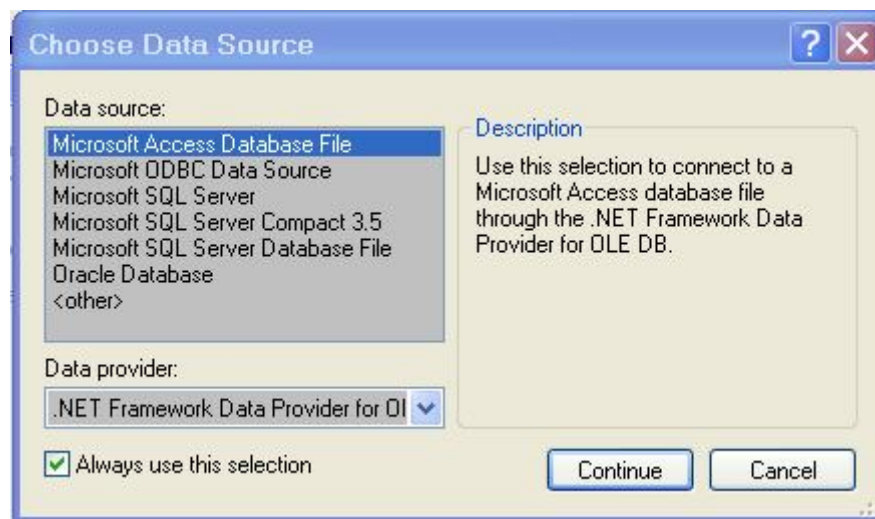
انقر السهم الصغير لتظهر أمامك نافذة :

في حالة عدم الاتصال بقاعدة البيانات اختر الخيار Add Project data Source تظهر النافذة التالية ثم اختر Next





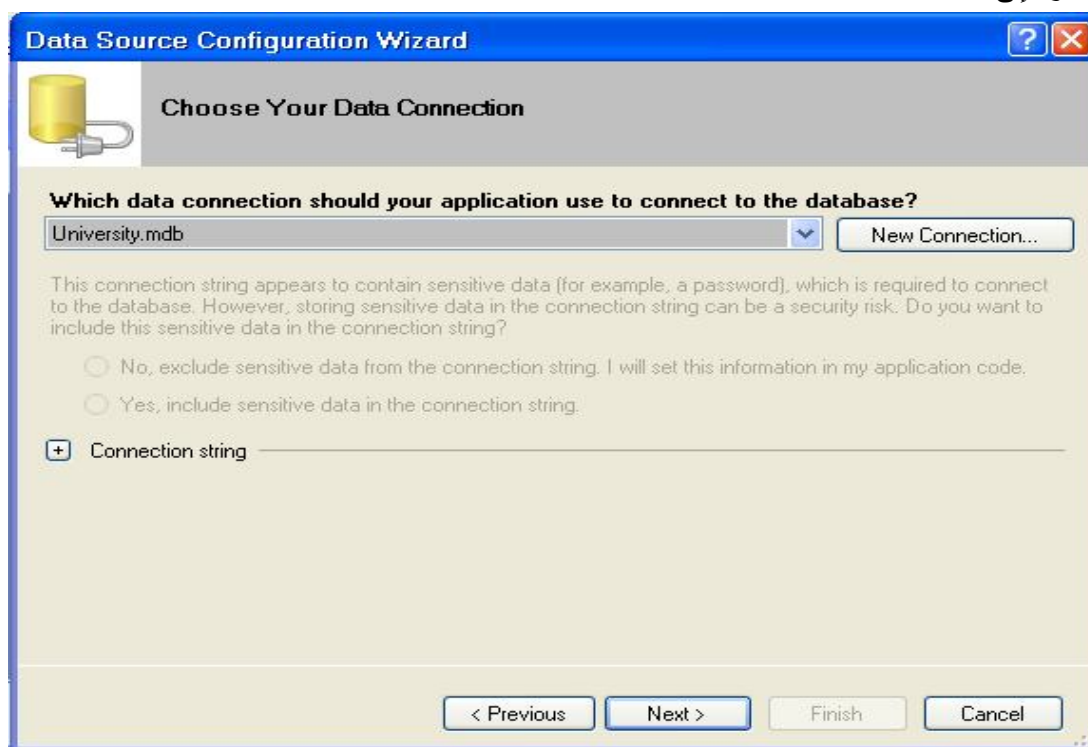
انقر زر New Connection تظهر النافذة التالية



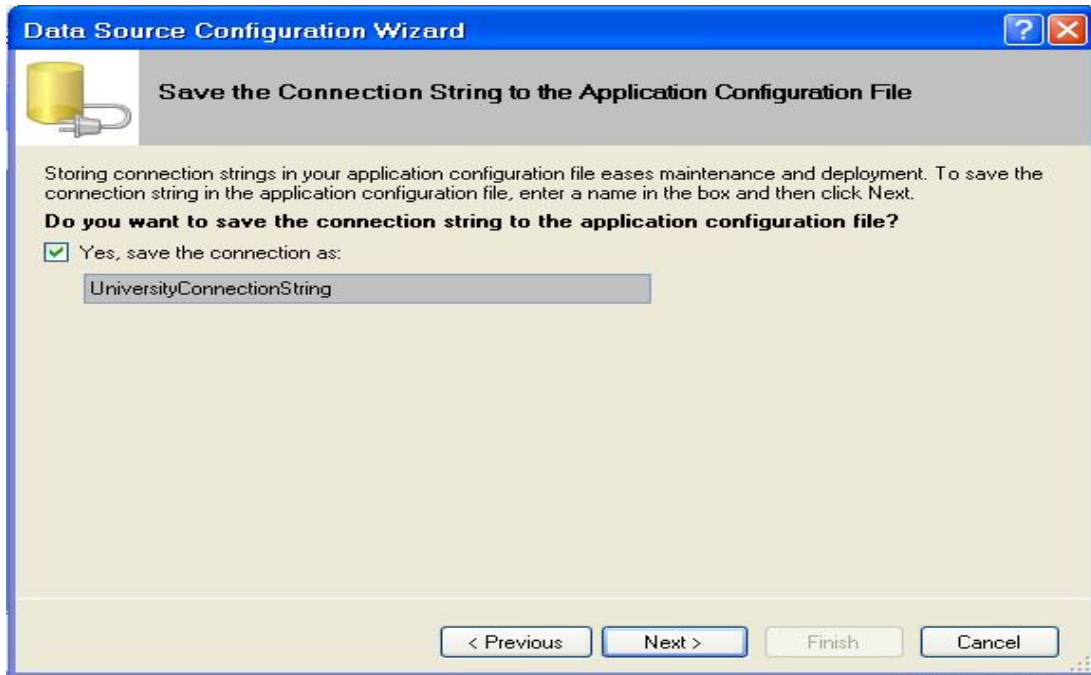
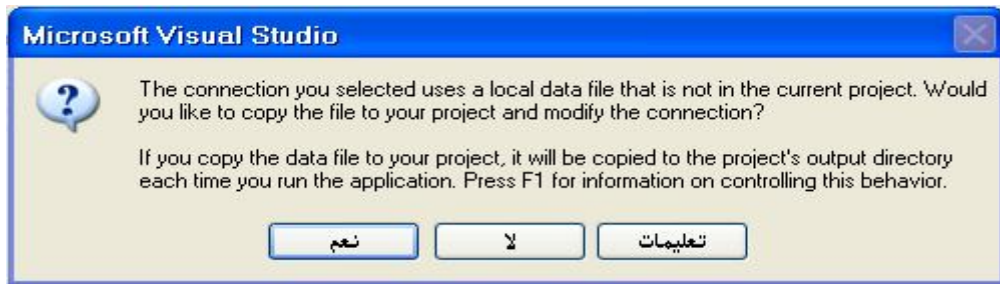
منها نختار نوع قاعدة البيانات . نختار الخيار الأول ثم انقر Continue تظهر النافذة



انقر زر استعراض Browse و اختر اسم قاعدة البيانات من مربع حوار فتح أو اكتب مسار القاعدة في الخانة Database File Name و إذا كانت محمية بكلمة مرور ادخل كلمة مرور في الخانة Password . ثم انقر Ok .
ستعود إلى النافذة



انقر Next يظهر مربع الحوار التالي انقر نعم تظهر نافذة أخرى

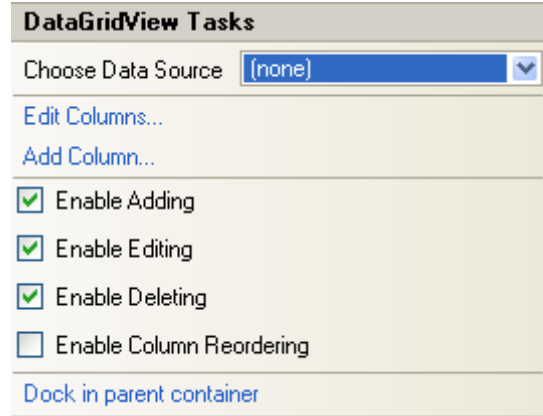




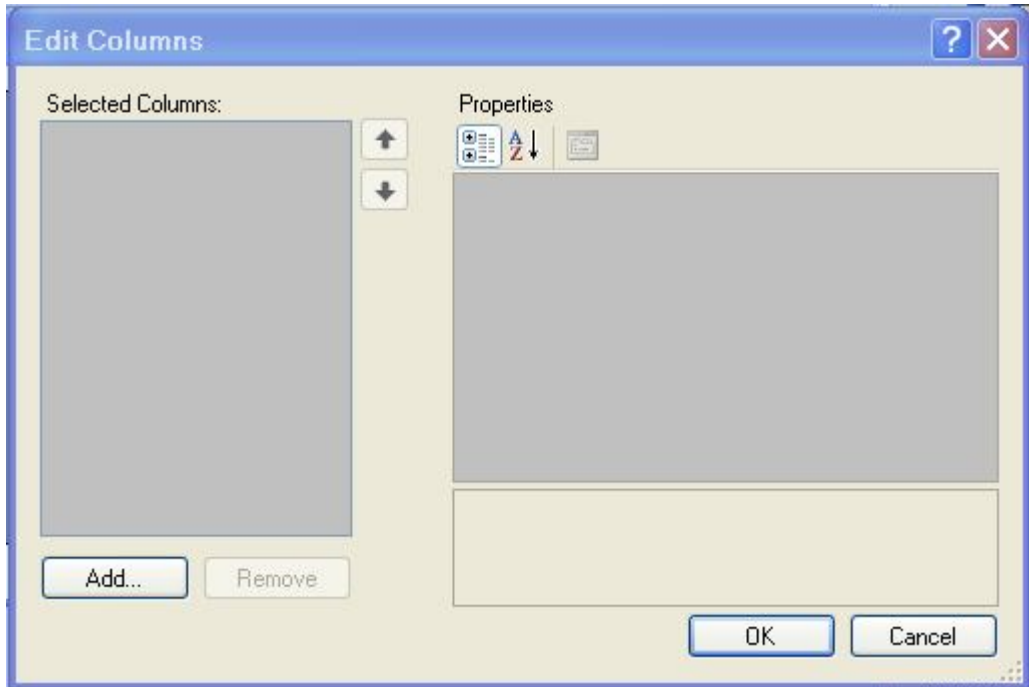
نختار الجدول المراد عرض بياناته ثم نضغط Finch
شغل البرنامج و راقب النتيجة : تم تحميل الجدول بنجاح.

رقم التسجيل	الاسم	اللقب	تاريخ الميلاد	نة الميلاد
1	فوزي	بلباشه	16-03-2003	2003
2	كريمه	قافي	16-11-2003	2003
3	ورده	بلباشه	01-12-2003	2003
4	جميله	بن باله	01-09-2003	2003
5	سعاد	ستوتوي	00-00-2003	2003
7	بسمه	خرشوش	04-12-2002	2002
8	مريم	قافي	23-09-2002	2002
9	سمية	ستوتوي	17-08-2003	2003
10	سهيله	بن باله	19-05-2003	2003
11	محمد	مني	05-07-2003	2003
12	وليد	بغوره	21-04-2003	2003
13	عبد الحميد	حمشه	15-10-2003	2003
14	ربيعة	خرشوش	23-02-2003	2003
15	صالح	حمشه	23-09-2003	2003

تحرير و إضافة أعمدة لجدول فارغ



انقر الخيار Edit Columns تظهر النافذة التالية :



لا شك و أنك اعتدت على استعمال هذه النافذة المألوفة
انقر زر Add لإضافة أعمدة Columns. تظهر نافذة أخرى

الفصل الثالث الأدوات

Name : الاسم البرمجي

Header text : عنوان العمود

Type : نمط العمود

- >ataGridViewButtonColumn
- >ataGridViewCheckBoxColumn
- >ataGridViewComboBoxColumn
- >ataGridViewImageColumn
- >ataGridViewLinkColumn
- >ataGridViewTextBoxColumn

أمثلة عن أنماط الحقول

Form1				
	Column4	Column5	رابط الموقع	Column7
♂	<input checked="" type="checkbox"/>		WWW.google.com	<input type="checkbox"/>
*	<input type="checkbox"/>			<input type="checkbox"/>

Visible : مرئي أم لا

Read only : للقراءة فقط و لا يمكن تغيير

Add Column

Databound column

Columns in the DataSource

Unbound column

Name:

Type:

Header text:

Visible Read Only Frozen

قم بإضافة الأعمدة التالية حيث يبدو الشكل

العنوان	النوع
الرقم	نص (TextBox Column)
الاسم	نص (TextBox Column)
متزوج	مربع خيار (Check Box Column)
الحساب الالكتروني	Link Column

Edit Columns

Selected Columns:

- الرقم
- الاسم
- متزوج
- الحساب لالكتروني

Unbound Column Properties

DefaultCellStyle	DataGridViewCellStyle { }
HeaderText	الحساب الالكتروني
LinkColor	0; 0; 255
Text	w.w.w.@SalahMeddo
ToolTipText	
UseColumnTextForLink	False
Visible	True
VisitedLinkColor	128; 0; 128

Behavior

Text
The link text displayed in all of the column's cells.

الفصل الثالث الأدوات

قم بتحميل الحساب الالكتروني من الخاصية Text لأنه لا يمكن تحميله أثناء التنفيذ مثل بقية الأعمدة
شغل البرنامج واكتب الرقم ثم الاسم وحدد خانة الاختيار كما يلي

	الرقم	الاسم	متزوج	الحساب الالكتروني
▶	1	الصالح	<input checked="" type="checkbox"/>	
*			<input type="checkbox"/>	

أما الخيار الثاني Add Column لإضافة أعمدة إلى جدول موجود مسبقا يحتوي على أعمدة أخرى

Dialog box titled "Add Column" with the following settings:

- Databound column
- Unbound column
- Name: Column5
- Type: DataGridViewComboBoxColumn
- Header text: الحالة العائلية
- Visible
- Read Only
- Frozen

نضيف عمود جديد و ليكن الحالة العائلية من نوع ComboBoxColumn Type ثم انقر Add
نعود إلى نافذة التحرير Edit Column لتحميل عمود (الحالة العائلية)
من الخاصية Items (Collection) تظهر نافذة أخرى نكتب فيها

Dialog box titled "Enter the strings in the collection (one per line):" with the following content:

- متزوج
- أعزب
- مطلق

ثم انقر Ok . شغل البرنامج و لاحظ انسدال قائمة عند نقرك على خانة العمود الحالة العائلية بها العناصر التي كتبتها.

الفصل الثالث الأدوات

و يمكن تلوين خانات الأعمدة : نختار العمود و من خلال الخاصية Default Cellstyle التابعة لنافذة التحرير تظهر نافذة أخرى ومن خلال الخاصية BackColor نختار اللون المناسب .و أخيرا نحصل على شكل التالي

الرقم	الاسم	متزوج	الحساب الالكتروني	الحالة العائلية
1245	الصالح	<input checked="" type="checkbox"/>		متزوج
*		<input type="checkbox"/>		

أحداث جدول عرض البيانات

الحدث	الشرح
CellEnter	يحدث عند دخول المؤشر إلى الخلية
Cell Formatting	تنسيق الخلية

الحدث CellEnter

يمكنك من تحديد موقع الخلية (العمود ، الصف) مثل برنامج Excel .لاحظ المثال

```
VB.Net Code
Private Sub DataGridView1_CellEnter(ByVal sender As Object _
, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellEnter
    MsgBox("Row" & e.RowIndex & "-" & "Col" & e.ColumnIndex)
End Sub
```

الحدث Cell Formatting

يستعمل في حالة تغيير و تنسيق الخلايا .

```
VB.Net Code
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    DataGridView1.Columns.Add("Col1", "الرقم ")
    DataGridView1.Columns.Add("Col2", "الاسم ")
    DataGridView1.Columns.Add("Col3", "المعدل ")
    DataGridView1.Rows.Add(3)
    Dim Numbers() As Integer = {1, 2, 3}
    Dim Names() As String = {"وسيم", "الخليم عبد", "سامح"}
    Dim Score() As Double = {12.53, 14.75, 16.32}
    For i As Integer = 0 To DataGridView1.Rows.Count - 2
        DataGridView1.Rows(i).Cells(0).Value = Numbers(i)
        DataGridView1.Rows(i).Cells(1).Value = Names(i)
        DataGridView1.Rows(i).Cells(2).Value = Score(i)
    Next

    DataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect
End Sub
Private Sub DataGridView1_CellFormatting(ByVal sender As Object, ByVal e As
```

```
System.Windows.Forms.DataGridViewCellFormattingEventArgs) Handles
DataGridView1.CellFormatting
    e.CellStyle.SelectionForeColor = Color.Red
    e.CellStyle.ForeColor = Color.White
    e.CellStyle.SelectionBackColor = Color.Azure
    DataGridView1.RowsDefaultCellStyle.BackColor = Color.Olive
    DataGridView1.AlternatingRowsDefaultCellStyle.BackColor = Color.Teal
End Sub
```

في نموذج التحميل قمنا بتعبئة الأداة كما تعلمنا سابقاً ولكن الجديد هو جعل الأداة تقوم بتحديد كامل السطر عند النقر على أي خلية بواسطة سطر الأوامر التالي :

```
DataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect
```

أما بحدث تعديل إحدى الخلايا Cell Formatting فقد قمنا بجعل الخط المحدد يصبح احمر اللون. ولونه الأساسي هو الأبيض. ولون الخلفية للخلية Azure. والسطران التاليان يقومان على تبديل لون الصفوف بالتناوب وبلونين محددين.

التعامل مع الجدول بالكود

تحميل الجدول من قاعدة البيانات

VB.Net Code

```
Me.StudentsTableAdapter.Fill(Me.Db1DataSet.Students)
DataGridView1.DataSource = Db1DataSet.Tables("Students")
-----
DataGridView1.DataSource = Db1DataSet.Tables(0)
```

ملاحظة

DataSet : Db1Dataset

اسم الجدول : Students

الرقم الترتيبي للجدول في مجموعة DataSet : 0

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
DataGridView1.AllowUserToResizeColumns = True
DataGridView1.AllowUserToDeleteRows = False
DataGridView1.AllowUserToAddRows = True
DataGridView1.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.Raised
DataGridView1.ColumnHeadersHeight = 23
```

الفصل الثالث الأدوات

End Sub

كل الخصائص التي تحدثنا عنها سابقاً ([خصائص الجدول](#)) يمكن الوصول لها بالطريقة التي ذكرناها هنا والتغيير فيها أو القراءة منها.

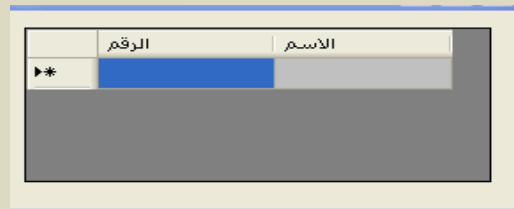
إضافة أعمدة لجدول

الصيغة العامة 

```
DataGridView1.Columns.Add(عنوان العمود, الاسم البرمجي)
```

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    DataGridView1.Columns.Add("", "الرقم")
    DataGridView1.Columns.Add("", "الاسم")
End Sub
```



الوصول إلى الأعمدة: عن طريق الاسم البرمجي

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    DataGridView1.Columns.Add("Col1", "")
    DataGridView1.Columns.Add("Col2", "الاسم")
    DataGridView1.Columns.Add("Col3", "المعدل")
    DataGridView1.Columns("Col1").HeaderText = "تغيير اسم العمود 'الرقم'"
    DataGridView1.Columns("Col2").Width = 80 ' تحديد عرض العمود الثاني
    DataGridView1.Columns("Col3").Name = "Col4" ' تغيير الاسم البرمجي
End Sub
```

تحميل صف من جدول عن طريق مصفوفة

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
```

الفصل الثالث الأدوات

```
Dim value() As String = {"1", "الصالح", "14.50"}
    DataGridView1.Columns.Add("Col1", "الرقم")
    DataGridView1.Columns.Add("Col2", "الاسم")
    DataGridView1.Columns.Add("Col3", "المعدل")
    DataGridView1.Rows.Add(value) ' تحميل الصف
End Sub
```

 VB.Net Code حذف عمود

```
DataGridView1.Columns.Remove("Col1")
```

 VB.Net Code عدد الأعمدة

```
MsgBox(DataGridView1.Columns.Count)
```

 VB.Net Code مسح كل الأعمدة

```
DataGridView1.Columns.Clear
```

 VB.Net Code إضافة صفوف فارغة

```
DataGridView1.Rows.Add(3)
```

أمور أخرى مع الصفوف

 VB.Net Code

```
DataGridView1.Rows(2).Selected = True '3 تحديد الصف
```

```
DataGridView1.Rows.Clear() ' مسح كل الصفوف
```

```
MsgBox(DataGridView1.Rows.Count) ' عدد الصفوف
```

تكوين جدول بأعمدته و صفوفه و خلاياه .

 VB.Net Code

```
DataGridView1.Columns.Add("Col1", "الرقم")
```

```
    DataGridView1.Columns.Add("Col2", "الاسم")
```

```
    DataGridView1.Columns.Add("Col3", "المعدل")
```

```
    DataGridView1.Rows.Add(3)
```

الفصل الثالث الأدوات

```
DataGridView1.Rows(0).Cells(0).Value = "1"  
DataGridView1.Rows(1).Cells(0).Value = "سامح"  
DataGridView1.Rows(2).Cells(0).Value = "12.53"  
  
DataGridView1.Rows(0).Cells(1).Value = "2"  
DataGridView1.Rows(1).Cells(1).Value = "الحليم عبد"  
DataGridView1.Rows(2).Cells(1).Value = "14.75"  
  
DataGridView1.Rows(0).Cells(2).Value = "3"  
DataGridView1.Rows(1).Cells(2).Value = "وسيم"  
DataGridView1.Rows(2).Cells(2).Value = "16.32"
```

شغل البرنامج ستحصل على الناتج

	الرقم	الاسم	المعدل
▶	1	2	3
	سامح	عبد الحليم	وسيم
	12.53	14.75	16.32
*			

كما يمكنك تنسيق خلايا الجدول وتغيير لونها من خلال Style مثلا: نضيف للكود السابق التعليمات التالية

VB.Net Code

```
DataGridView1.Rows(0).Cells(0).Style.BackColor = Color.Yellow  
DataGridView1.Rows(2).Cells(2).Style.ForeColor = Color.Red  
DataGridView1.Rows(1).Cells(1).Style.BackColor = Color.Green
```

يكون الناتج

	الرقم	الاسم	المعدل
▶	1	2	3
	سامح	عبد الحليم	وسيم
	12.53	14.75	16.32
▶*			

تحميل الجدول عن طريق الحلقات التكرارية

VB.Net Code

```
DataGridView1.Columns.Add("Col1", "الرقم")
```



```

DataGridView1.Columns.Add("Col2", "الاسم")
DataGridView1.Columns.Add("Col3", "المعدل")
DataGridView1.Rows.Add(3)
Dim Numbers() As Integer = {1, 2, 3}
Dim Names() As String = {"وسيم", "الحليم عبد", "سامح"}
Dim Score() As Double = {12.53, 14.75, 16.32}
For i As Integer = 0 To DataGridView1.Rows.Count - 2
    DataGridView1.Rows(i).Cells(0).Value = Numbers(i)
    DataGridView1.Rows(i).Cells(1).Value = Names(i)
    DataGridView1.Rows(i).Cells(2).Value = Score(i)
Next
    
```

نحصل على نفس الجدول السابق

ولعرض كل حقول جدول العرض نكتب :

DGV.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill

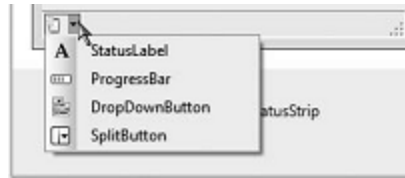
أداة شريط الحالة StatusBar

يستعمل لعرض حالة الأداة . أو شروحات عن الأداة



كيف نعمل ذلك ؟

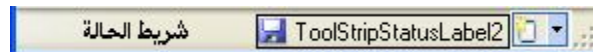
1. قم بإضافة الأداة StatusBar إلى النموذج
2. انقر السهم الصغير تظهر قائمة الأدوات التي يمكن إضافتها



3. قم بإضافة StatusLabel1 ، StatusLabel2



4. من نافذة الخصائص اختر الخاصية Text واكتب عنوان الأول و ليكن شريط الحالة
5. من نافذة الخصائص اختر الخاصية Image لإضافة أيقونة للعنوان الثاني



تدريب نموذجي :

1. أنشئ مشروعا جديدا
2. قم بإضافة الأداة StatusBar

3. ثم اكتب التعليمات التالية

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Load
    StatusStrip1.ImageList = ImageList1
    StatusStrip1.Items.Add("حفظ", ImageList1.Images(0))
    StatusStrip1.Items(0).BackColor = Color.Yellow
    StatusStrip1.Items.Add("اليوم تاريخ |" & Today)
    StatusStrip1.Items(1).BackColor = Color.Blue
    StatusStrip1.Items.Add("الوقت |" & TimeString)
    StatusStrip1.Items(2).BackColor = Color.Green
End Sub
```

حفظ | تاريخ اليوم: 23/03/2012 | الوقت: 19:44:10

أداة شريط الأدوات ToolStrip

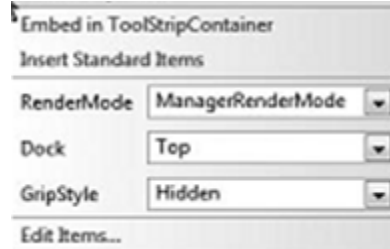
تستعمل لاحتواء أدوات (أيقونات) للوصول السريع إلى المراد . مثل الشريط القياسي لبرنامجنا Vs.Net



كيف نعمل ذلك ؟

1. قم بإضافة الأداة ToolStrip إلى النموذج

2. انقر السهم الصغير تظهر نافذة صغيرة




3. اختر الخيار Insert Standard Items لإدراج الشريط القياسي الافتراضي



طبعاً يمكنك تعديله حسب رغبتك .

أما إذا أردت أن تتجز شريطاً جديداً

1. قم بإضافة الأداة ToolStrip إلى النموذج

2. انقر على السهم الصغير  تظهر قائمة منسدلة لتختار منها الأداة التي تريد إضافتها

الفصل الثالث الأدوات

الشرح	الأداة
زر	Button
عنوان	Label
	SplitButton
قائمة منسدلة	DropDownButton
فاصل	Separator
صندوق سرد	ComboBox
مربع نص	textBox
شريط التقدم	ProgressBar

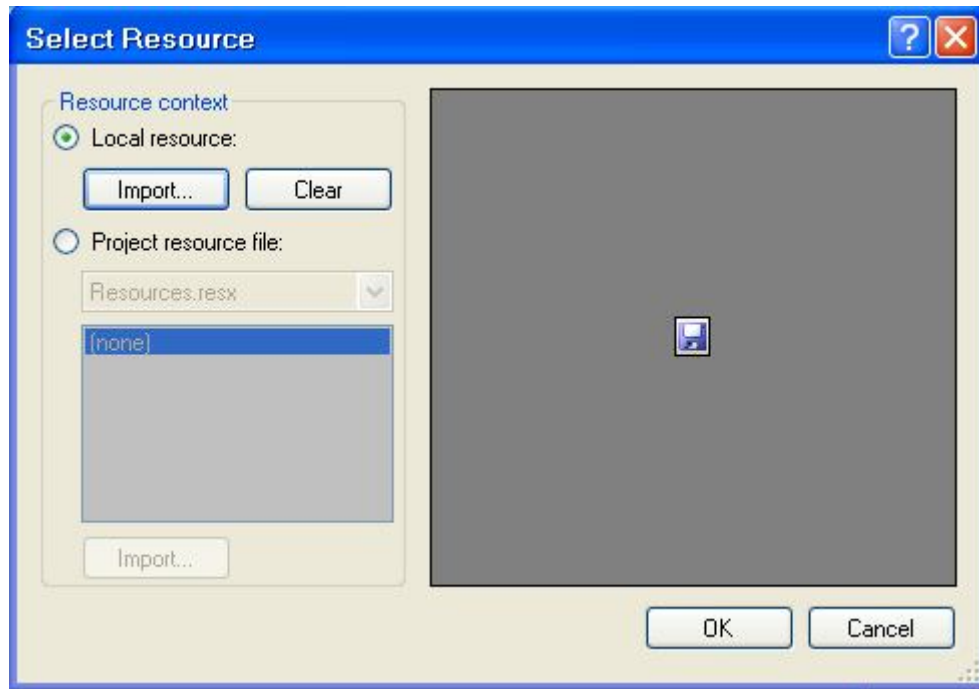
انقر زر Button لإضافته ثم اضغط السهم مرة أخرى وقم بإضافة فاصل Separator ثم ضف



ComboBox بحيث يبدو الشكل .
و يمكنك حذف الأدوات المضافة بالنقر على الأداة ثم اضغط مفتاح Suppr من لوحة المفاتيح

إدراج أيقونة :

انقر على الزر الأول بيمين الفأرة ثم اختر من القائمة Set Image تظهر نافذة مععادة لاستيراد الصورة أو الأيقونة . قم بإضافة الأيقونة بالضغط زر Import و اختر الأيقونة من مربع حوار حفظ كما في الشكل



ثم انقر Ok و لاحظ الناتج

و هكذا تفعل مع باقي الأدوات الأخرى.

و لمزيد من التحسينات انظر نافذة الخصائص .

تدريب نموذجي

1. أنشئ مشروعاً جديداً

2. قم بإدراج الأدوات التالية ImageList ، Toolstrip

3. أضف أيقونات إلى ImageList (راجع الدرس الأداة)

4. اكتب التعليمات التالية في حدث تحميل النموذج

VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As _
    System.EventArgs) Handles Me.Load
    ToolStrip1.ImageList = ImageList1
    ToolStrip1.Items.Add("حفظ", ImageList1.Images(0))
    ToolStrip1.Items.Add("نقطتي تعداد", ImageList1.Images(1))
End Sub
```

و لاحظ الناتج



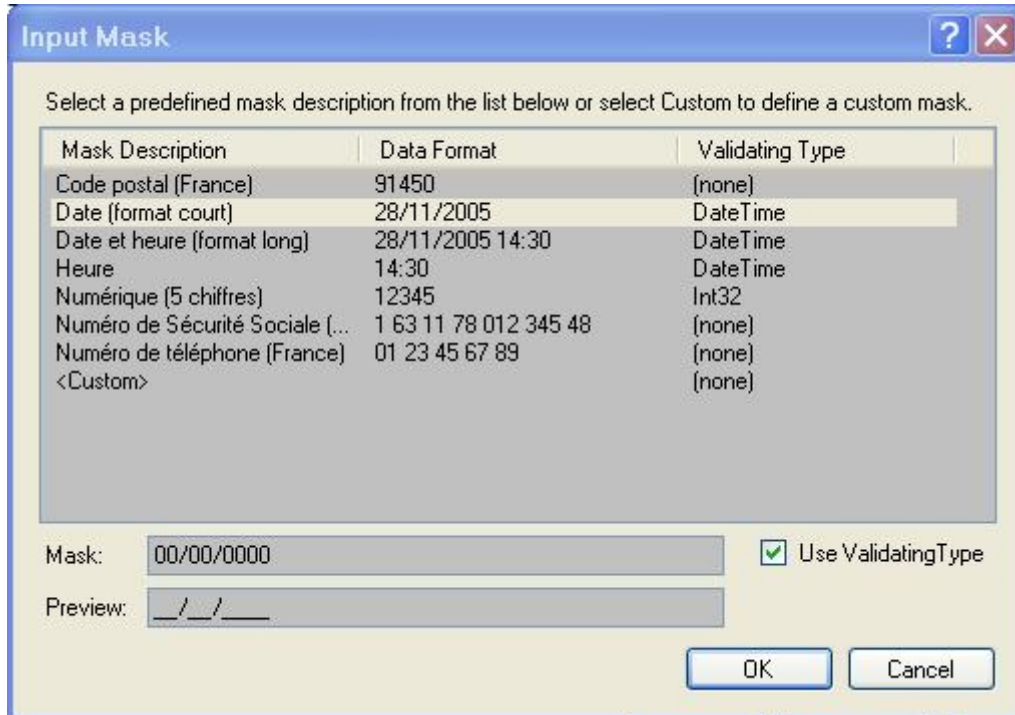
أداة MaskedTextBox

تستعمل لتنسيق البيانات الرقمية أو التاريخ و الوقت .و ذلك بتقييد و إرغام المستخدم على إدخال البيانات حسب التنسيق المحدد لتجنب حدوث الأخطاء.

كيف نعمل ذلك ؟

1. قم بإدراج الأداة MaskedTextBox إلى النموذج

2. من نافذة الخصائص اختر الخاصية Mask تظهر النافذة التالية



يمكنك من هذا التحديد بتنسيق التاريخ .انظر الخيار Mask يحدد لك نمط التنسيق

00 يمثلان رقمين للدلالة على اليوم / فاصل

00 يمثلان رقمان للدلالة على الشهر / فاصل

0000 أربعة أرقام تعبر عن السنة

مثال : 25/02/2012

كما يمكنك تخصيص تنسيق وذلك باختيار Custom فاكتب التنسيق المراد في خانة Mask و ليكن 0000 هذا يجبر المستخدم على إدخال أربعة أرقام أو أقل .

الفصل الرابع : النماذج و تطبيقات أخرى

- التعامل مع تطبيق الاختبار Console Application
- التعامل مع تطبيق الويب Web Application
- مستعرض الانترنت Explorer Web
- التعامل مع النماذج Forms
 - خصائص النموذج في وضع التصميم
 - خصائص النموذج من خلال الكود
 - الخصائص الوظيفية
 - وسائل النموذج
 - أحداث النموذج
 - نموذج بدء التشغيل
 - نماذج متعددة الوثائق Mdi

التعامل مع تطبيق الاختبار Console Application

هو تطبيق ذو واجهة سوداء وهي تشبه شاشة DOS حيث يتم فيها إدخال البيانات و إخراجها على سبيل الاختبار

لا يحتوي على أدوات فبمجرد التنفيذ تظهر لك النتائج دون استعمال الأدوات كالأزرار .

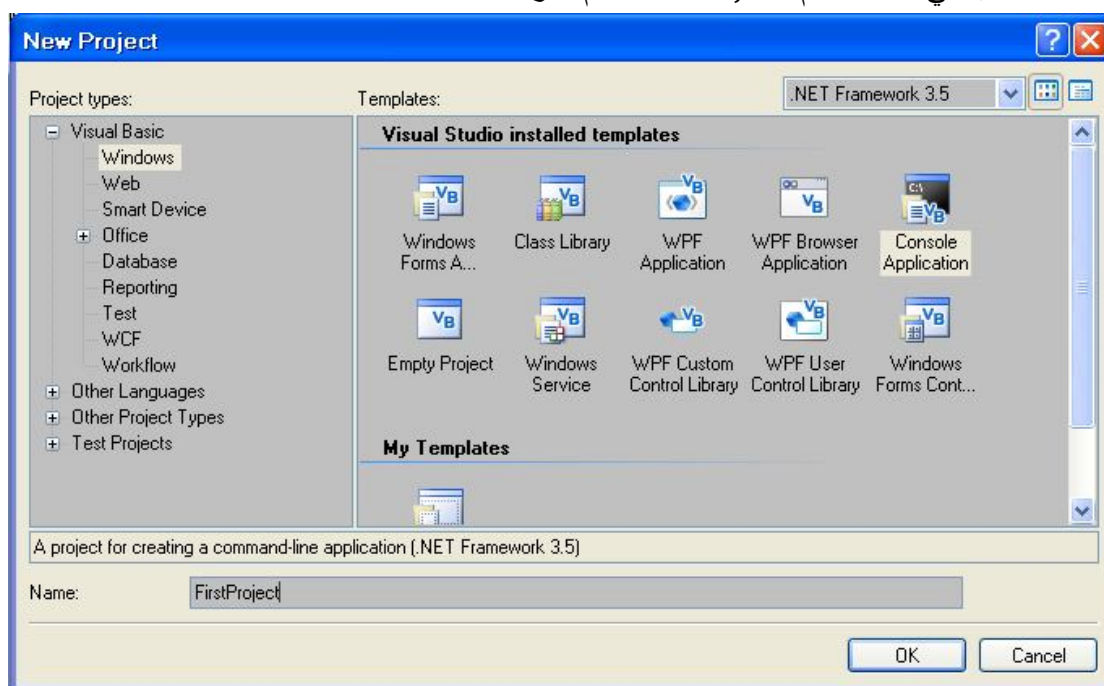
إنشاء تطبيق اختبار جديد Create New Console Application

1. من قائمة File اختر New Project

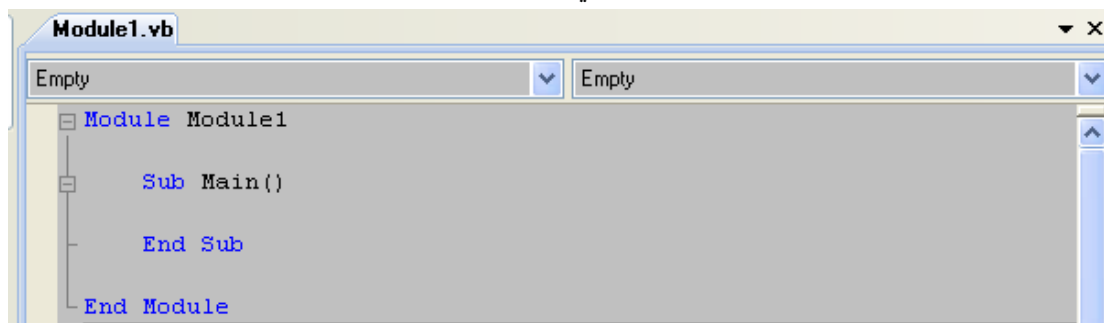
يظهر مربع حوار إنشاء مشروع جديد

2. اختر التطبيق Console Application

اكتب في خانة الاسم FirstProject ثم انقر OK



3. تظهر نافذة محرر الكود كالتالي



إدخال و إخراج البيانات

لإدخال البيانات إلى تطبيق الاختبار Console Application استعمال الكلمتين المحجوزتين

WriteLine : لإدخال البيانات في سطر جديد

الفصل الرابع النماذج و تطبيقات لأخرى

Input : لإدخال البيانات مع إمكانية الكتابة كما في صندوق الإدخال **Write** ►
Box

لقراءة و عرض البيانات استعمل الكلمتين المحجوزتين

ReadLine ► : لقراءة البيانات في هذا سطر

Read ► : لقراءة البيانات

4. كتابة التعليمات



VB.Net Code

```
Sub Main()
```

```
Console.WriteLine("In the name of Allah the most gracious and the most merciful") ' البسمة فاتحة خير
```

```
Console.Write("Entre Your Name") ' ادخل اسمك
```

```
Dim S As String = Console.ReadLine ' حفظ اسمك في متغير
```

```
Console.Write("Entre Your Age") ' ادخل عمرك
```

```
Dim A As Integer = Console.ReadLine ' تخزين قيمة العمر في متغير
```

```
Console.WriteLine("Hello Mr : {0}, You are {1} years Old", S, A) ' عرض
```

الناتج

```
Console.ReadLine() ' قراءة و عرض الناتج
```

```
End Sub
```

5. تنفيذ البرنامج

بعد كتابة اسمك و عمرك تحصل على الناتج

```
In the name of Allah the most gracious and the most merciful
Entre Your Name Salah
Entre Your Age 45
Hello Mr : Salah, You are 45 years Old
```



ملاحظة : في السطر ما قبل الأخير في المعامل الأول نلاحظ الرمزين

{0} : يعني ما يوجد في المعامل الأول أي S

{1} : يعني ما يوجد في المعامل الثاني أي A

تنسيق البيانات في تطبيق الاختبار

أولا : ترتيب المخرجات

لو عندنا التعليمة التالية :

```
Console.WriteLine("{1}, {0}, {2}", 10, 20, 30)
```

المخرجات تكون مرتبة على النحو التالي :

20 , 10 , 30

تنسيق الأرقام :

الفصل الرابع النماذج و تطبيقات لأخرى

يستعمل Vb.Net رموز خاصة في تنسيق الأرقام أنظر الجدول أدناه

الرمز	المدلول
C	عملة Currency
D	عشري decimal
E	علمي Scientific
F	مقربة لأقرب 3 علامات عشرية
G	عام General
N	تنسيق رقم Number
x	عدد سداسي عشر

تطبيق :



VB.Net Code

```
Sub Main()  
Console.WriteLine("C format: {0:C}", 99989.987)  
Console.WriteLine("D9 format: {0:D9}", 99999)  
Console.WriteLine("E format: {0:E}", 99999.76543)  
Console.WriteLine("F3 format: {0:F3}", 99999.9999)  
Console.WriteLine("N format: {0:N}", 99999)  
Console.WriteLine("X format: {0:X}", 99999)  
Console.WriteLine("x format: {0:x}", 99999)  
Console.ReadLine()  
End Sub
```

يكون الناتج :

```
C format: 99 989.99 ?  
D9 format: 000099999  
E format: 9.999977E+004  
F3 format: 100000.000  
N format: 99 999.00  
X format: 1869F  
x format: 1869f
```

تنسيق الأعداد باستعمال String.Format

باستعمال الرموز المبينة في الجدول السابق نستطيع تنسيق النصوص مع الدالة String.Format

كما يلي :



VB.Net Code

```
Sub Main()  
Dim formatStr As String  
formatStr = _  
String.Format("Number Is {0:E} ", 99989.987)
```

```
Console.WriteLine(formatStr)
```

```
Console.ReadLine()
```

```
End Sub
```

التعامل مع تطبيق الويب Web Application

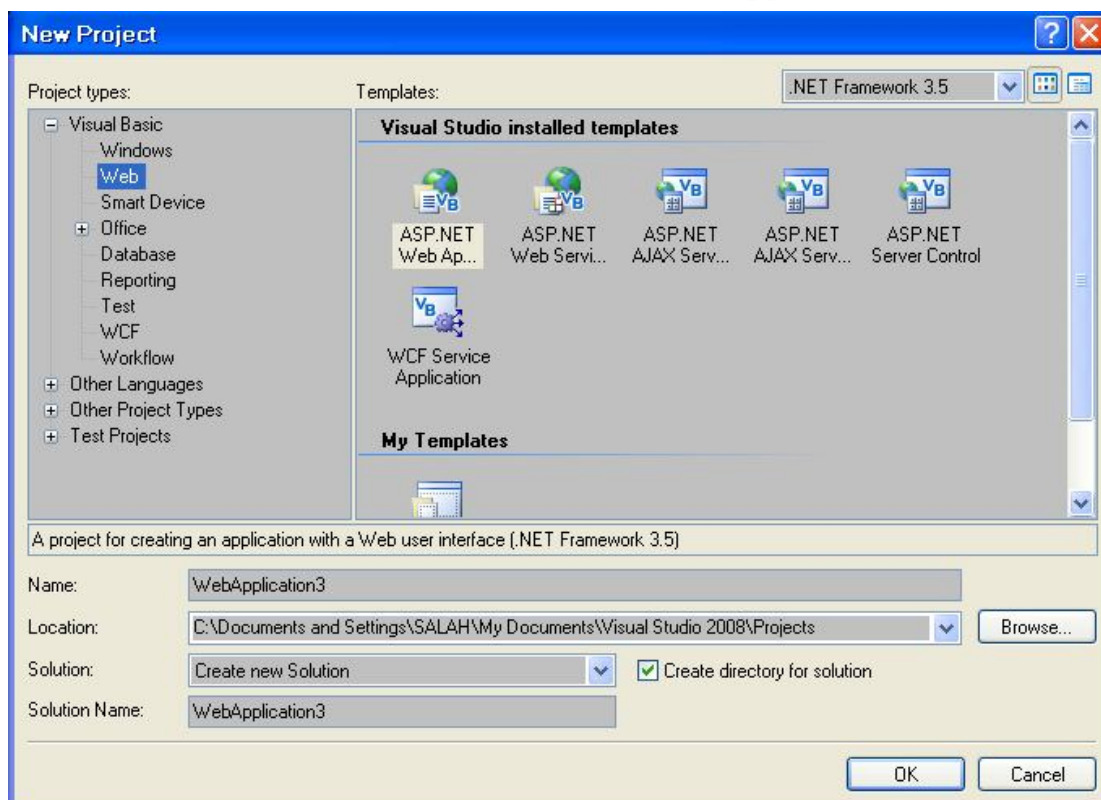
تممكنك مثل هذه التطبيقات في إنشاء صفحات ويب و كذا صفحات المساعدة Help لإنشاء مشروع ويب اتبع الخطوات التالية :

1. من قائمة File اختر الأمر New Project

يظهر مربع حوار New Project

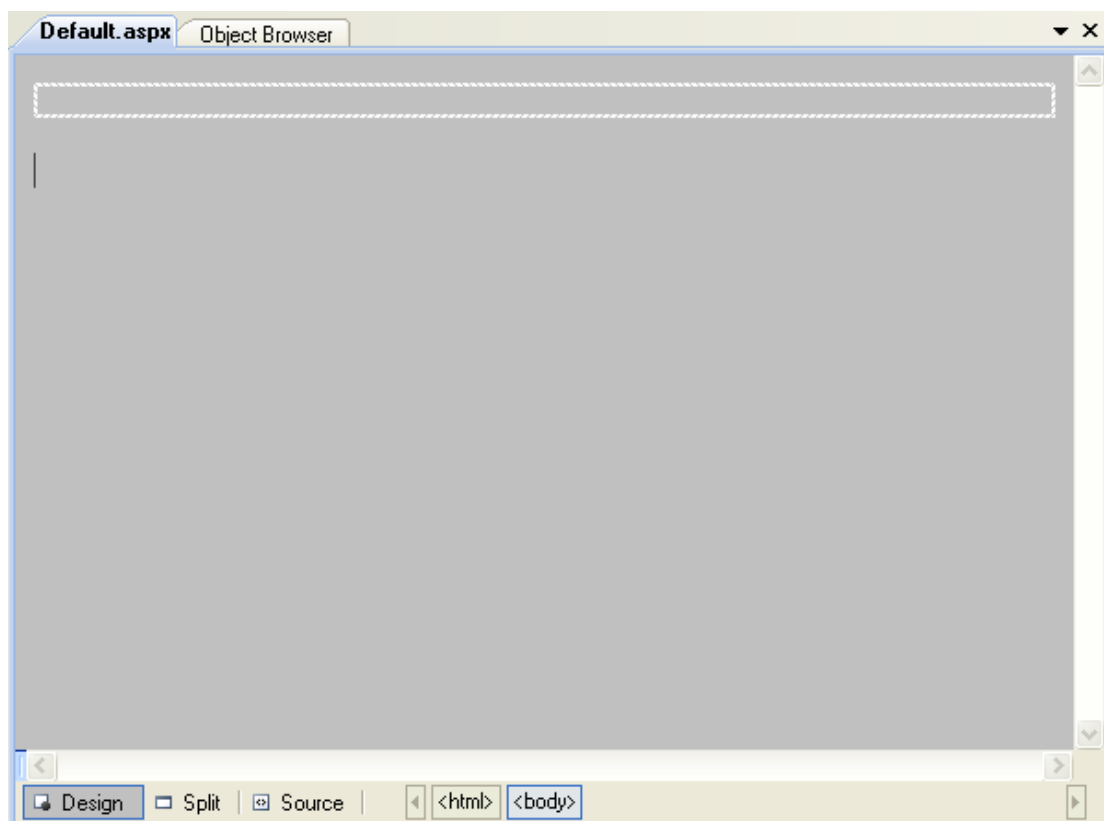
2. اختر نوع المشروع Web من القسم الأيسر

3. اختر التطبيق ASP.Net Application

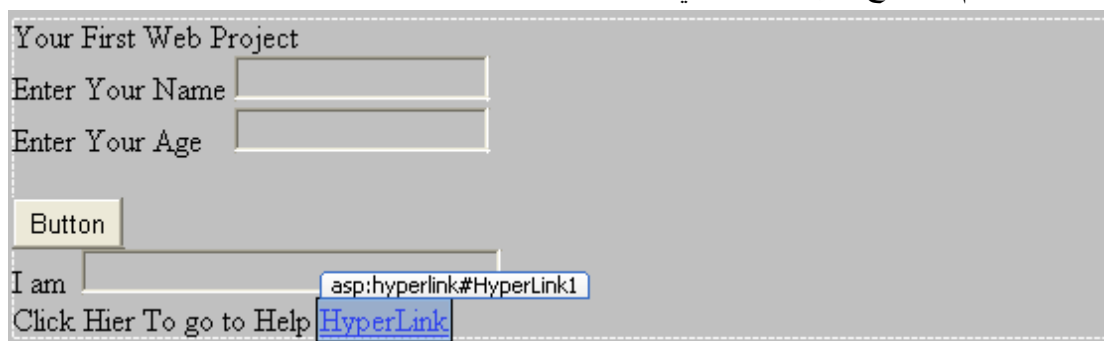


4. انقر Ok و ترك كل الافتراضيات كما هي

تظهر النافذة الرئيسية Default و أدوات خاصة بصفحات ويب



5. نضيف الأدوات التالية من صندوق الأدوات Tool Box من تبويب Standard و هذا بالضبط المزدوج على الأدوات: 3 مربعات نصوص TextBox و زر Button و الرابط HyperLink بحيث نصمم النموذج على النحو التالي :



بالضبط المزدوج على الزر Button ننتقل إلى محرر الكود و نكتب التعليمات

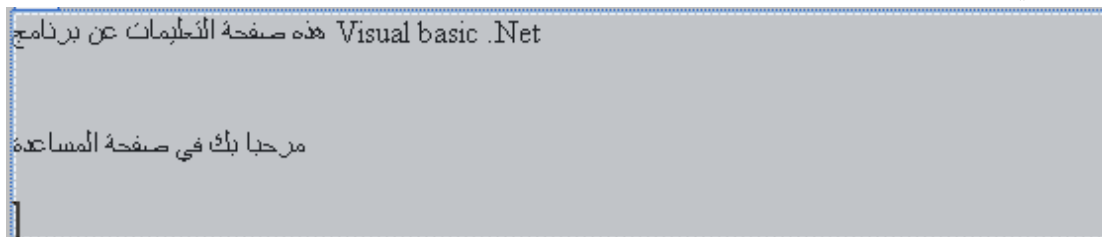
```

VB.Net Code
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles Button1.Click
    TextBox3.Text = TextBox1.Text + " My age is " + TextBox2.Text + "
Old"
End Sub
    
```

5. نضيف صفحة جديدة WebForm من القائمة Project ثم Add Component

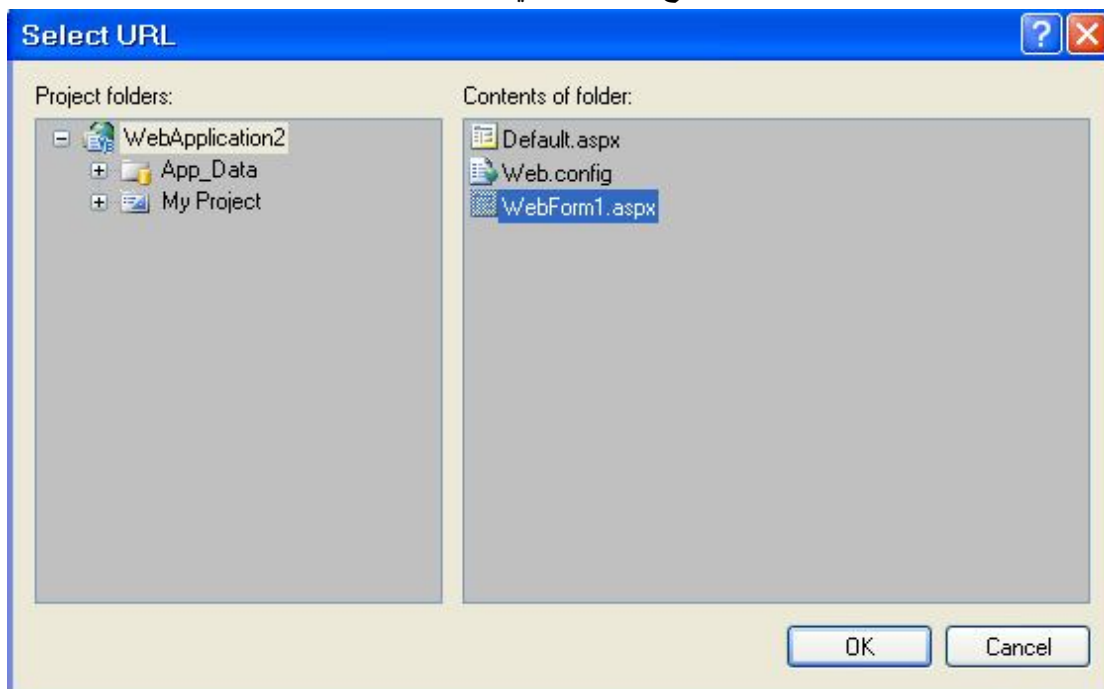
يظهر مربع حوار Add New Item

6. اختر الأيقونة Web Form ثم انقر Ok دون تغيير اسمها تظهر صفحة جديدة اكتب فيها ما يلي



7. حدد أداة الربط HyperLink من نافذة الخصائص اختر الخاصية NavigateUrl

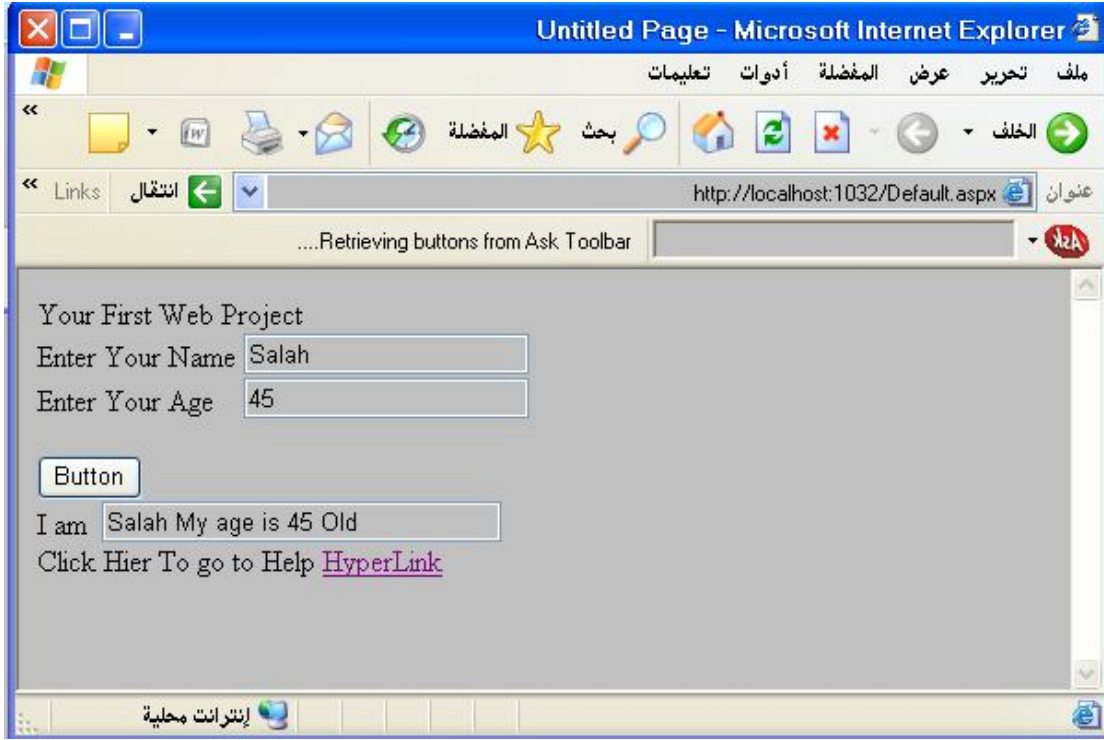
8. انقر زر الاستعراض ... تظهر مربع الحوار التالي



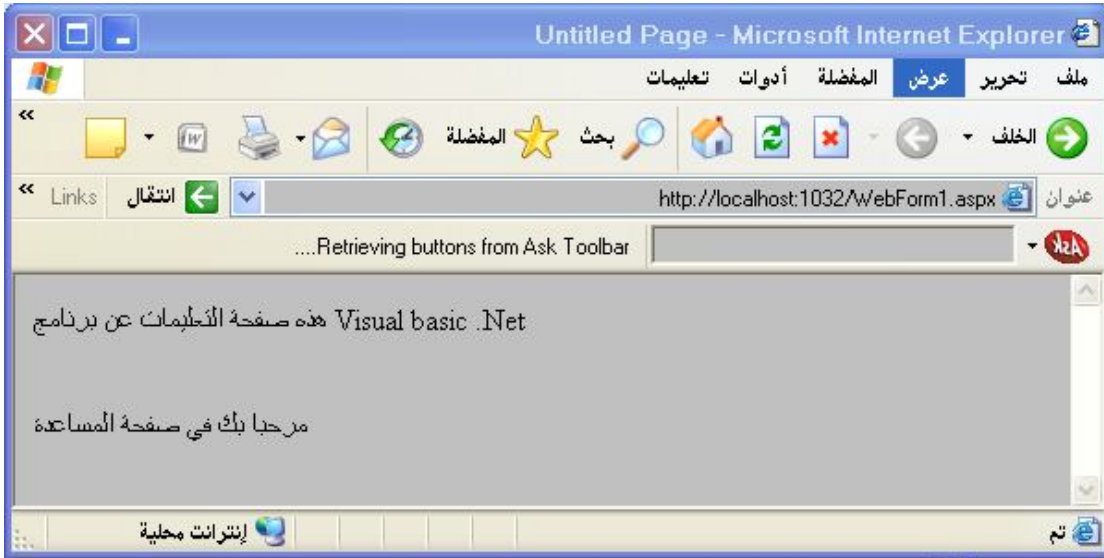
اختر WebForm1 كما هو محدد في الصورة ثم انقر Ok

9. شغل البرنامج بالضغط F5 ثم اكتب اسمك وعمرك و انقر الزر و لاحظ الناتج في

صفحة مستعرض الانترنت يظهر مربع حوار انقر Ok



و عند النقر على الرابط HyperLink تظهر الصفحة التالية



مستعرض الانترنت Explorer Web

1. من قائمة File اختر New Project يظهر مربع حوار
2. اختر التطبيق Windows Forms Application و اكتب في خانة الاسم MyProjetWeb
3. انقر Ok ثم نقوم بإضافة الأدوات التالية من تبويب All Windows Forms :
3 أزرار Button و TextBox و WebBrowser
4. نحدد خصائصها حسب الجدول التالي :

الفصل الرابع النماذج و تطبيقات لأخرى

الأداة	الخاصية	القيمة
Button1	Text	Go
Button2	Text	Back
Button3	Text	ForWard
Textbox	Text	اكتب الرابط

بحيث يبدو الشكل ما يلي :



5. انقر نقرًا مزدوجًا على الزر Go و اكتب التعليمات التالية في حدث نقر الزر.



VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
    WebBrowser1.Navigate(TextBox1.Text)  
End Sub
```

و في حدث نقر زر Back نكتب



VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
    WebBrowser1.GoBack  
End Sub
```

و في حدث نقر زر ForWard نكتب



VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
    WebBrowser1.GoForward  
End Sub
```

6. نفذ البرنامج بالضغط F5

الفصل الرابع النماذج و تطبيقات لأخرى

7. اكتب في خانة مربع النص www.google.com وانقر Ok سيتم فتح صفحة الويب إن كان جهازك متصلا بالإنترنت . وعند النقر على زر Back تنتقل إلى الصفحة السابقة و النقر على Forward تنتقل إلى الصفحة التالية .

التعامل مع النماذج Forms

النموذج هي النافذة الرمادية المنقطة التي تحوي الأدوات التي تستعملها في مشاريعك .
أهم خصائص النموذج في وضع التصميم

الخاصية	وظيفتها
Name	الاسم البرمجي للأداة الذي نستعمله في الكود
Acceptbutton زر الموافقة	إذا كانت True تجعل الزر يستجيب لحدث النقر على Enter من لوحة المفاتيح مثل : زر Ok أو موافق ...
AllowDrop	هل يتم السماح او عدم السماح بعمليات السحب والإفلات
AutoScroll	إذا كان حجم الأداة على النموذج أكبر من حجم النموذج يتم إضافة أداة انزلاق إلى النموذج حتى تتمكن من استعراض ما عليه.
Autosize تحجيم تلقائي	إذا كان حجم النموذج أصغر يجعل بعض الأدوات لا تظهر فعند تفعيل هذه الخاصية إلى True يتم تحجيم النموذج تلقائيا بحيث تظهر كل الأدوات
BackColor	لون الخلفية للأداة.
Backgroundimage	يمكنك من هنا اختيار صورة ليتم وضعها في خلفية الأداة.
BackgroundImageLayout	اختيار طريقة التخطيط للصورة، هل يكون بوضعها العادي أم يتم تمددها لتسع كامل المساحة أم يتم تكرارها ...
CancelButton	اختيار الزر الذي يستجيب بحال الضغط على المفتاح Esc او زر الهروب مثل : Cancel أو إلغاء الأمر
ConTextMenuStrip القائمة المختصرة	اختيار القائمة التي سيتم عرضها بحال تم النقر بزر الفأرة الأيمن على الأداة
ControlBox صندوق أدوات النموذج	هل يتم عرض او لا يتم عرض شريط أدوات التحكم الخاص بالنموذج.
Cursor	اختيار مؤشر الفأرة الذي ترغب في ظهوره على الأداة
Enabled	هل يتم تمكين أم عدم تمكين الأداة

الفصل الرابع النماذج و تطبيقات لأخرى

اختيار نوعية الخط وحجمه ولونه وما إلى ذلك	Font
لون الخط	FontColor
اختيار نوعية الإطار للنموذج أو الأداة	FormBorderStyle
اختيار أيقونة ليتم وضعها للنموذج	Icon
هل تريد ان يكون هذا النموذج من النوع الحامل متعدد النوافذ MDI	IsMdiContainer
هل تريد عرض زر النموذج الخاص بتكبير النموذج	MaximizeBox
هل تريد عرض الزر النموذج الخاص بتصغير النموذج	MinimizeBox
اختيار درجة شفافية النموذج بوضع أرقام عشرية من 0 إلى 1	Opacity
من اليمين الى اليسار	RightToLeft
تقنية المرآة أي قلب كل شئ لجهة اليمين أو العكس	RightToLeftLayout
هل أقوم بعرض الأيقونة أم لا	ShowIcon
هل تريد ان ترى أيقونة النموذج في شريط المهام الخاص بنظام التشغيل والملتصق بزر أبداً أم لا	ShowInTaskpar
هل تريد عرض أداة تغيير الحجم التي تظهر في أسفل النموذج	SizeGripStyle
ما هي الطريقة التي تريد أن يتم عرض النموذج بها، هل يتم ذلك يدوياً أم في منتصف الشاشة الخ..	StartPosition
النص الذي سيظهر على الأداة أو العنوان	Text
هل تريد أن يظهر النموذج فوق كل النماذج الأخرى	Topmost
اختيار اللون الذي تريد أن يكون شفافاً ، اذا كان لون النموذج أبيض واخترت الأبيض فسيكون شفافاً.	Transparenckey
هل تريد ان يتحول مؤشر الفأرة الى وضع الانتظار على الأداة	UseWaitCousor
كيف تريد عرض النموذج هل عادي، أم بكامل حجمه، أم في وضع التصغير	WindowState

طراز الإطار FormBorderStyle:

تتحكم هذه الخاصية في مظهر النموذج، وهي تأخذ واحدة من القيم الموضحة في الجدول التالي:

سيظهر النموذج بدون إطار وبدون شريط عنوان، بحيث لا يمكن	بدون None
--	-----------

الفصل الرابع النماذج و تطبيقات لأخرى

تغيير حجمه أو تحريكه.. حاول أن تتجنب هذه القيمة.	متغير الحجم Sizable
وهي القيمة الافتراضية، حيث يتم عرض النموذج التقليدي، القابل للتحريك وتغيير الحجم.	
نموذج بإطار مرئي، مرتفع عن مساحة النموذج الرئيسية، ولكن لا يمكن تغيير حجمه.	مثبت مجسم Fixed3D
نموذج يستخدم في مربعات الحوار، لا يمكن تغيير حجمه.	مثبت حوار FixedDialog
نموذج لا يمكن تغيير حجمه، بإطار عبارة عن خط مفرد.	مثبت مفرد FixedSingle
نموذج لا يمكن تغيير حجمه، بزر إغلاق فقط، وبدون زري تصغير وتكبير، وبدون الأيقونة ومربع التحكم.	نافذة أدوات مثبتة FixedToolWindow
مماثل للطراز السابق، لكنه قابل لتغيير حجمه.	نافذة أدوات متغيرة الحجم SizableToolWindow

طراز علامة تغيير الحجم SizeGripStyle:

تتمكنك هذه الخاصية من عرض علامة في ركن النموذج السفلي الأيمن، لتدل على قابلية النموذج لتغيير

حجمه.. وهي تأخذ واحدة من هذه القيم

علامة تغيير الحجم ستظهر عند الاحتياج إليها.. وهي القيمة الافتراضية.	تلقائي Auto
سيتم عرض علامة تغيير الحجم دائما.	عرض Show
علامة تغيير الحجم لا تعرض على الإطلاق.. ولكن هذا لا يمنع تغيير حجم النموذج بلوحة المفاتيح كالمعتاد.	إخفاء Hide

موضع البداية StartPosition:

تحدد هذه الخاصية موضع النموذج عندما يتم عرضه للمرة الأولى.. وهي تأخذ القيم التالية:

سيتم توسيط النموذج حول مركز النافذة التي تحتويه.	مركز محتويه CenterParent
سيتم توسيط النموذج حول مركز الشاشة.	مركز الشاشة CenterScreen
سيعرض النموذج تبعا لموضعه وحجمه كما تحددتها أنت.	يدوي Manual
سيعرض النموذج تبعا للموضع والحجم الافتراضيين كما يحددهما الويندوز.	حدود الويندوز الافتراضية WindowsDefaultBounds
سيتم عرض النموذج تبعا للموضع الافتراضي كما يحدده	موضع الويندوز الافتراضي WindowsDefaultLocation

الويندوز، ولكن بالحجم الذي حدّدته أنت.

حالة النافذة WindowState:

يمكنك بهذه الخاصية قراءة أو تغيير حالة النافذة.. ولهذه الخاصية 3 قيم

عادي	Normal
مكبرة	Maximized
مصغرة	Minimized

التحكم في ترتيب الانتقال بين الأدوات

هنا ميزة أخرى يمنحها لك Vb.Net غير الخاصيتين Tab Index و Tab Stop لترتيب انتقال الأدوات في النموذج . اضغط قائمة View ثم اختر الأمر Tab Order سيتم ترقيم الأدوات على النموذج حسب ترتيب إضافتها أو ترتيبها أنت وفق الخاصية TabIndex . ثم اضغط بالفأرة على الأدوات بالترتيب الذي تريده للانتقال إليها . وحتى اللافتات Lables بعد الانتهاء عد إلى قائمة View ثم الأمر Tab Order للعودة إلى الوضع العادي.



أزرار الوصول Access Keys

هي الأزرار التي تستعمل عند ضغط على Alt + حرف . في المثال السابق نريد الوصول إلى خانة الرقم ماذا نعمل ؟

نضع العلامة "&" قبل الحرف الذي تريده في خاصية Text للافتة فمثلا : نضع العلامة "&" في خاصية لافتة الرقم قبل الحرف ق لاحظ وجود سطر تحت حرف ق . يعني هذا إذا ضغط على Alt + ق يتم الانتقال إلى مربع النص المقابل للافتة الرقم . وهذا لأن الانتقال لا يختص للافتة و ينتقل إلى الأداة الموالية في الترتيب وهي مربع النص Textbox1 و يمكنك بنفس الطريقة أن تضع أزرار الوصول لبقية صناديق النصوص حتى يبدو الشكل هكذا :



أهم خصائص النموذج من خلال الكود



VB.Net Code

```

Me.Text = "النموذج عنوان"
Me.AcceptButton = Button1 ' زر الموافقة
Me.CancelButton = Button2 ' زر إلغاء الأمر
Me.FormBorderStyle = FormBorderStyle.FixedDialog ' تحديد نوع الإطار
Me.MaximizeBox = False ' عدم إظهار زر التكبير
Me.StartPosition = FormStartPosition.CenterScreen ' موضع الانطلاق للنموذج
Me.AutoScroll = True ' ظهور المنزلاقات تلقائيا
Me.AutoSize = True ' تحجيم تلقائي يجعل كل الأدوات ظاهرة
Me.BackColor = Color.Blue ' لون خلفية النموذج
Me.BackgroundImage = imageList1.Images(0) ' صورة لخلفية النموذج
Me.BackgroundImageLayout = ImageLayout.Center ' موضع الصورة
Me.Enabled = False ' عدم تمكين النموذج
Me.Opacity = 0.5 ' شفافية
Me.MinimumSize = New Size(400, 300) ' أصغر حجم ممكن
Me.Cursor = Cursors.WaitCursor ' نوع المؤشر

لجعل النموذج متعدد النوافذ MDI
Me.IsMdiContainer = True ' جعل النموذج متعدد النوافذ
ChildForm.MdiParent = Me ' تحديد النموذج الأب
ChildForm.Show() ' عرض النموذج الفرعي
    
```

أهم خصائص النموذج الوظيفية

يأسر الفأرة Capture

تعيد True إذا كان مؤشر الفأرة فوق النموذج

معرفة النموذج النشط



VB.Net Code

```
Dim F As New Form1
```

```
F.Text = "النشط النموذج أنا"
```

```
F.Activate() ' تنشيط النموذج
```

```
F.Show() ' عرض النموذج
```

```
MsgBox(Form.ActiveForm.ToString) ' عرض النموذج النشط
```

مستطيل النموذج ClientRectangle:

استخدم هذه الوسيلة للحصول على كائن مستطيل Rectangle يمثل مساحة النموذج.



VB.Net Code

```
Dim r As Rectangle = Me.ClientRectangle
```

```
Me.Bounds = Me.RectangleToScreen(r) ' جعل
```

حجم النموذج ClientSize:

تستخدم هذه الوسيلة للحصول على كائن حجم Size يمثل أبعاد النموذج.



VB.Net Code

```
If (Me.ClientSize.Width = 0) Or _ '0 = العرض الخاص بالنموذج الخاص بالعرض
```

```
(Me.ClientSize.Height = 0) Then Exit Sub '0 = الارتفاع الخاص بالارتفاع
```

البرنامج

```
MsgBox(Me.ClientSize.ToString) ' عرض احاثيات حجم النموذج
```

موضع النموذج Location:

تستطيع بهذه الخاصية قراءة وتغيير موضع النموذج على الشاشة.



VB.Net Code

```
MsgBox(Me.Location.ToString) ' قراءة موضع النموذج
```

```
Me.Location = New Point(100, 200) ' تغيير موضع النموذج
```

أزرار الفأرة MouseButton:

تستعمل لمعرفة الزر المضغوط من الفأرة عندما يكون المؤشر فوق النموذج.



VB.Net Code

```
Private Sub Form1_MouseDown(ByVal sender As Object, _  
    ByVal e As System.Windows.Forms.MouseEventArgs) Handles  
Me.MouseDown  
    MsgBox(MouseButtons.ToString)  
End Sub
```

موضع الفأرة MousePosition:

تعيد لك موضع الفأرة بالنسبة لإحداثيات الشاشة.



VB.Net Code

```
Private Sub Form1_MouseClick(ByVal sender As Object, _  
    ByVal e As System.Windows.Forms.MouseEventArgs) Handles  
Me.MouseClick  
    Me.Text = MousePosition.ToString  
End Sub
```

أزرار التحكم ModifierKeys:

تعيد ما إذا كان أيا من أزرار الفأرة مضغوطا .

الأدوات Controls:

هذه الخاصية عبارة عن مجموعة Collection، تحتوي كل الأدوات الموجودة على النموذج وتمكنك من إضافة الأدوات إلى النموذج.



VB.Net Code

```
Dim BtOk As New Button  
BtOk.Text = "موافق"  
BtOk.Location = New Point(10, 40)  
Me.Controls.Add(BtOk)
```

أهم وسائل النموذج Methods

النموذج النشط Active

لجعل النموذج هو الذي يستقبل أحداث لوحة المفاتيح أي النموذج النشط



VB.Net Code

```
Dim form2 As New Form1  
form2.Visible = True  
form2.Text = "آخر نموذج"  
Me.Activate()  
Me.Text = "النشط النموذج أنا"
```

إغلاق Close

تستعمل هذه الوسيلة لغلاق النموذج.

التخلص من النموذج Dispose:

تغلق النموذج وتحرر أي مساحات كان يشغلها في الذاكرة.

معرفة الأداة الموجودة **GetChildAtPoint**:

تستخدم هذه الوسيلة لكي تعرف الأداة التي تقع على النموذج عند نقطة معينة.



VB.Net Code

```
MsgBox(Me.GetChildAtPoint(New Point(10, 40)).ToString)
```

تحديث **Refresh**:

تعيد رسم النموذج لإعاشته و تحديثه .

تحديد موضع النموذج **SetDesktopLocation**:

تحدّد موضع النموذج على الشاشة.



VB.Net Code

```
Me.SetDesktopBounds(10, 20, 100, 200)
```

إخفاء **Hide**

تستعمل لإخفاء النموذج و لتقلقه و يبقى في الذاكرة.

عرض مربع حوار **ShowDialog**

تستعمل لعرض النموذج على شكل مربع حوار كنموذج مربع الرسائل Messgbox

أهم أحداث النموذج **Events**

حدثا التنشيط **Activated** والإخماد **Deactivated**:

حدث الإغلاق **Closing**:

يحدث هذا الحدث عند إغلاق النموذج.

ويمكنك الاستفادة منه، مثلا في إظهار رسالة تذكّر المستخدم بحفظ التغييرات أو إغلاق النموذج



VB.Net Code

```
Private Sub Form1_FormClosing(ByVal sender As Object, _  
    ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles  
    Me.FormClosing  
    If MsgBox("التغييرات حفظ تريد هل", MsgBoxStyle.YesNo) =  
    MsgBoxResult.Yes Then  
        e.Cancel = True ' البرنامج إغلاق تلغي  
    End If  
End Sub
```

حدث تغيير الحجم **Resize**:

ويحدث كلما تغيّر حجم النموذج.

حدث رسم النموذج **Paint**:

يمكنك من إعادة رسم الأشكال . و هذا مثال لرسم مربع



VB.Net Code

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
```

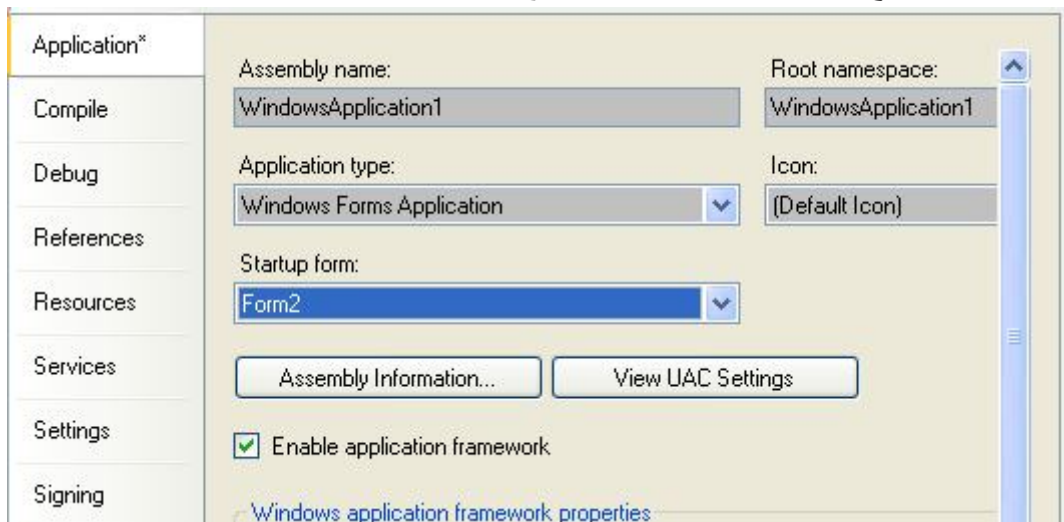
الفصل الرابع النماذج و تطبيقات لأخرى

```
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
    Me.CreateGraphics.FillRectangle(Brushes.Blue, New Rectangle(0, 0, _
Me.Width / 2, Me.Height / 2)) ' رسم مربع '
End Sub
```

الحدث	وظيفته
Click	عند النقر
DoubleClick	عند النقر المزدوج
Enter	يشبه الحدث Load
Load	يحدث عند تحميل النموذج
GotFocus	عند انتقال التركيز الى الاداة
KeyDown	يحدث عند ضغط احد المفاتيح
KeyPress	يحدث عند ضغط احد المفاتيح
KeyUp	يحدث لحظة رفع إصبعك عن المفتاح
Leave	يحدث لحظة المغادرة
LostFocus	يحدث لحظة انتقال التركيز إلى أداة أخرى
MouseDown	النقر بالفأرة
MouseDoubleClick	النقر بالفأرة المزدوج
MouseDown	لحظة النقر بالفأرة ولاحظ أن تشابه الأحداث يختلف فمثلاً هذا الحدث يتم تنفيذه قبل أن تفلت الزر
MouseEnter	لحظة الدخول بالفأرة
MouseHover	شبه مرور الفأرة
MouseLeave	لحظة ترك الفأرة
MouseMove	يحدث عند مرور الفأرة على النموذج أو الأداة
MouseUp	انتقال الفأرة فوق الأداة
MouseWheel	عند تحريك عجلة الفأرة
Resize	لحظة تغيير الحجم
KeyDown	يحدث عند تكون ضاغطا للمفتاح و تستعمل مفاتيح التحكم ك Alt+ ..
KeyPress	يحدث عند الضغط على المفتاح
KeyUp	يحدث عند رفع الضغط على المفتاح
Validating	يستعمل للتحقق من صحة تعبير

نموذج بدء التشغيل Startup Form:

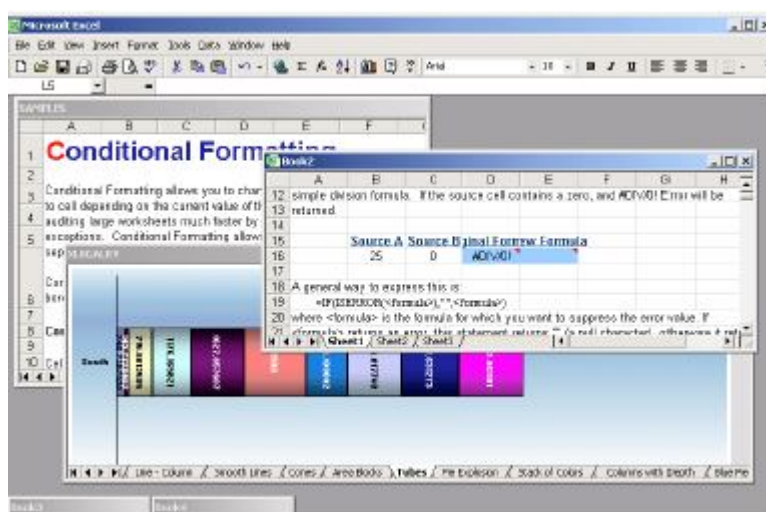
لاختيار نموذج بدء التشغيل، اضغط اسم المشروع في متصفح المشاريع Solution Explorer بزر الفأرة الأيمن، وضغط خصائص Properties من القائمة الموضعية، ليظهر لك مربع حوار خصائص المشروع، حيث يمكنك من اختيار نموذج بدء التشغيل.



النماذج متعددة الوثائق MDI

تحتوي واجهة النماذج المتعددة الوثائق على نموذجي رئيسي يشمل النماذج الفرعية الأخرى و تحتوي أيضا على القوائم Menu و الأشرطة القياسية Standard Bar وهي ما تعرف Multiple Document Interface (MDI).. فكل إجراء تقوم به على النافذة الرئيسية ينطبق على النوافذ الأخرى مثلا : فإذا أراد المستخدم تصغير أو غلق كل نماذج التطبيق، فما عليه سوى تصغير أو غلق النافذة الرئيسية

و أمثلة هذه التطبيقات كثيرة مثل : برنامج Word و Excel



إنشاء واجهة متعددة الوثائق MDI

الفصل الرابع النماذج و تطبيقات لأخرى

1. انشأ مشروعاً جديداً و سمه MDIProject
2. حدد النموذج Form1 و غير الخصائص التالية له حسب الجدول

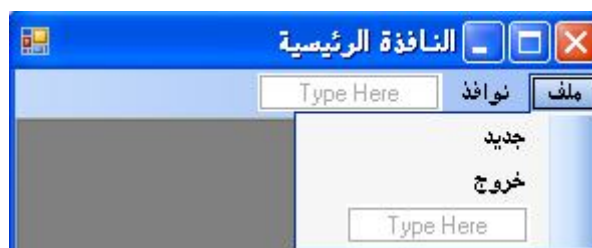
الخاصية	القيمة
Text	النافذة الرئيسية
Name	MDIForm
IsMdiContainer	True لجعل النموذج يمثل النافذة الرئيسية
Right to Left	True

إنشاء القوائم Menus

من صندوق الأدوات Tool Box قم بإضافة الأداة MenuStrip من تبويب & Menus & ToolBars إلى نموذجك ثم اكتب في خانة Type Here ملف بمجرد أن تبدأ الكتابة، سيظهر أمران جديان من أوامر القائمة، أحدهما في مستوى الأمر الأول (على يساره)، والآخر فرعي من الأمر الأول (أسفله).

اكتب في الأمر الفرعي "جديد" New ثم استمر في كتابة عناوين القوائم بنفس الطريقة كما في الجدول

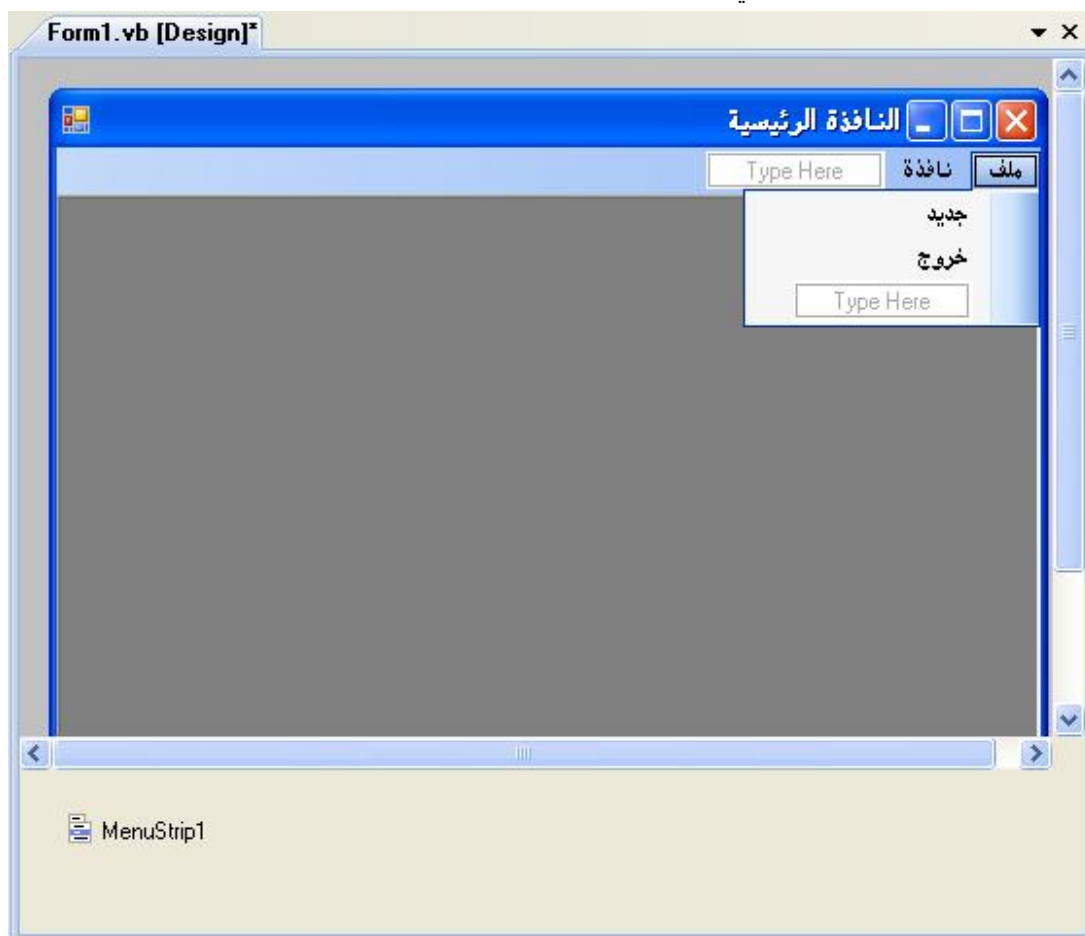
قائمة ملف	الاسم البرمجي
ملف	FileMenu
جديد	FileNew
خروج	FileExit



قائمة النوافذ	الاسم البرمجي
تجاوز أفقي	MenuH
تجاوز رأسي	MenuV
تتالي	MenuCas
ترتيب أيقونات	MenuIcon



و أخيرا نحصل على الشكل التالي :



كتابة التعليمات

نضيف نموذج آخر Form2 كنموذج فرعي

انقر نقرًا مزدوجًا على قائمة جديد ثم اكتب التعليمات



VB.Net Code

```
Dim F As New Form2 ' إنشاء نسخة جديدة
Static i As Integer ' متغير لعدد النسخ
i += 1 ' زيادة عدد النسخ ب 1
F.Text = "رقم نموذج" & I ' عرض عنوان النموذج
F.MdiParent = Me ' جعل النموذج فرعيا
```

عرض النموذج ' F.Show()

انقر نقرا مزدوجا على قائمة إنهاء و اكتب End بين السطرين
ننتقل إلى قائمة نوافذ
انقر نقرا مزدوجا على قائمة تجاوز أفقي و نكتب :



VB.Net Code

```
Me.LayoutMdi(MdiLayout.TileHorizontal)
```

وهكذا بالنسبة لبقية القوائم

```
Me.LayoutMdi(MdiLayout.TileVertical) ' تجاوز عمودي
```

```
Me.LayoutMdi(MdiLayout.Cascade) ' تتالي
```

```
Me.LayoutMdi(MdiLayout.ArrangeIcons) ' ترتيب أيقونات
```

شغل البرنامج ثم اضغط جديد لتحصل على العديد من النسخ ثم اختر الأمر مثلا : تجاوز رأسي و
لاحظ الناتج



مزج قوائم النوافذ الداخلية بقوائم النافذة الرئيسية

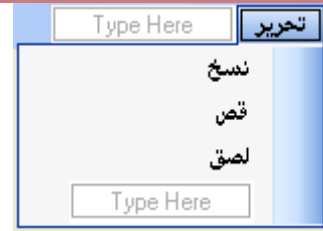
أحيانا تكون قوائم في النافذة الرئيسية متاحة للاستعمال بدون تأثير إلا إذا كانت نافذة أخرى نشيطة
تتعلق أوامر هذه القائمة بها .

فمثلا : في محرر النصوص لا توجد أي نافذة مفتوحة للكتابة رغم ذلك نجد أوامر قائمة التحرير
متاحة في قوائم النافذة الرئيسية فأى شيء ننسخه أو نلصقه أو نلصقه ..؟ لا شيء . فما هو الحل
الأفضل ؟

الفصل الرابع النماذج و تطبيقات لأخرى

ننجز قائمة التحرير ضمن نموذج الوثيقة الخاصة بالتحرير و عند ظهور هذه الوثيقة نمزج قائمة التحرير بقوائم النافذة الرئيسية . كيف نفعل ذلك ؟
أضف القائمة تحرير تحتوى على الأوامر التالية كما تعلمنا سابقا

القائمة	الاسم
تحرير	MenuEdit
نسخ	MenuCopy
قص	MenuCut
لصق	MenuPast





شغل البرنامج بالضغط F5 و لاحظ أنه تم إضافة قائمة تحرير في الأخير
ماذا إذا أرت أن تدمجها في موضع ترغبه ؟
في هذه الحالة استخدم خاصية "ترتيب المزج" MergeIndex لكل قائمة (سواء أكانت قائمة رئيسية أو فرعية) فمثلا، لعرض القوائم بالترتيب التالي: File ثم Edit ثم Window، اجعل قيمة هذه الخاصية في قائمة File = 0، وفي قائمة Edit = 1، وفي قائمة Window = 2.
و مع القائمة Edit التي نريد دمجها أو إدراجها بين القوائم نغير الخاصية Merge إلى Insert و الآن شغل البرنامج و قم بإضافة نسخة جديد و لاحظ الناتج . ألف مبروك !





الفصل الخامس : الفئات الأساسية


Arrays المصفوفات

تعريفها 


Initializing Arrays إسناد القيم 


حدود مصفوفة 

المصفوفة متعددة الأبعاد 


المصفوفات المرنة 

عرض قيم المصفوفة 


ترتيب مصفوفة 


عكس ترتيب مصفوفة 


دمج عناصر مصفوفة 

البحث في مصفوفة 

نسخ مصفوفة 

Object مصفوفة من نوع 

جدول ملخص 

تطبيقات 

Class String فئة النصوص

الخصائص 

▪ Lenght الطول

▪ Chares الحروف

▪ وسائل فئة النصوص 

▪ Compare مقارنة

▪ Concat دمج

▪ Copy نسخ

▪ Copy To نسخ إلى

▪ Join ربط



















▪ Split تجزئة

▪ SubString أخذ أجزاء

▪ Start With يبدأ ب

▪ EndWith ينتهي ب

▪ البحث

- Insert إدراج
- Remove حذف
- المحاذاة
- فئة بانتي النصوص String Builder 
- تعريفها 
- استعمالها 
- ملخص عن النصوص 
- تطبيقات 
- فئة التاريخ و الوقت 
- خصائصها 
- وسائلها 
- تنسيق التاريخ و الوقت 
- تطبيقات 
- المعاملات 
- الرياضية 
- المقاربة و المقارنة 
- المعامل Like 
- المعاملات المنطقية 
- الدوال الرياضية 
- تعريفها و استخدامها 
- جدول لأهم الدوال الرياضية 

المصفوفات Arrays

المصفوفة : نوع المتغيرات Variables تخزن مجموعة من البيانات من نفس النوع تستعمل لعرض بيانات رتيبة متكررة من نفس النوع . كحساب المتوسط الحسابي لمعدلات مادة من المواد لطلاب صف . تتكوّن المصفوفة من مجموعة من الخانات، كل خانة تحتفظ بقيمة معينة، بحيث يمكن الوصول لهذه القيمة عن طريق رقم خانتها Index. و لكن كيف نعرف المصفوفات ؟

تعرف مثل المتغير العادي و لكن نضيف قوسين فارغين أو بينهما رقم يمثل عدد خاناتها بعد اسم المتغير

Dim NameArrays(Index) As Type


نوع البيانات As (رقم الخانة) اسم المصفوفة Dim

و الآن نعرف مصفوفة طلاب تتكون من 10 خانات

Dim Students(9) As String

Dim Students(0 To 9) As String

عرفنا مصفوفة طلاب تحتوي على 10 خانات أو عناصر .

 ملاحظة : الترقيم في المصفوفات يبدأ من 0

إسناد القيم للمصفوفات Initializing Arrays

قلنا أن المصفوفات عبارة عن سلسلة من متغيرات لها أسماء و قيما من نفس النوع و كل متغير يمثل خانة لها رقم و قيمة انظر المثال

0	1	2	3	4	5	6	7	8	9
سامح	عبد الله	سلمى	محمد	نزيهة	خديجة	سعيد	خالد	سعاد	علي

لاحظ : رقم الخانة 3 في المصفوفة تمثل الطالب محمد
ونكتب هذا بالكود :

Students(3) = "محمد"

Students(6) = "سعيد"

و هكذا بالنسبة لباقي الخانات

و يمكنك تخصيص قيما لمصفوفة في سطر واحد بكيفية سريعة و هذا ما يعرف بالمصفوفات المرنة الديناميكية .

كيف نعرف مصفوفة مرنة و نسد لها قيما ؟

نعرفها مثل المصفوفات العادية و لكن دون وضع طول لها أي نترك القوسين فارغين و لتخصيص قيما لها بعد نوعها نكتب = و رمز المجموعات { } و نضع العناصر داخل المجموعة و نفصل بينها بفاصلة

`Dim Students() As String = {"علي", ".....", "الله عبد", "سامح"}`

في حالة الأرقام

`Dim Numbers () As Integer = {1,2,3,.....}`

وحتى الكائنات و الألوان يمكن أن نضعها في مصفوفات

Code Vb.Net

```
Dim Colors(2) As Color
Colors (0) = Color.Red
Colors (1) = Color.Blue
Colors (2) = Color.Green
```

حدود المصفوفة Array Limits

يمكنك أن تعرف رقم الخانة الأخيرة في المصفوفة، باستخدام الدالة "الحد الأعلى" UBound()، التي جاء اسمها اختصارا لتعبير Upper Bound، وهي تأخذ اسم المصفوفة كمعامل، وترجع رقم آخر خانة بها.. ويمكنك أن تحصل على نفس النتيجة، باستخدام الدالة "اقرأ الحد الأعلى" GetUpperBound و لمعرفة الحد الأدنى نستعمل الوسيلتين GetLowerBound أو LBound

Code Vb.Net

```
Dim Students(9) As Integer
MsgBox(UBound(Students)) ' 19
```


الفصل الخامس الفئات الأساسية

```
MsgBox(Students.GetUpperBound(0)) ' 19
MsgBox(LBound(Students)) ' 0
MsgBox(Students.GetLowerBound(0)) ' 0
```

ويمكنك استخدام الخاصية "طول" Length لمعرفة طول المصفوفة
طول المصفوفة هو 10 MsgBox(Students.Length)'
الطريف أيضاً، أنك تستطيع استخدام الوسائل الخاصة بنوع خانات المصفوفة، فلو كانت المصفوفة من النوع
عدد صحيح Integer مثلاً، يمكن استخدام وسائل الأعداد الصحيحة لكل خانة، كالتالي

Code Vb.Net

```
Dim Students (4) As Integer
```

```
Students (0) = 1
```

```
Students (2) = 2
```

```
Students (4) = 15
```

```
MsgBox(Students.Max)' المصفوفة
```

المصفوفة متعددة الأبعاد :Multidimensional Arrays

كل ما تعلمناه من المصفوفات حتى الآن، هو المصفوفات أحادية البعد.. ولكنك كثيراً ما ستحتاج
لاستخدام مصفوفات متعددة البعد، ثنائية وثلاثية وأكثر
فمثلاً، ستحتاج لمصفوفة ثنائية لتمثيل خمس طلاب و معدلاتهم في الصول الثلاثة .. إن المصفوفة ثنائية
البعد تمثل جدولاً: صفوفاً وأعمدة.. وفي حالتنا هذه، سيكون الصف هو رقم الطالب، والعمود هو الفصل.

	0	1	2
رقم الطالب	الفصل الأول	الفصل الثاني	الفصل الثالث
0	(0,0) 12	(0,1) 11	(0,2) 14
1	(1,0)	(1,1)	(1,2)
2	(2,0) 15	(2,1) 14.5	(2,2)
3	(3,0)	(3,1)	(3,2)
4	(4,0)	(4,1)	(4,2) 13

كما تلاحظ معدل الطالب الأول رقمه 0 في الفصل الأول تحصل على معدل 12
معدل الطالب الثالث رقمه 2 في الفصل الثاني تحصل على معدل 14.5
و يمكن تمثل ذلك عن طريقة مصفوفة ثنائية البعد حيث نحتاج لرقم الصف و رقم العمود للوصول إلى الخلية .

Code Vb.Net

```
Dim Score(4, 2) As Double
```

```
Score(0, 0) = 12 ' معدل الطالب الأول في الفصل الأول 12
```

```
Score(0, 1) = 11
```

```
Score(0, 2) = 14
```

```
Score(2, 0) = 15
```

```
Score(2, 1) = 14.5 ' معدل الطالب الثالث في الفصل الثاني 14.5
```

```
Score(4, 2) = 13
```

المصفوفات المرنة :Dynamic Arrays

الفصل الخامس الفئات الأساسية

كثيرا ما تجد نفسك عاجزا عن أن تعرف بالضبط حجم مصفوفتك قبل تشغيل التطبيق، أو قد يحتاج المستخدم لأن يُدخل قيمة جديدة باستمرار، مما يجعلك في حاجة لزيادة طول المصفوفة.. في هذه الحالة تحتاج إلى مصفوفة مرنة، قابلة لزيادة حجمها أو إنقاصه في أي وقت، لتتلاءم مع عدد الخانات التي لديك. ولفعل ذلك، ابدأ بتعريف المصفوفة بقوسين فارغين، بمعنى ألا تحدّد لها طولاً معيناً

Code Vb.Net

```
Dim Score() As Double
```

```
Dim score As Double()
```

مثال عن استعمال المصفوفات المرنة

Code Vb.Net

```
Dim Month() As String 'تعريفها
```

```
ReDim Month(2) 'طولها تحيد
```

```
Month(0) = "جانفي"
```

```
Month(1) = "فيفري"
```

```
MsgBox(Month(1)) 'فيفري يطبع
```

حينما تجد نفسك أثناء البرنامج محتاجا لحجز عدد من الخانات، استخدم تعبير "أعد تعريف" ReDim

Code Vb.Net

```
ReDim Month(3) 'تغيير حجم المصفوفة
```

```
Month(2) = "مارس"
```

```
Month(3) = "أفريل"
```

```
MsgBox(Month(2)) 'مارس يطبع
```

```
MsgBox(Month(1)) 'بقيمتها تحتفظ لم أي شيئا يطبع لا
```

و إذا أردت الحفاظ على البيانات السابقة استعمل ReDim Preserve

Code Vb.Net

```
ReDim Preserve Month(3) 'السابقة البيانات على للحفاظ استعمل
```

```
Month(2) = "مارس"
```

```
Month(3) = "أفريل"
```

```
MsgBox(Month(2)) 'مارس يطبع
```

```
MsgBox(Month(1)) 'القيم على محافظة تبقى
```

عند استعمال Preserve لا يمكن تغيير المصفوفة

و لتدميرها و تفريغ محتوى الذاكرة اكتب Erase Month .

عرض قيم المصفوفات

أولاً : عرض عنصر واحد

لعرض عنصر واحد نحتاج إلى رقم العنصر كما يلي:

Code Vb.Net

```
Dim Myarray() As String = {"A", "b", "c"}  
Debug.Print(Myarray(1)) 'b
```

```
Dim Students (2) As String  
Array1(0) = "Ali"  
Array1(1) = "Salah"  
Array1(2) = "Said"  
Debug.Print(Array1(1)) 'Salah
```

```
Dim Score(,) As Integer = {{0, 12}, {1, 14}, {2, 10}}  
Debug.Print(Score(2, 1)) '10
```

```
Dim Score1(4, 1) As Integer  
Score1(0, 0) = 12 ' الأول الطالب معدل  
Score1(0, 1) = 14  
Score1(1, 0) = 10  
Score1(4, 1) = 8.5 ' الأخير الطالب معدل  
Debug.Print(0, 1) '14
```

ثانياً عرض عناصر كل المجموعة دفعة واحدة

1- بطريقة For...Next

Code Vb.Net

```
Private strFriends(4) As String  
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    strFriends(0) = "Ali"  
    strFriends(1) = "Said"  
    strFriends(2) = "Nadia"  
    strFriends(3) = "Faid"  
    strFriends(4) = "Sara"  
    For i As Integer = 0 To strFriends.GetUpperBound(0)  
        lstFriends.Items.Add(strFriends(i))  
    Next  
    '-----  
    For i As Integer = 0 To UBound(strFriends)  
        lstFriends.Items.Add(strFriends(i))  
    Next  
    '-----  
    For i As Integer = 0 To strFriends.Length - 1  
        lstFriends.Items.Add(strFriends(i))  
    Next  
End Sub
```

2- بطريقة For Each

Code Vb.Net

```
Private strFriends(4) As String
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    strFriends(0) = "Ali"
    strFriends(1) = "Said"
    strFriends(2) = "Nadia"
    strFriends(3) = "Faid"
    strFriends(4) = "Sara"
    For Each strName As String In strFriends
        'Add the array item to the list
        lstFriends.Items.Add(strName)
    Next
End Sub
```

ترتيب المصفوفات Sorting Arrays

يمكنك ترتيب المصفوفة مباشرة، بمجرد استدعاء الوسيلة "ترتيب" Sort الخاصة بخليّة المصفوفات



الصيغة العامة

```
Array.Sort(اسم المصفوفة)
Array.Sort(عدد الخانات, خانة البداية, اسم المصفوفة)
Array.Sort(المصفوفة المراد ترتيبها, المصفوفة القائدة للترتيب)
Array.Sort(عدد الخانات, البداية, المصفوفة المراد ترتيبها, المصفوفة القائدة)
```

Code Vb.Net

```
Dim Alfb() As String = {"a", "z", "b", "x"}
Array.Sort(Alfb)
For i As Integer = 0 To Alfb.Length - 1
    Console.WriteLine(Alfb(i))
Next

-----

Dim Numbers() As Integer = {9, 1, 5, 3}
Array.Sort(Numbers)
For i As Integer = 0 To Numbers.Length - 1
    Debug.Print(Numbers(i)) ' 1,3,5,9
Next
```

ويتم ترتيب العناصر تبعاً لنوعها: نصية أم رقمية أم تواريخ.. وخلاف ذلك
أما لو أحببت أن ترتب جزءاً فقط من المصفوفة، فاستخدم الصيغة التالية من وسيلة الترتيب:



الصيغة العامة

الفصل الخامس الفئات الأساسية

Array.Sort(عدد الخانات ,خانة البداية ,اسم المصفوفة)

الأروع من هذا، هو قدرتك على ترتيب مصفوفة تبعاً لقيم مصفوفة أخرى!
افتراض أن لديك مصفوفة لأسماء الطلبة، وأخرى لدرجاتهم في الامتحان، وأردت أن ترتب أسماء الطلبة تبعاً لدرجاتهم في الامتحان انظر الصيغة التالية:

 الصيغة العامة

Array.Sort(المصفوفة المراد ترتيبها ,المصفوفة القائدة للترتيب)

Code Vb.Net

```
Dim Students() As String = {"Ali", "Salah", "Nadia"}
Dim Score() As Integer = {13, 15, 10}
Array.Sort(Score, Students)
For i As Integer = 0 To Score.Length - 1
    Console.WriteLine(Students(i) & " Score " & Score(i))
Next
```

Nadia Score 10
Ali Score 13
Salah Score 15

الوسيلة "عكس" Reverse:

لإرجاع مصفوفة جديدة ترتيب عناصرها عكس ترتيب المصفوفة الأصلية.

 الصيغة العامة

Array.Reverse(array[, startindex, endindex])
(عنصر النهاية , عنصر البداية , اسم المصفوفة)

Code Vb.Net

```
Dim Alfb(2) As String
Alfb(0) = "a"
Alfb(1) = "b"
Alfb(2) = "c"
Array.Reverse(Alfb)
For i = 0 To UBound(Alfb)
    ListBox1.Items.Add(Alfb(i)) ' c ,b ,a
Next
```

```
Dim Numbers() As Integer = {0, 1, 2}
Array.Reverse(Numbers)
For i As Integer = 0 To UBound(Numbers)
    Debug.Print(Numbers(i)) '2,1,0
```

Next i

دمج عناصر مصفوفة

إذا أردنا عرض كل عناصر المصفوفة في نص واحد استعمل الوسيلة Join

 الصيغة العامة

```
Dim TestJoin As String = Join("رمز", مصفوفة)
```

مثال :

Code Vb.Net

```
Dim myArray() As String = {"سلم و عليه الله صلى", "محمد"}  
Dim TestJoin As String = Join(myArray, ",")  
Debug.Print(TestJoin)
```

البحث في المصفوفة

الطريقة العامّة للبحث في المصفوفة، سواءً أكانت مرتبة أم لا، هي استخدام الوسيلة "رقم الخانة" IndexOf للبحث عن أول خانة يوجد فيها عنصر معين، أو الوسيلة "رقم آخر خانة" LastIndexOf للبحث عن آخر خانة يوجد فيها عنصر معين (وهي تبدأ البحث من نهاية المصفوفة بالطبع).. وفي حالة عدم العثور على عنصر البحث، تُرجع هاتان الوسيلتان -1.

 الصيغة العامة

```
(عصر البحث, المصفوفة) Array.IndexOf = رقم العنصر  
Dim result As Integer = Array.IndexOf(array, _  
value[, startIndex[, count]])
```

أو الصيغة التالية:

 الصيغة العامة

```
(خانة بداية البحث, عنصر البحث, المصفوفة) System.Array.IndexOf = رقم العنصر  
(عدد الخانات, خانة البداية, عنصر البحث, المصفوفة) System.Array.IndexOf = رقم العنصر
```

Code Vb.Net

```
Dim Alfb() As String = {"A", "b", "c"}  
MsgBox(Array.IndexOf(Alfb, "b"))' تطبع 1 رقم العنصر  
MsgBox(Array.LastIndexOf(Alfb, "A"))' تطبع 0 رقم العنصر
```

الفصل الخامس الفئات الأساسية

```
Dim Names() As String = {"Ali", "Salah", "Said"}
Dim FindName As String = InputBox("الاسم ادخل")
MsgBox("رقم الخانة في يوجد" & Array.IndexOf(Names, FindName))
```

وهناك طريقة خاصة فقط بالبحث في المصفوفات المرتبة، هي "البحث الثنائي" BinarySearch ولهذه الوسيلة الصيغة التالية:



الصيغة العامة

```
عنصر البحث, المصفوفة) = Array.BinarySearch(الخانة = رقم الخانة
عدد الخانات, خانة البداية, عنصر البحث, المصفوفة) = Array.BinarySearch(رقم الخانة
Dim result As Integer = Array.BinarySearch(array, value[, comparer])
or:
Dim result As Integer = Array.BinarySearch(array, index, _
length, value[, comparer])
```

Code Vb.Net

```
Dim Alfb() As String = {"a", "b", "c"}
Dim f As String
Dim h As String
h = InputBox("البحث كلمة ادخل")
f = Array.BinarySearch(Alfb, h)
If f < 0 Then
    MsgBox("العنصر غير موجود")
Else
    MsgBox("موجود" & h & " العنصر ")
End If

Dim Alfb() As String = {"a", "b", "c"}
Dim S As String = InputBox("ادخل حرف للبحث عنها")
If S = "" Then Exit Sub
Dim Found As Boolean, I As Integer
For I = 0 To Alfb.GetUpperBound(0)
    If S = Alfb(I) Then
        MsgBox("رقم السطر في توجد البحث كلمة" + Str(I + 1))
        Found = True
    End If
Next
If Found = False Then MsgBox("موجودة غير البحث كلمة")
```

الوسيلة "نسخ" Copy:

لنسخ جزء من المصفوفة إلى مصفوفة أخرى، ولها الصيغة التالية:

System.Array.Copy(عدد الخانات المنسوخة, المصفوفة المستهدفة, مصفوفة المصدر)

أو الصيغة التالية:

System.Array.Copy(خانة بداية النسخ في مصفوفة المصدر, مصفوفة المصدر)

الفصل الخامس الفئات الأساسية

(عدد الخانات، خانة بداية النسخ في المصفوفة المستهدفة، المصفوفة المستهدفة)

Code Vb.Net

```
Dim Prenom() As String = {"Ali", "Salah", "Said"}
Dim Nom() As String = {"Aksa", "Meddour", "Saada"}
Array.Copy(Prenom, Nom, 3)
For i As Integer = 0 To UBound(Prenom)
    Console.WriteLine(Nom(i))
Next
```

الناتج يكون : Ali , Salah , Said

الوسيلة "نسخ إلى" **CopyTo**:

وهي تنسخ مصفوفة بكاملها إلى مصفوفة أخرى، ولها الصيغة التالية:

Array.CopyTo(المصفوفة المستهدفة, مصفوفة المصدر)

تعريف مصفوفة من نوع كائن **Object**

Code Vb.Net

```
Dim myObjects(3) As Object
myObjects(0) = 10
myObjects(1) = False
myObjects(2) = New DateTime(1969, 3, 24)
myObjects(3) = "Text"
```

استعمال المصفوفات كمعاملات لإجراء

Code Vb.Net

```
Function Avg(ByVal Numbers() As Integer) As Double
    Dim I As Integer, Sum As Integer
    For I = 0 To Numbers.GetUpperBound(0)
        Sum += Numbers(I)
    Next
    Return Sum / Numbers.Length
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Values() As Integer = {4, 5, 6}
    Debug.Print(Avg(Values)) '5
End Sub
```

جدول لأهم دوال المصفوفات

Array Class	Represents an array
Array.BinarySearch Method	Searches for a value in a sorted one-dimensional array

	<p>بحث عن عنصر في مصفوفة بطريقة Binary عنصر ,المصفوفة)System.Array.BinarySearch = رقم خانة (البحث)</p>
Array.Copy Method	<p>Copies all or part of an array لنسخ جزء من المصفوفة إلى مصفوفة أخرى، عدد الخانات ,المصفوفة المستهدفة ,مصفوفة المصدر)Array.Copy (المنسوخة)</p>
Array.IndexOf Method	<p>Searches for the first occurrence of a value in an unsorted one-dimensional array للبحث عن أول خانة يوجد فيها عنصر معين (عنصر البحث ,المصفوفة)Array.IndexOf = رقم العنصر</p>
Array.LastIndexOf Method	<p>Searches for the last occurrence of a value in an unsorted one-dimensional array للبحث عن آخر خانة يوجد فيها عنصر معين (وهي تبدأ البحث من نهاية المصفوفة بالطبع).. (عنصر البحث ,المصفوفة)Array.LastIndexOf = رقم العنصر</p>
Array.Reverse Method	<p>Reverses the order of elements in an array dimension لإرجاع مصفوفة جديدة ترتيب عناصرها عكس ترتيب المصفوفة الأصلية Array.Reverse(مصفوفة)</p>
Array.Sort Method	<p>Sorts the elements of an array dimension ترتيب و فرز مصفوفة Array.Sort(عدد الخانات ,خانة البداية ,اسم المصفوفة)</p>
Erase Statement	<p>Resets an array to its uninitialized state لتدميرها و تفريغ محتوى الذاكرة مصفوفة Erase</p>
isArray Function	<p>Indicates whether a variable is an array إنها مصفوفة</p>
Join Function	<p>Concatenates an array of values into a delimited string دمج عناصر مصفوفة باختيار رمز يفصل بينها Dim TestJoin As String = Join (مصفوفة , "رمز")</p>
LBound Function	<p>Returns the lower boundary of an array تعيد رقم أصغر عنصر في المصفوفة (1 ,مصفوفة)LBound = رقم العنصر</p>
ReDim Statement	<p>Adjusts the bounds of an array dimension يستعمل لتغيير حجم أو طول مصفوفة</p>

الفصل الخامس الفئات الأساسية

UBound Function

Returns the upper boundary of an array

تعيد رقم أعلى عنصر في المصفوفة

UBound(مصفوفة, 1) = رقم العنصر

تدريب نموذجي :

قم بتصميم النموذج على النحو التالي :



في قسم التعريفات العامة نكتب المتغير العام و الذي يعبر عن مصفوفة كما يلي

مصفوفة الإخوة ' As String (4) strFriends Private

في حدث تحميل النموذج Form_Load نكتب التعليمات التالية

VB.Net Code

```
strFriends(0) = "Ali"  
strFriends(1) = "Said"  
strFriends(2) = "Nadia"  
strFriends(3) = "Faid"  
strFriends(4) = "Sara"
```

الفصل الخامس الفئات الأساسية

تحميل المصفوفة ببعض العناصر '

في إجراء الخاص بالعرض نكتب:

 VB.Net Code

```
Sub ShowItems()  
    For Each strName As String In strFriends  
        'Add the array item to the list  
        ListBox1.Items.Add(strName)  
    Next  
End Sub
```

و في حدث نقر زر العرض نكتب

 VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    ShowItems() ' عرض عناصر المصفوفة '  
End Sub
```

و في حدث زر الترتيب نكتب

 VB.Net Code

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button2.Click  
    ListBox1.Items.Clear() ' مسح محتوى القائمة '  
    Array.Sort(strFriends) ' ترتيب عناصر المصفوفة تصاعديا '  
    ShowItems() ' عرض عناصر المصفوفة '
```

و في حدث نقر زر عكس الترتيب نكتب

 VB.Net Code

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button2.Click  
    ListBox1.Items.Clear() ' مسح محتوى القائمة '  
    Array.Reverse(strFriends) ' عكس ترتيب عناصر المصفوفة تصاعديا '  
    ShowItems() ' عرض عناصر المصفوفة '
```

و في حدث نقر زر البحث نكتب

 VB.Net Code

```
Dim x As String = InputBox("الاسم ادخل")  
Dim N As Integer = Array.BinarySearch(strFriends, x) ' دالة البحث '  
If N > 0 Then  
    ListBox1.SelectedIndex = N ' تحديد عنصر البحث في القائمة '
```

```
ListBox1.Focus()  
Else  
    MsgBox("موجود غير الاسم")  
End If
```

عرض محتوى المصفوفة حسب الرقم

1. قم بإضافة زر و صندوق النص لعرض عناصر المصفوفة إلى نافذة التصميم
2. ثم اكتب التعليمات التالية في حدث نقر النموذج

VB.Net Code

```
Dim N As Integer  
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    Dim Names(3) As String  
    Names(0) = "Ali"  
    Names(1) = "Salah"  
    Names(2) = "Said"  
    Names(3) = "Ahmed"  
    'Dim n As Integer = CInt(TextBox1.Text)  
    N = N + 1  
    If N > 4 Then MsgBox("5 من اقل رقما اختر") : Exit Sub  
    TextBox2.Text = Names(n - 1)  
  
End Sub
```

فئة النصوص Class String

للتعامل مع النصوص يمنحك إطار العمل Net Frame Work فئة النصوص و فئة بانتي النصوص.

البيانات النصية تشتمل على كل أنواع البيانات نصية أو رقمية أو حرفية أو تاريخ و وقت ...
توضع بين علامتي تنصيص " " . يمكنك تعريفها كما يلي :

```
Dim MyString As String = "البيانات أنواع كل"
```

خصائص فئة النصوص

الطول Length :

تعيد لك عدد الحروف الموجودة في النص .

الصيغة العامة



```
Dim Value As Integer = Stringtxt.Length
```

Value = ترجع عدد الحروف أو طول النص

StringTxt = النص

 VB.Net Code

```
Dim s As String = "abcdef"
```

```
Debug.Print("هو النص الحروف عدد" & s.Length) '6
```

الحروف Chares

تعتبر هذه الخاصية مصفوفة تستقبل رقم وتعيد الحرف الذي يوجد في هذا الموضع من النص

 الصيغة العامة

```
Dim Value As Char = Stringtxt.Chares(Index)
```

Value = ترجع حرفا

StringTxt = النص

Index : رقم يمثل موضع الحرف في النص

مثال

 VB.Net Code

```
Dim s As String = "abcdef"
```

```
Debug.Print("هو النص من الثالث الموقع في يوجد الذي الحرف" & s.Chars(2)) 'c
```

مثال آخر

 VB.Net Code

```
Dim s As String = "abcdef"
```

```
For i As Integer = 0 To s.Length - 1
```

```
    Console.WriteLine(s.Chars(i)) ' الحروف كل تطبع
```

```
Next
```

وسائل فئة النصوص

مقارنة Compare

تقارن بين نصين، فإذا كان أولهما أصغر (في ترتيبه الهجائي) كان ناتجها -1، وإذا كانا متساويين

كان ناتجها صفرا، وإذا كان أولهما أكبر كان ناتجها 1

 الصيغة العامة

```
Dim Value As Integer = String.Compare(S1, S2)
```

```
Dim Value As Integer = String.Compare(S1, S2, Boolean)
```

Value = ترجع رقما

S1 = النص الأول

النص الثاني : S2

Boolean : قيمة منطقية و هذا لمرعاة حالة الأحرف

مثال

VB.Net Code

```
Dim S1 As String = "Salah"
Dim S2 As String = "Salim"
Dim resulta As Integer = String.Compare(S1, S2)
Select Case resulta
    Case -1 : Console.WriteLine("s1 < s2")
    Case 0 : Console.WriteLine("s1 = s2")
    Case 1 : Console.WriteLine("s1 > s2")
End Select
```

و هناك صيغة أخرى يمكنك من مقارنة جزء من النصين

String.Compare(طول الجزء المقارن ,بدايته ,النص الثاني ,بدايته ,النص الأول)

دمج النصوص Concat

الصيغة العامة

```
Dim Value As String = String.Concat(S1, S2)
```

النص 1 و النص 2 : S1 , S2

تقوم بدمج النصوص في نص جديد و هي مكافئة للصيغتين

Value = S1 + S2

Value = S1 & S2

و ليس هذا فقط بل يمكنك دمج عدة نصوص لأنها تقبل مصفوفة كمعامل كما في المثال

Value = **String.Concat**(S1, S2,S3,S4....)

أو تكتب :

VB.Net Code

```
Dim S() As String = {"A", "b", "c", "d", "e", "f"}
Dim Resulta As String
Resulta = String.Concat(S)
Console.WriteLine(Resulta)
```

نسخ Copy

تستقبل نصا و ترجع نسخة منه في متغير نصي آخر وهي تعادل العملية =

الصيغة العامة

```
Dim Value As String = String.Copy(S)
```

النص المنسوخ : S

Value : النسخة الجديدة

مثال

VB.Net Code

```
Dim S1 As String = "abcdef"
Dim S2 As String
S2 = String.Copy(S1) ' S2 = s1
Console.WriteLine(S2)
```

نسخ إلى Copy To

تنسخ نص و تضعه في مصفوفة حروف تحدد لها موضع النسخ
تعين رقم البداية في مصفوفة الحروف ثم متغير مصفوفة الحروف ثم الموضع الذي سيتم نسخ النص
في مصفوفة الحروف ثم عدد الحروف التي تنسخها من النص
أي كأننا نقوم بإدراج نص في وسط مجموعة حروف

الصيغة العامة

عدد حروف من مصفوفة الحروف, مصفوفة الحروف, رقم بداية CopyTo(مصفوفة المصدر
(طول النص المنسوخ)

مثال :

VB.Net Code

```
Dim strSource As String = "abcdefj"
Dim destination As Char() = {"T"c, "h"c, "e"c, " "c, "i"c, "n"c, "i"c, _
    "t"c, "i"c, "a"c, "l"c, " "c, "a"c, "r"c, "r"c, "a"c, "y"c}
strSource.CopyTo(0, destination, 4, strSource.Length)
Console.WriteLine(destination)
```

ربط ودمج Join

هذه الدالة مماثلة للوسيلة Concat تقوم بدمج النصوص و تختلف عنها بأنها تمنحك بإدراج حرف
فاصل بين النصوص مثل فاصل أو فاصلة ...

الصيغة العامة

```
Dim Value As String = String.Join(Sparator, ArrayString)
```

Sparator : حرف يمثل فاصل

ArrayString : مصفوفة نصوص

```
Dim Value As String = String.Join(Sparator,
ArrayString, StartIndex, LenArray)
```

StartIndex : خانة البداية

LenArray : عدد الخانات

مثال :

VB.Net Code

```
Dim s() As String = {"Visual", "Basic", ".Net"}
Dim Value As String = String.Join(" ", s)
```

Console.WriteLine(Value) ' Visual Basic .Net

تقسيم و تجزئة Split

هي عكس الدالة السابقة تقوم بتقسيم نص إلى أجزاء وتضعها في مصفوفة اعتمادا على حرف فاصل يفصل بين الأجزاء

الصيغة العامة 

```
Dim result( ) As String = String . Split(expression[, delimiter[, _  
limit[, compare]]])
```

نفس المثال السابق و لكن الآن نقوم بتقسيمه إلى أجزاء

 VB.Net Code

```
Dim Value As String = "Visual Basic .Net"  
Dim Splittxt As String()  
Splittxt = Value.Split(" ")  
For i As Integer = 0 To UBound(Splittxt)  
    Console.WriteLine(Splittxt(i))  
Next
```

Visual
Basic
.Net

مثال آخر لتشطير النص إلى أجزاء حسب مصفوفة من الفواصل

 VB.Net Code

```
Dim words As String = "Exmple For: String,Split."  
Dim split As String() = words.Split(New [Char]() {" "c, ","c, "."c,  
":"c})  
  
Dim s As String  
For Each s In split  
    If s.Trim() <> "" Then  
        Console.WriteLine(s)  
    End If  
Next s
```

Exmple
For
String
Split

تجزئة نص SubString

تقوم بأخذ جزء من نص وهذا بتحديد موضع بدأ الإقطاع وطوله.

الصيغة العامة 


```
Dim Resulta As String = S1.Substring(StartIndex, Length)
```

مثال :

 VB.Net Code

```
Dim S1 As String = "Visual Basic .Net"  
MsgBox(S1.Substring(7, 5)) 'Basic
```

Start With يبدأ ب

تستعمل للتأكد ما إذا كان النص يبدأ بحرف أو نص محدد.

 الصيغة العامة

```
Dim Resulta As Boolean = S.StartsWith(Value)
```

Value : النص كله , S : كلمة بداية النص

EndWith ينتهي ب

تستعمل للتأكد ما إذا كان النص ينتهي بحرف أو كلمة

 الصيغة العامة

```
Dim Resulta As Boolean = S.EndsWith(Value)
```

Value : كلمة بداية النص

S : النص

مثال

 VB.Net Code

```
Dim S As String = "Visual Basic .Net"  
Dim Value As Boolean = S.StartsWith("Visual") _  
And S.EndsWith(".Net")  
MsgBox(Value) ' True
```

Index Of البحث في موضع

تبحث عن كلمة في نص و ترجع موضع أو حرف من الكلمة في النص.

 الصيغة العامة

```
Dim Resulta As Integer = S.IndexOf(Value)
```

S : النص الذي نبحث فيه

Value : الكلمة التي نبحث عنها

مثال

 VB.Net Code

```
Dim S As String = "Visual Basic .Net"  
Dim Resulta As Integer = S.IndexOf(".Net")  
MsgBox(Resulta) ' 13
```

آخر موضع Last Index Of

وهي مثالة للوسيلة السابقة إلا أنها تبدأ البحث من آخر النص .

موضع أي منها Index Of Any

هذه الوسيلة شقيقة، فهي تمكنك من البحث عن مجموعة منفصلة من الحروف (نرسلها للدالة في مصفوفة)، بحيث تُعيد هذه الدالة موضع أول حرف تجده من هذه الحروف في النص..

الصيغة العامة

```
Dim Resulta As Integer = S.IndexOfAny(ArrayChars, StartIndex, Count)
```

S : النص

ArrayChars : مصفوفة حروف

StartIndex : تحديد موضع البدء

مثال

VB.Net Code

```
Dim chars() As Char = {".", "c", " ", "c", ";", "c"}
```

```
Dim myString As String = "This Is a short sentence"
```

```
Console.WriteLine(myString.IndexOfAny(chars)) ' الفصل تجده حرف أول
```

4 الموضع في

إدراج Insert

تمكنك هذه الوسيلة من إدراج نص في موضع محدد في نص آخر .

الصيغة العامة

```
Dim Resulta As String = S.Insert(Index, Value)
```

S: النص , Value : النص المدرج , Index : موضع الإدراج

مثال :

VB.Net Code

```
Dim S As String = "Visual .Net"
```

```
Dim Resulta As String = S.Insert(7, "Basic")
```

```
Console.WriteLine(Resulta) ' Visual Basic .Net
```

حذف Remove

يمكنك استخدام هذه الوسيلة لإزالة جزء من النص، بدءاً من موضع معين وبطول معين

الصيغة العامة

```
Dim Resulta As String = S.Remove(Index, Count)
```

S: النص , Count : طول النص المحذوف , Index : موضع بداية الحذف

مثال

VB.Net Code

```
Dim S As String = "Visual .Net"
Dim Resulta As String = S.Remove(7, 4)
Console.WriteLine(Resulta) ' Visual
```

محاذاة يميني PadRight ومحاذاة يسرى PadLeft:

يمكنك استخدام الوسيلة PadRight للحصول على نسخة من النص وقد أضيف في نهايته عدد من المسافات لإكمال طول النص إلى العدد الذي ترغبه من الحروف.. ويحدث المثل عند استخدام الوسيلة PadLeft ولكن المسافات تُضاف في بداية النص. وتأخذ هذه الوسيلة معاملاً، هو الطول الكلي الذي تريده للنص.. انظر للمثال التالي:

VB.Net Code

```
Dim S As String = "Visual .Net"
```

فئة بائي النصوص StringBuilder Class

تستعمل لبناء نصّ طويل جداً، عن طريق تشبيك أجزائه عبر جملة تكرارية و تستعمل الوسيلة الهامة وهي Append لإضافة نص إلى نهاية النص الحالي

الصيغة العامة 

```
Dim SB As New System.Text.StringBuilder
SB.Append(Str)
SB.Append(ArrayChar, StartIndex, CountIndex)
```

عدد : CountIndex , خانة البداية : StartIndex , مصفوفة حروف : ArrayChar
الخانات

VB.Net Code

```
Dim sb As New System.Text.StringBuilder
sb.Append("الحمد") ' إضافة نص
sb.AppendLine("العالمين رب") ' إضافة سطر
sb.Insert(5, " الله ") ' إدراج نص في موضع
sb.AppendLine("الرحيم الرحمن") ' إضافة سطر
Console.WriteLine(sb) ' تحرير كامل النص
```

يكون الناتج :

العالمين رب الله الحمد

الرحيم الرحمن

الفصل الخامس الفئات الأساسية

مثال : عن استعمال باني النصوص في الحلقات

VB.Net Code

```
Dim Sb As New System.Text.StringBuilder
For i As Integer = 0 To TextBox1.Lines.Count - 1
    Sb.AppendLine(i + 1 & " / " & TextBox1.Lines(i))
Next
TextBox1.Text = Sb.ToString
```

ملخص عن فئة النصوص خصائصها و وسائلها و دوالها

الخصائص

Syntax	Description	Exp
X = StrA.Chars(Value)	تعيد مصفوفة تتكون من الحروف الذي يكون منه النص	<pre>Dim x As String = "Salah" For i = 0 To x.Length - 1 Debug.Print(x.Chars(i)) Next</pre>
X = StrA.Length	تعيد عدد حروف النص	<pre>Dim x As String = "Salah" MsgBox(x.Length) '5</pre>

الوسائل

Syntax	Description	Exp
String.Compare(StrA, StrB)	مقارنة نصين تعيد 0 في حالة التساوي -1 في حالة وجوده قبل في الترتيب الهجائي و +1 العكس	<pre>MsgBox(String.Compare("Salah", "Salah")) '0 MsgBox(String.Compare("Salah", "Zin")) '-1 MsgBox(String.Compare("Zin", "Salah")) '1</pre>
String.Compare(StrA, StrB, Boolean)	إذا كان true مراعاة حالة الحروف و Fase لعدم مراعاتها	<pre>MsgBox(String.Compare("SALAH", "Salah", True)) '0 MsgBox(String.Compare("SALAH", "Salah", False)) '1</pre>
String.Compare(StrA, Start1, StrB, Start2, LengthCompare)	تسمح لك هذه الصيغة بمقارنة جزء من النص	<pre>MsgBox(String.Compare("Salah", 0, "Salim", 0, 3)) '0</pre>
	يمكنك إضافة المعامل true أو false	لمراعاة حالة الأحرف أو لا
String.Concat(StrA, StrB) String.Concat(Str1, Str2)	لتشبيك نصين	<pre>MsgBox(String.Concat("Sal", "ah")) 'Salah MsgBox(String.Concat("Sal",</pre>

الفصل الخامس الفئات الأساسية

,.....)		"ah", " Like", " Vb")) 'Salah Like Vb
StrA.Contains(Strb)	هل هذا النص محتوى في النص لآخر	MsgBox(x.Contains("ala")) 'True
String.Copy(str)	نسخ نص	Dim x As String = "Salah" MsgBox(String.Copy(x)) 'Salah
StrA.EndsWith(Value)	هل ينتهي ب	MsgBox("Salah".EndsWith("ah"))
StrA.Equals(Strb)	هل النصان متساويان	MsgBox("Salah".Equals("Salah")) 'True
String.Equals(StrA,Str B)	هل النصان متساويان	MsgBox(String.Equals("Salah", "Ali")) 'False
Format		
StrA.IndexOf(Value)	تعيد موضع أول حرف من Value أو 1- إذا لم تجده	Dim X As String = "Salah" MsgBox(X.IndexOf("lah")) '2
StrA.IndexOf(Value,Start)	كما يمكن تحديد موضع البداية	MsgBox(X.IndexOf("lah", 3)) '-1
StrA.IndexofAny (Arrayx , Start ,Count)	تعيد موضع أول حرف من مصفوفة تجده في النص المحدد و تختار موضع بدأ البحث و طول الكلمة من النص	Dim Chars() As Char = {",", " ", ";", "."} Dim StrA As String = "Salah Meddour" MsgBox(StrA.IndexOfAny(Chars,0,6)) '5
StrA.Insert(Start,Value)	إدراج نص بتحديد موضع البداية و النص المدرج	Dim X As String = "Sa" MsgBox(X.Insert(2, "lah"))
Join(Arrayx ,Separator)	دمج نصوص مصفوفة في نص واحد وبفاصل تحده	Dim x() As String = {"S", "a", "l", "a", "h"} Dim y As String = Join(x, ",") Debug.WriteLine(y)
String.Join(Separator,Arrayx)	دمج النص بكيفية أخرى	Dim x() As String = {"S", "a", "l", "a", "h"} Debug.WriteLine(String.Join(", ", x))

الفصل الخامس الفئات الأساسية

String.Join(Separator ,Arrayx ,Start,Count)	يمكنك إضافة موضع البدء و عدد الحروف	Debug.WriteLine(String.Join(";", x, 0, 5))
LastIndexOf	مماثلة للطريقة IndexOf و لكنها تبدأ من النهاية إلى البداية	
StrA.PadRight(totalWidth , paddingChar)	لإكمال نص إلى عدد محدد برموز تحده من جهة اليمين	Debug.WriteLine("Salah".PadRight(20, ".")) Salah.....
StrA.PadLeft(Totalwidth h, paddingChar)	لإكمال نص إلى عدد محدد برموز تحده من جهة اليمين	Debug.WriteLine("Salah".PadLeft(20, "."))Salah
StrA.Remove(Start,Count)	تزيل حروف من نص بتحديد موضع البداية و عدد حروف الكلمة التي تحذفها	MsgBox("Salah Meddour".Remove(6, 7)) 'Salah
StrA.Replace(OldValue , NewValue)	تستبدل كلمة مكان كلمة أخرى	Dim X As String = "Salah Aksa" MsgBox(X.Replace("Aksa", "Meddour"))
StrA.Split(Arrayx,Separator)	عكس الطريقة Join فهي تقوم بتقسيم النص حسب رمز تحده	Dim x As String = "Salah , Meddour" Dim LINE_split() As String LINE_split = Split(x, ",") For i As Integer = 0 To UBound(LINE_split) Debug.WriteLine(LINE_split(i)) Next
StrA.StartsWith(Value)	هل يبدأ ب	MsgBox("Salah".StartsWith("S"))'True
StrA.Substring (Start ,Count)	تقطع النص إلى جزء حدد بدايته و طوله	MsgBox("Salah Meddour".Substring(6, 7))'Meddour
Arrayx = StrA.ToCharArray	تنسخ نص إلى مصفوفة تقطع النص إلى حروف	Dim x As String = "Salah" Dim y() As Char = x.ToCharArray For i As Integer = 0 To UBound(y)

الفصل الخامس الفئات الأساسية

		Debug.WriteLine(y(i)) Next
StrA.ToLower	تحول النص إلى حروف صغيرة	MsgBox("Salah".ToLower)' salah
StrA.ToUpper	تحول النص إلى حروف كبيرة	sgBox("Salah".ToUpper) 'SALAH
StrA.Trim	تمحو المسافات من الجهتين	MsgBox(" Salah ".ToUpper)

دوال النصوص

Element	Description	
Asc (ChrA)	تعيد الرقم Ascii لأول حرف من النص	MsgBox(Asc("A"))'65
Chr (NumberA)	تعيد الحرف المقابل للرقم Ascii	MsgBox(Chr(65)) 'A
Dim result As String() = Filter(source, match[, include[, compare]])	ترسل لها مصفوفة نصية تحتوي على كلمة البحث و True ثم نوع المقارنة وتعيد لك مصفوفة تشمل على كلمة البحث إذا كتب False فتعيد لك الكلمات المخالفة لنص البحث	Dim X() = {"الله عبد", "عبد", "الصالح", "الرحمن"} Dim Y() As String = Filter(X, "عبد", True, CompareMethod.Text) For I As Integer = 0 To Ubound(Y) Debug.WriteLine(Y(i)) Next
Format Function	Format(expression[, style])	سنعرض لها لاحقا
FormatCurrency , FormatNumber, FormatPercent	تنسق العدد إلى عملة نسق العدد إلى رقم نسق العدد إلى نسبة مئوية	MsgBox(FormatCurrency(12 000))'12 000,00 €MsgBox(FormatNumber(1 200, 2)) '1 200,00 MsgBox (FormatPercent(50)) '5 000,00%
Dim result As Char = GetChar(str, index)	اذهب إلى الحرف المحدد بموضعه في النص	MsgBox(GetChar("Salah", 4)) 'h
InStr([start,	تعيد موقع أول حرف من النص	Dim X As String =

[string1, string2[, compare])	الذي تبحث عنه String1 النص String2 كلمة البحث Compare نوع المقارنة	"ABCDEFD" MsgBox(InStr(X, "D")) '4 MsgBox(InStrRev(X, "D")) '7
InStrRev(stringCheck, stringMatch[, _start[, compare]])	نفس الدالة السابقة و لكنها بدأ البحث من نهاية النص	
Join(sourceArray, [delimiter])	مماثلة للوسيلة Join تستقبل مصفوفة من النصوص و فاصل ثم نعيد نصا يضم كل الكلمات و يفصل بينهما بالفاصل المحدد	Dim X() As String = {"S", "a", "l", "a", "h"} MsgBox(Join(X, ",")) 'S,a,l,a,h
LCase (Value)	حول النص إلى حروف صغيرة	MsgBox(LCase("AbcD"))'abcd
Left(str, length)	تقطع جزءا من النص من اليسار بقدر عدد الحروف المحددة	MsgBox(Microsoft.VisualBasic.Left("ABCD", 2))'AB
Len	حسب عدد حروف النص	MsgBox(Len("ABCD")) '4
string Like pattern	يقارن نص بصيغة # للأرقام و ? للحروف	MsgBox("10.00" Like "###.###")'True
LSet(source, length)	قطع حروف من نص من اليسار بقدر عدد الحروف المحددة	MsgBox(LSet("ABCD", 3))'ABC
Ltrim(Str)	محو المسافات من اليسار	MsgBox(LTrim(" ABCD ")) "ABCD "
Mid(target, start[, length]) = expression	تقطع جزءا من نص من أي موضع وسطي	MsgBox(Mid("ABCDEF", 2, 3)) 'BCD
Replace(expression, find, replacement _[, start[, count[, compare]])	تستقبل نصا ثم الكلمة التي يبحث عنها و يستبدلها ب كلمة أخرى و يعيد نصا جديدا معدلا	MsgBox(Replace("Salah Aksa", "Aksa", "Meddour"))

الفصل الخامس الفئات الأساسية

Right(str, length)	تقتطع جزءاً من النص من اليمين	MsgBox(Microsoft.VisualBasic.Right("ABCDEF", 3)) 'DEF
RSet(source, length)	تقتطع جزءاً من النص من اليسار	MsgBox(RSet("ABCDEF", 3)) 'ABC
RTrim(Str)	محو المسافات من اليمين	MsgBox(RTrim(" ABCD ")) "' ABCD"
Str(number)	تحول الأرقام إلى نصوص	Str(5) ' "5"
SPC(count)		
Space Function	ترك فاصل بقدر	MsgBox("ABCD" & Space(3) & "EF")'ABCD EF
Split(expression[, delimiter[, _limit[, compare]])]	تشطير النص حسب فاصل حدده و تعيد مصفوفة	Dim x As String = "ABCD/EF" Dim y() As String = Split(x, "/") For i As Integer = 0 To UBound(y) Debug.WriteLine(y(i)) Next
StrComp(string1, string2[, compare])]	تعيد ناتج مقارنة نصين 0 في حالة التساوي -1 لا يساوي و يسبقه في الترتيب 1 لا يساوي و بعده في الترتيب	MsgBox(StrComp("ABC", "ABC")) '0 MsgBox(StrComp("ABC", "Abc")) '-1 MsgBox(StrComp("BBC", "ABC")) '1
StrConv(str, conversion[, localeID])]	تقوم بتحويل النص إلى حالة أخرى تحددها حسب نوع التحويل	MsgBox(StrConv("Abcd", VbStrConv.Lowercase)) 'abcd
StrDup(number, character)	تكرر الحرف حسب العدد المحدد	MsgBox(StrDup(5, "=")) '=====
StrReverse(expression)	تعكس حروف النص	MsgBox(StrReverse("ABCD")) 'DVBA
Trim(str)	تمحو المسافات من الجهتين	Trim (" ABCD ") 'ABCD
UCase (Str)	تحول الحروف إلى حروف كبيرة	MsgBox(ucase("Abcd")) 'ABCD
Val (Str)	تحول النص إلى رقم	MsgBox(Val("123")) '123

تطبيقات عن النصوص

الشريط السريع

1. أنشئ مشروعاً جديداً
2. ضف أداة المؤقت إلى النموذج و غير خصائصه كما يلي
Interval = 75 و Enabled = True
3. غير خصائص النموذج RightToLeft = Yes و Right To Left = True
4. اكتب التعليمات التالية

 VB.Net Code

```
Public Class Form1

    Private txt As String
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        txt = "الله رسول محمد أن و الله إلا إله لا أن أشهد" & Space(50) & txt
        Timer1.Enabled = True
        Timer1.Interval = 75
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        txt = Mid(txt, 2) & LSet(txt, 1)
        Me.Text = txt
    End Sub
End Class
```

عند تنفيذ البرنامج يظهر الشريط السريع على عنوان النموذج.

كتابة نص حرف حرف

1. أنشئ مشروعاً جديداً
2. ضف أداة المؤقت إلى النموذج و غير خصائصه كما يلي
Interval = 75 و Enabled = True
3. اكتب التعليمات التالية

 VB.Net Code

```
Private txt As String = "Visual Basic .Net"
Dim i As Integer
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    i = i + 1
    If i = Len(txt) Then Exit Sub
    Me.Text = LSet(txt, i)
End Sub
```

تشفير النصوص

VB.Net Code

```
Dim st As String
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    For i As Integer = 1 To Len(TextBox1.Text)
        'النص صندوق في النص حروف كل عبر المرور'
        Dim st1 As String = Mid(TextBox1.Text, i, 1)
        'جدول حسب أرقام إلى الحروف تحويل'
        Dim as1 As Integer = Asc(st1)
        'النص تشفير'
        Dim ch1 As String = Chr(255 - as1)
        st = st + ch1
    Next
    TextBox1.Text = st
End Sub
```

فئة التاريخ و الوقت Date And Time

للتعامل مع التاريخ و الوقت يمنحك VB.Net الخلية DateAndTime. أما Date فهو نوع من أنواع البيانات فحسب وفي كلتا الحالتين، سواء عرّفت المتغير من هذا أم من ذلك، فأنت تستخدم خواصّ ووسائل خلية التاريخ والوقت. و لتعريف متغير من نوع الزمن نكتب :

Code Vb.Net

```
Dim date1 As Date = #5/15/2014#
Dim date2 As Date = #5/15/2014 10:30:45 AM#
Dim Date3 As Date = "May 15 2004"
```

خصائصها :

DateString : لكتابة التاريخ اليوم قصير

 الصيغة العامة

```
Dim result As String = DateAndTime.DateString
```

or:

```
DateAndTime.DateString = newDate
```

newDate (required in second syntax; String)

VB.Net Code

```
MsgBox("The current date is " & DateString)
```

```
MsgBox("The current date is " & DateAndTime.DateString)
```

لاحظ أنه يمكننا أن نستغني عن كتابة DateAndTime لأن خصائصها مشتركة .
Now : تعود بتاريخ اليوم الحالي و الوقت الحالي

VB.Net Code

```
MsgBox("الحالي الوقت و التاريخ" & Now)
```

09/03/2012 17:12:18 الحالي الوقت و التاريخ

TimeOfDay : تعود بالوقت الحالي

VB.Net Code

```
Debug.Print("الحالي الوقت " & DateAndTime.TimeOfDay)
```

17:13:56 الحالي الوقت

Timer :تعود بعدد الثواني المنقضية من منتصف الليل.

VB.Net Code

```
Debug.Print(" الليل منتصف من المنقضي الثواني " & DateAndTime.Timer) '  
62479,65625
```

TimeString :تعود بالوقت الحالي

VB.Net Code

```
Debug.Print("الحالي الوقت " & TimeString)'17:24:10
```

Today : تعود بتاريخ اليوم الحالي بالتنسيق الانجليزي

VB.Net Code

```
Debug.Print(" بالانجليزي اليوم تاريخ " & Today) ' 20/03/2012
```

دوال التاريخ و الوقت

إضافة تاريخ **DateAdd** : دالة تعود بنتائج جمع تاريخين .

 الصيغة العامة

```
Dim result As Date = DateAdd (interval, number, dateValue)
```

(التاريخ, القيمة المضافة, المدى الزمني) = DateAdd (التاريخ الجديد

حيث المدى الزمنيّ هو أحد قيم المرقّم DateInterval التالية:

الفصل الخامس الفئات الأساسية

الحرف	المدى	الشرح
Y	Year	عام.
Q	Quarter	فصل (ربيع عام: 1 - 4 شهور).
M	Month	شهر.
Y	DayOfYear	يوم من السنة (1 - 266 يوما).
D	Day	يوم من الشهر (1 - 31 يوما).
W	WeekDay	يوم من الأسبوع (1 - 7 أيام).
ww	WeekOfYear	أسبوع من العام (1 - 53 أسبوعا).
H	Hour	ساعة.
N	Minute	دقيقة.
s	Second	ثانية.

VB.Net Code

أضافة 120 يوما للتاريخ '

```
Debug.Print(DateAdd(DateInterval.Day, 120, #3/3/2005#)) '
01/07/2005
```

إضافة 3 سنوات للتاريخ '

```
Debug.Print(DateAdd(DateInterval.Year, 3, #3/3/2005#))'
03/03/2008
```

إضافة 15 شهرا للتاريخ '

```
Debug.Print(DateAdd(DateInterval.Month, 15, #3/3/2005#))
'03/06/2006
```

لاحظ أن VB.Net يقوم بالتحويل اللازمة عند الضرورة.

فرق زمني **DateDiff**: دالة تعود بنتائج فرق تاريخين .

 الصيغة العامة

```
Dim result As Long = DateDiff(interval, date1, date2[,dayOfWeek[,
weekOfYear]])
```

_ التاريخ الثاني, التاريخ الأول, المدى الزمني) **DateDiff** = الفارق الزمني

([أول أسبوع في السنة, أول يوم في الأسبوع,]

حيث المدى الزمني هو أحد قيم المرقم DateInterval و التي تعرفنا عليها سابقا .

VB.Net Code

```
Dim dateBirth As Date = InputBox("ميلادك تاريخ ادخل")
Debug.Print(DateDiff(DateInterval.Year, dateBirth, Today, _
    FirstDayOfWeek.Saturday, FirstWeekOfYear.Jan1))
```

المثال السابق يرجع عمرك بعد أن تحدد تاريخ ميلادك و أول يوم في الأسبوع السبت و أول أسبوع في السنة الأسبوع الأول من جانفي .

📌 **ملاحظة** : إذا كان التاريخ الأول أحدث من التاريخ الثاني يكون الناتج سالب .

DatePart : جزء من التاريخ

 الصيغة العامة

```
DatePart(interval, dateValue[,
    firstDayOfWeekValue[,firstWeekOfYearValue]])
```

```
DatePart([أول أسابيع السنة],[أول أيام الأسبوع],[التاريخ,الجزء المطلوب])
```

وهي تعيد جزءا من التاريخ، على حسب قيمة المعامل الأول، وهو بالمناسبة معامل نصي.. انظر للأمثلة التالية:

String	DateInterval	Description
d	Day	The day of the month, from 1 to 31
y	DayOfYear	The day of the year, from 1 to 366
h	Hour	The hour of the day, from 0 to 23
n	Minute	The minute of the hour, from 0 to 59
m	Month	The month of the year, from 1 to 12
q	Quarter	The quarter of the year, from 1 to 4
s	Second	The second of the minute, from 0 to 59
w	Weekday	The day of the week, from 1 to 7; the value 1 is normally used for Sunday, but it may differ based on the <i>firstDayOfWeekValue</i> parameter
ww	WeekOfYear	The week number of the year, from 1 to 53; the <i>firstWeekOfYearValue</i> parameter impacts this value
yyyy	Year	The year of the Gregorian calendar, from 1 AD to 9999 AD

أما المعامل الأخير أول أسابيع السنة يأخذ القيم التالية

Value	Description
System	Uses the system-defined value
Jan1	Uses the week in which January 1 appears
FirstFourDays	Uses the first week of the year that has at least four days in it
FirstFullWeek	Uses the first week of the year that has a full seven days in it

VB.Net Code

```

Console.WriteLine(DatePart("yyyy", Now())) 'العام 2012
Console.WriteLine(DatePart("q", Now())) 'العام في الفصل رقم 1
Console.WriteLine(DatePart("m", Now())) 'العام في الشهر رقم 3
Console.WriteLine(DatePart("y", Now())) 'العام في اليوم رقم 69
Console.WriteLine(DatePart("d", Now())) 'الشهر في اليوم رقم 9
Console.WriteLine(DatePart("w", Now())) 'الأسبوع في اليوم رقم 6
Console.WriteLine(DatePart("ww", Now())) 'السنة في الأسبوع رقم 10
Console.WriteLine(DatePart("h", Now())) 'الساعة 18
Console.WriteLine(DatePart("n", Now())) 'الدقيقة 37
Console.WriteLine(DatePart("s", Now())) 'الثانية 30
    
```

تكوين التاريخ DateSerial:

تستقبل هذه الدالة السنة والشهر واليوم، وتعيد لك التاريخ الذي يتكوّن منها.. مثال



```
Dim result As Date = DateSerial(year, month, day)
```

 التاريخ الجديد = DateSerial(السنة , الشهر , اليوم)

VB.Net Code

```
MsgBox(DateSerial(2012, 10, 1)) ' 1/10/2012
```

قيمة التاريخ DateValue:

تستقبل هذه الدالة نصًا، وتعيد التاريخ الذي يمثّله هذا النصّ.

VB.Net Code

```
Debug.Print(DateValue("February 12, 2012")) '12/02/2012
```

وللحصول على أيّ جزء من أجزاء التاريخ، استخدم الخواصّ التالية:

العام **Year** – الشهر **Month** – اليوم **Day** – الساعة **Hour** – الدقيقة **Minute** – الثانية
Second – الجزء من الألف الثانية **Millisecond** – الجزء من مئة ألف من الثانية **Ticks**.

VB.Net Code

```

Dim MyDate As Date = #1/22/2012#
Debug.Print(DateAndTime.Day(MyDate)) '22
Debug.Print(Month(MyDate)) '1
Debug.Print(Year(MyDate)) '2012
    
```

اسم الشهر MonthName

دالة تعود باسم الشهر و تستقبل رقم الشهر

VB.Net Code

```
Debug.Print(DateAndTime.MonthName(5, True)) ' Mai
```

تكوين الوقت TimeSerial:

تستقبل هذه الدالة ثلاثة معاملات: الساعة والدقيقة والثانية، وتعيد الوقت الذي يتكوّن منها.

VB.Net Code

```
Debug.Print(DateAndTime.TimeSerial(15, 30, 20)) '15:30:20
```

قيمة الوقت TimeValue

تستقبل قيمة نصية و تعيد الوقت الذي يمثل النص

VB.Net Code

```
Debug.Print(DateAndTime.TimeValue("15:40")) '15:40:00
```

Weekday

دالة تستقبل التاريخ و تعيد رقم اليوم بالنسبة لأيام الأسبوع مع تحديد أول يوم في الأسبوع



```
Dim result As Integer = Weekday(dateValue[, dayOfWeek])
```

(أول يوم في الأسبوع , قيمة التاريخ) = Weekday رقم

VB.Net Code

```
Debug.Print(DateAndTime.Weekday(Today, _  
FirstDayOfWeek.Saturday)) '7
```

WeekdayName

دالة تستقبل رقم اليوم و تعيد اسمه

VB.Net Code

```
Debug.Print(WeekdayName(4, True)) 'Jeu
```

وسائل التعامل مع التاريخ

للتعامل مع هذه الوسائل نستعين بالخلية DateTime

Compare

تقوم بمقارنة تاريخين وفي حالة التساوي ترجع 0 و في حالة الاختلاف ترجع -1 .

VB.Net Code


```
Debug.Print(DateTime.Compare(#1/1/1995#, #1/2/1995#)) ' -1
```

عدد أيام الشهر **DaysInMonth**:

تخبرك هذه الدالة بعدد أيام الشهر الذي تريده.. ونظرا لأنّ شهر فبراير يكون 28 يوما في السنة البسيطة و 29 يوما في السنة الكبيسة، فإنّ من الطبيعي أنّ تحتاج هذه الدالة لإرسال السنة إليها كمعامل، بجانب إرسال اسم الشهر:



```
monDays = DateTime.DaysInMonth(السنة, رقم الشهر)
```

VB.Net Code

```
Debug.Print(DateTime.DaysInMonth(2012, 2)) '29
```

IsLeapYear

دالة تستقبل رقم السنة وتعيد ما إذا كانت السنة كبيسة أو لا .

VB.Net Code

```
Debug.Print(DateTime.IsLeapYear(2012)) 'True
```

الوقت المنقضي **TimeSpan**

تمتلك هذه الخلية من حساب الوقت المنقضي عن الوقت الحالي للجهاز بأي وحدة تريد . فهي تستقبل المعاملات الأيام و الساعات و الدقائق و الثواني

VB.Net Code

```
Dim TS As New TimeSpan  
Dim date1 As Date = #4/11/1985#  
Dim date2 As Date = Now()  
TS = date2.Subtract(date1)  
Console.WriteLine(TS)'5992.15:58:14.4766848
```

تعرض عدد الأيام ثم الوقت

والمثال التالي يريك كيف تستخدم هذه الخلية لحساب الوقت المنقضي

VB.Net Code

```
Dim T1, T2 As Date  
T1 = Now ' البداية وقت  
MsgBox("انتظرته الذي الوقت لمعرفة موافق اضغط ثم لفترة، انتظر")  
T2 = Now  
Dim TS As TimeSpan = T2.Subtract(T1)
```

```

Console.WriteLine("الكليّة المدّة = " & TS.Duration.ToString)
Console.WriteLine("الدقائق = " & TS.Minutes.ToString)
Console.WriteLine("الثواني = " & TS.Seconds.ToString)
Console.WriteLine("الثانية من الألف من الأجزاء = " & _
TS.TotalMilliseconds.ToString)
Console.WriteLine("الثانية من ألف مئة من الأجزاء = " & TS.Ticks.ToString)
Console.WriteLine("بالثواني الزمنية المدّة = " &
TS.TotalSeconds.ToString)

```

```

-----
الكليّة المدّة = 00:00:05.0312500
الدقائق = 0
الثواني = 5
الثانية من الألف من الأجزاء = 5031,25
الثانية من ألف مئة من الأجزاء = 50312500
بالثواني الزمنية المدّة = 5,03125

```

إضافة Add



(المدّة الثانية) add . المدّة = 1 = مدّة زمنية جديدة

VB.Net Code

```

Dim T1, T2 As TimeSpan
T1 = New TimeSpan(1, 12, 20, 0)
T2 = New TimeSpan(2, 12, 40, 0, 0)
Dim TS As TimeSpan = T2.Add(T1)
Debug.Print(TS.ToString) ' 4.01:00:00

```

طرح Subtract:

عكس العملية السابقة، حيث تُرجع مدّة زمنيّة جديدة، ناتجة عن طرح مدّة زمنيّة معيّنة من المدّة الزمنيّة الحالية.. مثال:

VB.Net Code

```

Dim T1, T2 As TimeSpan
T1 = New TimeSpan(1, 12, 20, 0)
T2 = New TimeSpan(2, 12, 40, 0, 0)
Dim TS As TimeSpan = T2.Subtract(T1)
Debug.Print(TS.ToString) ' 1.00:20:00

```

تنسيق التاريخ و الوقت

كتابة تاريخ قصير أو طويل

```
VB.Net Code
MsgBox(FormatDateTime(Today, DateFormat.ShortDate))
MsgBox(FormatDateTime(Today, DateFormat.LongDate))
```

كتابة وقت قصير أو طويل

```
VB.Net Code
MsgBox(FormatDateTime(Now, DateFormat.ShortTime))
MsgBox(FormatDateTime(Now, DateFormat.LongTime))
```

و الجدول التالي يبين لنا الرموز المستعملة لتنسيق البيانات الزمنية

	Description
General Date or G	يعرض التاريخ بصورة عامة تاريخ قصير و وقت طويل
Long Date or Medium Date or D	يعرض تاريخ طويل
Short Date or d	يعرض تاريخ قصير
Long Time or Medium Time or T	يعرض وقت طويل
Short Time or t	يعرض وقت قصير
f	يعرض تاريخ طويل و وقت قصير
F	يعرض تاريخ طويل و وقت طويل
g	تاريخ قصير و وقت قصير
M or m	تاريخ طويل و اليوم "August 23"
R or r	Date and time, adjusted to Greenwich Mean Time, and in the sample format "Tue, 23 Aug 2005 17:33:11 GMT"
s	Formats the date and time in a format that allows for easy sorting: "2005-08-23T17:33:11"
u	Adjusts the date and time to Greenwich Mean Time and then formats the result for easy sorting: "2005-08-23 17:33:11Z"
U	Adjusts the date and time to Greenwich Mean Time, and then formats the result using the locale-specific long date and long time format
Y or y	Full month name, a comma, and then the year, as in "August, 2005"

أمثلة

 Code Vb.Net

```
Dim d As Date = #1/1/2014#
Console.WriteLine(d.ToString("d"))
Console.WriteLine(d.ToString("D"))
Console.WriteLine(d.ToString("f"))
Console.WriteLine(d.ToString("s"))
Console.WriteLine(d.ToString("y"))
```

و يكون الناتج

```
01/01/2014
mercredi 1 janvier 2014
mercredi 1 janvier 2014 00:00
2014-01-01T00:00:00
janvier 2014
```

Code Vb.Net

```
Dim VarDate As Date = #1/15/2000 12:30:10 AM#
Console.WriteLine(Format(VarDate, "G"))
Console.WriteLine(Format(VarDate, "D"))
Console.WriteLine(Format(VarDate, "d"))
Console.WriteLine(Format(VarDate, "T"))
Console.WriteLine(Format(VarDate, "t"))
Console.WriteLine(Format(VarDate, "f"))
Console.WriteLine(Format(VarDate, "F"))
Console.WriteLine(Format(VarDate, "g"))
Console.WriteLine(Format(VarDate, "M"))
Console.WriteLine(Format(VarDate, "y"))
```

يكون الناتج

```
15/01/2000 00:30:10
samedi 15 janvier 2000
15/01/2000
00:30:10
00:30
samedi 15 janvier 2000 00:30
samedi 15 janvier 2000 00:30:10
15/01/2000 00:30
15 janvier
janvier 2000
```

وهناك رموز أخرى لتنسيق الزمن و هي حسب الجدول التالي :

code	Description
d	التاريخ قصير
dd	اليوم بالأرقام
ddd	اليوم بالحروف مختصر
dddd	اليوم كامل بالحروف
M	اليوم بالأرقام و الشهر كامل بالحروف

الفصل الخامس الفئات الأساسية

MM	الشهر بالأرقام
MMM	الشهر مختصرا بالحروف
MMMM	الشهر كاملا بالحروف
h	Hour of the day with no leading zero, using a 12-hour clock.
hh	Two-digit hour of the day, with a leading zero if needed, using a 12-hour clock.
H	Hour of the day with no leading zero, using a 24-hour clock.
HH	Two-digit hour of the day, with a leading zero if needed, using a 24-hour clock.
mm	الدقائق
ss	الثواني
f	Fractions of a second. Use multiple "f" characters for more digits. For example, "s.fff" displays thousandths of a second. Uppercase or lowercase "f" will work.
t	الساعة و الدقائق
tt	Displays "AM" for prenoon times or "PM" for noon and beyond. الساعة و الوقت
y	الشهر و السنة
yy	Displays the last two digits of the year, adding a leading zero if needed.
yyy	Same as "yyyy."
yyyy	السنة بالأرقام

أمثلة :

Code Vb.Net

```
Dim VarDate As Date = #1/15/2000 12:30:10 PM#
Console.WriteLine(Format(VarDate, "d"))
Console.WriteLine(Format(VarDate, "dd"))
Console.WriteLine(Format(VarDate, "ddd"))
Console.WriteLine(Format(VarDate, "dddd"))
Console.WriteLine(Format(VarDate, "M"))
Console.WriteLine(Format(VarDate, "MM"))
Console.WriteLine(Format(VarDate, "MMM"))
Console.WriteLine(Format(VarDate, "MMMM"))
Console.WriteLine(Format(VarDate, "mm"))
Console.WriteLine(Format(VarDate, "t"))
Console.WriteLine(Format(VarDate, "tt"))
Console.WriteLine(Format(VarDate, "y"))
Console.WriteLine(Format(VarDate, "yy"))
Console.WriteLine(Format(VarDate, "yyy"))
Console.WriteLine(Format(VarDate, "yyyy"))
```

يكون الناتج

15/01/2000
15
sam.
samedi
15 janvier
01
janv.
janvier
30
12:30

janvier 2000
00
2000
2000

جزء من التاريخ DatePart:

لهذه الدالة الصيغة التالية:

DatePart([أول أسابيع السنة, [أول أيام الأسبوع, التاريخ, الجزء المطلوب])
وهي تعيد جزءاً من التاريخ، على حسب قيمة المعامل الأول، وهو بالمناسبة معامل نصّي..
لقد تطرقنا إلى الصيغة الأولى مع الخصائص و الآن ندرس الصيغة الأخرى
حيث إنّ معاملها الأول من نوع المرقّم DateInterval

VB.Net Code

```
Dim MyDate As Date = #12/5/2011#  
Debug.Print(DatePart(DateInterval.Year, MyDate)) '2011  
Debug.Print(DatePart(DateInterval.Day, MyDate)) '5  
Debug.Print(DatePart(DateInterval.Month, MyDate)) '12  
Debug.Print(DatePart(DateInterval.Weekday, MyDate)) 'من اليوم رقم  
2 هو الاسبوع  
Debug.Print(DatePart(DateInterval.WeekOfYear, MyDate))  
Debug.Print(DatePart(DateInterval.Quarter, MyDate)) '4 الفصل
```

تطبيقات

كتابة اليوم بالعربي

VB.Net Code

```
Select Case DateAndTime.Weekday((Today),  
FirstDayOfWeek.Saturday)  
Case 1 : Label1.Text = "السبت"  
Case 2 : Label1.Text = "الأحد"  
Case 3 : Label1.Text = "الاثنين"  
Case 4 : Label1.Text = "الثلاثاء"
```

```
Case 5 : Label1.Text = "الأربعاء"  
Case 6 : Label1.Text = "الخميس"  
Case 7 : Label1.Text = "الجمعة"  
End Select
```

كتابة الشهر بالعربي

VB.Net Code

```
Select Case (DateAndTime.Month(TextBox1.Text))  
Case 1 : Label1.Text = "جانفي"  
Case 2 : Label1.Text = "فيفري"  
Case 3 : Label1.Text = "مارس"  
Case 4 : Label1.Text = "أفريل"  
Case 5 : Label1.Text = "ماي"  
Case 6 : Label1.Text = "جوان"  
Case 7 : Label1.Text = "جويلية"  
Case 8 : Label1.Text = "أوت"  
Case 9 : Label1.Text = "سبتمبر"  
Case 10 : Label1.Text = "أكتوبر"  
Case 11 : Label1.Text = "نوفمبر"  
Case 12 : Label1.Text = "ديسمبر"  
  
End Select
```

كتابة التاريخ كاملا بالعربي

VB.Net Code

```
Private Function SHorof(ByVal X)  
Dim Letter1, Letter2, Letter3, Letter4 As String  
Dim N As Integer = Int(X)  
Dim C As String = Format(N, "0000")  
Dim C1 As Integer = Val(Mid(C, 4, 1))  
Select Case C1  
Case Is = 1 : Letter1 = "واحد"  
Case Is = 2 : Letter1 = "اثنان"
```

Case Is = 3 : Letter1 = "ثلاثة"

Case Is = 4 : Letter1 = "أربعة"

Case Is = 5 : Letter1 = "خمسة"

Case Is = 6 : Letter1 = "ستة"

Case Is = 7 : Letter1 = "سبعة"

Case Is = 8 : Letter1 = "ثمانية"

Case Is = 9 : Letter1 = "تسعة"

End Select

Dim C2 As Integer = Val(Mid(C, 3, 1))

Select Case C2

Case Is = 1 : Letter2 = "عشرة"

Case Is = 2 : Letter2 = "عشرون"

Case Is = 3 : Letter2 = "ثلاثون"

Case Is = 4 : Letter2 = "أربعون"

Case Is = 5 : Letter2 = "خمسون"

Case Is = 6 : Letter2 = "ستون"

Case Is = 7 : Letter2 = "سبعون"

Case Is = 8 : Letter2 = "ثمانون"

Case Is = 9 : Letter2 = "تسعون"

End Select

If Letter1 <> "" And C2 > 1 Then Letter2 = Letter1 + " و " + Letter2

If Letter2 = "" Then Letter2 = Letter1

If C1 = 0 And C2 = 1 Then Letter2 = Letter2 + "ة"

If C1 = 1 And C2 = 1 Then Letter2 = "عشرة إحدى"

If C1 = 2 And C2 = 1 Then Letter2 = "عشرة إثنًا"

If C1 > 2 And C2 = 1 Then Letter2 = Letter1 + " " + Letter2

Dim C3 As Integer = Val(Mid(C, 2, 1))

Select Case C3


```

Case Is = 1 : Letter3 = "مئة"
Case Is = 2 : Letter3 = "مئتان"
Case Is > 2 : Letter3 = LSet(SHorof(C3), Len(SHorof(C3)) - 1) +
"مائة"
End Select
If Letter3 <> "" And Letter2 <> "" Then Letter3 = Letter3 + "و" +
Letter2
If Letter3 = "" Then Letter3 = Letter2

Dim C4 As Integer = Val(Mid(C, 1, 1))
Select Case C4
Case Is = 1 : Letter4 = "ألف"
Case Is = 2 : Letter4 = "ألفان"
Case 3 To 10 : Letter4 = SHorof(C4) + "آلاف"
Case Is > 10 : Letter4 = SHorof(C4) + "ألف"
End Select
If Letter4 <> "" And Letter3 <> "" Then Letter4 = Letter4 + "و" +
Letter3
If Letter4 = "" Then Letter4 = Letter3
SHorof = Letter4
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
Select Case (DateAndTime.Month(TextBox1.Text))
Case 1 : Label1.Text = "جانفي"
Case 2 : Label1.Text = "فيفري"
Case 3 : Label1.Text = "مارس"
Case 4 : Label1.Text = "أفريل"
Case 5 : Label1.Text = "ماي"
Case 6 : Label1.Text = "جوان"
Case 7 : Label1.Text = "جويلية"

```

Case 8 : Label1.Text = "أوت"

Case 9 : Label1.Text = "سبتمبر"

Case 10 : Label1.Text = "أكتوبر"

Case 11 : Label1.Text = "نوفمبر"

Case 12 : Label1.Text = "ديسمبر"

End Select

Label1.Text = SHorof(DateAndTime.Day(TextBox1.Text)) + " " +
Label1.Text + " " + (SHorof(Year(TextBox1.Text)))

End Sub

المعاملات Operators

يمنحك Vb.Net أنواع مختلفة من المعاملات .

المعاملات الرياضية Arithmetic Operators

المعامل	Operator	المثال	النتائج
^	Exponentiation	3^2	8
-	Negation	2-	2-
*	Multiplication	3*2	6
/	Division	3/2	1.5
/	Integer division	2/3	3
/	Modulus	5 \ 17	2
\	Addition	Mod 5 17	5
Mod	Subtraction	3+2	1
+	Bit left shift	2-3	01101110
-	Bit right shift	>> 10110111	01011011
>>		1	
<<		<< 10110111	
		1	

معاملات المقارنة Comparison Operators

المعامل	Operator	الشرح	المثال
=	Equals	يساوي	5 = 5
<>	Not equals	يختلف	5 <> 2
<	Less than	أكبر من	5 > 2
>	Less than or equal to	أكبر من	5 >= 2
=>	Greater than	أكبر أو يساوي	2 < 5
<=>	Greater than or equal to	أصغر من	

الفصل الخامس الفئات الأساسية

2 <=5 A is B (متغيرات) A is Not B Typeof(obj) Is textbox A Like A	أصغر من إنه ليس نوع من مثل	Equality of two objects Inequality of two objects Object is of a certain type Matches a text pattern	=< Is IsNot TypeOf...Is Like
--	--	---	--

المعاملات الخاصة ب Like

أمثلة	الدور	الرمز
A* تعرض كل الأسماء التي تبدأ ب A A* تعرض كل الأسماء التي تنتهي ب A *Ali* تعرض كل الأسماء التي تشمل على الحروف في أي موضع كانت	تحل محل مجموعة من الحروف	*
A?i تعرض مثلاً Sai ، Ami ، Ali	تحل محل حرف واحد	?
AL[i,y] تعرض Ali ، ALy	حرف واحد من مدى معين	[]
F# تعرض F1 ، F2 ،	تحل محل رقم واحد	#
*A[!q] تعرض كل الأسماء التي تنتهي بحرفين A و أي حرف آخر	تستثني الحرف	[!]

المعاملات المنطقية

مثال	الشرح	المعامل
Not true 'False True and True ' True True Or False ' True False Xor False ' True True And True ' True	النفى و أو عملية استثناء منطقية	Not And Or Xor AndAlso
	تتحقق من صحة القضية 1 فإذا كانت صحيحة تنتقل لاختبار القضية 2	OrElse

الدوال الرياضية

تعريفها و استخدامها

تستخدم هذه الخلية على الصيغة التالية:



(المعاملات) اسم الدالة. Math. = الناتج
Value = Math . Function (Parameters)

VB.Net Code

```
Dim X As Integer = 10
Dim Y As Integer = 5
MsgBox(Math.Max(X, Y)) ' 10 الرقم الرسالة ستعرض
```

وهذا الجدول يوضح لنا معظم الدوال الرياضية

أمثلة :

مثال	عملها	الدالة
MsgBox(Math.PI)	الثابت PI	PI
MsgBox(Math.Abs(-3)) ' 3	القيمة المطلقة لعدد	ABS
MsgBox(Math.Ceiling(105.0589)) ' 106	تلغي القيمة العشرية و تضيف 1 الى الجزء الصحيح	Ceiling
MsgBox(Math.Floor(1.915))	تلغي الجزء العشري و تبقي الجزء الصحيح كما كان	Floor
MsgBox(Math.Pow(2, 3))	حساب الأس	Pow
MsgBox(Math.Sqrt(36))	الجزر التربيعي	Sqrt
MsgBox(Math.Sign(-2)) ' سالب MsgBox(Math.Sign(2)) ' موجب MsgBox(Math.Sign(0)) ' صفر	تعيد إشارة العدد و تفيد في حساب المميز	Sign
MsgBox(Math.Log(1, 10))	حساب لوغاريتم عدد في أساس	Log
MsgBox(Math.Round(2.912)) ' تقرب 3 فتعطي العدد MsgBox(Math.Round(2.912, 1)) ' تظهر 2.9 العشري الجزء من واحد رقما	تقرب تحدد عدد أرقام الجزء العشري التي ستظهر	Round
MsgBox(Math.Min(7, 5)) ' 5	اختيار أصغر عدد	Min
MsgBox(Math.Max(7, 5)) ' 7	اختيار أكبر عدد	Max
MsgBox(Math.IEEERemainder(13,	حساب باقي القسمة	IEEERemainder

الفصل الخامس الفئات الأساسية

2)) MsgBox(13 Mod 2)		
MsgBox(Math.Sin(45))	حساب جيب زاوية	Sin
MsgBox(Math.Cos(45))	حساب تجب زاوية	Cos
MsgBox(Math.Tan(45))	تظل زاوية	Tan
MsgBox(Rnd() * 2.5)	توليد أرقام عشرية	Rnd

الفصل السادس : الملفات و المجلدات

- فئة المجلدات
- فئة الملفات
- فئة معلومات المجلد
- فئة المسار
- التعامل مع ملفات Stream
- تطبيقات
- أنواع الملفات
- الملفات و الدالة Open
- تطبيقات
- الملفات FileSystem
- تطبيقات

الملفات والمجلدات Files and Directorys

عند التعامل مع الملفات و المجلدات نسترد مجال الأسماء System . Io و يحتوى على مجموعة من الفئات يمكنك من إدارة و تحرير الملفات و مجلدات الجهاز كالنسخ و النقل و الحذف

```
Imports System.IO
Public Class Form1
End Class
```

فئة المجلدات Directory

تمتلك الفئة Directory مجموعة من الوسائل المشتركة يمكنك استعمالها مباشرة من الخلية Directory . Property / Methods

الوسائل

إنشاء مجلد Create Directory

Directory.CreateDirectory(Path)

Path : المسار

Exp

Directory.CreateDirectory("C:\folder1\folder2\folder3")

حذف مجلد Delete Directory

Directory.Delete(Path) ' حذف مجلد إن كان موجود و بدون مجلدات فرعية '

Directory.Delete(Path, True) ' حذف المجلد مع مجلداته الفرعية '

Directory.Delete("C:\Folder1", True)

نقل مجلد Move Directory

تستعمل لنقل مجلد بكل محتوياته إلى مجلد آخر في نفس القرص

Directory.Move(sourceDirName, destDirName)

sourceDirName :: مسار المجلد المصدر : destDirName : مسار المجلد المنقول الجديد

Exp

Directory.Move("c:\folder1", "c:\Test")

كما يمكنك من إعادة تسمية المجلد

مجلد موجود Exists Directory

تستقبل هذه الوسيلة المسار فإذا كان موجود تعيد True

Directory.Exists(Path) ' ترجع صواب في حالة وجود الملف '

If Directory.Exists("C:\folder1") = True Then

MsgBox("موجود الملف")

Else

MsgBox("موجود غير الملف")

End If

اقرأ المجلدات GetDirectories

تعيد مصفوفة نصية تحتوي على كل أسماء المجلدات الموجودة في المجلد المطلوب دون المجلدات الفرعية أي تحصل على مجلدات المستوى الأول فقط

```
Dim result( ) As String = Directory.GetDirectories(path, _
    searchPattern[, searchOption])
```

SearchPattern : صيغة البحث

SearchOption : خيارات البحث

المعامل الثاني (صيغة البحث) يمكن أن يحتوى على الحروف كما في الجدول

الحرف	الدلالة	مثال
*	يدل على جميع	C:/P ' عرض كل المجلدات التي تنتهي ب ' P '*
?	يدل على حرف واحد مهما كان	C:\Progr?m

Exp

```
Dim Dir() As String
Dim counter As Integer
Dir = Directory.GetDirectories("c:\", "P*")
For counter = 0 To UBound(Dir)
    Console.WriteLine(Dir(counter))
Next counter
```

اقرأ الملفات GetFiles

تعيد مصفوفة نصية لجميع أسماء الملفات الموجودة في المجلد المرسل

```
Dim result( ) As String = Directory.GetFiles
(path, _
    searchPattern[, searchOption])
```

Exp :

```
Dim Files() As String = Directory.GetFiles("C:\", "*.txt") ' كل الملفات النصية
For i As Integer = 0 To UBound(Files)
    Console.WriteLine(Files(i))
Next
```

اقرأ عناصر المجلد GetFileSystemEntries

هذه الوسيلة هي دمج للوسيلتين السابقتين و تعيد مصفوفة لجميع أسماء المجلدات و الملفات الموجود في المسار المحدد

```
Dim result( ) As String = Directory.GetFileSystemEntries(path, _
```


searchPattern])

و لنميز بين المجلدات و الملفات نستعمل الوسيلة Exists الخاصة بالمجلدات انظر المثال :

```
Dim AllFiles() As String = Directory.GetFileSystemEntries _  
("C:\Program Files")  
For Each items As String In AllFiles  
If Directory.Exists(items) Then  
Console.WriteLine(" المجلد " & items)  
Else  
Console.WriteLine(" الملف " & items)  
End If  
Next
```

اقرأ المجلد الحالي GetCurrentDirectory

تعيد مسار المجلد الحالي

```
Dim Dir As String = Directory.GetCurrentDirectory()
```

اقرأ الجذر الرئيسي (محرك الأقراص) GetDirectoryRoot

تعيد لك محرك الأقراص المستعمل حالياً في مسارك

و ناتج هذه الدالة هو نص مثل : C:\ أو D:\

```
Dim result As String = Directory.GetDirectoryRoot(path)
```

اقرأ المجلد الأب GetParent

تعيد كائن معلومات المجلد للمجلد المحدد

```
Dim parent As DirectoryInfo = Directory.GetParent(مسار المجلد)
```

Exp

```
Dim parent As DirectoryInfo = Directory.GetParent("c:/test/Folder")
```

```
Debug.Print(parent.ToString) 'Output C:\test
```

اقرأ محركات الأقراص

تعيد مصفوفة نصية لجميع أسماء محركات الأقراص المحلية الموجودة في جهازك

```
Dim result( ) As String = Directory.GetLogicalDrives
```

Exp :

```
Dim Drivers() As String = Directory.GetLogicalDrives
```

```
For Each items As String In Drivers
```

```
Console.WriteLine(items)
```

```
Next
```

ملاحظة : يفضل استعمال الوسيلة Exists للتأكد من وجود المجلد قبل القيام بالعمليات السابقة

```
If Directory.Exists(Path) Then
```

```
End If
```

أو استعمال طريقة صيد الأخطاء و هذا تجنب للوقوع في الاستثناءات .

قراءة تاريخ و وقت إنشاء مجلد `GetCreationTime`

قراءة تاريخ و وقت آخر استعمال `GetLastAccessTime`

قراءة تاريخ و وقت آخر عملية كتابة `GetLastWriteTime`

كل هذه الوسائل تستقبل مسار مجلد و تعيد بيانات تدل عن التاريخ انظر المثال

Exp :

```
Console.WriteLine(Directory.GetCreationTime("C:\Test"))
Console.WriteLine(Directory.GetLastAccessTime("C:\Test"))
Console.WriteLine(Directory.GetLastWriteTime("C:\Test"))
```

يمكنك تعديل القيم السابقة باستعمال الوسائل :

تعديل تاريخ و وقت إنشاء المجلد `SetCreationTime`

تعديل تاريخ و وقت آخر استعمال `SetLastAccessTime`

تعديل تاريخ و وقت آخر عملية كتابة `SetLastWriteTime`

كل هذه الوسائل تستقبل مسار المجلد و تاريخ التعديل و تعيد بيانات تدل عن تاريخ انظر المثال

```
Directory.SetCreationTime("C:/Folder", #1/25/2010#)
```

فئة الملف `Class File`

تحتوي على مجموعة من الوسائل المشتركة يمكن الوصول إليها بتعريف نسخة من الفئة مباشرة و كما أريد أن أذكر بأن هناك طرق موجودة في الفئة السابقة `Directory` مدعومة في أيضا في فئة `File` بنفس الطريقة .

إنشاء ملف `Create File`

تقوم بإنشاء ملف جديد و هذا بتحديد مساره و تعيد كائن `Stream` يمكننا من القراءة و الكتابة في ملف

```
Dim FS As FileStream = File.Create(Path)
```

Exp :

```
Dim Fs As FileStream = File.Create("C:\TestFile.txt")
```

إنشاء نص `Create Text`

مماثلة للوسيلة السابقة إلا أنها تنشأ ملفا نصيا و ترجع كائن من نوع `Stream Write`

```
Dim SW As StreamWriter = File.CreateText("C:\Test.txt")
```

للكتابة في الملفات باستعمال فئة `Stream` نستعمل :

`Write` : للكتابة في سطر دون فواصل

`WriteLine` : للكتابة في سطر جديد

Exp :

```
Dim SW As StreamWriter = File.CreateText("C:\Test.txt")
```

```
SW.Write("العالمين رب الله الحمد")
```

```
SW.WriteLine("الرحيم الرحمن")
SW.Close()
```

إضافة نص Append Text

تمتلك من إضافة نصوص إلى آخر ملف . فإذا كان الملف غير موجود تقوم بإنشائه و تعيد كائن من نوع Stream Write

```
Dim SW As StreamWriter = File.AppendText(Path)
```

نضيف التعليمات التالية للكود السابق

```
Dim SW As StreamWriter = File.CreateText("C:\Test.txt")
SW.WriteLine("العالمين رب الله الحمد")
SW.WriteLine("الرحيم الرحمن")
SW.Close()
'الملف جديدة نصوص إضافة'
SW = File.AppendText("C:\Test.txt")
SW.WriteLine("الدين يوم ملك")
SW.Close()
```

إضافة كل النص AppendAllText

تضيف نص في آخر ملف دفعة واحدة و تستقبل مسار الملف و النص الذي تضيف و ترجع File.AppendAllText(Path, contents)

Contents : محتوى النص المضاف

```
File.AppendAllText("C:/Test.txt", TextBox1.Text)
```

فتح الملفات Open File

تستعمل لفتح ملف للقراءة أو الكتابة و ترجع كائن من نوع File Stream

```
Dim Fs As FileStream = File.Open(Path, mode, access, share)
```

المعامل الثاني طور الفتح Mode يمثل أعضاء المرقم كما يلي :

للإضافة إلى نهاية الملف مع المحافظة على المحتويات السابقة.	Append
تنشئ ملفا جديدا، ولو كان هناك ملف بنفس الاسم يتم استبداله.	إنشاء Create
تنشئ ملفا جديدا، فإذا كان هناك ملف بنفس الاسم ينطلق استثناء ولا يُنشأ الملف.	إنشاء ملف جديد CreateNew
لفتح ملف موجود.	فتح Open
لفتح ملف موجود، فإن لم يكن موجودا يتم إنشاؤه.	فتح أو إنشاء OpenOrCreate
تفتح ملفا موجودا وتمحو محتوياته.	تقليص Truncate

الفصل السادس المجلدات و الملفات

أما المعامل الثالث Access طور التعامل مع الملف و يمثل أعضاء المرقم التالي

الملفّ مفتوح للقراءة فقط.. لهذا فإنّ محاولة تغيير محتويات الملفّ تؤدي لإطلاق استثناء.	Read
الملفّ مفتوح للقراءة والكتابة معا.	ReadWrite
الملفّ مفتوح للكتابة فقط، دون أن تستطيع قراءة محتوياته.	Write

أما المعامل الثالث Share طور المشاركة يمثل أعضاء المرقم التالي

الملفّ غير قابل للاشتراك بين أكثر من تطبيق..	None
عندما تفتح الملفّ، يمكن للتطبيقات الأخرى أن تفتحه للقراءة فقط، دون أن تمتلك القدرة على تغيير محتوياته.	Read
يمكن للتطبيقات الأخرى أن تفتح الملفّ للقراءة أو الكتابة.	ReadWrite
يمكن للتطبيقات الأخرى أن تفتح الملفّ الذي تفتحه حاليا، للكتابة فيه دون القراءة منه!!	Write

مثال

```
Dim Fs As FileStream = File.Open("C:/test.txt",
FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.ReadWrite)
```

قراءة الملف '

```
Dim Sr As New StreamReader(Fs)
```

```
TextBox1.Text = Sr.ReadToEnd
```

فتح للقراءة Open Read

هذه الوسيلة تفتح ملفا موجودا للقراءة فقط فهي مكافئة للوسيلة Open في طور القراءة فقط

```
Dim Fs As FileStream = File.OpenRead(Path)
```

مثال

للقراءة الملف فتح '

```
Dim Fs As FileStream = File.OpenRead("C:\Test.txt")
```

الملف قراءة '

```
Dim Sr As New StreamReader(Fs)
```

```
TextBox1.Text = Sr.ReadToEnd
```

* فتح نص OpenText

تفتح ملفا نصا و تعيد كائن من نوع Stream Reader ليقوم بالتحويلات اللازمة حتى تتمكن من قراءة الملف .

```
Dim Sr As StreamReader = File.OpenText(Path)
```

مثال :

```
Dim Sr As StreamReader = File.OpenText("C:\Test.txt")
```

الملف قراءة'

```
TextBox1.Text = Sr.ReadToEnd
```

فتح للكتابة Open Write

هذه الوسيلة تفتح ملفا موجودا للكتابة فقط و ترجع كائنا من نوع Stream Write

```
Dim Fs As FileStream = File.OpenWrite("C:\test.txt")
```

* **ReadAllText** اقرأ كل النص

تقرأ محتوى الملف كله و تغلقه

```
Dim TxtFile As String = File.ReadAllText(Path)
```

مثال

```
TextBox1.Text = File.ReadAllText("C:\Test.txt")
```

ReadAllLines اقرأ كل الأسطر

تقرأ محتوى الملف كله سطرا سطرا و تعيد مصفوفة نصية

مثال :

للقراءة الملف فتح'

```
Dim ArrTxt() As String = File.ReadAllLines("C:\Test.txt")
```

2 السطر تقرأ'

```
Console.WriteLine(ArrTxt(1))
```

الملف كل قراءة'

```
For i As Integer = 0 To UBound(ArrTxt)
```

```
    TextBox1.Text += ArrTxt(i) & vbCrLf
```

```
Next
```

نسخ ملف Copy

```
File.Copy (SourceFileName ,DestFileName)
```

SourceFileName : ملف المصدر , DestFileName : الملف الجديد

نقل ملف Move

```
File.Move (SourceFileName ,DestFileName)
```

استبدال ملف Replace

```
File.Replace (SourceFileName ,DestFileName)
```

حذف ملف Delete

```
File.Delete(Path)
```

ملف موجود Exists

الفصل السادس المجلدات و الملفات

التحقق من وجود ملف و تعيد True في حالة وجوده

```
Dim Value As Boolean = File.Exists(Path)
```

```
If File.Exists("C:\TestFile.txt") Then
```

```
    MsgBox("موجود الملف")
```

```
End If
```

وهناك وسائل أخرى خاصة بقراءة تاريخ و وقت إنشاء و آخر كتابة و آخر استعمال و كذا تعديل هذه التواريخ مماثلة تماما لوسائل المجلد Directory فلا داعي لإعادتها .

اقرأ السمات Get Attributes

تعيد كائن من نوع FileAttributes يحتوي على كل سمات الملف حسب الجدول التالي

الملفّ سجّل.. معظم ملفات النظام تمتلك هذه السمة.	Archive
الملفّ مضغوط.	Compressed
الملفّ مشفّر.	Encrypted
الملفّ مخفي.	Hidden
الملفّ ليس مرقيماً بواسطة خدمة ترقيم محتويات النظام system's content indexing service.	NotContentIndexed
الملفّ غير متّصل عبر الشبكة	Offline
الملفّ للقراءة فقط، ولا يمكن تغيير محتوياته.	ReadOnly
الملفّ ضئيل المحتوى.	SparseFile
الملفّ من ملفات النظام.	System
الملفّ مؤقت	Temporary
الملفّ عاديّ، لا يمتلك أيّ سمة ممّا سبق.	Normal

مثال

```
MsgBox(File.GetAttributes("C:/test.txt").ToString)
```

تغيير سمات ملف SetAttributes

```
File.SetAttributes(Path, FileAttributes)
```

Exp

```
File.SetAttributes("C:/test.txt", FileAttributes.ReadOnly +  
FileAttributes.Hidden)
```

جعلنا الملف للقراءة فقط و مخفي

خلية معلومات المجلد Directory Info

هذه الخلية تمنحك معلومات عن مجلد بعينه، لهذا لا بدّ أن تعرّف نسخة من هذه الخلية وتربطها بمجلد ما أولاً، حتّى يمكنك أن تستخدم خصائصها ووسائلها.
ولإنشاء نسخة جديدة من كائن معلومات المجلد استخدم الصيغة التالية:

```
Dim Di As New DirectoryInfo(Path)
```

الوسائل

معظم الوسائل ممتثلة لوسائل فئة المجلد Directory لهذا نكتفي بذكر الوسائل الغير مذكورة و

الخاصة بخلية معلومات المجلد Directory Info

إنشاء مجلد فرعي CreateSubdirectory

تنشأ مجلداً أو مجموعة من المجلدات للمجلد المختار .

```
Dim Dir As New DirectoryInfo("C:\Test")
```

```
Dir.CreateSubdirectory("Folder1")
```

```
Dir.CreateSubdirectory("Folder2")
```

المسار FullName

تعيد المسار الكامل لمجلد المعلومات

```
Dim Di As New DirectoryInfo("C:/test")
```

```
Console.WriteLine(Di.FullName) 'C:\Test
```

اقرأ معلومات نظام الملفات

تعيد مصفوفة نصية لجميع أسماء المجلدات و الملفات الموجودة في النسخة المحددة .

```
Dim Dir As New DirectoryInfo("C:\Test")
```

```
Dim Items() As FileSystemInfo = Dir.GetFileSystemInfos()
```

```
For i As Integer = 0 To UBound(Items)
```

```
Console.WriteLine(Items(i))
```

```
Next
```

فئة معلومات الملف File Info

كثير من وسائل هذه الخلية ممتثلة لوسائل خلية الملف File فلا داعي لإعادة ذكرها . نتطرق للوسائل و الخصائص الغير مذكورة فقط .

هذه الخلية تتعامل مع الملف بعينه فوسائلها ليست مشتركة و حتى تتمكن من استعمالها لابد من تعريف متغير لإنشاء نسخة من الفئة كما يلي :

```
Dim FInfo As New FileInfo(FileName)
```

FileName : مسار الملف كامل

```
Dim FInfo As New FileInfo("C:\Test.txt")
```

الخصائص

```
Dim FInfo As New FileInfo("C:\Test.txt")
```

```
Console.WriteLine(FInfo.Directory) ' قراءة المجلدات التي يوجد فيها الملف
```

```
Console.WriteLine(FInfo.DirectoryName) ' قراءة المجلدات التي يوجد فيها الملف '
Console.WriteLine(FInfo.Extension) ' .Txt الامتداد '
Console.WriteLine(FInfo.Length) ' الملف طول '
```

الوسائل

نسخ إلى Copy To

```
Dim FInfo As New FileInfo(Path)
FInfo.CopyTo(DestFileName)
```

DestFileName : مسار الملف الجديد

نقل إلى Move To

```
Dim FInfo As New FileInfo(Path)
FInfo.MoveTo(DestFileName)
```

خلفية المسار Path

وسائلها مشتركة يمكن استعمالها من الخلية مباشرة . يمكنك من التعامل مع المسار و التعرف على محرك الأقراص و الملف و الامتداد

قراءة الرموز الخاصة بالمسار

```
Console.WriteLine(Path.AltDirectorySeparatorChar) ' => /
Console.WriteLine(Path.DirectorySeparatorChar) ' => \
Console.WriteLine(Path.PathSeparator) ' => ;
Console.WriteLine(Path.VolumeSeparatorChar) ' => :
```

قراءة مكونات المسار من المسار كاملا و اسم الملف و امتداده

```
Dim file As String = "C:\MyApp\Bin\MyApp.exe"
```

قراءة المسار كاملا

```
Console.WriteLine(Path.GetFullPath(file))
```

قراءة المجلدات فقط

```
Console.WriteLine(Path.GetDirectoryName(file)) ' => C:\MyApp\Bin
```

قراءة اسم الملف بامتداده

```
Console.WriteLine(Path.GetFileName(file)) ' => MyApp.exe
```

قراءة امتداد الملف فقط

```
Console.WriteLine(Path.GetExtension(file)) ' => .exe
```

قراءة اسم الملف دون امتداد

```
Console.WriteLine(Path.GetFileNameWithoutExtension(File)) ' => MyApp
```

قراءة محرك الأقراص

```
Console.WriteLine(Path.GetPathRoot(File)) ' => C:\
```

هل له امتداد


```
Console.WriteLine(Path.HasExtension(File)) ' => True
```

هل له جذر

```
Console.WriteLine(Path.IsPathRooted(File)) ' => True
```

ChangeExtension تغيير الامتداد

```
Dim NewPath As String = Path.ChangeExtension(Path, Extension)
```

Exp

```
Dim NewPath As String = Path.ChangeExtension("C:/Test.txt", ".dat")
```

```
Console.WriteLine(NewPath) 'C:/Test.dat
```

Combine دمج

تدمج مسار المجلدات مع اسم الملف .

```
Path.Combine(Path1, Path2)
```

Exp

```
Console.WriteLine(Path.Combine("C:\MyApp", "MyApp.Dat")) ' =>
```

```
C:\MyApp\MyApp.Dat
```

التعامل مع الكائن Stream

يمكنك هذا الكائن التعامل مع الملفات و القيام بعمليات القراءة و الكتابة فهو شامل للكائنات :
FileStream و StreamReader و StreamWriter، حيث ستجد أنه يجمع معظم خصائصها
ووسائلها

أنواع الملفات

في إطار العمل هناك نوعان من الملفات :

الملفات النصية Text File و الملفات الثنائية Binary File التي تقوم بتخزين كل البيانات
كالصور و الأرقام

كائن بيانات الملف FileStream

يعتبر حلقة وصل بينك و بين الملفات رغم أنه لا يمكنك القراءة و الكتابة بمفرده .

عند إنشاء نسخة منه نستعمل نفس معاملات الوسيلة Open الخاصة بالملفات (راجع الدرس)

```
Dim FS As New FileStream(Path, Mode, Access, Share)
```

```
Dim FS As New FileStream(الملف، مسار الملف، مشارَكة، طور التعامل مع الملف، طور فتح الملف، مسار الملف، (الملف))
```

الخصائص:

يمكن قراءته CanRead – يمكن الانتقال لموضع فيه CanSeek – يمكن الكتابة فيه CanWrite
– طول الملف (حجمه) Length – الموضع الحالي (الموضع الذي تقرأ منه أو تكتب فيه في
الملف) Position.

الوسائل

الخروج Close ، إمكانية القراءة CanRead ، إمكانية الكتابة CanWrite ، إمكانية البحث
CanSeek

حجم الملف Length ، الموضع Position ، محو Flush ، قراءة Read ، قراءة وحدة ReadByte
كتابة write ، كتابة وحدة WriteByte ، بحث Seek

كاتب البيانات النصية Stream Write

يمكنك من الكتابة في الملفات النصية ولإنشائه هناك عدة صيغ :

```
Dim SW As New StreamWriter(Fs) ' إنشاءه من FileStream
```

```
Dim SW As New StreamWriter(Path, Append, Encoding, BufferSize)
```

```
Dim SW As New StreamWriter(المسار, استبدال, ترميز, حجم المخزن)
```

Exp

```
Dim SW As New StreamWriter("C:\Test.txt", True,  
System.Text.Encoding.Default)
```

الخصائص

كتابة Write

SW.Write(Value)

تستعمل هذه الوسيلة للكتابة في الملفات النصية تضيف نصا في نهاية الملف ملتحما مع النص السابق بدون فاصل.

كتابة سطر جديد Write Line

مماثلة للوسيلة السابقة إلا أنها تضيف النص في سطر جديد.

الخروج Close

محو Fluch

مثال : كيف نكتب أنواع البيانات في ملف

```
Dim SW As New StreamWriter("C:\Test1.txt", True,  
System.Text.Encoding.Default)
```

```
SW.Write("الرقم")
```

```
SW.WriteLine(1)
```

```
SW.Write("الميلاد تاريخ")
```

```
SW.Write(DateAndTime.DateSerial(2000, 5, 15))
```

```
SW.WriteLine() ' فارغ سطر
```

```
SW.WriteLine(New RectangleF(1, 1, 100, 200))
```

```
SW.WriteLine(Color.Yellow)
```

```
SW.Close()
```

الناتج

لرقم 1

تاريخ الميلاد 00:00:00 2000/05/15

```
}X=1,Y=1,Width=100,Height=200{  
Color [Yellow]
```

قارئ البيانات النصية Stream Reader

يمكنك من قراءة بيانات نصية من ملف و لتعريفه له صيغ مماثلة للكائن Stream Write

```
Dim Sr As New StreamReader(Fs) ' إنشاءه من FileStream
```

```
Dim Sr As New StreamReader (Path,Append, Encoding, BufferSize)
```

```
Dim Sr As New StreamReader (حجم المخزن بترميز, استبدال, المسار)
```

الخصائص

القمة Peek

تمتلك هذه الوسيلة من قراءة السطر التالي دون أن تغير من موضعها . فإذا كان لا يوجد أي حرف أو نص تال تعيد -1

```
Dim Sr As New StreamReader("C:\test.txt")
```

```
Sr.Peek() ' تقرأ السطر الأول
```

```
Console.WriteLine(Sr.ReadLine)
```

```
Sr.Peek() ' تقرأ السطر التالي
```

```
Console.WriteLine(Sr.ReadLine)
```

قراءة Read

تمتلك هذه الوسيلة من قراءة السطر التالي و الانتقال إلى موضع الحالي بزيادة واحد . فإذا كان لا يوجد سطر أو نص تال تعيد -1. و تأخذ الصيغة التالية :

```
Dim Value As Integer = Sr.Read
```

```
Dim Sr As New StreamReader("C:\test.txt")
```

```
Do While Sr.Peek <> -1
```

```
Console.Write(Convert.ToChar(Sr.Read()))
```

```
Loop
```

و هناك صيغة أخرى تعيد عددا محدد من الحروف و تأخذ الصيغة التالية :

عدد الحروف ,رقم خانة البداية في مصفوفة الحروف ,مصفوفة حروف)
charsRead = Read (المطلوب قراءتها)

```
Dim Sr As New StreamReader("C:\test.txt")
```

```
Do While Sr.Peek() >= 0
```

```
Dim c(35) As Char
```

```
Sr.Read(c, 0, c.Length)
```

```
Console.WriteLine(c) ' تعيد 35 حرفا من بداية الملف
```

```
Loop
```

```
Sr.Close()
```

قراءة مقطع ReadBlock

مماثلة للوسيلة السابقة

قراءة سطر ReadLine

تقرأ سطرا من ملف . فإذا كان الموضع الحالي هو نهاية الملف فإنها ترجع القيمة Null

```
Dim TxtFile As String = Sr.ReadLine
```

يمكنك قراءة كل سطور الملف كما يلي :

```
Dim Sr As New StreamReader("C:\test.txt")
Do While Sr.Peek <> -1
    Console.WriteLine(Sr.ReadLine)
Loop
Sr.Close()
```

اقرأ لنهاية الملف ReadToEnd

تقرأ من الموضع الحالي إلى نهاية الملف . فإذا كان في بداية الملف فإنها تقرأ كل الملف.

```
Dim TxtFile As String = Sr.ReadToEnd
```

Exp

```
Dim Sr As New StreamReader("C:\test.txt")
Console.WriteLine(Sr.ReadToEnd)
```

كاتب البيانات الثنائية Binary Writer

يمكنك هذا الكائن من قراءة بيانات ثنائية من ملف . و لتعريفه استعمل نستعمل الصيغ

```
Dim Br As New BinaryWriter(Fs,Encoding) ' FileStream إنشاؤه من
```

الكتابة Write

استعمل الوسيلة Write للكتابة إلى الملف و تحتوي على صيغ كثيرة حتى تتمكن من إضافة مختلف البيانات

```
Dim Fs As New FileStream("C:/test.bin", FileMode.OpenOrCreate)
Dim bw As New BinaryWriter(Fs)
Dim aDouble As Double = 1234.67
Dim anInt As Integer = 34567
Dim aString As String = "A, B, C"
BW.Write(aDouble)
BW.Write(anInt)
BW.Write(aString)
BW.Close()
```

وطبعا أن تألف هذه الوسائل:

إغلاق Close – محو Flush – انتقال Seek – كتابة Write.

قارئ البيانات الثنائية Binary Reader

يمكنك هذا الكائن من قراءة الملفات الثنائية و لتعريفه نكتب

```
Dim Br As New BinaryReader(Fs, Encoding)
```

وسائل القراءة كثيرة لقراءة مختلف البيانات منها

الفصل السادس المجلدات و الملفات

لراءة متغير منطقي.	ReadBoolean
لراءة وحدة واحدة.	ReadByte
لراءة مجموعة من الوحدات في مصفوفة: (عدد الوحدات ,مصفوفة وحدات) ReadBytes	ReadBytes
لراءة حرف، سواء أكان مخزنا في وحدة أم اثنتين، تبعا لنوع الترميز.	ReadChar
لراءة مجموعة من الحروف في مصفوفة: (عدد الحروف ,مصفوفة حروف) ReadChars	
لراءة عدد عشري.	ReadDecimal
لراءة عدد مزدوج.	ReadDouble
لراءة عدد صحيح قصير Short (وحدتان).	ReadInt16
لراءة عدد صحيح Integer (4 وحدات).	ReadInt32
لراءة عدد صحيح طويل Long (8 وحدات).	ReadInt64
لراءة وحدة ذات إشارة Signed Byte.	ReadSByte
لراءة عدد مفرد (4 وحدات).	ReadSingle
لراءة نص.	ReadString
لراءة عدد صحيح قصير بدون إشارة Unsigned Short Integer (وحدتان).	ReadUInt16
لراءة عدد صحيح بدون إشارة Unsigned Integer (4 وحدات)	ReadUInt32

نقرأ بيانات الملف التي أضفناها سابقا

Exp

```
Dim Br As New BinaryReader(Fs)
Console.WriteLine(br.ReadDouble())
Console.WriteLine(br.ReadInt32())
Console.WriteLine(br.ReadString())
```

تطبيقات File Stream

أولا : مع مربعات الحوار

استدعاء مجال الأسماء الخاص بالملفات ' Imports System.IO

Public Class Form1

زر حفظ

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
SaveFileDialog1.Filter = "All txt (*.txt)|*.txt| All File(*.*)|*.*"
SaveFileDialog1.ShowDialog()
If SaveFileDialog1.Filter = "" Then Exit Sub
Dim SW As New StreamWriter(SaveFileDialog1.FileName)
```

```
SW.WriteLine(TextBox1.Text)
SW.Close()
```

```
End Sub
```

زر فتح أو عرض

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    OpenFileDialog1.Filter = "All txt (*.txt)|*.txt| All File (*.*)|*.*"
    OpenFileDialog1.ShowDialog()
    If OpenFileDialog1.FileName = "" Then Exit Sub
    TextBox1.Text = File.ReadAllText(OpenFileDialog1.FileName)
End Sub
End Class
```

كيفية أخرى

```
Imports System.IO
Public Class Form1
```

زر حفظ

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    SaveFileDialog1.Filter = "All txt (*.txt) |*.txt| All File (*.*) |*.*"
    If (SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK)
Then
    Dim Sw As StreamWriter =
File.CreateText(SaveFileDialog1.FileName)
    Sw.WriteLine(TextBox1.Text)
    Sw.Close()
    End If
End Sub
```

زر فتح

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    OpenFileDialog1.Filter = "All txt (*.txt) |*.txt| All File (*.*) |*.*"
    If OpenFileDialog1.FileName = "" Then Exit Sub
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK
Then
    Dim Sr As StreamReader =
File.OpenText(OpenFileDialog1.FileName)
    TextBox1.Text = Sr.ReadToEnd
    Sr.Close()
    End If
End Sub
End Class
```

ثانيا : مع مربعات الحوار و النص الغني

Imports System.IO
Public Class Form1

' حفظ ملف '

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    SaveFileDialog1.Filter = "All txt (*.txt)|*.txt| Richtxt (*.rtf)|*.rtf| All File(*.*)|*.*"
    If SaveFileDialog1.Filter = "" Then Exit Sub

    If (SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
        Select Case SaveFileDialog1.FilterIndex
            Case 1
                RichTextBox1.SaveFile(SaveFileDialog1.FileName, RichTextBoxStreamType.PlainText)
            Case Else
                RichTextBox1.SaveFile(SaveFileDialog1.FileName, RichTextBoxStreamType.RichText)

        End Select
    End If
End Sub
```

' فتح ملف '

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    OpenFileDialog1.Filter = "All txt (*.txt)|*.txt| Richtxt (*.rtf)|*.rtf| All File(*.*)|*.*"

    If OpenFileDialog1.FileName = "" Then Exit Sub
    If (OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
        Select Case OpenFileDialog1.FilterIndex
            Case 1
                RichTextBox1.LoadFile(OpenFileDialog1.FileName, RichTextBoxStreamType.PlainText)
            Case Else
                RichTextBox1.LoadFile(OpenFileDialog1.FileName, RichTextBoxStreamType.RichText)

        End Select
    End If
End Sub

End Class
```

المثال 3



```
Imports System.IO
Public Class Form1
    Public x As Integer
```

' زر إضافة البيانات

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Fs As New FileStream("C:/Test11.txt", FileMode.Append)
    Dim SW As New StreamWriter(Fs)
    SW.WriteLine(TextBox1.Text)
    SW.WriteLine(TextBox2.Text)
    SW.Close()
    Fs.Close()
    TextBox1.Clear()
    TextBox2.Clear()
    TextBox1.Focus()
End Sub
```

' زر التالي

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim Fs As New FileStream("C:/Test11.txt", FileMode.Open)
    Dim Sr As New StreamReader(Fs)
    x = x + 1
    For i As Integer = 1 To x
        TextBox1.Text = Sr.ReadLine
        TextBox2.Text = Sr.ReadLine
    Next
    Sr.Close()
    Fs.Close()
End Sub
```

' زر الخلف

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```



```

Dim Fs As New FileStream("C:/Test11.txt", FileMode.Open)
Dim Sr As New StreamReader(Fs)
x = x - 1
If x = 0 Then MsgBox("سجل أول") : Exit Sub
For i As Integer = 1 To x
    TextBox1.Text = Sr.ReadLine
    TextBox2.Text = Sr.ReadLine
Next
Sr.Close()
Fs.Close()
End Sub

```

' زر البحث

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim Fs As New FileStream("C:/Test11.txt", FileMode.Open)
    Dim Sr As New StreamReader(Fs)
    Dim m As String = InputBox("z")
    Dim f As Boolean, N As Integer, P As String
    Do While Sr.Peek <> -1
        N = Sr.ReadLine
        P = Sr.ReadLine
        If P = m Then
            f = True
            Exit Do
        End If
    Loop
    If f = True Then
        TextBox1.Text = N
        TextBox2.Text = P
    Else
        MsgBox("موجود غير")
    End If
    Sr.Close()
    Fs.Close()
End Sub

```

' زر حذف

```

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    Dim num As Integer, Prénom
    Dim foundFlag As Boolean = False
    Dim sr As StreamReader = File.OpenText("c:\Test11.TXT")
    Dim sw As StreamWriter = File.CreateText("TEMP.TXT")
    Do While (sr.Peek <> -1)

```

```
num = CInt(sr.ReadLine)
Prénom = (sr.ReadLine)

If (num <> TextBox1.Text) Then
    sw.WriteLine(num)
    sw.WriteLine(Prénom)
Else
    foundFlag = True
End If
Loop
sr.Close()
sw.Close()
File.Delete("c:\Testz.TXT")
File.Move("TEMP.TXT", "c:\Test2.TXT") 'Rename "TEMP.TXT" As
"YOB.TXT"
If foundFlag = False Then
    MsgBox("The name was not found.", 0, "")
Else
    TextBox1.Clear()
    TextBox2.Clear()

End If
End Sub
End Class
```

التعامل مع الملفات باستعمال الدوال

يمنحك Vb الفرصة للتعامل مع الملفات بالطرق القديمة مع مبرمجي Vb6

أنواع الملفات

يمكن تقسيم الملفات إلى ثلاثة أنواع على خلاف الملفات الخاصة بإطار العمل NetWork.

ملفات الوصول المتتابع Sequential File

تعتبر ملفات نصية . يمكنك من قراءة و كتابة البيانات فيها بالتوالي من بداية الملف إلى نهايته و لا يمكنك فتح الملف للقراءة و الكتابة في آن واحد . و تستعمل في نقل بيانات ملف في مربع نص أو العكس

الملفات العشوائية Random Files

وتستخدم هذه الملفات في حفظ البيانات التي تتكوّن من سجلات Records ذات أطوال ثابتة لهذا تستخدم المصفوفات والسجلات Structures في كتابة البيانات في هذا النوع.. إن تساوي أطوال وحدات الملف يجعل التعامل معه شبيهاً بالتعامل مع المصفوفات، حيث يمكن الوصول إلى أيّ وحدة في الملف عن طريق معرفة رقمها Index.. ويمكنك فتح هذه الملفات للقراءة والكتابة في وقت واحد.

الملفات الثنائية Binary Files

الفصل السادس المجلدات و الملفات

وهي مشابهة لملفات الوصول المتتابع، مع خلاف واحد، هو أن كل البيانات توضع فيها في صورة وحدات ذاكرة Bytes تحتوي على أرقام (سواء أكانت تمثل حروفا أو أرقاما أو غير ذلك)

التعامل مع الملفات باستعمال الدالة Open

لفتح ملف أو إنشائه استعمال الدالة Open و هي تأخذ الصيغة التالية :

FileOpen (fileNumber, fileName, mode[, access[,
share[, recordLength]]])

FileOpen([طول السجل],[طور المشاركة],[طور التعامل],[طور الفتح,مساره,رقم الملف])

تتكون الصيغة من المعامل رقم الملف و هذا نوع من الاختصار فبدلا نتعامل مع أسماء الملفات الطويلة نستعمل أرقما . فإذا كنت بصدد فتح العديد من الملفات، فيمكنك أن تستخدم الدالة FreeFile لتحصل على رقم جديد غير مستعمل .

أما المعامل الثالث طور التعامل فيأخذ قيم المرقم التالي :

فتح الملف للقراءة منه (وصول متتابع).	Input
فتح الملف للكتابة فيه (وصول متتابع).. لاحظ فتح الملف بهذه الطريقة يُلغي محتوياته السابقة، حتى ولو لم تكتب فيه!	Output
فتح الملف للإضافة إلى محتوياته السابقة (وصول متتابع).	Append
فتح الملف للقراءة أو الكتابة فيه في موضع أي سجل تختاره (وصول عشوائي).	Random
فتح الملف كملف ثنائي.	Binary

ويأخذ المعامل الرابع (طور التعامل) إحدى قيم المرقم OpenAccess التالية:

فتح الملف للقراءة فقط.	Read
فتح الملف للكتابة فقط.	Write
فتح الملف للقراءة والكتابة.	ReadWrite

لاحظ أن هذا المعامل لن يُفيدك عند التعامل مع ملفات الوصول المتتابع، وإنما سيفيدك عند التعامل مع ملفات الوصول العشوائي والملفات الثنائية.

أما المعامل الخامس (طور المشاركة)، فيأخذ إحدى قيم المرقم OpenShare التالية:

يمكن للتطبيقات الأخرى فتح الملف أثناء فتحك له.. ويجب أن تكون حذرا عند استخدام هذه الطريقة، حتى لا تتسبب التطبيقات الأخرى في تغيير محتويات	Shared
---	--------


```
Do Until EOF(1)
  FileGet(1, P)
  Console.WriteLine(P.Id)
  Console.WriteLine(P.Name)
```

```
Loop
FileClose(1)
```

=====

الملفات الثنائية Binary Files:

يمكنك التعامل مع الملفات الثنائية بنفس دوال ملفات الوصول المتتابع، ما عدا بالطبع تلك التي تقرأ أو تكتب سطوراً، فالملف الثنائي لا يحتوي على سطور.

تطبيقات

أولاً : الملفات العشوائية :

نقوم بإنشاء مشروع جديد و نصمم نافذة النموذج حسب الشكل



=====

تعريف سجل الشخص Person

```
Structure Person
  Dim Id As Integer
  <VBFixedString(20)> Dim Name As String 'السجل طول تحديد'
End Structure
Public P As New Person
Public x As Integer 'الحالي السجل رقم'
'الملف إلى سجلات إضافة'
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
  FileOpen(1, "c:/txtFile.txt", OpenMode.Random, , , Len(P))
  x = x + 1
  P.Id = CInt(TextBox1.Text)
```

```
P.Name = TextBox2.Text  
FilePut(1, P, x)  
TextBox1.Text = ""  
TextBox2.Text = ""  
TextBox1.Focus()  
FileClose(1)
```

End Sub

=====

'التالي'

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button2.Click  
FileOpen(1, "c:/txtFile.txt", OpenMode.Random, , , Len(P))  
x = x + 1  
FileGet(1, P, x)  
TextBox1.Text = P.Id  
TextBox2.Text = P.Name
```

If EOF(1) Then MsgBox("سجل آخر") : Exit Sub

```
FileClose(1)
```

End Sub

=====

'السابق'

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button3.Click  
FileOpen(1, "c:/txtFile.txt", OpenMode.Random, , , Len(P))  
x = x - 1  
If x < 1 Then MsgBox("سجل أول") : Exit Sub  
FileGet(1, P, x)  
TextBox1.Text = P.Id  
TextBox2.Text = P.Name  
FileClose(1)
```

End Sub

=====

'البحث'

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button4.Click  
FileOpen(1, "c:/txtFile.txt", OpenMode.Random, , , Len(P))  
Dim m As String = InputBox("الاسم ادخل")  
Dim found As Boolean  
السجلات كل طول حساب'  
x = LOF(1) / Len(P)
```

```
For i As Integer = 1 To x
    FileGet(1, P, i)
    If P.Name.Trim = m Then
        found = True
        Exit For
    End If
Next
If found = True Then
    TextBox1.Text = P.Id
    TextBox2.Text = P.Name
Else
    MsgBox("موجود غير")
End If
End Sub
```

ثانيا :الملفات التسلسلية

```
Structure Person
    Dim Id As Integer
    Dim Name As String
    Dim D_N As Date
End Structure
Public P As New Person
```

إضافة

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    FileOpen(1, "C:/SeqFile.txt", OpenMode.Append)
    P.Id = CInt(TextBox1.Text)
    P.Name = TextBox2.Text
    WriteLine(1, P.Id)
    WriteLine(1, P.Name)
    TextBox1.Clear()
    TextBox2.Clear()
    TextBox1.Focus()
    FileClose(1)
End Sub
```

=====

قراءة وعرض

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    FileOpen(1, "C:/SeqFile.txt", OpenMode.Input)
    Input(1, P.Id)
    TextBox1.Text = P.Id
    Input(1, P.Name)
    TextBox2.Text = P.Name
    FileClose(1)
End Sub
```

=====

قراءة محتوى كل الملف

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    FileOpen(1, "C:/SeqFile.txt", OpenMode.Input)
    Do Until EOF(1)
        TextBox3.Text += LineInput(1) & vbNewLine
    Loop
End Sub
```

الملفات التسلسلية و مربعات الحوار

فتح ملف

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    OpenFileDialog1.Filter = "All txt (*.txt)|*.txt| All File (*.*)|*.*"
    OpenFileDialog1.ShowDialog()
    If OpenFileDialog1.FileName = "" Then Exit Sub
    FileOpen(1, OpenFileDialog1.FileName, OpenMode.Input)
    Do Until EOF(1)
        TextBox1.Text = LineInput(1)
    Loop
    FileClose(1)
End Sub
```

=====

حفظ ملف

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    SaveFileDialog1.Filter = "All txt (*.txt)|*.txt| All File (*.*)|*.*"
    SaveFileDialog1.ShowDialog()
    If SaveFileDialog1.FileName = "" Then Exit Sub
```

```
FileOpen(1, SaveFileDialog1.FileName, OpenMode.Append)
PrintLine(1, TextBox1.Text)
FileClose(1)
End Sub
```

التعامل مع My.Computer.FileSystem

أولا : مع المجلدات :

إنشاء مجلد : Creating a New Directory

VB.Net Code

```
' ----- Create the directory requested by the user.
If (My.Computer.FileSystem.DirectoryExists( _
    "C:\test")) Then
    MsgBox("مسبقاً موجود الملف")
Else
    Try
        My.Computer.FileSystem.CreateDirectory("C:\test")
        MsgBox("إنشأؤه تم الملف.")
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End If
```

التأكد من وجود مجلد Determining if a Directory Exists

VB.Net Code

```
If (My.Computer.FileSystem.DirectoryExists( _
    "C:\Windows")) Then
    MsgBox("موجود المجلد", MsgBoxStyle.Information)
Else
    MsgBox("موجود غير المجلد", MsgBoxStyle.Information)
End If
```

نسخ مجلد Copying Directories

VB.Net Code

```
My.Computer.FileSystem.CopyDirectory( _
    sourceDirectory, destDirectory)
```

sourceDirectory : المجلد المصدر

destDirectory : المجلد الجديد

VB.Net Code

```
My.Computer.FileSystem.CopyDirectory(_  
"C:\test", "d:\test")
```

نقل مجلد : **Moving Directories**

 VB.Net Code

```
My.Computer.FileSystem.MoveDirectory(_  
sourceDirectory, destDirectory)
```

sourceDirectory : المجلد المصدر
destDirectory : المجلد الجديد

 VB.Net Code

```
My.Computer.FileSystem.MoveDirectory(_  
"C:\test", "d:\test")
```

 الطريقة Copy و Move متشابهتان إلى حد كبير

إعادة تسمية مجلد : **Renaming Directories**

 VB.Net Code

```
My.Computer.FileSystem.RenameDirectory(_  
sourceDirectory, newName)
```

sourceDirectory : المجلد المصدر
newName : اسم المجلد الجديد


 VB.Net Code

```
My.Computer.FileSystem.RenameDirectory(_  
"c:/test", "Mytest")
```

 يمكن استخدام الطريقة MoveDirectory لإعادة تسمية المجلد

 VB.Net Code

```
My.Computer.FileSystem.MoveDirectory(_  
"C:\test", "C:\MyTest")
```

 و يمكن استخدام الطريقة Rename


 VB.Net Code

```
Rename("C:\MyTest", "C:\test")
```

دمج الملفات مع المجلدات Parsing File and Directory

 VB.Net Code

```
Dim newPath As String = _
    My.Computer.FileSystem.CombinePath( _
        "C:\temp", "WorkFiles\TodaysWork.txt")
    MsgBox(newPath)
    ' Displays: "C:\temp\WorkFiles\TodaysWork.txt"
```

سينفذ التعليمة حتى ولو لم يكن المجد موجودا 

عرض اسم المجد الأخير

 VB.Net Code

```
MsgBox(My.Computer.FileSystem.GetName( _
    "C:\temp\part1\part2"))
    ' ----- Displays: part2
```

عرض اسم المجلدات الرئيسية

 VB.Net Code

```
' ----- Displays: C:\temp\part1
    MsgBox(My.Computer.FileSystem.GetParentPath( _
        "C:\temp\part1\part2"))
```

عرض ملفات مجلد :

- عرض كل ملفات مجلد

 VB.Net Code

```
For Each oneFile As String In _
    My.Computer.FileSystem.GetFiles("C:\Windows")
    MatchResults.Items.Add(oneFile)
Next
```

- عرض ملفات من نوع خاص لمجلد

 VB.Net Code

```
For Each oneFile As String In _
    My.Computer.FileSystem.GetFiles("C:\Windows", _
        FileIO.SearchOption.SearchTopLevelOnly, _
        "*.Exe")
    MatchResults.Items.Add(oneFile)
Next
```

- عرض ملفات من نوعين لمجلد

 VB.Net Code

```
For Each oneFile As String In _
    My.Computer.FileSystem.GetFiles("C:\Windows", _
        FileIO.SearchOption.SearchTopLevelOnly, _
        "*.LOG", "*.TXT")
    MatchResults.Items.Add(oneFile)
```

Next

عرض ملفات خاصة من مجلد خاص SpecialDirectories & GetDirectories

VB.Net Code

```
For Each foundDirectory As String In _
    My.Computer.FileSystem.GetDirectories _
        (My.Computer.FileSystem.SpecialDirectories.MyDocuments,
        FileIO.SearchOption.SearchAllSubDirectories, "**.*")
    ListBox1.Items.Add(foundDirectory)
Next
```

مسار المجلد الحالي CurrentDirectory

VB.Net Code

```
MsgBox(My.Computer.FileSystem.CurrentDirectory)
```

حذف مجلد رئيسي بلا فروع

VB.Net Code

```
My.Computer.FileSystem.DeleteDirectory("c:/test",
FileIO.DeleteDirectoryOption.ThrowIfDirectoryNotEmpty)
```

حذف مجلد رئيسي بفروعه

VB.Net Code

```
My.Computer.FileSystem.DeleteDirectory("c:/test",
FileIO.DeleteDirectoryOption.DeleteAllContents)
```

اقرأ مجلد المعومات و اعرض بعض خصائصه GetDirectoryInfo

VB.Net Code

```
Dim getInfo As System.IO.DirectoryInfo
getInfo = My.Computer.FileSystem.GetDirectoryInfo("C:/test")
MsgBox("الانشاء تاريخ " & getInfo.CreationTime)
MsgBox("استعمال آخر " & getInfo.LastAccessTime)
MsgBox("للكتابه وقت تاريخ آخر " & getInfo.LastWriteTime)
```

و هذا مثال آخر عن مجلد خاص GetDirectoryInfo

VB.Net Code

```
Dim getInfo As System.IO.DirectoryInfo
getInfo = My.Computer.FileSystem.GetDirectoryInfo _
(My.Computer.FileSystem.SpecialDirectories.MyDocuments)

MsgBox("الانشاء تاريخ " & getInfo.CreationTime)
MsgBox("استعمال آخر " & getInfo.LastAccessTime)
MsgBox("للكتابه وقت تاريخ آخر " & getInfo.LastWriteTime)
```


ثانيا : مع الملفات

التأكد من وجود ملف Determining If a File Exists

 VB.Net Code

```
If (My.Computer.FileSystem.DirectoryExists( _  
    userSuppliedPath) = True) Then  
    MsgBox("Invalid directory specified.")  
Else  
    '----- Process directory here.  
End If
```

 VB.Net Code

```
If (My.Computer.FileSystem.FileExists( _  
    "C:/test.txt")) Then  
    MsgBox("فعلا موجود الملف")  
Else  
    MsgBox("موجود غير الملف")  
End If
```


نسخ ملف CopyFile

 VB.Net Code

```
My.Computer.FileSystem.CopyFile(sourceFileName  
,destinationFileName ,overwrite)
```

 VB.Net Code

```
My.Computer.FileSystem.CopyFile("c:/test.txt", "d:/test.txt", True)
```

 عند كتابة True فمعنى هذا أن الملف المنسوخ و إن كان موجودا سيتم استبداله

حذف ملف DeleteFile

و يشتمل على عدة صيغ و نكتفي بذكر الصيغة البسيطة

 VB.Net Code

```
My.Computer.FileSystem.DeleteFile("C:\test.txt")
```

حذف ملف و إرساله إلى سلة المحذوفات

 VB.Net Code

```
My.Computer.FileSystem.DeleteFile("C:\Test.txt",  
FileIO.UIOption.AllDialogs, FileIO.RecycleOption.SendToRecycleBin) '
```

اقرأ معلومات الملف و اعرض بعض خصائصه GetFileInfo

 VB.Net Code

```
Dim information As System.IO.FileInfo
information = My.Computer.FileSystem.GetFileInfo("c:/test.txt")
MsgBox("The file's full name is " & information.FullName & ".")
MsgBox("Last access time is " & information.LastAccessTime & ".")
MsgBox("The length is " & information.Length & ".")
```

نقل ملف

 VB.Net Code

```
My.Computer.FileSystem.MoveFile("C:\test.txt", "d:\TestDir2\test.txt")
```

الكتابة والتحرير في الملفات



VB.Net Code

```
My.Computer.FileSystem.WriteAllText(file, Text, append)
My.Computer.FileSystem.WriteAllText(file, Text, append, encoding)
```


File : اسم و مسار الملف

Text : النص

Appnd : يأخذ قيمة منطقية :

True : يضيف و يحافظ على البيانات

False : يمسح البيانات القديمة و يضيف الجديدة :

 حتى و إن لم يكن الملف موجودا سيتم إنشاؤه

 VB.Net Code

```
My.Computer.FileSystem.WriteAllText("c:/test.txt", "Visual basic",
True)
```

عرض و قراءة الملفات



VB.Net Code

```
My.Computer.FileSystem.ReadAllText(file)
My.Computer.FileSystem.ReadAllText(file,encoding)
```

File : اسم و مسار الملف

 VB.Net Code

```
TextBox1.Text = My.Computer.FileSystem.ReadAllText("C:/test.txt")
```

الكتابة إلى الملفات الثنائية

 VB.Net Code

```
Dim fileContents() As Byte = {244, 123, 56, 34}
```

```
My.Computer.FileSystem.WriteAllBytes("C:\Output.bin", _
fileContents, True)
```

قراءة الملفات الثنائية

VB.Net Code

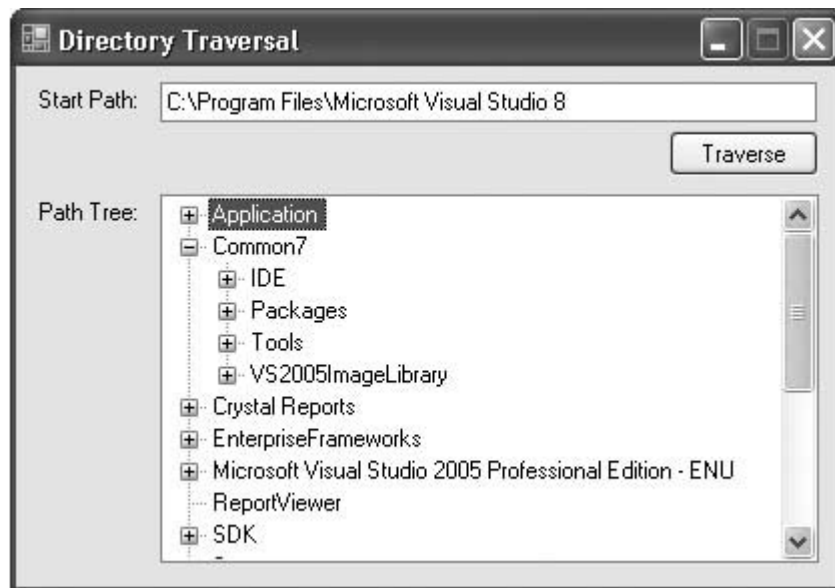
```
Dim fileData() As Byte = _
My.Computer.FileSystem.ReadAllBytes("c:/test.bin")
For i As Integer = 0 To UBound(fileData)
    TextBox1.Text = fileData(i)
Next
```

تطبيقات :

البحث في المجلدات و المجلدات الفرعية

قم بإضافة الأدوات التالية

الأداة	الاسم البرمجي
TextBox	StartPath
Button	Button
TreeView	PathTree



VB.Net Code

```
Imports System.IO
Public Class Form1
    Private Sub BuildDirectoryTree(ByVal fromNode As TreeNode, _
        ByVal basePath As String)
        ' ----- Attach all of the subdirectories found in
```

```

' basePath to the supplied node. If fromNode is
' Nothing, create root entries.
Dim newDirectory As TreeNode
Dim justTheSubdirectory As String

' ----- Retrieve all directories in this path.
For Each oneDirectory As String In _
    My.Computer. _
        FileSystem.GetDirectories(basePath)
    ' ----- Extract just the final directory name.
    justTheSubdirectory = My.Computer.FileSystem.GetName( _
        oneDirectory)

    If (fromNode Is Nothing) Then
        ' ----- Add a top-level subdirectory.
        newDirectory = PathTree.Nodes.Add( _
            justTheSubdirectory)
    Else
        ' ----- Add a subordinate node.
        newDirectory = fromNode.Nodes.Add( _
            justTheSubdirectory)
    End If

    ' ----- Recurse into the subdirectory.
    BuildDirectoryTree(newDirectory, My.Computer.FileSystem. _
        CombinePath(basePath, justTheSubdirectory))
Next oneDirectory

End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    If (My.Computer. _
        FileSystem.DirectoryExists( _
            StartPath.Text) = False) Then
        MsgBox("Please supply a valid directory path.", _
            MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, _
            "Invalid Path")
    End Sub
End If

' ----- Clear any previous tree.
PathTree.Nodes.Clear()
' ----- Call the scanning routine, a recursive routine.

```

```
BuildDirectoryTree(Nothing, StartPath.Text)
End Sub
End Class
```

البحث عن الملفات و المجلدات

قم بإدراج الأدوات التالية :

الأداة	الاسم البرمجي
TextBox	StartPath
TextBox	Pattern
ListBox	Matches



VB.Net Code

```
' ----- Make sure the supplied path is valid.
If (My.Computer._
FileSystem.DirectoryExists( _
StartPath.Text) = False) Then
MsgBox("Please supply a valid directory path.", _
MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation, _
"Invalid Path")
Exit Sub
End If

' ----- Clear any previous items.
MatchResults.Items.Clear()

' ----- First, add in the subdirectories.
For Each oneItem As String In _
My.Computer.FileSystem.GetDirectories( _
```

```

StartPath.Text, _
FileIO.SearchOption.SearchTopLevelOnly, _
PathPattern.Text)
MatchResults.Items.Add("[ " & _
    My.Computer.FileSystem.GetName(oneItem) & "]")
Next oneItem

' ----- Second, add in the files.
For Each oneItem As String In _
    My.Computer.FileSystem.GetFiles(StartPath.Text, _
    FileIO.SearchOption.SearchTopLevelOnly, _
    PathPattern.Text)
    MatchResults.Items.Add( _
    My.Computer.FileSystem.GetName(oneItem))
Next oneItem
    
```

إضافة محتوى مصفوفة إلى ملف

VB.Net Code

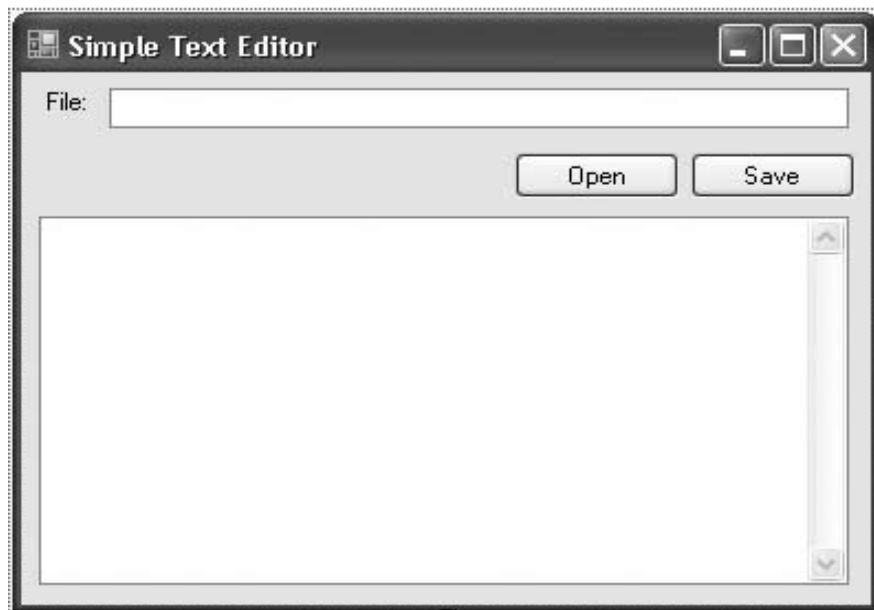
```

Dim fileContents() As String = {244, 123, 56, 34}
For i As Integer = 0 To fileContents.Count - 1
My.Computer.FileSystem.WriteAllText("c:/test1.txt", fileContents(i) & "
",
True)
Next
    
```

محرر الملفات 1

قم بإضافة الأدوات التالية حسب الشكل :

الأداة	الاسم البرمجي Name
Button(Save)	Button1
Button(Open)	Button2
Textbox1	FilePath
Textbox2	Editor



VB.Net Code

```
Imports System.IO
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        ' ----- Open an existing file and load its text.
        Try
            Editor.Text = My.Computer. _
                FileSystem.ReadAllText(FilePath.Text)
        Catch ex As Exception
            MsgBox("Could not open the file due to the " & _
                "following error:" & vbCrLf & vbCrLf & ex.Message)
        End Try

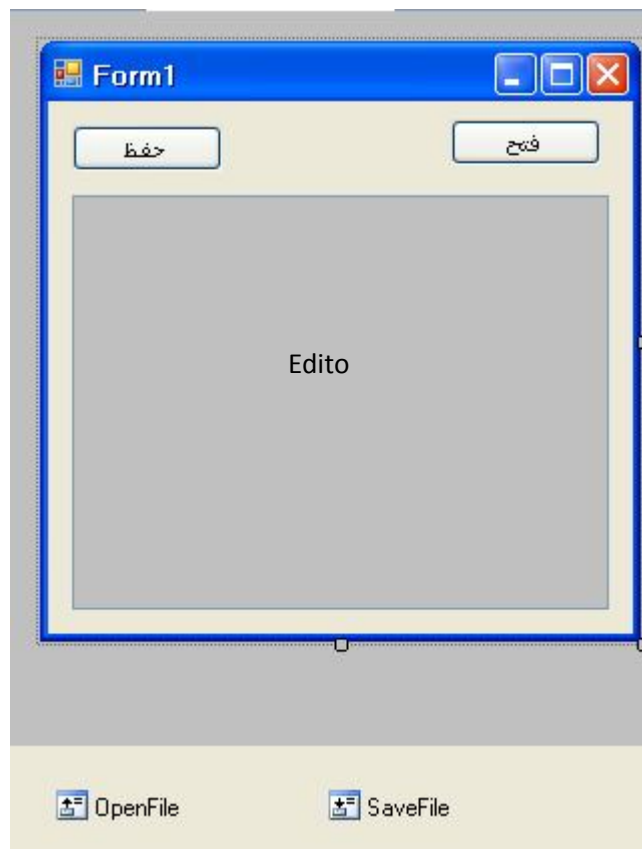
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        ' ----- Save the edited data.
        If (My.Computer.FileSystem.FileExists(FilePath.Text) = _
            True) Then
            If (MsgBox("File exists. Overwrite?", _
                MsgBoxStyle.YesNo Or MsgBoxStyle.Question, _
                "Overwrite") <> MsgBoxResult.Yes) Then Exit Sub
        End If
    End Sub
End Class
```

```
' ----- Save the data.
Try
    My.Computer.FileSystem.WriteAllText(FilePath.Text, _
        Editor.Text, False)
Catch ex As Exception
    MsgBox("Could not save the file due to the " & _
        "following error:" & vbCrLf & vbCrLf & ex.Message)
End Try
End Sub
End Class
```

محرر النصوص 2

قم بإضافة الأدوات حسب الشكل



VB.Net Code

```
Imports System.IO
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        With OpenFile
            .Title = "فتح"
            .Filter = "All text (*.txt) |*.txt"
            If .ShowDialog = Windows.Forms.DialogResult.OK Then
```



```

If .FileName = "" Then Exit Sub
' ----- Open an existing file and load its text.
Try
    Editor.Text = My.Computer. _
        FileSystem.ReadAllText(.FileName)
Catch ex As Exception
MsgBox("Could not open the file due to the " & _
"following error:" & vbCrLf & vbCrLf & ex.Message)
End Try
End If
End With
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
' ----- Save the edited data.
With SaveFile
    .Title = "حفظ"
    .Filter = "All text (*.txt) |*.txt"
    If .ShowDialog = Windows.Forms.DialogResult.OK Then
        If .FileName = "" Then Exit Sub
        If My.Computer.FileSystem.FileExists(.FileName) = _
            True) Then
            If (MsgBox("File exists. Overwrite?", _
                MsgBoxStyle.YesNo Or MsgBoxStyle.Question, _
                "Overwrite") <> MsgBoxResult.Yes) Then Exit Sub
            End If
        Try
            My.Computer.FileSystem.WriteAllText(.FileName, _
                Editor.Text, False)
        Catch ex As Exception
            MsgBox("Could not save the file due to the " & _
                "following error:" & vbCrLf & vbCrLf & ex.Message)
        End Try
    End If
End With

End Sub

End Class

```

الفصل السابع : المجموعات Collection

- المصفوفة القائمة Array List
- الجداول المختلطة Hash Table
- القائمة المرتبة Sorted List
- المجموعات الخاصة Collection

أولا : مجموعة المصفوفة القائمة **ArrayList Collection**

المصفوفة القائمة هي عبارة عن مصفوفة لكنها تخزن عدد كبير من العناصر من نفس النوع حيث يمكنك من ترتيبها و البحث عنها.و يمكنك إضافة عناصر لها أو حذفها من أي موضع تريد مثل القوائم لذلك سميت بمصفوفة القائمة. كيف نعرفها ؟

Dim MyAList As New ArrayList

لاحظ أننا لمن نحدد طول المصفوفة و لم نضع الأقواس .

فهي لا تحتاج لتحديد طولها فهو مبدئيا 16 خانة .فكلما زادت عدد الخانات زاد طولها

إضافة عناصر **Add**

نستعمل الوسيلة **Add** لإضافة عنصر جديد إلى معظم المجموعات .

Dim Index As Integer = MyAList.Add(Value)

 VB.Net Code

```
Dim MyAList As New ArrayList
```

```
MyAList.Add("الصالح")
```

```
MyAList.Add("محمد")
```

```
MsgBox(MyAList(1))
```

إضافة مجموعة عناصر **AddRange**

 VB.Net Code

```
Dim MyAList As New ArrayList
```

```
MyAList.AddRange(New String() {"الصالح", "محمد"})
```

و للتحقق من وجود عنصر ضمن المصفوفة القائمة استعمل الوسيلة **Contain**

 VB.Net Code

```
If MyAList.Contains("الصالح") Then
```

```
MsgBox("المصفوفة في موجود العنصر")
```

```
End If
```

و لمعرفة عدد عناصرها نستعمل الوسيلة **Count**

عرض محتوى مصفوفة قائمة

لقراءة العنصر الأول من المصفوفة القائمة نكتب

```
MsgBox(MyAList(0).ToString) ' الصالح
```

و لعرض كل عناصر المصفوفة دفعة واحدة نستعمل الحلقات التكرارية

 VB.Net Code

```
For i As Integer = 0 To MyAList.Count - 1
```

```
Console.WriteLine(MyAList(i))
```

Next

For Each x As String In MyAList

Console.WriteLine(x)

Next

إدراج عنصر Insert

MyAList.Insert(1, "أحمد")

و لإدراج مجموعة من العناصر في موضع محدد استعمل الوسيلة InsertRange

MyAList.InsertRange(1, New String() {"سعد", "أحمد"})

استبدال مجموعة عناصر SetRange

MyAList.AddRange(New String() {"الصالح", "محمد", "سعد"})

MyAList.SetRange(1, New String() {"أحمد"})

حذف عناصر

إذا أردت حذف عنصر من المصفوفة القائمة نستعمل الصيغة

Arrlist . Remove (العنصر)

و لحذف حذف العنصر محمد من المصفوفة القائمة نكتب

MyAList.Remove("محمد")

و إذا أردت حذف عنصر من موضع محدد بتحديد رقم خانته نستعمل الصيغة

ArrList . RemoveAt(رقم الخانة)

و لحذف العنصر رقم 2 من المصفوفة نكتب

MyAList.RemoveAt(2)

و لحذف مجموعة متتالية من العناصر نستعمل الصيغة

ArrList . RemoveRange (عدد الخانات , رقم خانة البداية)

ترتيب العناصر

لترتيب عناصر المصفوفة القائمة نستعمل الصيغة :

ArrList.Sort

نسخ و نقل العناصر

اقرأ مجموعة عناصر "GetRange":

تنسخ مجموعة من العناصر المتتالية من مصفوفة قائمة إلى أخرى.. ولها الصيغة التالية:

newList = ArrayList.GetRange(عدد الخانات , رقم بداية الخانات)

 VB.Net Code

Dim MyAlist, NewList As New ArrayList

MyAlist.AddRange(New String() {"الصالح", "محمد", "سعد"})

NewList = MyAlist.GetRange(1, 2)

```
For Each x As String In NewList
```

```
    Console.WriteLine(x) ' سعيد ، محمد ، سعد
```

```
Next
```

"استنساخ" Clone:

تصنع نسخة مطابقة من المصفوفة القائمة:

```
NewList = OledList.Clone
```

إلى مصفوفة "ToArray":

تنسخ مصفوفة إلى مصفوفة أخرى و لها الصيغة التالية

 VB.Net Code

```
Dim MyAlist As New ArrayList
```

```
MyAlist.AddRange(New String() {"الصالح", "محمد", "سعد"})
```

```
Dim NewList As Array = MyAlist.ToArray()
```

```
For Each x As String In NewList
```

```
    Console.WriteLine(x)
```

```
Next
```

نسخ إلى CopyTo

مماثلة للوسيلة السابقة إلا أنها تتطلب تحديد عدد عناصرها انظر المثال

 VB.Net Code

```
Dim MyAList As New ArrayList()
```

```
MyAList.AddRange(New String() {"الصالح", "محمد", "سعد"})
```

```
Dim NewList(MyAList.Count - 1) As String ' أو لا المصفوفة حجم حددنا
```

```
MyAList.CopyTo(NewList)
```

```
MsgBox(NewList(1)) ' "محمد" الرسالة تعرض
```

و لها صيغة يمكنك من تحديد بداية خانة النسخ و عدد العناصر المنسوخة

(عدد العناصر , المصفوفة الجديدة , رقم خانة البداية) .CopyTo المصفوفة القديمة

"تكرار" Repeat:

تملاً مصفوفة قائمة بنفس العنصر:

```
NewList = OledList.Repeat(العدد , العنصر)
```

"عكس" Reverse:

تعكس ترتيب العناصر في المصفوفة القائمة.. ولها الصيغة:

```
ArrList.Reverse()
```

أو الصيغة:

```
ArrList.Reverse(عدد الخانات , رقم بداية الخانات)
```

البحث

الفصل السابع المجموعات

نستعمل الوسيلة IndexOf و لها الصيغة التالية

(رقم خانة بداية البحث , كلمة البحث) = ArrList . IndexOf = رقم العنصر

```
MsgBox(MyAList.IndexOf("محمد", 0)) '1
```

أو نستعمل الوسيلة LastIndexOf و هي مماثلة للوسيلة السابقة إلا أنها تبدأ البحث من الخانة الأخيرة

أو نستعمل الوسيلة BinarySearch و لها الصيغة

(عدد الخانات , خانة البداية , عنصر البحث) = Array.BinarySearch = رقم العنصر

مثال

 VB.Net Code

```
Dim MyAList As New ArrayList()
MyAList.AddRange(New String() {"الصالح", "محمد", "سعد"})
Dim FindX As String = InputBox("الاسم ادخل")
If FindX = "" Then Exit Sub
Dim I As Integer = MyAList.BinarySearch(FindX)
If I > 0 Then
    MsgBox("رقم الخانة في موجود الاسم" & I+1)
Else
    MsgBox("موجود غير الاسم")
End If
```

محتوى Contains

ترجع True في حالة احتواء المصفوفة على العنصر

```
If MyAList.Contains("الصالح") Then
```

```
    MsgBox("موجود العنصر")
```

```
End If
```

ملاحظه هامة : هناك تشابه في وسائل و خصائص و طرق المجموعات **Collection** و

المصفوفات **Arrays** و القوائم **Lists**

ثانيا : مجموعة الجدول المختلط HashTable Collection

عند تعاملنا مع المصفوفات استعملنا خانة أو عنصر المصفوفة للقيام بجميع العمليات من إضافة و

حذف و بحث... أنا لي أن أعرف رقم العنصر في مصفوفة كبيرة ؟ فما الحل ؟

يمنحك Vb.Net خلية الجدول المختلط تمتاز بقدرتك التعامل مع العنصر بالرمز الدال عليه **Key**

بدلا من رقم العنصر **Index** و قيمته **Value**

ولاستخدام هذه الخلية، ابدأ بتعريف متغير من نوعها، كالتالي:

```
Dim hTable As New Hashtable
```



```

Dim i As Integer
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Students As New Hashtable
    Students.Add("سامح", 14.5)
    Students.Add("وسيم", 12.67)
    Dim InputX As String = InputBox("الاسم ادخل")
    If InputX = "" Then Exit Sub
    Dim findX As Boolean
    For i = 0 To Students.Keys.Count - 1
        If Students.Keys(i) = InputX Then
            findX = True
            Exit For
        End If
    Next
    If findX = True Then
        MsgBox(InputX & " معدله " & Students.Values(i))
    Else
        MsgBox("موجود غير")
    End If
End Sub

```

ثالثا : خلية القائمة المرتبة SortedList Class:

القائمة المرتبة هي خليط من المصفوفة والجدول المختلط، حيث يمكنك التعامل مع عناصرها برقم الخانة (كالمصفوفة) أو بالرمز (كالجدول المختلط).. هذا بالإضافة إلى أنّ عناصرها مرتبة دائما على حسب رموزها، وحتى لو أدخلت الرموز غير مرتبة، فسيتم ترتيبها آليا. ويمكنك تعريف متغير منها كالتالي:

```

Dim sList As New SortedList
Add لإضافة عناصر إلى القائمة المرتبة واضح نستعمل الوسيلة
SList.Add("وسيم", 13.67)
SList.Add("سامح", 14.5)
لقراءة مفتاح Key العنصر الأول 0 من القائمة المرتبة نستعمل الوسيلة
MsgBox(SList.GetKey(0)) ' سامح

```


يمكنك تعريف مجموعة خاصة بك Custom Collection بجملته كالتالية:

Public MyCollection As New Collection

وبهذا أصبح للمتغير MyCollection الخصائص والوسائل التقليدية المعروفة للمجموعات:

إضافة **Add** – عدد العناصر **Count** – العنصر **Item** – حذف **Remove**.....

يمكنك إضافة أي نوع البيانات لأن عناصرها من نوع Object

ملاحظة : يبدأ ترقيم الخانات و العناصر من 1 و ليس من 0

الفصل الثامن : قواعد البيانات Data base

- تعريف
- إنشاء قاعدة بيانات عن طريق Access
- مكونات Ado.Net
- مزودات البيانات
- كائن الاتصال
- الوضع المتصل
- كائن الأوامر
- الوضع المنفصل
- مجموعة البيانات Data Set
- موصل البيانات
- العمليات الأساسية مع Data Set
- كائن بيانات الجدول
- التعامل مع Data Binding
- التعامل مع Data Set من خلال النوافذ
- التعامل مع SQL Server
- التعامل مع تقرير Ceystal Report
- التعامل مع تقرير Report
- التعامل مع مكتبة ADO

قواعد البيانات Data Base

قواعد البيانات هي ملف تخزن فيه البيانات بشكل منظم في جداول يسهل التعامل معها كالإضافة و التعديل و الحذف و القراءة . باستخدام الوسائل التي تمنحها لك برامج قواعد البيانات و هناك أنواع كثيرة من قواعد البيانات مثل :

برنامج Access ، SQL Server ، Oracle ، My SQL ،

و تسمى هذه البرامج أنظمة إدارة قواعد البيانات (DBMS) Database Management Systems ، فهي تسهل عليك التعامل مع قواعد البيانات بحيث توفر لك وسائل و أدوات لإنشائها و إدارتها بكل سهولة .

كما أنها تستعمل لغة الاستعلامات المهيكلة (SQL) (Structured Query Language)، وهي لغة خاصة لحفظ واسترجاع وتحديث البيانات في قواعد البيانات.

مكونات قاعدة البيانات :

تتكون أساسا من جداول (Tables) فيها يتم تخزين البيانات . و كل جدول يتكون من أعمدة (Columns) و تسمى أيضا حقول (Fields) و كذا من صفوف (Rows) و تعرف أيضا بالسجلات (Records)

الخطوات الأساسية لتصميم قواعد البيانات

1 - تحديد الغرض من قاعدة البيانات .

الفصل الثامن قواعد البيانات

- 2 - تحديد الجداول التي تحتاج إليها .
- 3 - تحديد الحقول التي تحتاج إليها .
- 4 - تعريف الحقول و معرفة أنواعها .
- 5 - تحديد العلاقات بين الجداول .

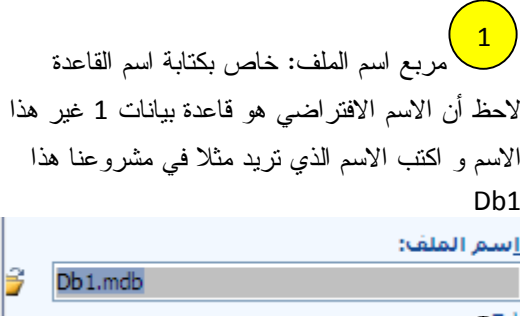
إنشاء قاعدة بيانات جديدة :

- لإنشاء قاعدة بيانات جديدة نتبع الخطوات التالية :
1. انقر زر أوفيس ثم اختر جديد

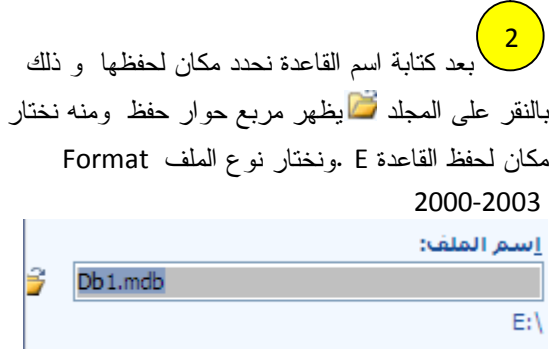


قاعدة بيانات فار

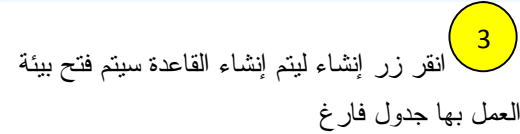
أو انقر على قاعدة بيانات فارغة من صفحة الشروع في العمل
يظهر مربع الحوار على اليسار كما يلي :



1 مربع اسم الملف: خاص بكتابة اسم القاعدة
لاحظ أن الاسم الافتراضي هو قاعدة بيانات 1 غير هذا
الاسم و اكتب الاسم الذي تريد مثلاً في مشروعنا هذا
Db1



2 بعد كتابة اسم القاعدة نحدد مكان لحفظها و ذلك
بالنقر على المجلد يظهر مربع حوار حفظ ومنه نختار
مكان لحفظ القاعدة E. ونختار نوع الملف Format
2000-2003



3 انقر زر إنشاء ليتم إنشاء القاعدة سيتم فتح بيئة
العمل بها جدول فارغ



ثم انقر زر عرض يظهر مربع حوار يخيرك فيما إذا كنت تريد حفظ الجدول



اكتب اسم الجدول Students
ثم انقر موافق تظهر نافذة منها نضيف الحقول



الفصل الثامن قواعد البيانات

لاحظ أن النافذة تتكون من أربعة أجزاء

اسم الحقل	كتابة اسم الحقل أو العمود Columns
نوع الحقل	لاختيار نوع الحقل
وصف الحقل	معلومات سريعة تصف دور الحقل
خصائص الحقل	لتحديد خصائص الحقل

شروط تسمية الحقل

- يستحسن أن يكون بالانجليزية أو الفرنسية
- أن لا يتعدى طوله 255 حرفاً
- أن يكون غير منقطع (و لربط مقطعين نستعمل (_) كما في الحقل Date_N)
- أن يكون مفهوماً ليسهل على المستخدم فهمه
- لا يبدأ بعدد أو رمز أو نقطة

أنواع الحقول

نوع الحقل	الشرح
نص	نص صغير لا يتعدى طوله 255 حرفاً
مذكرة	نص طويل
رقم	عدد
تاريخ أو وقت	بيانات تدل على التاريخ أو الوقت
عملة	بيانات تدل على العملة
ترقيم تلقائي	رقم يزيد تلقائياً ب 1 كلما أضفت صفاً

الفصل الثامن قواعد البيانات

نعم/لا	لاختيار بين أمرين نعم أو لا
كائن Ole	لإدراج صورة أو أي ملف آخر
ارتباط تشعبي	لربط الحقل بموضع آخر انترنت ...
معالج البحث	لاختيار قيم من قائمة

خصائص الحقل

الخاصية	الشرح
حجم_الحقل	تعيين أقصى حجم للبيانات المخزونة كنوع البيانات "نص" أو "رقم" أو "ترقيم تلقائي".
تنسيق	تخصيص طريقة لظهور الحقل عند عرضه أو طبعه.
منازل_عشرية	تحديد عدد المنازل العشرية لاستخدامها عند عرض أرقام.
قيم_جديدة	تعيين ما إذا كان الحقل "ترقيم تلقائي" تزايدياً أو يقوم بتعيين قيمة عشوائية.
قناع_إدخال	عرض أحرف التحرير التي ترشد إلى طريقة إدخال البيانات.
تسمية توضيحية	تعيين النص المعروف افتراضياً في التسميات للنماذج والتقارير والاستعلامات .
قيمة_افتراضية	تعيين قيمة افتراضية للحقل تلقائياً عند إضافة سجلات جديدة.
قاعدة_التحقق_من_الصحة	توفير تعبير يجب أن يكون صحيحاً عند إضافة قيمة أو تغييرها في هذا الحقل.
نص_التحقق_من_الصحة	إدخال نص يظهر عند مخالفة إحدى القيم لتعبير "قاعدة التحقق من الصحة".
مطلوب	لطلب إدخال البيانات في الحقل.
السماح بالطول الصفري	السماح بإدخال (بواسطة تعيينها إلى "نعم") سلسلة ذات طول صفري ("") في الحقل "نص" أو "مذكرة".
مفهرس	تسريع الوصول إلى البيانات الموجودة في هذا الحقل بواسطة إنشاء فهرس واستخدامه.
ضغطUnicode	ضغط النص المخزن في هذا الحقل عند تخزين كمية كبيرة من النص (> 4069 حرفاً)
وضعIME	التحكم في تحويل الأحرف في إصدار Windows آسيوي.
وضع جملةIME	التحكم في تحويل الأحرف في إصدار Windows آسيوي.
العلامات_الذكية	إرفاق علامة ذكية بهذا الحقل .
إلحاق فقط	السماح بتعيين إصدارات) بواسطة تعيينها إلى "نعم" الحقل "مذكرة"
تنسيق_النص	اختيار نص "منسق" لتخزين النص بتنسيق HTML والاحتفاظ بميزات النص المنسق. اختيار نص "عادي" لتخزين النص فقط.
محاذاة النص	تحديد المحاذاة الرئيسية للنص من خلال عنصر تحكم.

الفصل الثامن قواعد البيانات

الدقة	تعيين العدد الإجمالي للأرقام المسموح بها متضمناً تلك الأرقام على يمين أو يسار العلامة العشرية.
مقياس	تعيين أقصى عدد من الأرقام يمكن تخزينها على يمين الفاصلة العشرية.

وفي مشروعنا هذا نحدد ثلاثة حقول لجدول Students كما يلي

اسم الحقل	النوع	الخاصية
Num	رقم	عدد صحيح (حجم الحقل)
prénom	نص	(حجم الحقل) 25
Classe	رقم	عدد صحيح
Sexe	معالج البحث	

بالنسبة لحقل الجنس Sexe نوعه معالج البحث بمجرد اختيار هذا النوع يظهر معالج البحث كما يلي :

معالج البحث

ينشئ المعالج عمود البحث، الذي يعرض قائمة قيم يمكن الاختيار منها، ما هي الطريقة التي تريد بها عمود البحث الحصول على القيم؟

أريد أن يبحث عمود البحث عن القيم في جدول أو استعلام.

سوف أكتب القيم التي أريدها.

إلغاء
التالي <
> السياق
إلغاء الأمر

حدد الخيار الثاني ثم انقر التالي

الفصل الثامن قواعد البيانات

معالج البحث

ما هي القيم التي ترغب في مشاهدتها في عمود البحث؟ أدخل عدد الأعمدة التي تريدها في القائمة. ثم اكتب القيم التي تريدها في كل خلية.

لتعديل عرض أحد الأعمدة، اسحب حافته اليسرى إلى العرض الذي تريده، أو انقر نقرًا مزدوجًا فوق الحافة اليسرى للعمود للحصول على أفضل احتواء.

عدد الأعمدة: 1

عمود 1	↓
ذ	↑
	*

إيحاء < السابق > التالي إلغاء الأمر

ثم أكتب في القائمة (ذ) و يرمز للذكور ثم (أ) و يرمز للإناث ثم انقر ثم انقر إنها نختار الحقل Num كمفتاح أساسي وذلك بتحديد الحقل ثم انقر الزر المفتاح الأساسي من شريط الأدوات و بنفس الطريقة نقوم بإنشاء جدول آخر باسم Score (النتائج) و يحتوي على الحقول التالية

اسم الحقل	النوع	الخاصية
Num	رقم	عدد صحيح
Prénom	نص	25 حرفا
Arabe	رقم	عدد مزدوج
Math	رقم	عدد مزدوج
Franc	رقم	عدد مزدوج
Moyen	رقم	عدد مزدوج

و نختار حقل الرقم Num كمفتاح أساسي كما فعلنا مع جدول الطلاب Students

بعد التصميم قم بتحميل الجدولين ببعض البيانات كما في الشكل

Num	Prénom	Arabe	Math	Fra	Moye
1	سامح	12	14	11	
2	سعاد	13	12	12	
3	داوود	14	14	14	
*					

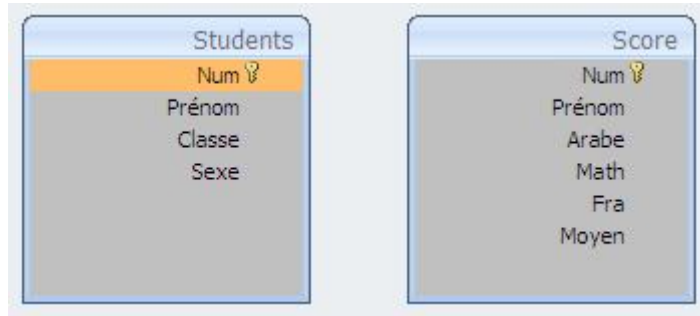
Num	Prénom	Classe	Sex
1	سامح	1	ذ
2	سعاد	1	أ
3	داوود	2	ذ
4	وردة	2	أ
5	وسليم	3	ذ
*			

لاحظ أنه عند تحميل حقل الجنس Sexe تظهر قائمة منسدلة بها الخيارين ذ أو أ و لا يمكنك كتابة غيرهما يبقى الآن الربط بين الجدولين .

من تبويب أدوات قاعدة البيانات نختار علاقات تظهر أمامك النافذة التالية



نضغط إضافة لإضافة الجدول المحدد ثم حدد الجدول الثاني ثم انقر إضافة و هكذا نحصل على الشكل

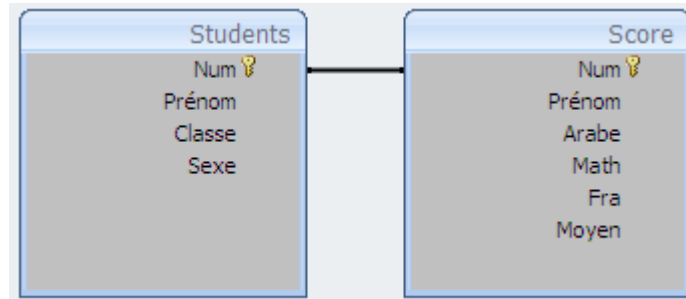


انقر الحقل Num من جدول الطلاب Students ثم اسحب باستمرار نحو حقل الرقم Num من جدول النتائج Score تظهر النافذة التالية

الفصل الثامن قواعد البيانات

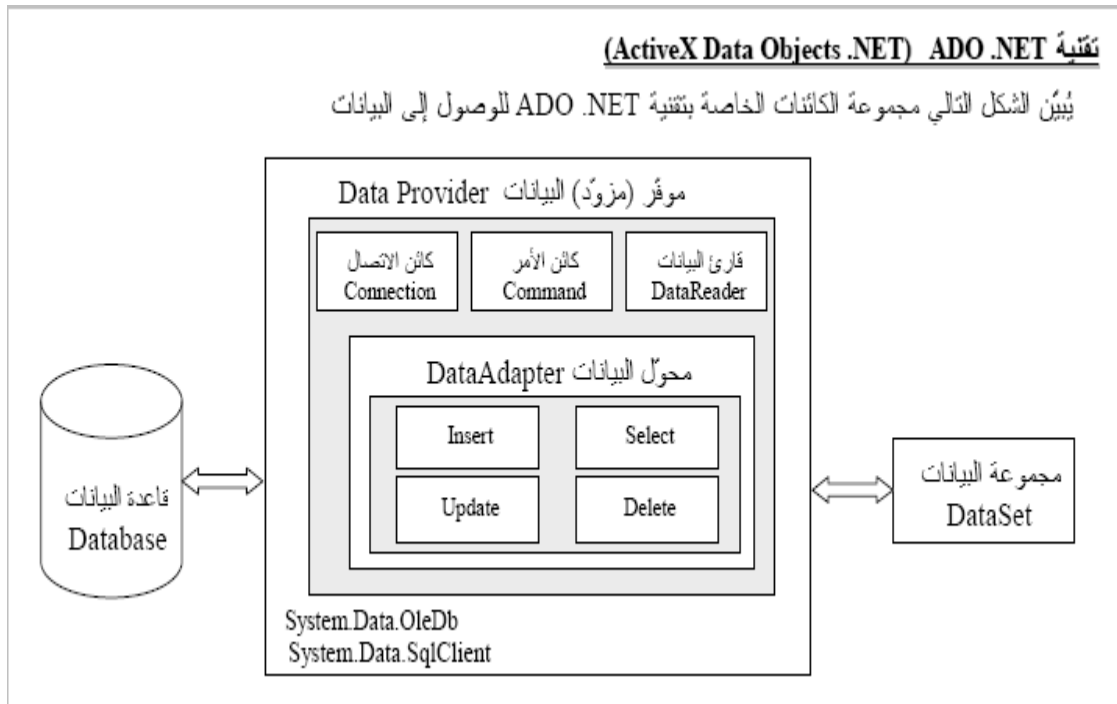


انقر إنشاء لإنشاء علاقة بين الجدولين.



التعامل مع الكائن ADO.NET

فهي مجموعة من الفئات Classes للتعامل مع مصادر البيانات (إضافة و التحديث و الحذف ...).
تتمكنك الاتصال بقاعدة بيانات مثل Access أو Oracle ... من خلال بيئة Vs.Net بواسطة ADO.NET



مزودات البيانات .NET Data Providers

بعد إنشاء القاعدة و اختيار نوع الاتصال نختار مزود البيانات . فما هي مزودات البيانات ؟

الفصل الثامن قواعد البيانات

ما هي إلاحقة وصل بين ADO.NET و مصادر البيانات حسب أنظمة القواعد المستخدمة (البرامج المستخدمة).

توجد ثلاثة أنواع من مزودات البيانات مدعومة في إطار عمل Frame Net Work : OLE DB .NET Data Provider

هذا النوع من المزودات يمكنك من التعامل مع قواعد البيانات القديمة التابعة لتقنية COM المنجزة قبل صدور برنامج Vb.Net 2008. تستطيع الاتصال بهذا المزود مع برنامج Access و Oracle و MySQL ...

:SQL Server .NET Data Provider

هذا النوع من المزودات يتناسب مع SQL Server التابع للبرنامج Vb2008 ODBC .NET Data Provider قبل تقنية .Net و قبل تقنيات الوصول إلى البيانات كانت تقنية ODBC أول تقنية تمكن المبرمجين من الوصول إلى مصادر البيانات المختلفة عن طريق مشغلات ODBC Drivers و تدرج هذه المزودات تحت فضاء الأسماء حسب ما يبينه الجدول :

البرنامج	فضاء الأسماء
Access	System.Data.OleDb
SQL Server	System.Data.SqlClient
ملفات Odbc	System.Data.Odbc
Oracle	System.Data.OracleClient

فلا تنسى استيراد أو لا مجال الأسماء المستعمل

الميزة الهامة التي تجدها في ADO.NET هو أنك تستعمل نفس الكود مع البيانات المصدرية ل Access و البيانات الخاصة ب SQL Server و كذا مع Oracle . الاختلاف الطفيف فقط يكون في نوع المزود

كائن الاتصال Connection

بعدما تعرفنا على نوع المزود أو البرنامج الذي نتعامل معه آن الأوان أن نتصل و نربط البرنامج بقاعدة البيانات المحددة طبعا سنحتاج لمعلومات خاصة عن القاعدة كاسمها أو بالأحرى مسارها و كلمة مرور و اسم المستخدم وهذا ما يعرف بسلسلة أو نص الاتصال و هو عبارة عن عنوان يقودنا إلى القاعدة مباشرة . وعليه لإنشاء كائن الاتصال نحتاج للعوامل التالية يبينها الجدول التالي :

المعامل	الشرح
Provider	اسم المزود
Data Source	اسم و مسار قاعدة البيانات مع Access
Intial Catalog	اسم و مسار قاعدة البيانات مع SQL Server
Integrated Secutity	يحدد ما إذا كان الاتصال كاملا مع ويندوز
Connection TiemOut	يحدد وقت محاولة الاتصال
UserId/PassWord	كلمة المرور و اسم المستخدم لقاعدة البيانات

المعامل الأول / المزود يمكن أن يأخذ الوسيطات حسب الجدول التالي :

الفصل الثامن قواعد البيانات

اسم مزود البيانات	قاعدة البيانات
Microsoft.Jet.OLEDB.4.0	Microsoft Access
SQLOLEDB.1	Microsoft SQL Server 2005
MSDAORA.1	Oracle

أمثلة على نص الاتصال

نص الاتصال مع Access

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = E:\Db1.mdb"
```

و إذا كانت قاعدة البيانات Access محمية نكتب :

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = E:\Db101.mdb;" & _  
"Jet Oledb:Database Password=MyPassWord"
```

و إذا أردت تخصيص وقت محدد للاتصال نكتب :

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = E:\Db1.mdb;" & _  
"Connection Timeout=10"
```

كما يمكنك جعل الاتصال كاملا مع ويندوز

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = E:\Db1.mdb;" & _  
"Persist Security Info=True;"
```

إما إذا كانت قاعدة البيانات في نفس محرك الأقراص يمكن أن نكتب :

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = Db1.mdb;"
```

و إذا كانت في نفس المجلد نكتب

VB.Net Code

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source =" & Application.StartupPath & "\db1.mdb"
```

حيث Application.StartupPath تعبر عن المجلد الحالي

نص الاتصال بقاعدة البيانات SQL Server 2005

VB.Net Code

```
Dim StrCon As String = "Data Source=.\SQLEXPRESS;Initial  
Catalog=SQLDb1;Integrated Security=True;"
```

نص الاتصال مع Access 2007 بكلمة مرور

VB.Net Code

```
con.ConnectionString = "Provider =Microsoft.ACE.oledb.12.0;" & _  
"data source =c:\test.accdb;" & _  
"Jet Oledb:Database Password=MyPassWord"
```

عند التعامل مع SQL Server عليك تجاهل اسم المزود Provider بعد أن تعرفنا على نص الإتصال (العنوان) علينا أن ننشأ كائن الإتصال (موزع البريد) و نرفقه بنص الإتصال حتى نصل القاعدة المرغوبة

ملاحظة هامة جدا : سنتعامل مع قواعد البيانات Access أما إذا أردت التعامل مع SQL Server فمعاليتك إلا أن تقوم بتغيير طفيف فقط :

1. تغيير مجال الأسماء

2. و استبدال كلمة OleDb بـ SqlClien في جميع الأكواد

إنشاء كائن الإتصال و ربطه بنص الإتصال

بعد أن تعرفنا على نص الاتصال بقاعدة البيانات نحتاج إلى كائن الإتصال ليربطنا و يوصلنا إلى القاعدة و الذي يتطلب منا تحديد نص الاتصال Connection String و كذا نوع المزود Data Provider

VB.Net Code

```
Dim Con As New OleDbConnection(connectionString) ' Access مزود  
Dim con As New SqlConnection(connectionString) ' SQLServer مزود  
Or  
Dim Con As New OleDbConnection()  
Dim con1 As New SqlConnection()  
Con.ConnectionString = "الاتصال نص"  
con1.ConnectionString = "SQLserver الاتصال نص"
```

و إذا أردت الاستعلام و التأكد من القيم المدخلة في نص الإتصال نستعمل مجموعة من تاخصائص للقراءة فقط نكتب:

VB.Net Code

```
Console.WriteLine(Con.Provider) ' Microsft.jet.oledb.4.0  
Console.WriteLine(Con.DataSource) 'E:\Db1.mdb  
Console.WriteLine(Con.ConnectionTimeout) ' افتراضيا 15
```

بعد الإتصال بقاعدة البيانات نحتاج الآن لفتحها .

فتح و غلق الإتصال

الفصل الثامن قواعد البيانات

لفتح الاتصال واضح استعمال الوسيلة Open

فتح القاعدة ' Con.Open ()

و يجب غلق الاتصال بعد تحميل مجموعة البيانات DataSet لتفادي الاستثناءات باستعمال الوسيلة Close
Con.Close

تفادي الاستثناءات

لتفادي الاستثناءات و الوقوع في الأخطاء نستعمل طريقة صيد الأخطاء Try ... Catch
و يأخذ الصيغة التالية

Try

{ مقطع من الكود }

Catch نوع الاستثناء As اسم_الاستثناء

{ مقطع معالجة هذا النوع من الاستثناءات }

.....

Catch يمكن أن يتكرر مقطع

لأي عدد من المرات، بأنواع مختلفة من الاستثناءات

.....

Catch Exception As اسم_الاستثناء

{ مقطع معالجة باقي أنواع الاستثناءات }

Finally

{ مقطع إلزامي، ينفذ في كل الأحوال }

End Try

مثال

VB.Net Code

Try

Con.Open() ' فتح الاتصال

Catch ex As Exception

عرض رسالة تبيين الخطأ أو الاستثناء ' MsgBox(ex.Message)

Finally

Con.Close() ' غلق الاتصال

End Try

التأكد من حالة الاتصال

تستطيع أن تتعرف على حالة الاتصال باستعمال الخاصية State التابعة لكائن الاتصال Connection
و التي تأخذ القيم التالية

مغلق	Closed
جاري فتح الاتصال	Connecting
الاتصال مفتوح	Open
يتم تنفيذ أمر	Executing
جاري الحصول على البيانات من سجلات مصدر البيانات	Fetching

و هذا مثال يوضح ما قلناه



VB.Net Code

```
If con.State = ConnectionState.Open Then
    MsgBox("تم فتح القاعدة بنجاح")
Elseif con.State = ConnectionState.Closed Then
    MsgBox("تم غلق القاعدة بنجاح")
Elseif con.State = ConnectionState.Connecting Then
    MsgBox("جاري الاتصال")
Elseif con.State = ConnectionState.Executing Then
    MsgBox("يتم تنفيذ الأمر")
Elseif con.State = ConnectionState.Fetching Then
    هنا تنفيذه تريد شئ اي'
End If
```

فتح قاعدة البيانات عن طريق مربع الحوار فتح



VB.Net Code

```
Dim con As New OleDbConnection

Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    With OpenFileDialog1
        .Title = "فتح"
        .Filter = "All database (*.mdb)|*.mdb"

        If .ShowDialog = Windows.Forms.DialogResult.OK Then
            con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
                "Data Source = " & OpenFileDialog1.FileName
            con.Open()

            If con.State Then MsgBox("بنجاح الاتصال فتح تم")
        End If
    End With
End Sub
```

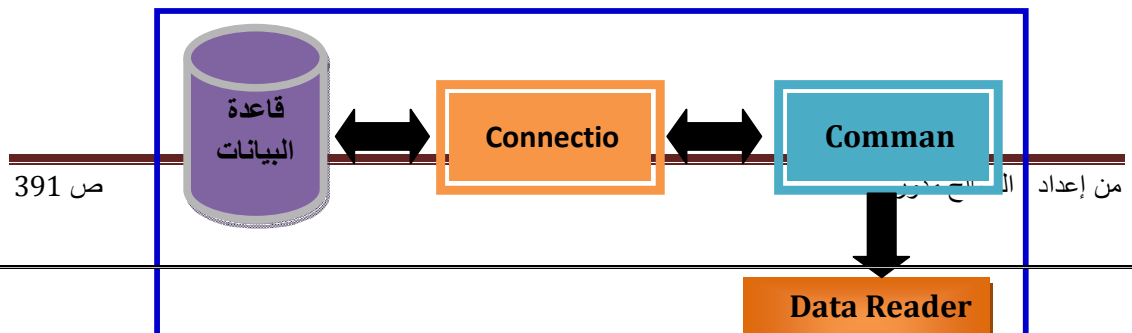
بعد الاتصال بقاعدة البيانات و فتحها نختار طريقة الربط و الاتصال . إما الاتصال المباشر Connected أو الاتصال الغير مباشر Disconnected

الوضع المتصل Connected Mode

و يتم الاتصال مباشرة مع قاعدة البيانات اتصالاً مستمراً يتم من خلاله القيام بالتعامل المباشر مع البيانات فكلما تم تعديل أو إضافة أو حذف ... يتم تنفيذها مباشرة على قاعدة البيانات الأصلية.

مكونات الوضع المتصل

إليك هذا المخطط يوضح لك فئات الوضع المتصل



يتضح لنا من الشكل أن فئات الوضع المتصل هي : Connection ، Command ، Data Reader
 الفئة Connection المختصة بالاتصال بقاعدة البيانات و كل ما يرفق هذا الاتصال من خصائص
 ومواصفات الاتصال (كاسم المستخدم و كلمة المرور و المزود المستخدم)
 بعد الاتصال بمصادر البيانات نحتاج لكائن آخر (Command) لتنفيذ أوامر جمل الاستعلام SQL
 (إضافة ، تحديث ، حذف) .

بعد تنفيذ الأوامر ستحصل على نتائج نصل إليها و نقرأها باستعمال الكائن Data Reader

كائن الأوامر Command

بعد الاتصال مع قاعدة البيانات نحتاج إلى القيام بعمليات مختلفة عن طريق جمل الاستعلامات المهيكلة SQL و تنفيذها
 وتعديل بيانات القاعدة وهذا باستعمال كائن تنفيذ الأوامر Command

خصائصه

الخاصية	القيمة	الشرح
Name	أي اسم تفهمه	الاسم البرمجي الذي نتعامل معه بالكود
Connection	كائن الاتصال المستخدم	اسم الاتصال الذي يقوم به كائن الأمر
CommandType	text	لغة الاستفسار المهيكلة SQL
	StoredProcedur	تنفيذ إجراء موجود في قاعدة البيانات
CommandText	TableDirect	التعامل مع جدول واحد
	تعتمد على الخاصية السابقة	يقوم بتنفيذ من الأوامر الثلاثة السابقة

طرق كائن الأمر Command

الطريقة	الشرح
ExecuteReader	ينفذ استعلام و يرجع أكثر من سجلات و لقراءتها نستعمل قارئ البيانات
ExecuteNonQuery	تنفيذ أمر يقوم بإنشاء أو تعديل في قاعدة البيانات و يرجع عدد الصفوف المتأثرة
ExecuteScalar	تنفيذ أمر يرجع قيمة واحدة فقط كعدد السجلات

إنشاء كائن الأوامر و ربطه بالاتصال

وهذا يتوقف على نوع المزود المختار فإن كان مزود Access من نوع OLE DB .NET Data
 Provider سنعرف الكائن هكذا :

VB.Net Code

Dim Cmd As New OleDbCommand(Sql, connection)

جملة الاستعلامات المهيكلة Sql :

ربط كائن الأوامر بكائن الاتصال Connection

وهذا مثال يوضح لنا كيف ننشئ كائن الأوامر

VB.Net Code

```
Imports System.Data.OleDb ' مزود البيانات أكسيس
Public Class Form1
    Dim StrCon As String = "Provider = Microsoft.jet.oledb.4.0;" & _
        "data source = e:/db1.mdb" ' نص الاتصال
    Dim Con As New OleDbConnection(StrCon) ' إنشاء كائن الاتصال

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Load
        Try
            Con.Open() ' فتح الاتصال
            Dim Cmd As New OleDbCommand(Sql, connection) ' إنشاء كائن الأوامر و ربطه
            ' بكائن الاتصال
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        End Sub
```

تنفيذ أوامر جمل الاستعلامات

عند تنفيذ جمل استعلام بواسطة الكائن Command لابد من تحديد نوع جمل الاستعلام التي نتعامل معها هل هي جمل استرجاعية أو جملة تنفيذية
الجمل الاستعلامية الاسترجاعية هي تلك الجمل التي لا تؤثر على سجلات قاعدة البيانات و إنما تقوم بقراءة محتوياتها و نستعمل جملة Select لتحديد هذا النوع من الجمل. يمكنك تنفيذها باستعمال الطريقة ExecuteScalar أو ExecuteReader
أما الجمل التنفيذية هي التي تؤثر مباشرة على سجلات قاعدة البيانات و هي إما أن تكون Insert ، Update ، Create ، Delet ، Into ... و يتم تنفيذها باستعمال الطريقة ExecuteNonQuery

تنفيذ الأمر ExecuteScalar

تقوم بقراءة عمود واحد و تعيد عدد يمثل عدد السجلات المتأثرة . و تستعمل لتنفيذ الأوامر التي ترجع قيمة فريدة وحيدة . كحساب عدد كل السجلات ، و كذا في دوال التجميع.

VB.Net Code

```
Dim SQL As String = "Select count(*)from students"
Dim cmd As New OleDbCommand(SQL, Con)
MsgBox("المسجلين عدد " & cmd.ExecuteScalar)
```

الفصل الثامن قواعد البيانات

تنفيذ الأمر Execute Reader

عند تنفيذ هذا الأمر يعود بمجموعة من السجلات يمكن قراءتها بكائن قارئ البيانات SqlDataReader أهم خصائصه

الخاصية	الشرح
Read	ترجع قيمة منطقية True في حالة وجود سجل موالى و False إذا لم يوجد
Item	ترجع سجلات الحقل (العمود) باستخدام اسم أو الرقم الفهرسي للعمود
GetValues	ترجع قيمة الصف كاملاً
isDBNull	هل تحتوي على قيمة فارغة
close	إغلاق قارئ البيانات
RecordsAffected	عدد السجلات المتأثرة عند تنفيذ جمل SQL

قراءة السجلات

لقراءة السجلات يجب استدعاء الطريقة Execute Reader و اسنادها إلى قارئ البيانات الذي يعيد مصفوفة من السجلات لقراءتها نستعمل الوسيلة Read في حلقة .

```
VB.Net Code
Dim SQL As String = "Select * From Students Where Prenom ='Salah'"
Dim Cmd As New OleDbCommand(SQL, Con)
Dim Dr As OleDbDataReader = Cmd.ExecuteReader ' اسناد الطريقة إلى كائن قارئ
البيانات
Do While Dr.Read ' قراءة السجلات في حلقة
    ListBox1.Items.Add(Dr("Prenom") & " " & Dr("Nom")) ' عرض حقل الاسم و
اللقب الموافقة للشرط
Loop
```

كما يمكنك قراءة السجلات بالوسيلة GetValue و هذا باستبدال السطر المظلل ب

```
Listbox1.Items.Add(Dr.GetValue("Prenom"))
```

و لقراء سجل واحد نكتب

```
Dr.Read()
MsgBox(Dr("Prenom") & " " & Dr("Nom"))
```

و هكذا نكتب الوسيلة Read الخاصة بقارئ البيانات و نقرأ السجل التالي

تنفيذ الأمر ExecuteNonQuery

تنفذ الأمر، وتعيد فقط عدد السجلات التي تأثرت بالأمر.. ويمكنك استخدام هذه الوسيلة لتنفيذ أوامر التحديث والحذف والإدراج مع الجمل التنفيذية

```
VB.Net Code
Dim SQL As String = "Update Students Set " & _
"prenom = 'Salim' Where Prenom ='Salima'"
Dim cmd As New OleDbCommand(SQL, Con)
Me.Text = cmd.ExecuteNonQuery ' ترجع عدد السجلات المتأثرة
```

و في إنشاء الجداول ترجع 0 في حالة نجاح العملية.

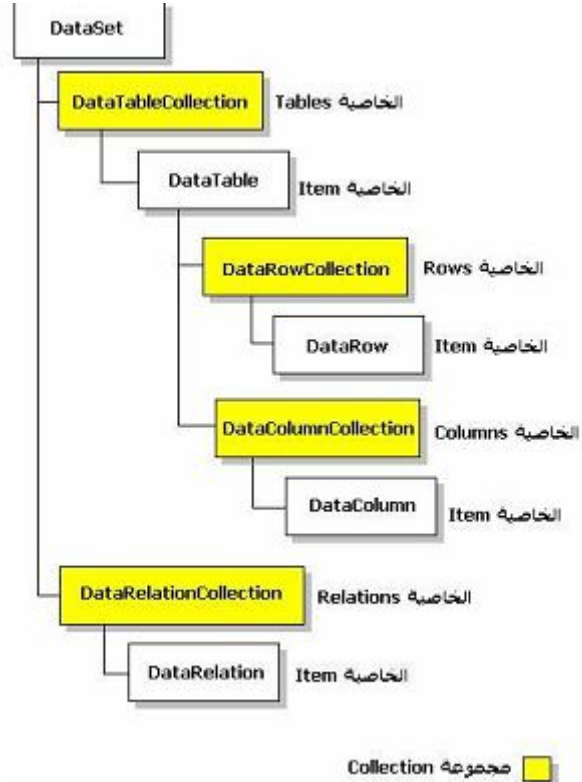
الوضع المنفصل

في هذا الوضع يتم الاتصال بقاعدة البيانات (الخادم Server) و نسخ البيانات المطلوبة إلى جهازك (العميل Client) في مخزن يسمى مجموعة البيانات DataSet ثم غلق الاتصال. و تتيح لك تقنية ADO.Net إمكانية التعامل مع القاعدة بعد غلق الاتصال Disconnected Mode، حيث يتم تحميل البيانات في ذاكرة الجهاز، ليتعامل معها المستخدم على جهازه دون التأثير على البيانات المصدرية ، وعندما يريد حفظ التغييرات، يتم فتح الاتصال مرّة أخرى للتحديث البيانات في قاعدة البيانات. و بذلك نخفف الضغط على خوادم قواعد البيانات فكل عميل يمكنه التعامل مع القاعدة بدون مشاكل.

مجموعة البيانات DataSet

تمثل قاعدة بيانات مصغرة يتم فيها نسخ البيانات من قاعدة البيانات الأصلية و حفظها في ذاكرة الجهاز وبذلك يمكنك التعامل مع هذه البيانات بتعديلها وحذفها و حفظها و كذا إنشاء جداول....و هذا دون تأثير على القاعدة الأصلية.

مكونات مجموعة البيانات



من الشكل يمكن تقسيم مجموعة البيانات إلى مجموعتين

1- مجموعة DataTableCollection: و تتكون من

جدول البيانات DataTable: يمثل جدول من قاعدة البيانات

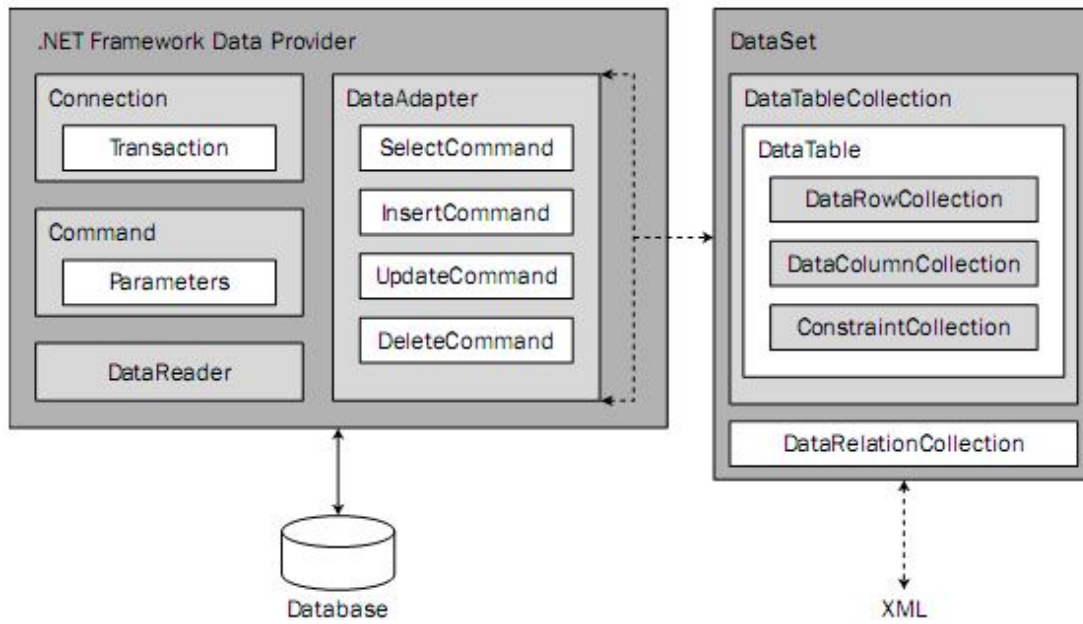
صفوف البيانات DataRow: تمثل صفوف أو سجلات جدول

الفصل الثامن قواعد البيانات

عمود البيانات DataColumn : يمثل عمود أو حقل من جدول
2- مجموعة DataRelationCollection و تتكون من
DataRelation : فئة تختص بربط بين الجداول عن طريق العلق

موصل البيانات DataAdepter

يقوم مقام ثلاث كائنات كائن الاتصال و كائن الأوامر و كائن قارئ البيانات . يقوم بفتح الاتصال بمصدر البيانات عن طريق كائن الاتصال ثم يستخدم كائن الأوامر لتنفيذ عملية ما ثم قارئ البيانات لقراءة البيانات و تعبئتها في مجموعة البيانات .



إنشاء كائن محول البيانات

عند إنشاء كائن موصل البيانات يجب اختيار نوع المزود الذي تتعامل معه . ثم ربطه بكائن الاتصال وكذا الجدول أو البيانات التي تتعامل معها.

VB.Net Code

```
Imports System.Data.OleDb 'مزود البيانات Access
Public Class Form1
    Dim StrCon As String = "الاتصال نص"
    Dim Con As New OleDbConnection(StrCon) ' كائن الاتصال
    Dim DA As New OleDbDataAdapter("SQL", Con) ' إنشاء كائن موصل البيانات وتحديد
    الجدول و ربطه بكائن الاتصال
```

و حتى تتمكن من التعامل مع البيانات و قراءتها يجب عليك تحميل مجموعة البيانات DataSet و هذا عن طريق الوسيلة Fill التابعة لكائن محول البيانات . و التي تتطلب تحديد مجموعة البيانات DataSet و اسم الجدول



VB.Net Code

```
Imports System.Data.OleDb
Public Class Form1
    Dim StrCon As String = "Provider = Microsoft.jet.oledb.4.0;" & _
        "Data source =E:/db1.mdb" ' نص الاتصال
    Dim con As New OleDbConnection(StrCon) ' ربط كائن الاتصال
    Dim Da As New OleDbDataAdapter("Select * From Students", con) ' تحديد الجدول
        و ربط موصل البيانات ب كائن الاتصال
    Dim Ds As New DataSet ' إنشاء مجموعة بيانات جديدة

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Load
        Try
            con.Open() ' فتح الاتصال
            Da.Fill(Ds, "Students") ' تحميل مجموعة البيانات عن طريق موصل البيانات مع تحديد
            ' الجدول
            ' ....
        Catch ex As Exception
            MsgBox(ex.Message) ' عرض رسالة الاستثناءات
        End Try
    End Sub
End Class
```

بعد تحميل مجموعة البيانات يمكنك غلق الاتصال دون خوف لأنك حصلت على نسخة مصغرة لقاعدة البيانات في DataSet مخزنة في ذاكرة جهازك .
والآن نحتاج لعرض البيانات إما في جدول أو صناديق النصوص أو القوائم

عرض البيانات في صناديق النصوص

يمكنك عرض البيانات بواسطة كائن الارتباط Binding يقوم هذا الكائن بربط أداة كمرجع نص بكائن البيانات DataSet بواسطة الوسيلة Add وطبعاً يجب أن نحدد الجدول ثم الحقل الذي ستعرض سجلاته
Control . DataBinding . Add ("Text" , مجموعة البيانات , اسم الحقل , اسم الجدول)
انظر المثال : نضيف هذه التعليمات مكان النقط ... في الكود السابق



VB.Net Code

```
TextBox1.DataBindings.Add("Text", Ds, "Students.Num") '
TextBox2.DataBindings.Add("Text", Ds, "Students.Prenom")
TextBox3.DataBindings.Add("Text", Ds, "Students.Classe")
' ...
```

عرض البيانات في جدول

يمكنك Vs.Net الأداة DataGridView1 لعرض بيانات جدول من قاعدة البيانات وهذا من خلال الخاصية DataSource و يجب أن نحدد لها مجموعة البيانات DataSet و منها نختار جدول مصدر البيانات



```
DataGridView1.DataSource = Ds.Tables("Students")
```

و يمكنك عرض البيانات باستعمال كائن العرض DataView يقوم بعرض بيانات جدول و يمتلك عدة خصائص و وسائل أهمها

ويملك كائن العرض الخصائص التالية:الجدول الذي يتم عرض سجلاته. Dim dv As New DataView dv.Table = Ds.Tables("TableName")	Table
تصفية سجلات حقل وفق شرط محدد Dim dv As New DataView(Ds.Tables(0)) dv.RowFilter = "Prenom like 'س*'" DataGridView1.DataSource = dv	RowFilter
حالة السجلات التي تريد عرضها (السجلات الأصليّة، المعدّلة، المحذوفة، المضافة...)	RowStateFilter
استخدم هذه الخاصية للتعامل مع السجلات المعروضة عن طريق أرقامها.. لاحظ أنّ هذه الخاصية هي الخاصية الافتراضية، ويمكن ألا تكتب اسمها.	Item
عدد السجلات المعروضة.	Count
اجعل هذه الخاصية True لتسمح بحذف السجلات المعروضة.	AllowDelete
اجعل هذه الخاصية True لتسمح بتحرير السجلات المعروضة.	AllowEdit
اجعل هذه الخاصية True لتسمح بإضافة بعض السجلات لتلك المعروضة.	AllowNew
نصّ يمثّل عمودا (أو أكثر) ليتمّ الترتيب تبعاً له.. ويمكن أن يحتوي هذا النصّ على اتجاه الترتيب (تصاعديّ ASC أو تنازليّ DESC).. مثال: DataView1.Sort = "Author DESC"	Sort
اجعل هذه الخاصية True لترتيب السجلات افتراضياً.	ApplyDefaultSort

وسائل DataView

تضيف سجلاً جديداً للعرض.. Dim dv As New DataView(Ds.Tables(0)) Dim rowView As DataRowView = dv.AddNew() rowView("Num") = 105 rowView("Prénom") = "علي" rowView("Moyen") = 10.5 DataGridView1.DataSource = dv	AddNew
تحذف السجلّ الذي تمّدها برقمه.	Delete
استخدم هذه الوسيلة للبحث عن قيمة ما في السجلات المعروضة، حيث تعيد لك رقم أول سجلّ يحتوي على القيمة المطلوبة.. ويمكن أن ترسل لها مصفوفة من القيم لتعثر على السجلّ الذي تحتوي خانته المناظرة على هذه القيم.. وفي حالة عدم العثور على أيّ سجلّ،	Find

الفصل الثامن قواعد البيانات

تعيد هذه الوسيلة -1.	
مماثلة للوسيلة السابقة، ولكنها تعيد مصفوفة (خاناتها من النوع DataRowView) تحتوي على كلّ السجلات التي عثرت بها على القيم المطلوبة.	FindRows

عرض البيانات عن طريق كائن العرض في جدول DataGridView



VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
DataGridView1.DataSource = DV
```

تصفية البيانات عن طريق كائن العرض



VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
DV.RowFilter = "Prenom = 'Salah'"
DataGridView1.DataSource = DV
```

في المثال السابق يتم تصفية الطلاب الذين لهم اسم Salah كما يمكنك معرفة السجلات الموافقة للشرط السابق أي ما هو عدد الطلاب الذين أسماءهم الصالح نضيف
MsgBox("السجلات عدد" & DV.Count)

إضافة سجل جديد لكائن العرض و هذا باستعمال كائن صف العرض DataRowView انظر المثال



VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
Dim DRV As DataRowView = DV.AddNew
DRV(0) = 104
DRV(1) = "Samir"
DRV("Classe") = 1
DataGridView1.DataSource = DV
```

البحث في الجدول عن طريق كائن العرض
استعمل الوسيلة Find الخاصة بكائن العرض حيث تعيد لك رقم أول سجل تعثر عليه و هذا بعد ترتيب
السجلات حسب حقل محدد .



VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
DV.Sort = "Prenom"
Dim i As Integer = DV.Find("Salah")
MsgBox("رقم السجل في يوجد" & i)
```

كما يمكنك أن ترسل مصفوفة من القيم . فإن عثر على أي قيمة من القيم تعرض هذا السجل أولاً .



VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
DV.Sort = "Prenom"
Dim vals(1) As Object
```



```
vals(0) = "Ali"
vals(1) = "Salah"
For i As Integer = 0 To UBound(vals)
    Console.WriteLine(DV.Find(vals(i))) ' تعيد موضع كل سجل تعثر عليه '
Next
```

و أيضا يمكنك استعمال الوسيلة FindRow و هي مماثلة للوسيلة السابقة إلا أنها تعيد كائن صف العرض DataGridView

VB.Net Code

```
Dim DV As New DataView(Ds.Tables("Students"))
DV.Sort = "Prenom"
Dim Drv() As DataGridView = DV.FindRows("Ali")
For i As Integer = 0 To UBound(Drv)
    Console.WriteLine(Drv(i).Row(2)) ' يعرض لقب الاسم علي '
Next
```

التنقل بين السجلات

نستعمل كائن ربط المحتوى BindingContext التابع لخصائص النموذج . فهو يمكنك من التنقل بين السجل و هذا بواسطة الخاصية Position التي تتطلب منا تحديد مجموعة البيانات DataSet و اسم الجدول

VB.Net Code

```
Me.BindingContext(DataSet, TableName).Position = Value
```

يمكنك الآن إسناد أي رقم للسجل لكي تنتقل إليه . أول سجل رقمه 0 و ما هو رقم آخر سجل ؟ يمكنك استخدام خاصية Count، حيث سيكون رقم السجل الأخير = عدد كل السجلات - 1

VB.Net Code

```
Me.BindingContext(DataSet, TableName).Count - 1
```

السجل التالي :

VB.Net Code

```
With Me.BindingContext(Ds, "Students")
    If .Position < .Count - 1 Then ' إذا كان الموضع الحالي أصغر من آخر سجل '
        .Position += 1 ' انتقل إلى السجل التالي '
    Else
        MsgBox("سجل آخر")
    End If
End With
```

السجل السابق

VB.Net Code

```
With Me.BindingContext(Ds, "Students")
    If .Position > 0 Then ' إذا كان الموضع الحالي أكبر من السجل الأول '

```

```
.Position -= 1 ' انتقل إلى السجل السابق
Else
    MsgBox("سجل أول")
End If
End With
```

السجل الأول



VB.Net Code

```
Me.BindingContext(DataSet, TableName).Position = 0
```

السجل الأخير



VB.Net Code

```
With Me.BindingContext(Ds, "Students")
    .Position = .Count - 1
End With
```

إضافة سجل جديد

نستعمل جمل الاستعلامات المهيكلة SQL مع الجملة Insert Into
1. باستعمال صناديق النصوص



VB.Net Code

```
Dim SQL As String = "Insert Into Students (Num,Prenom,Classe)" & _
"Values (" & TextBox1.Text & " , " & TextBox2.Text & " , " & TextBox3.Text & ")"
Dim cmd As New OleDbCommand(SQL, con) ' إنشاء كائن الأوامر
Me.Text = cmd.ExecuteNonQuery ' تنفيذ الأمر
```

2. باستعمال المعاملات Parameters

تتطلب هذه الطريقة وضع قيم الحقول على شكل معاملات وهذا بكتابتها على الشكل @Field ثم إضافة هذه المعاملات إلى كائن الأوامر مع الخاصية Parameters التي تتطلب تحديد اسم العامل و نوعه ثم إسناد قيمة له ثم تنفيذ أمر الإضافة



VB.Net Code

```
Dim SQL As String = "Insert into Students (Num,Prenom,Classe)" & _ ' الحقول
"Values (@Num ,@Prenom,@Classe)" ' تحديد أسماء المعاملات
Dim Cmd As New OleDbCommand(SQL, con) ' إنشاء كائن الأوامر
With Cmd
    .CommandType = CommandType.Text
    ' إضافة المعاملات و إسناد قيم لها
    .Parameters.Add("@Num", OleDbType.Integer).Value = TextBox1.Text
    .Parameters.Add("@Prenom", OleDbType.BSTR).Value = TextBox2.Text
    .Parameters.Add("@Classe", OleDbType.Integer).Value = TextBox3.Text
    .ExecuteNonQuery() ' تنفيذ الأمر
End With
```

3. إضافة قيم مباشرة



VB.Net Code

```
Dim SQL As String = "Insert Into Students (Num ,prenom ,Classe)" & _
    'الحقول' & _
    "Values (39,'Nadia',1)" 'إسناد القيم للحقول مباشرة'
Dim Cmd As New OleDbCommand(SQL, con) 'إنشاء كائن الأوامر'
Me.Text = Cmd.ExecuteNonQuery 'تنفيذ الأمر'
```

تعديل سجل

لتعديل سجل و حفظ التغييرات نستعمل جملة Update و التي تتطلب تحديد اسم الجدول و الحقول التي يتم تعديلها و كذا تحديد الشرط .



VB.Net Code

```
Dim SQL As String = "Update Students Set " & _
    "[Num] = " & Val(TextBox1.Text) & ", " & _
    "[Prenom] =" & TextBox2.Text & "" & _
    "Where [Num]= " & TextBox1.Text
Dim cmd As New OleDbCommand(SQL, con)
cmd.ExecuteNonQuery()
```

أما لتعديل سجل بإدخال قيم مباشرة نكتب



VB.Net Code

```
Dim SQL As String = "Update Students Set " & _
    "Prenom ='Salima' Where Prenom ='Salim'"
Dim cmd As New OleDbCommand(SQL, con)
cmd.ExecuteNonQuery()
```

في المثال السابق يتم استبدال كل أسماء Salim ب الاسم Salima

حذف سجل

لحذف سجل نستعمل جملة الاستعلامات المهيكلة Delete و التي تتطلب تحديد اسم الجدول و الشرط



VB.Net Code

```
Dim SQL As String = "DELETE FROM Students WHERE num = " &
    TextBox1.Text
Dim cmd As New OleDbCommand(SQL, con)
If MsgBox("السجل حذف في ترغب هل", MsgBoxStyle.YesNo) =
    MsgBoxResult.Yes Then
    cmd.ExecuteNonQuery()
End If
```

البحث عن السجل

نستعمل جملة الاستعلامات المهيكلة Select مع تحديد شرط يتم من خلاله تصفية السجلات المرغوبة.



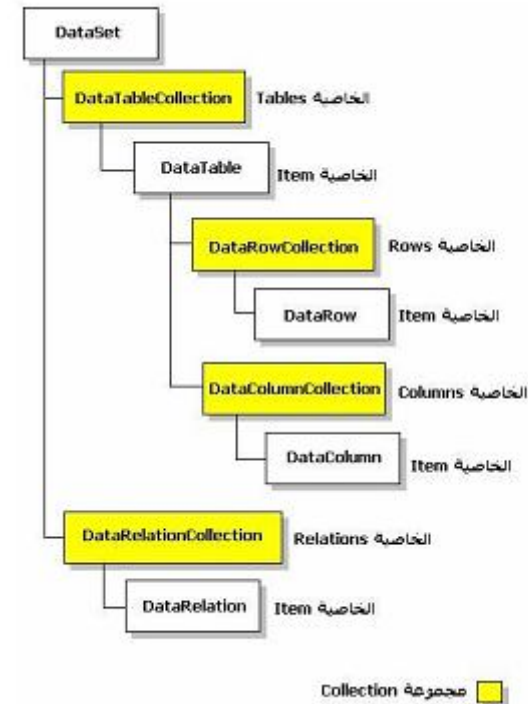
VB.Net Code

```
Dim XPrenom As String = InputBox("الاسم ادخل")
If XPrenom = "" Then Exit Sub
```

```
Dim SQL As String = "Select * From Students " & _
"where Prenom =" & XPrenom & "" '
Da = New OleDbDataAdapter(SQL, con) ' إنشاء كائن موصل البيانات
Da.Fill(Ds, "Students") ' تحميل مجموعة البيانات
' عرض البيانات حسب التصفية المحددة في صناديق النصوص
TextBox1.DataBindings.Add("Text", Ds, "Students.Num")
TextBox2.DataBindings.Add("Text", Ds, "Students.Prenom")
TextBox3.DataBindings.Add("Text", Ds, "Students.Classe")
' عرض البيانات المصفاة في جدول
DataGridView1.DataSource = Ds.Tables("Students")
```

التعامل مع كائن البيانات Data Set

يمنحك ADO.Net ميزة هامة يمكنك من التعامل مع قاعدة البيانات بعد غلق الاتصال Disconnected Mode، وهذا بتخزين البيانات في كائن البيانات DataSet فهو نسخة من قاعدة بيانات مصغرة يمكنك من التعامل مع البيانات وإدارتها كالإضافة و التعديل و الحفظ و الحذف ... دون أن تؤثر على قاعدة البيانات الأصلية . أما إذا أردت حفظ التغييرات ما عليك إلا أن تعيد الاتصال من جديد ثم تحديثها .
مكوناته :



من الشكل يمكن تقسيم مجموعة البيانات إلى مجموعتين

3- مجموعة DataTableCollection: و تتكون من

جدول البيانات DataTable: يمثل جدول من قاعدة البيانات

صفوف البيانات DataRow: تمثل صفوف أو سجلات جدول

عمود البيانات DataColumn: يمثل عمود أو حقل من جدول

الفصل الثامن قواعد البيانات

4- مجموعة DataRelationCollection و تتكون من

DataRelation : فئة تختص بربط بين الجداول عن طريق العلاقات

إنشاء و تعريف كائن البيانات **Crate Data Set**



Syntax

```
Dim Variable As New DataSet([DataSetName])
```

Or

```
Dim Variable As DataSet = New DataSet([DataSetName])
```

مثال



VB.Net Code

```
Dim Ds As New DataSet("Universty")
```

خصائصه

الخاصية	الشرح
Clear	مسح محتوى كائن البيانات DataSate
Copy	نسخ كائن البيانات إلى كائن بيانات آخر <code>Dim copyDataSet As DataSet = dataSet.Copy()</code>
DataSetName	تعود باسم مجموعة البيانات DataSet <code>Dim Ds As New DataSet("Univesty")</code> <code>MsgBox(Ds.DataSetName)</code>
Tables	تعود بمجموعة من الخصائص سنتعرف عليها لاحقاً فقط نكتفي باختيار جدول عن طريق اسمه أو رقم ترتيبه <code>MsgBox(Ds.Tables(0).TableName)</code>
Dispose	تنظيف مجموعة البيانات من الذاكرة <code>ds.Dispose()</code>
Relation	لإنشاء علاقة بين الجداول

أهم خصائص مجموعة الجداول التابعة لكائن DataSet

الخاصية	الشرح	مثال
Add	إضافة جدول	<code>Dim dt As New DataTable("Students")</code> <code>Ds.Tables.Add(dt)</code>
AddRange	إضافة مصفوفة من الجداول إلى DataSet	<code>Dim Table As New DataTable("Students")</code> <code>Dim Table2 As New DataTable("Score")</code> <code>ds.Tables.AddRange(New DataTable() {Table, Table2})</code>
CanRemove	إمكانية الحذف أم لا	<code>ds.Tables.Clear()</code>
Clear	تحذف كل الجداول الموجودة في مجموعة البيانات	<code>ds.Tables.Count.ToString()</code>
Count	عدد الجداول	

الفصل الثامن قواعد البيانات

MsgBox(Ds.Tables.Item(0).ToString)	تحديد اسم الجدول	Item
If Ds.Tables.Contains("Students") Then MsgBox("الجدول موجود") End If	هل الجدول موجود في DataSet	Contains
Ds.Tables.Remove(dt)	حذف جدول	Remove
	مماثلة للوسيلة السابقة، ولكنها تستقبل كعامل، رقم الجدول في مجموعة الجداول.	RemoveAt
	لنسخ الجداول من مجموعة الجداول إلى مصفوفة	CopyTo
	تستقبل هذه الوسيلة معاملا عبارة عن كائن جدول DataTable، وتعيد رقمه في مجموعة الجداول	IndexOf

إضافة جداول إلى كائن البيانات Adding Tables To Data Set



Syntax

```
Dim Table As New DataTable(tableName) 'الجدول إنشاء
Ds.Tables.Add(Table) 'إضافة الجدول إلى Data Set
Or
Dim Table As DataTable = Ds.Tables.Add(tablename)
```

VB.Net Code

```
Dim Ds As New DataSet("Universty")
Dim StdTable As DataTable = Ds.Tables.Add("Students") 'إضافة جدول الطلاب

Dim SCorTable As New DataTable("Score")
Ds.Tables.Add(SCorTable) 'إضافة جدول النتائج
```

عرض الجداول

و الآن تعال لنتأكد هل تم إضافة الجدولين أم لا ؟

VB.Net Code

```
Dim tb As DataTable
For Each tb In Ds.Tables
    Console.WriteLine(tb.TableName)
Next
Or
For I As Integer = 0 To Ds.Tables.Count - 1
    Console.WriteLine(Ds.Tables(I).TableName)
Next
```

نعم تم إضافة الجدولين و يتم عرض أسمائهم

قبل إضافة الحقول إلى الجدول نتطرق إلى الفئة Data Column التي تمكننا من إضافة الحقول و تعرفنا أيضا عن خصائص الحقول

الفصل الثامن قواعد البيانات

خصائص الحقول

إذا جعلت قيمة هذه الخاصية True فسيُسمح للمستخدم بترك أيّ خانة في هذا العمود فارغة.	AllowDBNull
إذا جعلت قيمة هذه الخاصية True، فسيصبح هذا العمود ترقيماً تلقائياً للسجلات.	AutoIncrement
تقرأ أو تُغيّر أصغر رقم يبدأ به الترقيم في العمود الذي يعمل كترقيم تلقائيّ للجدول.	AutoIncrementSeed
تحدّد مقدار الزيادة في عمود الترقيم التلقائيّ.	AutoIncrementStep
عنوان العمود.	Caption
اسم العمود.	ColumnName
نوع بيانات العمود.	DataType
القيمة الافتراضية لخانات العمود.	DefaultValue
صيغة العمود..	Expression
أقصى طول (بالبايت) لبيانات خانات العمود (لن تفيد مع البيانات الرقمية).	MaxLength
ترتيب العمود في مجموعة الأعمدة.	Ordinal
إذا جعلت قيمة هذه الخاصية True فلن يتمكّن المستخدم من تغيير قيمة الخانة بعد إضافتها للعمود.	ReadOnly
تعيد كائن الجدول الذي ينتمي إليه هذا العمود.	Table
إذا جعلت قيمة هذه الخاصية True فلن يسمح بتكرار قيم خانات هذا العمود.	Unique

إنشاء حقل و إضافته إلى الجدول Adding Columns



Syntax

Dim Colm As New DataColumn (columnName ,dataType,Expressions)

(تعبير , نوع بيانات الحقل , اسم الحقل)

Or

Dim Colm As New DataColumn

Colm.ColumnName = "الحقل اسم"

Colm.DataType = GetType(البيانات نوع)

.

.

(أخرى) خصائص

مثال

VB.Net Code

Dim NColm As New DataColumn

With NColm

.ColumnName = "Num" ' اسم الحقل'
 .DataType = GetType(Integer) ' نوع الحقل'
 .AllowDBNull = False ' السماح بنص فارغ'
 .AutoIncrement = True ' ترقيم تلقائي'
 .AutoIncrementSeed = 1 ' بداية الترقيم'
 .AutoIncrementStep = 1 ' مقدار الزيادة'
 .Unique = True ' مطلوب'

End With

StdTable.Columns.Add(NColm) ' إضافة الحقل إلى الجدول'

إضافة حقول أخرى

```
Dim ColPrenom As New DataColumn("Prenom", GetType(String)) ' حقل الاسم
Dim ColSex As New DataColumn("Sex", GetType(String)) ' حقل الجنس
ColSex.AllowDBNull = False ' السماح بنص فارغ
ColSex.Unique = False ' مطلوب
StdTable.Columns.AddRange(New DataColumn() {ColPrenom, ColSex}) '
إضافة الحقلين إلى جدول الطلاب
StdTable.PrimaryKey = New DataColumn() {NColm} ' جعل حقل الاسم مفتاحاً أساسياً
```

والآن دعنا نتأكد من الحقول المضافة

MsgBox(StdTable.Columns(1).ColumnName) ' قراءة الحقل 2 الاسم

عرض كل حقول الجدول

نقوم نعرض كل حقول جدول الطلاب StdTable

VB.Net Code

```
Dim col As DataColumn
For Each col In StdTable.Columns
    Console.WriteLine(col.ColumnName)
Next
Or
For i As Integer = 0 To StdTable.Columns.Count - 1
    Console.WriteLine(StdTable.Columns(i).ColumnName)
Next
```

النتائج

Num
Prenom
Sex

تدريب نموذجي :

1. قم بإضافة خمسة حقول لجدول النتائج ScoreTable حسب الجدول التالي

الفصل الثامن قواعد البيانات

اسم الحقل	نوعه
ID	Integer
Prenom	String
Arabe	decimal
Math	Decimal
Moyen	decimal

الحل :

VB.Net Code

```
Dim Numcol As New DataColumn("Id", GetType(Integer)) ' حقل الرقم
Dim prenomcol As New DataColumn("Prenom", GetType(String)) ' حقل الاسم
Dim arabecol As New DataColumn("Arabe", GetType(Decimal)) ' حقل العربية
Dim Mathcol As New DataColumn("Math", GetType(Decimal)) ' رياضيات
Dim Moyencol As New DataColumn("Moyen", GetType(Decimal),
"((Arabe*2)+(Math)*5)/7") ' حقل المعدل يحتوي على تعبير
'Moyencol.Expression = "((Arabe*2)+(Math)*5)/7"
SCorTable.Columns.AddRange(New DataColumn() {Numcol, prenomcol,
arabecol, Mathcol, Moyencol}) ' إضافة الحقول إلى جدول النتائج
```

2. قم بعرض حقول جدول النتائج

VB.Net Code

```
For Each col As DataColumn In SCorTable.Columns
Console.WriteLine(col.ColumnName)
Next
Id
Prenom
Arabe
Math
Moyen
```

بعد أن أنشأنا الجداول لقاعدة البيانات المصغرة Data Set و أضفنا الحقول أو الأعمدة إلى هذه الجداول يبقى أن نضيف صفوفًا إليه .

قبل أن نضيف الصفوف لا بد أن نتعرف على الفئة المختصة بكائن الصف Data Row

كائن الصف DataRow

يمثل أحد صفوف الجدول وهذه أهم خصائصه

تقرأ البيانات الموجودة في صف محدد بعمود محدد	Item
تقرأ أو تغيّر كلّ خانة هذا السجلّ من خلال وضع القيم في مصفوفة.	ItemArray
تقرأ حالة السجلّ من حيث كونه مضافًا أو محذوفًا أو معدّلًا... إلخ	RowState

الفصل الثامن قواعد البيانات

تعيد كائن الجدول الذي يحتوي على هذا السجل.. فإذا لم يكن هذا السجل منتمياً لجدول حتى الآن، فإنها تعيد كائن جدول، يماثل هذا السجل (له نفس تركيب سجلاته: نفس الحقول والقيم).	Table
---	-------

وسائله

تحفظ التغييرات التي حدثت بالسجل، بحيث لا يمكن التراجع عنها بعد ذلك، إلا إذا أعاد المستخدم تحميل الجدول من قاعدة البيانات.	AcceptChanges
تراجع عن التغييرات التي حدثت بالسجل، بحيث يعود لآخر وضع تم حفظه عليه.	RejectChanges
تعيد نصاً يصف الخطأ الذي حدث في أحد أعمدة هذا السجل.	GetColumnError
تسمح لك بوضع نص يصف الخطأ الذي حدث في أحد أعمدة هذا السجل.	SetColumnError
تعيد مصفوفة تحتوي على الأعمدة التي بها أخطاء في هذا السجل.	GetColumnsInError
تمحو كل المعلومات التي تشير إلى حدوث أخطاء في السجل (لا يعني هذا إجراء أي تعديل في البيانات.. فقط سيبدو أن السجل قد فقد ذاكرته بخصوص الأخطاء التي حدثت، و... "عفا الله عما سلف").	ClearErrors
تحذف السجل.	Delete
تعيد مصفوفة بها كل السجلات المرتبطة بعلاقة بهذا السجل في جداول أخرى.	GetChildRows
تعيد السجل الذي يرتبط به هذا السجل بعلاقة (تعيد السجل الموجود في الجدول الرئيسي).	GetParentRow
تعيد مصفوفة تحتوي على كل السجلات الرئيسية من الجداول الأخرى، المرتبطة بعلاقة بهذا السجل.	GetParentRows
تعيد True إذا كانت فارغة تلك الخانة المحددة بالسجل الحالي والعمود المرسل كمعامل.	IsNull
تسمح لك بتحديد قيم جديدة للسجل الرئيسي في الجدول الذي يرتبط معه هذا السجل بعلاقة.	SetParentRow

أهم خصائص الصف التابعة لكائن DataTable (DataTable.Rows)

الخاصية	الشرح	مثال
Add	إضافة سجل أو صف	<pre>Dim dr As DataRow = dt.NewRow dr(0) = 1 ' الصف الأول الخانة الأولى dr(1) = "a" ' الصف الأول الحقل الثاني dt.Rows.Add(dr)</pre>
Clear	مسح محتوى صف	<pre>dt.Rowe.Clear()</pre>

الفصل الثامن قواعد البيانات

MsgBox(dt.Rows.Count.ToString)	عدد الصفوف في الجدول	Count
MsgBox(dt.Rows(0).Item(1).ToString)	تحديد اسم الصف	Item
dt.Rows.Remove(Rows)	حذف الصف	Remove
If dt.Rows.Contains("Name") Then MsgBox("موجود الحقل") End If	هل السجل موجود في الجدول	Contains

إنشاء كائن الصف وإضافته إلى الجدول

لن تتمكن من إنشاء كائن الصف باستعمال كلمة New التابعة لفئة DataRow . يمكننا إنشاؤها بطريقتين

1. باستعمال الطريقة DataRow التابع للفئة



Syntax

```
Dim Row As DataRow = Table.AddNew  
Row(ColumnIndex) = "....."
```

2. تعريف مصفوفة من نوع Object تمثل قيم حقول في الصف ثم إضافتها إلى الخاصية Rows

التابعة للكائن DataTable



Syntax

```
Dim Row As DataRow = New Object () {Col1,Col2,...}  
Table.Rows(Row)
```

مثال : عن الكيفية الثانية

VB.Net Code

```
Dim Row As Object() = {1, "Salah", "ذ"}  
StdTable.Rows.Add(Row)  
Or  
StdTable.Rows.Add(New Object() {1, "Salah", "ذ"})
```

مثال : عن الكيفية الأولى

VB.Net Code

```
Dim Row As DataRow = StdTable.NewRow  
Row(0) = 2  
Row("Prenom") = "Ali"  
Row(2) = "ذ"
```

و الآن تعال نتعرف كيف نقرأ هذه الصفوف أو السجلات السابقة

VB.Net Code

```
Dim R As DataRow  
For Each R In StdTable.Rows  
Console.WriteLine(R(1)) ' عرض سجلات العمود 1 الاسم  
Next  
Or  
For i As Integer = 0 To StdTable.Rows.Count - 1  
Console.WriteLine(StdTable.Rows(i)("Prenom"))  
Next
```

Salah
Ali

تدريب نموذجي :

قم بتحميل جدول النتائج بالسجلات . دون تحميل عمود المعدل لأن هذا الحقل يحتوي على تعبير يملأ ألياً
الحل :

 VB.Net Code

```
SCorTable.Rows.Add(New Object() {1, "Salah", 14, 13.5})
SCorTable.Rows.Add(New Object() {2, "Ali", 13, 13})
```

```
For Each RW As DataRow In SCorTable.Rows
    Console.WriteLine("Prenom {0} , Moyen {1} ", RW(1), RW(4))
Next
```

```
Prenom Salah , Moyen 13,64
Prenom Ali , Moyen 13
```

إضافة سجلات عن طريق الحلقة

 VB.Net Code

```
Dim Names() As String = {"Saaid", "Nadia"}
For i As Integer = 0 To UBound(Names)
    Dim dr As DataRow = StdTable.NewRow
    dr(0) = i
    dr(1) = Names(i)
    StdTable.Rows.Add(dr)
    ListBox1.Items.Add(dr(0) & " " & dr(1))
Next
```

صيغة العمود Expression

رأينا كيف استعمالنا التعبير (الصيغة) Expression مع حقل المعدل في جدول النتائج و تستعمل كثيرا في العمليات الحسابية بين حقلين في نفس الجدول كما في المثال السابق

```
Dim Moyencol As New DataColumn("Moyen", GetType(Decimal),
```

```
"((Arabe*2)+(Math)*5)/7") ' حقل المعدل يحتوي على تعبير
```

```
'Moyencol.Expression = "((Arabe*2)+(Math)*5)/7"
```

كما يمكنك استعمال الوسيلة Expression في عرض سجلات موافقة لشرط معين

```
Dim cl As New DataColumn("x", GetType(Decimal), MappingType.Hidden)
```

```
cl.Expression = "Moyen < 10"
```

تطبيق دوال التجميع على العمود

نستعمل الوسيلة Compute التابعة لكائن الجدول DataTable و تعيد عدد يمثل عدد السجلات الموافقة للشرط المحدد.

 Syntax

```
Dim X As Integer = TableName.Compute(Exception, Filter)
```

المعامل الأول Expression يمثل العملية الحسابية

المعامل الثاني Filter الشرط المرافق للعملية الحسابية

 VB.Net Code

```
Dim X As Integer = SCorTable.Compute("Count(Id)", "Moyen > 10")  
MsgBox(X) ' 2
```

إنشاء علاقة بين الجداول Creating Relation

يمكنك إنشاء العلاقات بين الجداول باستعمال الفئة DataRelation التابعة للجدول اسند الجدول الأساسي مع الخاصية parentColumns و كائن الجدول التابع في الخاصية ChildColumns و لا تنسى اسم العلاقة في الخاصية Relation Names ثم إضافة إلى مجموعة البيانات DataSet



Syntax

```
Dim rel As New DataRelation(relationName ,parentColumn ,childColumn )  
DS.Relations.Add(rel)
```

مثال : ننشئ علاقة بين جدول الطلاب و النتائج



```
Dim rel As New DataRelation("R", StdTable.Columns("Num"), _  
SCorTable.Columns("Id"))  
Ds.Relations.Add(rel)
```

يمكنك عرض جميع العلاقات الموجودة في مجموعة البيانات كما يلي



```
For Each r As DataRelation In Ds.Relations  
Console.WriteLine(r.RelationName)  
Next
```

ملخص :



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
Dim Ds As New DataSet 'جديدة بيانات مجموعة إنشاء'  
Dim Employees As New DataTable("الموظفين") 'الموظفين جدول إنشاء'  
Dim Sales As New DataTable("المبيعات") 'المبيعات جدول إنشاء'  
الموظفين لجدول حقول إضافة'  
With Employees.Columns.Add("المعرف", GetType(Integer))  
.AutoIncrement = True  
.AutoIncrementSeed = 1  
.AllowDBNull = False  
.Unique = True  
End With  
With Employees.Columns.Add("الاسم", GetType(String))  
.MaxLength = 100  
End With  
المبيعات لجدول حقول إضافة'  
Sales.Columns.Add("الرقم", GetType(Integer))  
Sales.Columns.Add("تقدي دفع", GetType(Boolean))  
Sales.Columns.Add("المبلغ", GetType(Decimal))
```

```

صيغة على يحتوي حقل إنشاء'
Sales.Columns.Add("تخفيض", GetType(Decimal), "(المبلغ*5)/100")
الموظفين جدول من المعرف لحقل أساسي مفتاح إضافة'
Employees.PrimaryKey = New DataColumn() {Employees.Columns("المعرف")}

Data set إلى الجدولين إضافة'
Ds.Tables.Add(Employees)
Ds.Tables.Add(Sales)
أو'
'Ds.Tables.AddRange(New DataTable() {Employees, Sales})
الجدولين بين علاقة إضافة'
Ds.Relations.Add("Rel", Employees.Columns("المعرف"), _
    Sales.Columns("الرقم"))
الجدولين تعبئة و الصفوف إضافة'
Employees.Rows.Add(New Object() {1, "الصالح"})
Employees.Rows.Add(New Object() {2, "علي"})

Dim dr As DataRow = Sales.NewRow
dr(0) = 1
dr(1) = True
dr(2) = 1450
Sales.Rows.Add(dr)
Sales.Rows.Add(New Object() {2, True, 1540})
العروض'
DataSet في الموجودة الجداول عرض'
For Each Tb As DataTable In Ds.Tables
    Console.WriteLine(Tb.TableName)
Next
المبيعات جدول حقل عرض'
For i As Integer = 0 To Sales.Columns.Count - 1
    Console.WriteLine(Sales.Columns(i).ColumnName)
Next
الموظفين جدول حقل عرض'
For Each Col As DataColumn In Employees.Columns
    Console.WriteLine(Col.ColumnName)
Next
الموظفين جدول صفوف عرض'
For Each drow As DataRow In Employees.Rows
    Console.WriteLine(drow(1)) ' الاسم عمود سجلات عرض'
Next
التخفيض بعمود الخاصة و المبيعات جدول سجلات عرض'
For i As Integer = 0 To Sales.Rows.Count - 1
    Console.WriteLine(Sales.Rows(i)("تخفيض"))
Next
End Sub
    
```

الفصل الثامن قواعد البيانات

التعامل مع كائن بيانات الجدول Data Table

يمثل جدولا من جداول قاعدة البيانات و يشتمل على خصائص و وسائل كثيرة أهم خصائصه :

تعيد مجموعة العلاقات المتولدة من هذا الجدول (التي يدخل فيها كجدول رئيسي Master).	ParentRelations
تعيد مجموعة العلاقات الواصلة لهذا الجدول (التي يدخل فيها كجدول ثانوي Detail).	ChildRelations
تعيد مجموعة الأعمدة (الحقول) الموجودة في هذا الجدول.	Columns
تعيد مجموعة الصفوف (السجلات) الموجودة في هذا الجدول..	Rows
تعيد مجموعة القيود (الشروط) الموجودة في هذا الجدول.	Constraints
تعيد كائن مجموعة بيانات DataSet Object، يشير إلى مجموعة البيانات التي ينتمي إليها هذا الجدول.	DataSet
تعيد هذه الخاصية كائن العرض View الذي يمثل البيانات التي سيتم عرضها و تصفيتها وفق شرط معين.	DefaultView
تسمح لك هذه الخاصية بقراءة أو تغيير اسم الجدول.	TableName
تمثل هذه الخاصية النص الذي سيتم عرضه كعنوان للجدول في أدوات واجهة المستخدم (مثل جدول البيانات DataGridView).	DisplayExpression
تسمح لك هذه الخاصية بتحديد السعة المبدئية (أصغر حجم) للجدول بالبايت.	MinimumCapacity
هذه الخاصية عبارة عن مصفوفة من كائن الأعمدة DataColumn التي تريد استخدامها كمفتاح أساسي للجدول...	PrimaryKey

أهم وسائله

تمحو كل البيانات الموجودة في الجدول.	Clear
تستنسخ بنية الجدول بكل ما فيه من مخططات Schemas وقيود Constraints.. ولكن الجدول الناتج يكون فارغا.	Clone
تتنسخ الجدول ببنيته وبياناته، لتنتج جدولا مماثلا له تماما.	Copy
تنشئ صفاً جديداً له نفس تركيب صفوف الجدول، وتعيده لك، لكن دون إضافته للجدول (عليك أن تضيفه أنت بنفسك).	NewRow
تعيد مصفوفة تحتوي على كل صفوف الجدول. وتقوم أيضا بتصفية السجلات و وفق شرط محدد.	Select
تحدد الحقول التي ينطبق عليها مواصفات (شروط) معينة، وتجري عليها إحدى دوال التجميع Aggregate Function..	Compute

الفصل الثامن قواعد البيانات

نتعلم مع كائن الجدول كيف نتعامل مع قواعد البيانات و إدارتها من حيث الإضافة و التعديل و التصفية و البحث و عرض البيانات و الحذف ...
نبدأ أولاً بإنشاء قاعدة بيانات عن طريق Access حسب الجدول التالي

Num	Prénom	Classe
1	Salah	1
2	Ali	2
3	Salim	1
4	Nabil	2

ننشئ مشروعاً جديداً و نحفظ فيه قاعدة البيانات في مجلد Bin/Debug
نقوم بتصميم البرنامج على النحو التالي:



كتابة الكود و التعليمات

نبدأ باستيراد مجال الأسماء الخاص بقواعد البيانات Access في الأعلى كما يلي

```
Imports System.Data.OleDb
```

ثم نكتب في قسم التصريحات العامة للنموذج المتغيرات العامة

 VB.Net Code

نص الاتصال '

```
Dim StrCon As String = "provider = Microsoft.jet.oledb.4.0;" & _  
"data source = " & Application.StartupPath & "/db1.mdb"
```

تحميل كائن الاتصال بنص الاتصال ' Dim Con As New OleDbConnection(StrCon)

تعريف مجموعة البيانات ' Dim Ds As New DataSet

الفصل الثامن قواعد البيانات

متغير لتحديد موقع السجل ' -1 Dim X As Integer =

تعريف متغير خاص بالجدول ' Dim Dt As New DataTable

' تحميل محول البيانات باسم الجدول و كائن الاتصال

Dim da As New OleDbDataAdapter("Select * from Students", Con)

ملاحظة : الدالة Application.StartupPath تعيد لنا اسم المجلد الحالي

أما في حدث تحميل النموذج نكتب :



VB.Net Code

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
```

```
Try
```

```
Con.Open() ' فتح الاتصال
```

```
da.Fill(Ds, "Students") ' تحميل مجموعة البيانات باسم الجدول
```

```
Dt = Ds.Tables("Students") ' اسناد متغير الجدول باسم الجدول
```

```
loadTxt() ' إجراء خاص بعرض محتوى الحقول في مربع النصوص
```

```
DataGridView1.DataSource = Dt ' تحميل جدول العرض بجدول الطلاب
```

```
Catch ex As Exception
```

```
MsgBox(ex.Message) ' في حالة الخطأ تظهر رسالة الخطأ
```

```
End Try
```

```
End Sub
```

أما الآن فنكتب إجراء عرض محتوى الحقول في مربعات النصوص



VB.Net Code

```
Sub loadTxt()
```

```
On Error Resume Next ' تجاهل الأخطاء
```

```
TextBox1.Text = Dt.Rows(X)(0) ' عرض محتوى الحقل 1
```

```
TextBox2.Text = Dt.Rows(X)(1) ' عرض محتوى الحقل 2
```

```
TextBox3.Text = Dt.Rows(X)(2) ' عرض محتوى الحقل 3
```

```
Position() ' إجراء عرض موقع السجل
```

```
End Sub
```

إجراء عرض الموقع



VB.Net Code

```
Sub Position()
```

```
Me.Text = "رقم مسجل : " & X & "/" & Dt.Rows.Count - 1
```

```
End Sub
```

هذا الإجراء يعرض موقع السجل الحالي في عنوان النموذج بحيث

رقم السجل = موقع السجل X / عدد السجلات

زر السجل التالي :

 VB.Net Code

```
If X > Dt.Rows.Count - 1 Then ' في حالة الوصول إلى آخر سجل '
    MsgBox("سجل آخر") ' يعرض رسالة '
    Exit Sub
Else ' وإلا '
    X += 1 ' انتقل خطوة إلى الأمام '
    loadTxt() ' عرض السجلات في مربعات النصوص '
End If
```

زر السجل السابق

 VB.Net Code

```
If X = 0 Then ' في حالة الوصول إلى السجل الأول '
    MsgBox("مسجل أول", MsgBoxStyle.Information) ' يعرض رسالة '
Else ' وإلا '
    X -= 1 ' انتقل خطوة إلى الخلف '
    loadTxt() ' إجراء العرض '
End If
```

زر السجل الأول

 VB.Net Code

```
X = 0
loadTxt()
```

زر السجل الأخير

 VB.Net Code

```
X = Dt.Rows.Count - 1
loadTxt()
```

السجل الأخير = عدد كل السجل - 1

زر جديد : يقوم بمسح محتوى مربعات النصوص للكتابة

 VB.Net Code

```
Dim C As New Control ' متغير يمثل أداة '
For Each C In Me.Controls ' أي أداة في النموذج '
    If TypeOf C Is TextBox Then ' وإذا كان نوعها مربع نص '
        C.Text = "" ' امسح محتواها '
    End If
```

زر إضافة سجل جديد

 VB.Net Code

```
Dt.Rows.Add(New Object() {TextBox1.Text, TextBox2.Text, TextBox3.Text})
' نضيف مصفوفة من نوع كائن الخانة الأولى تمثل الحقل 1 أي مافي مربع النص 1 و الخانة 2 تمثل الحقل 2 النص هكذا '
' نضيف كائن تنفيذ الأوامر '
```

الفصل الثامن قواعد البيانات

```
Dim cmb As New OleDbCommandBuilder(da)
```

نفذ أمر الإضافة '

```
da.InsertCommand = cmb.GetInsertCommand
```

قم بحفظ التعديلات '

```
da.Update(Ds, "Students")
```

انقر زر جديد لمسح محتوى مربعات النصوص لإضافة سجل جديد '

```
Button9_Click(Me, e)
```

بكيفية أخرى

```
Dim row As DataRow = Dt.NewRow() تحضير لإضافة سجل جديد'
```

```
row(X)(0) = TextBox1.Text ' الحقل 1 يمثل ما في مربع النص 1
```

```
row(X)(1) = TextBox2.Text ' الحقل 2 يمثل ما في مربع النص 2
```

```
row(X)(2) = TextBox3.Text
```

```
Dt.Rows.Add(row) ' إضافة السجل إلى مجموعة البيانات
```

```
Dim cmb As New OleDbCommandBuilder(da) ' نضيف كائن تنفيذ الأوامر
```

```
da.InsertCommand = cmb.GetInsertCommand ' نفذ أمر الإضافة
```

```
da.Update(Ds, "Students") ' قم بحفظ التعديلات
```

```
Button9_Click(Me, e) ' انقر زر جديد لمسح محتوى مربعات النصوص لإضافة سجل جديد
```

زر تعديل البيانات



VB.Net Code

```
Dt.Rows(X)(0) = TextBox1.Text ' الحقل 1 يمثل ما في مربع النص 1
```

```
Dt.Rows(X)(1) = TextBox2.Text ' الحقل 2 يمثل ما في مربع النص 2
```

```
Dt.Rows(X)(2) = TextBox3.Text
```

```
Dim cmb As New OleDbCommandBuilder(da) ' نضيف كائن تنفيذ الأوامر
```

```
da.InsertCommand = cmb.GetInsertCommand ' نفذ أمر الإضافة
```

```
da.Update(Ds, "Students") ' قم بحفظ التعديلات
```

```
MsgBox("التعديل عملية تمت", MsgBoxStyle.Information)
```

بحث



VB.Net Code

```
Dim Names As String = Textbox4.text
```

نعرف مصفوفة من نوع صف تمثل ناتج الطريقة تحديد التابعة للجدول التي تصفي حقلًا حسب الشرط المحدد '

```
Dim FindRow() As DataRow = Dt.Select("Prénom=" & Names & "'")
```

```
X = x + 1
```

```
TextBox1.Text = (FindRow(X).Item(0)) ' مربع النص =1 سجل الحقل 1
```

```
TextBox2.Text = (FindRow(X).Item(1)) ' مربع النص =2 سجل الحقل 2
```

```
TextBox3.Text = (FindRow(X).Item(2))
```

بكيفية أخرى

```
Dim Found As Boolean = False ' متغير يحدد هل تمت عملية البحث
```

الفصل الثامن قواعد البيانات

```
Dim InputName = InputBox("اسم ادخل", "بحث")
For x As Integer = 0 To (Dt.Rows.Count - 1)
    ' في حالة تساوي قيمة سجل حقل الاسم مع ما تم إدخاله في صندوق الإدخال
    If CStr(Dt.Rows(x)("Prénom")) = InputName Then
        Found = True ' تمت عملية البحث
        TextBox1.Text = Dt.Rows(x).Item("Num") ' مربع النص = سجل حقل الرقم
        TextBox2.Text = Dt.Rows(x).Item("Prénom") ' مربع النص=2 سجل حقل الاسم
        TextBox3.Text = Dt.Rows(x).Item("Classe")
    End If
Next
If Found = False Then ' في حالة فشل عملية البحث
    MsgBox("غير موجود الاسم", MsgBoxStyle.Information) ' يعرض رسالة
End If
```

حذف سجل



VB.Net Code

```
If MsgBox("السجل حذف تريد هل", MsgBoxStyle.Question + MsgBoxStyle.YesNo) =
MsgBoxResult.Yes Then ' في حالة الموافقة
    Dt.Rows(X).Delete() ' يتم حذف السجل الحالي
    Dim cmb As New OleDbCommandBuilder(da) ' نضيف كائن تنفيذ الأوامر
    da.DeleteCommand = cmb.GetDeleteCommand ' نفذ أمر الإضافة
    da.Update(Ds, "Students") ' قم بحفظ التعديلات
    MsgBox("الحذف عملية تمت", MsgBoxStyle.Information)
End If
```

عرض أسماء حقول الجدول



VB.Net Code

```
For i As Integer = 0 To Dt.Columns.Count - 1
    ListBox1.Items.Add(Dt.Columns(i).ColumnName)
Next
Dim Col As New DataColumn
For Each Col In Dt.Columns
    ListBox1.Items.Add(Col.ColumnName)
Next
```

عرض سجلات عمود



VB.Net Code

```
For i As Integer = 0 To Dt.Rows.Count - 1
    ListBox1.Items.Add(Dt.Rows(i)(1))
Next
Dim Row As DataRow
For Each Row In Dt.Rows
    ListBox1.Items.Add(Row.Item(1))
Next
```

عرض البيانات في قائمة ذات أعمدة

 VB.Net Code

```
Dim fmtStr As String = "{0,15}{1,-10}{2,10:N1}"
For i As Integer = 0 To Dt.Rows.Count - 1
    ListBox1.Items.Add(String.Format(fmtStr, Dt.Rows(i)(0), _
    Dt.Rows(i)(1), Dt.Rows(i)(2), Dt.Rows(i)(3)))
Next
```

عرض بيانات حقل في قائمة

 VB.Net Code

```
Listbox1.DataSource = Dt 'Bind the list box to the data table.
Listbox1.DisplayMember = "Prénom" & " " & "Num"
```

حساب عدد طلاب قسم

 VB.Net Code

```
Dim Cls As Integer = InputBox("احصاء", "القسم رقم ادخل")
احسب عدد أرقام الطلاب حسب القسم المحدد الشرط '
MsgBox(Dt.Compute("Count(Num)", "Classe =" & Cls))
```

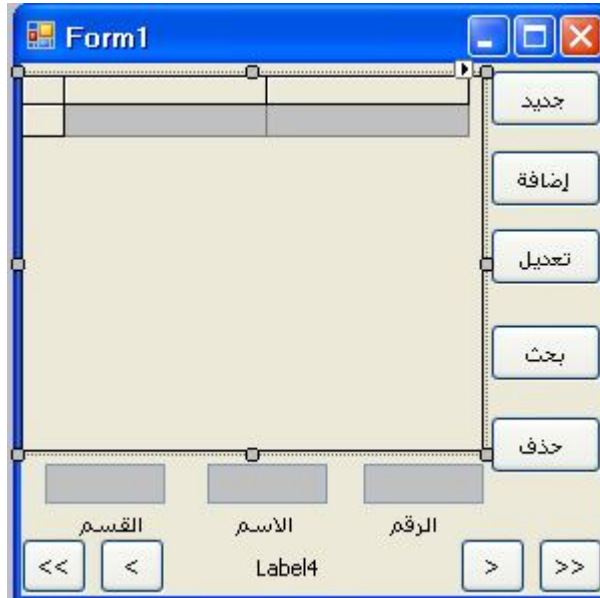
إضافة عمود مخصص للعمليات الحسابية

 VB.Net Code

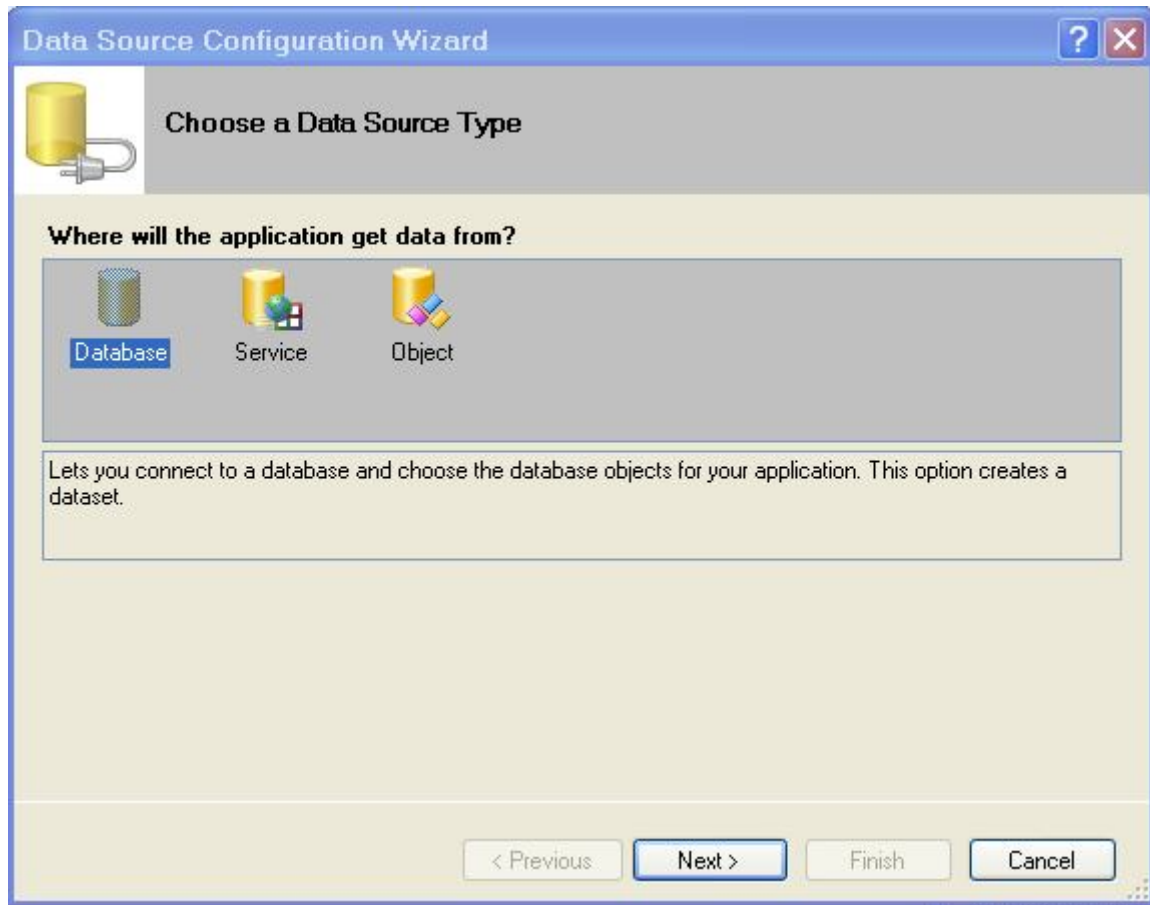
```
Dim ColCal As New DataColumn("Cal", GetType(System.Decimal), "Num*2")
dt.Columns.Add(ColMoyen)
DataGridView1.DataSource = dt
```

التعامل مع DataBinding

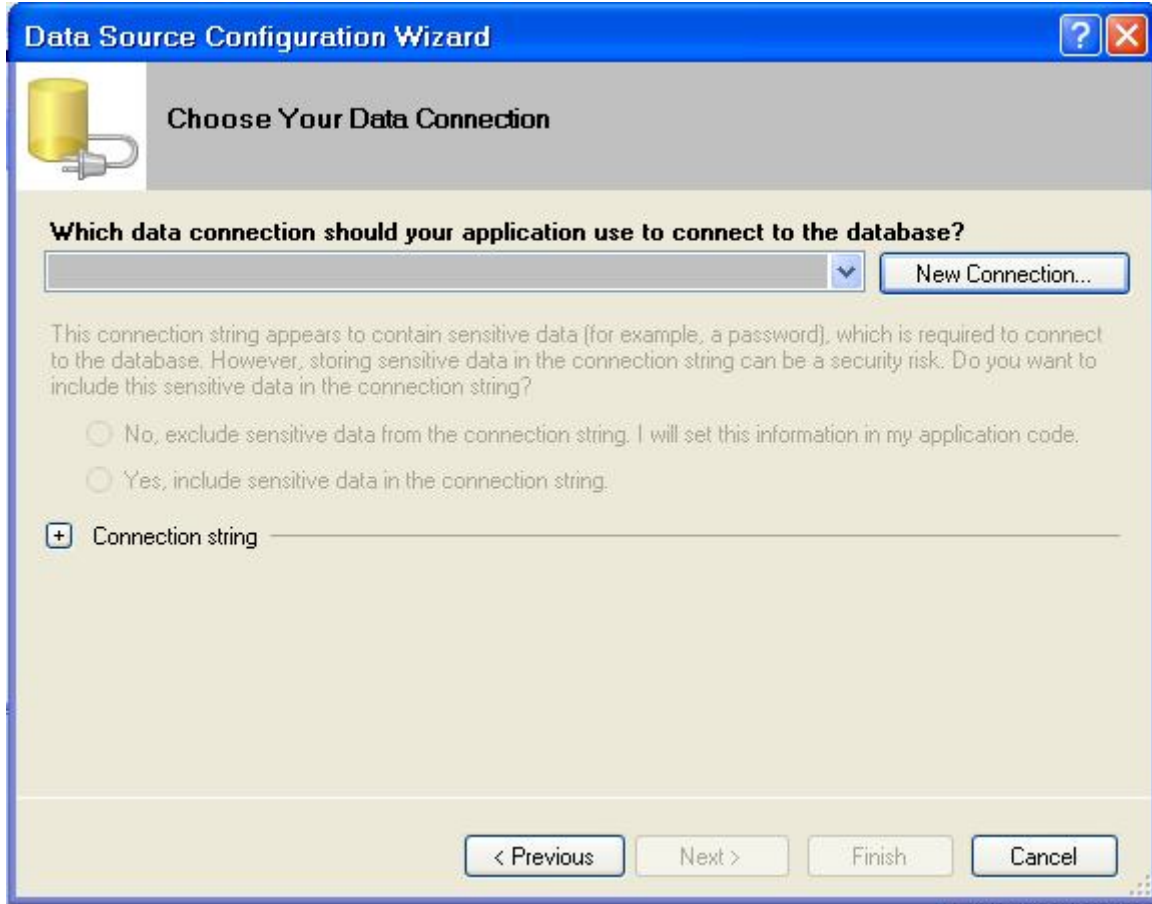
نقوم بإنشاء مشروع جديد ثم نصمم النموذج كما يلي :



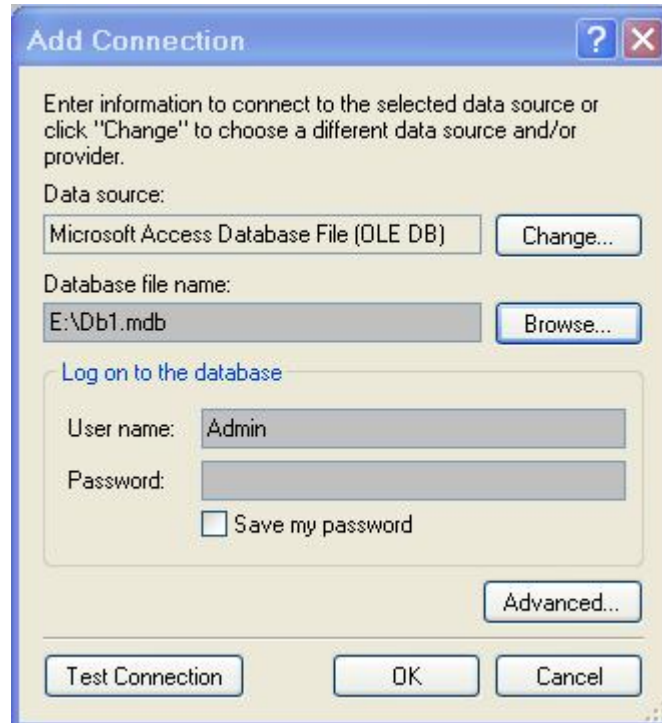
نقوم بإضافة قاعدة البيانات إلى المشروع و هذا من قائمة Data نختار Add New Data Source تظهر النافذة التالية :



انقر Next



انقر New Connection



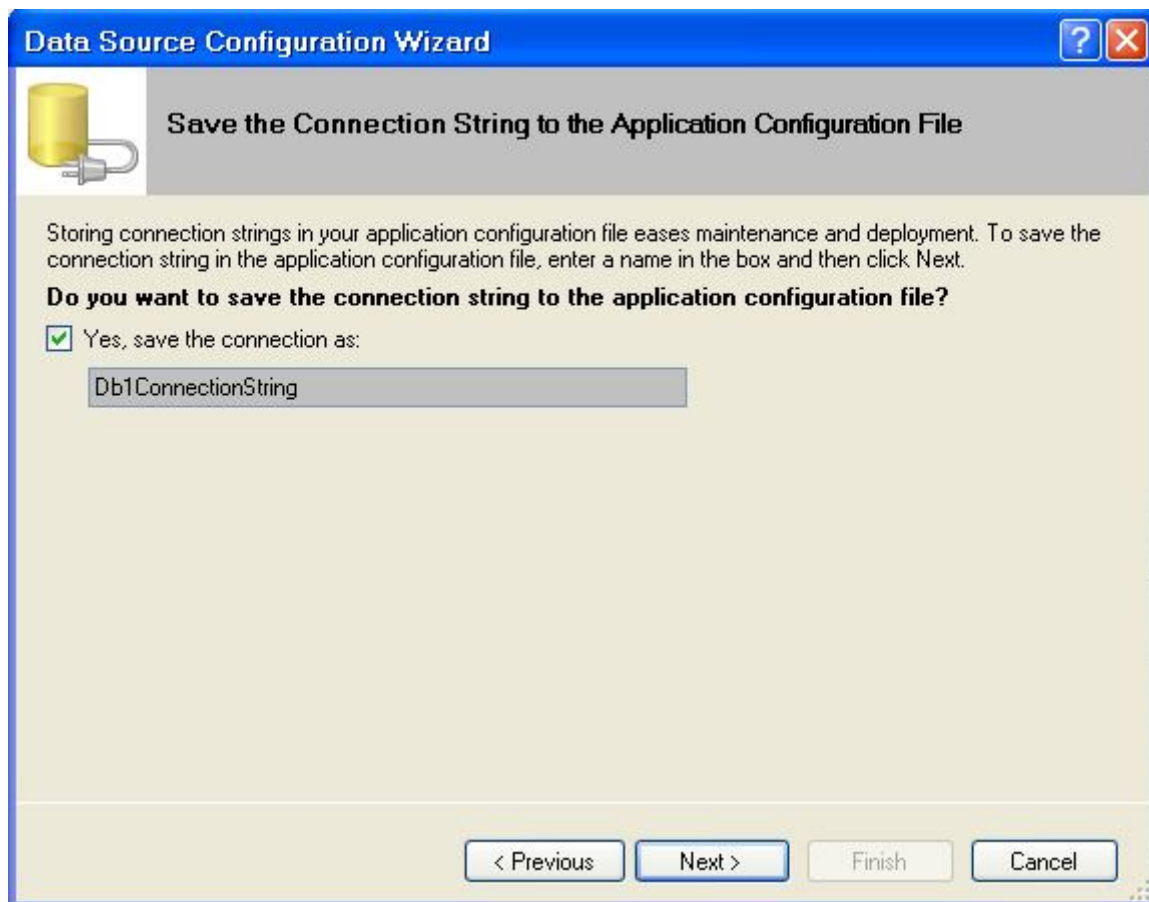
في خانة الاسم نكتب مسار القاعدة أو نختار Browse لتحديد مسار القاعدة
بعد تحديد المسار نضغط Ok



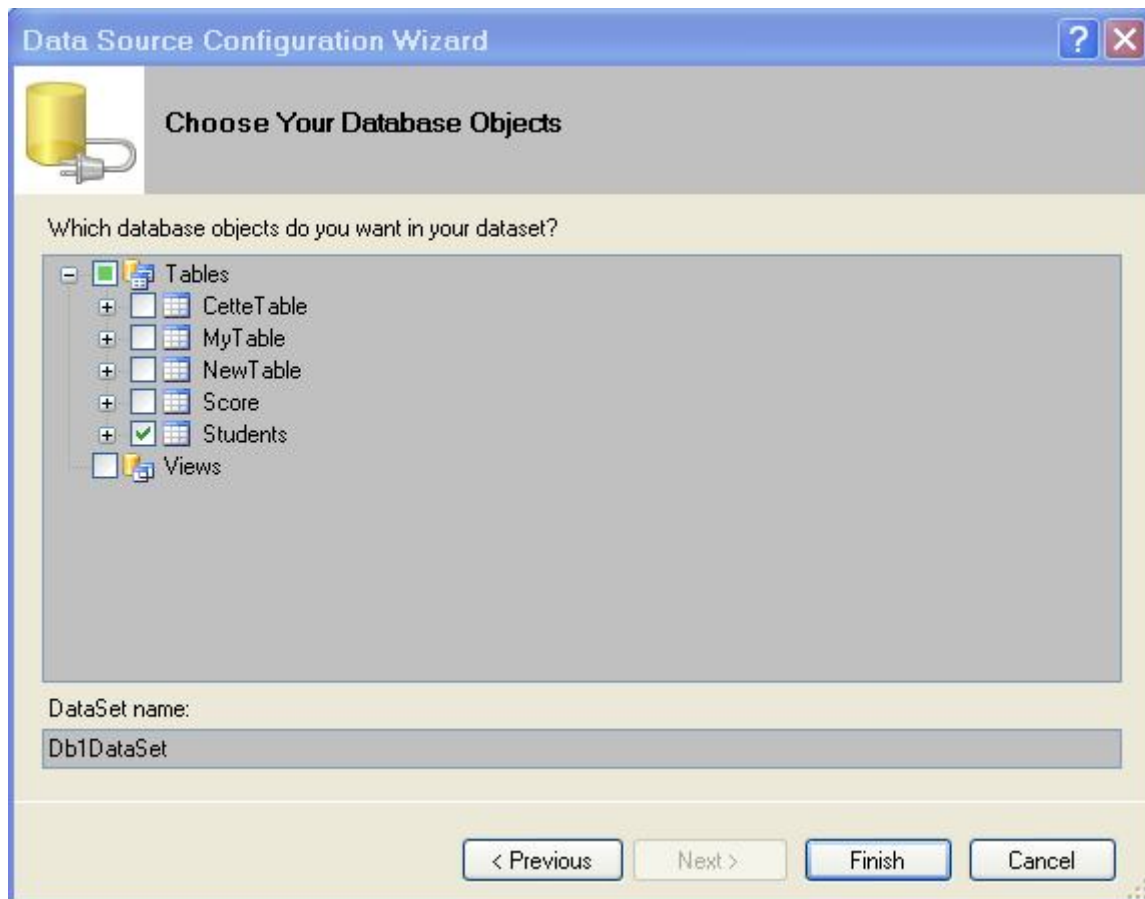
انقر Next



انقر نعم

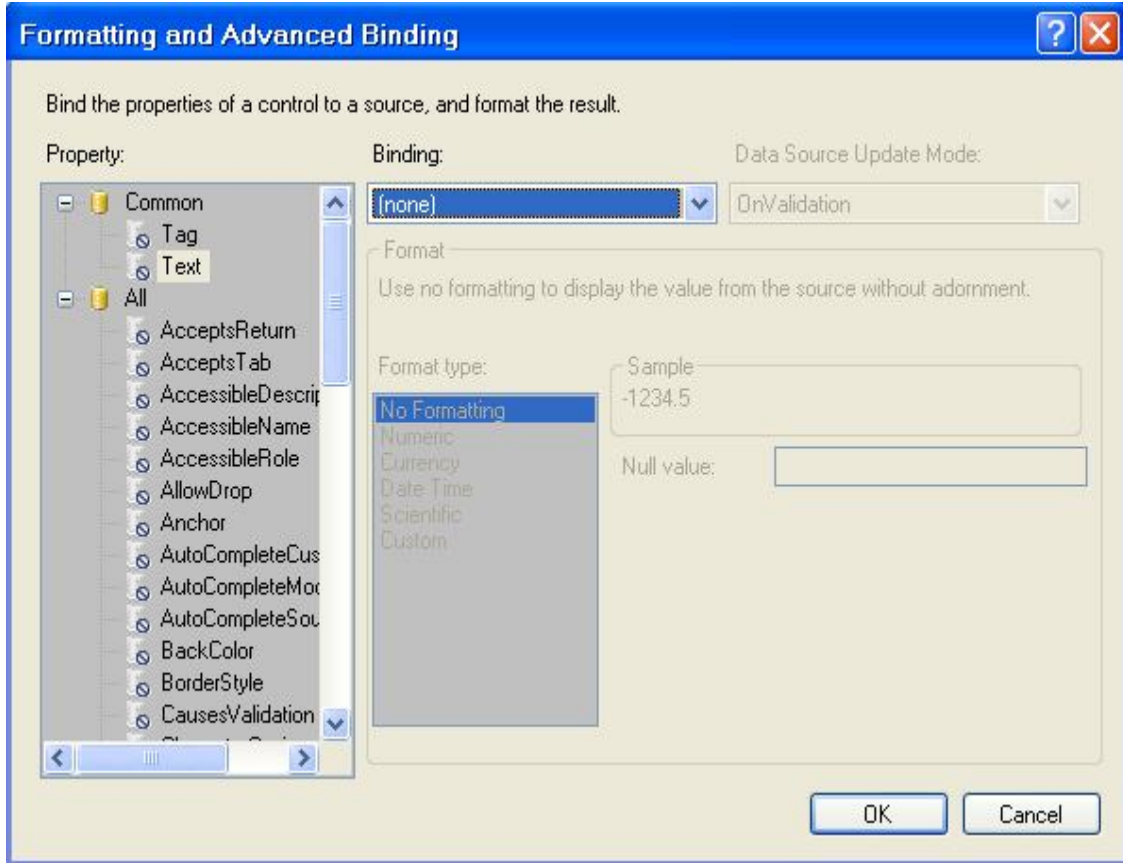


انقر Next

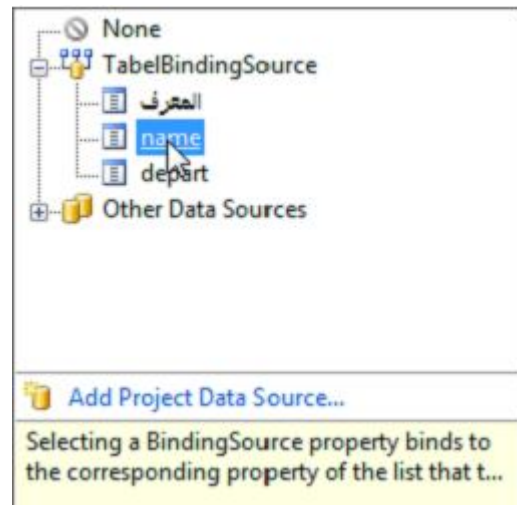


الفصل الثامن قواعد البيانات

اختر الجدول من القائمة وليكن Students ثم انقر Finish و لاحظ أنه تم إضافة القاعدة إلى المشروع و ظهورها في مستكشف الحل
أما الآن نقوم بربط مربعات النصوص بالحقول
نربط مربع النص1 Textbox1 مع الحقل1 (Num)
نختار الخاصية DataBindings لمربع النص Textbox1 ونضغط أمام + ثم نختار Advanced
تظهر النافذة :



اضغط صندوق السرد Binding تظهر النافذة التالية اختر منها الحقل Num ثم اضغط Ok



وبنفس الكيفية نربط مربعات النصوص بالحقول المناسبة.

 Code VB.Net **التالي**

```
StudentsBindingSource.MoveNext()
```

 Code VB.Net **السابق**

```
StudentsBindingSource.MovePrevious
```

 Code VB.Net **الأول**

```
StudentsBindingSource.MoveFirst()
```

 Code VB.Net **الأخير**

```
StudentsBindingSource.MoveLast()
```

العمليات الأساسية

 Code VB.Net **إضافة سجل جديد**

```
StudentsBindingSource.AddNew()
```

 Code VB.Net **تعديل**

```
StudentsBindingSource.EndEdit()  
Me.StudentsTableAdapter.Update(Me.Db1DataSet.Students)  
Me.StudentsTableAdapter.Fill(Me.Db1DataSet.Students)
```

 Code VB.Net **بحث**

```
StudentsBindingSource.Filter = "Prénom = 'وسيم'"
```

 Code VB.Net **موقع السجل**

```
Me.Text = StudentsBindingSource.Position + 1 & "/" & StudentsBindingSource.Count
```

 Code VB.Net **حذف السجل**

```
StudentsBindingSource.RemoveCurrent()
```

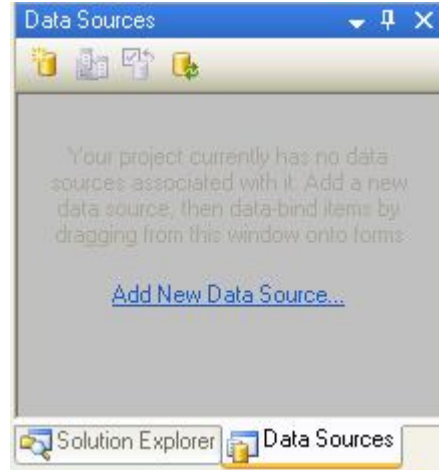
التعامل مع قواعد البيانات من خلال النوافذ

بناء مجموعة البيانات من خلال النوافذ

من قائمة البيانات Data اختر الأمر Show Data Source تظهر النافذة التالية مكان مستكشف الحل

الفصل الثامن قواعد البيانات

من اليمين إلى اليسار
 الزر الأول: لتنشيط القائمة وإعادة إنعاشها.
 الزر الثاني: إذا أردت إعادة التعديل في شكل البيانات
 المستورد أو ما شابه.
 الزر الثالث: إذا أردت أن تعدل في قاعدة البيانات مرئياً
 وباستخدام Designer
 الزر الرابع: لاتصال بالبيانات ، أو إنشاء اتصال بيانات
 جديد.

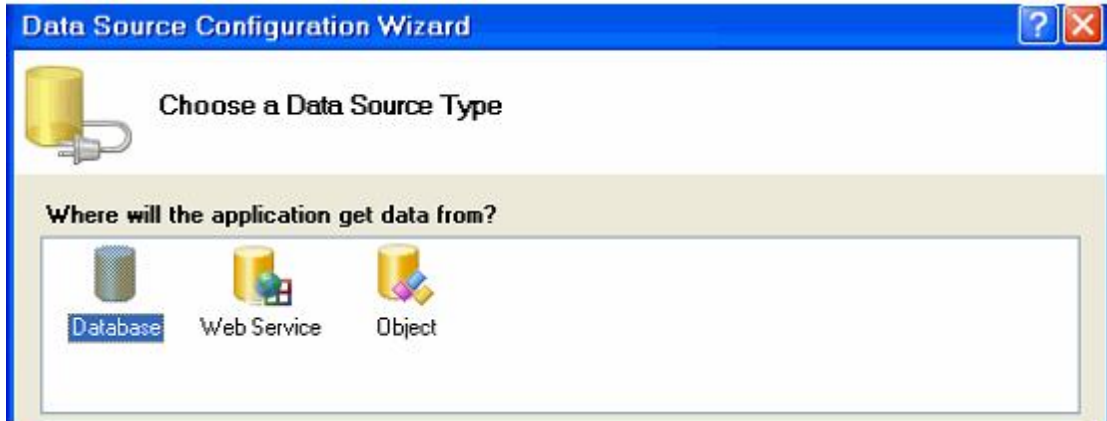


قبل كل شيء يجب أن تكون لدينا قاعدة بيانات ومن برنامج Access قم بإنشاء قاعدة بيانات جديدة وجدول
 بالاسم T1 قم بإنشاء هذه الحقول في الشكل التالي:

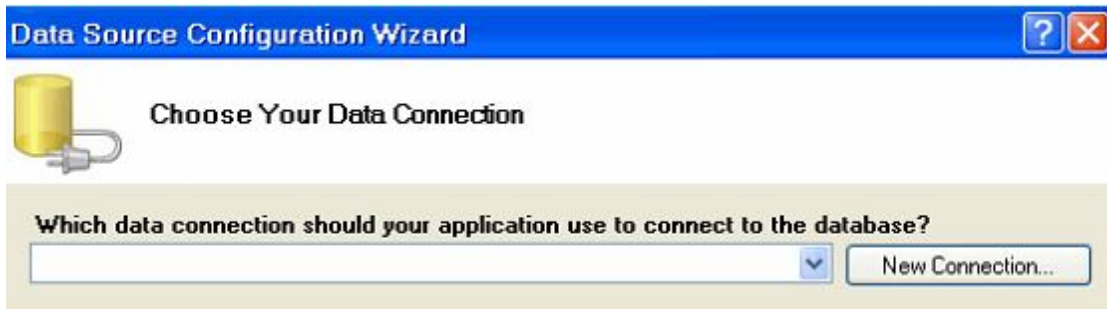
اسم الحقل	نوع البيانات
المعرف	ترقيم تلقائي
mname	نص
moude	نص
price	رقم
mdate	تاريخ/وقت
Pic	كائن OLE

طبعاً هناك درس عن كيفية إنشاء قاعدة بيانات جديدة راجعه بحال لا تعرف كيفية إنشاء قواعد البيانات
 ببرنامج Microsoft Access .

والآن دعنا ننشئ اتصالاً جديداً وبعد النقر على الخيار Add New Data Source سنرى النافذة التالية:



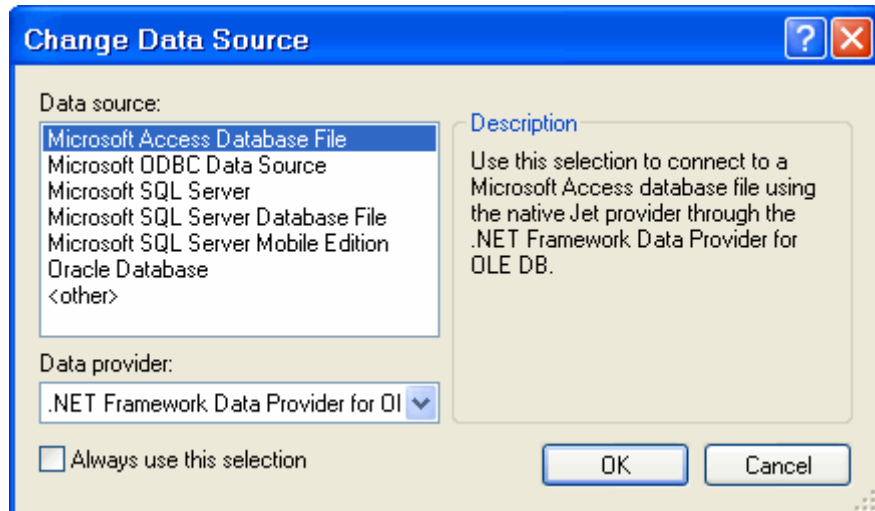
ما يهمنا هو الخيار المحدد حالياً Database وأنقر Next



أختر New Connection لتطالعك النافذة التالية:



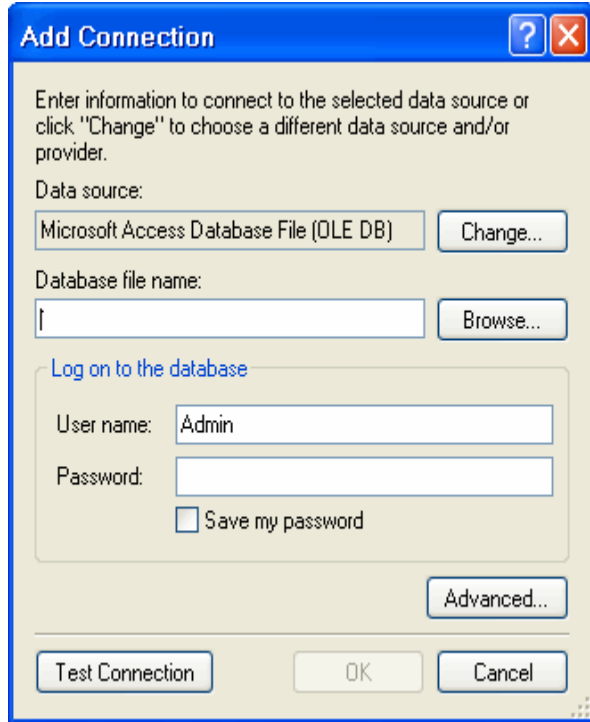
يمكنك من هنا اختيار نوع قاعدة البيانات التي سوف تتعامل معها وبحالنا هذه سوف نستخدم Access وأنقر على الزر Change لتطالعك النافذة التالية :



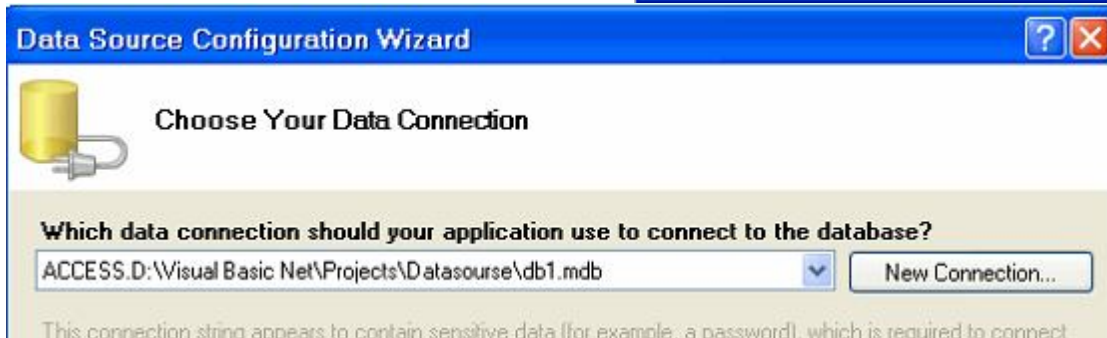
كما تلاحظ مزودات الخدمة لقواعد البيانات المختلفة وما يهمنا هنا هو مزود البيانات Access للتعامل مع قواعد البيانات Access حدده تم أنقر Ok . ستعود الى النافذة السابقة كما بالشكل التالي:

الفصل الثامن قواعد البيانات

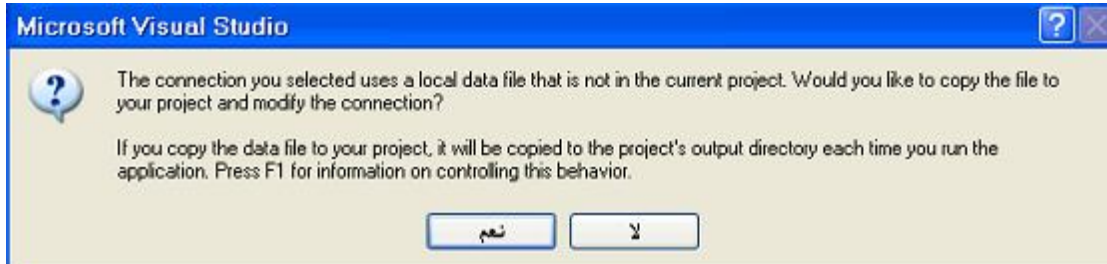
يوجد أمام أداة النص بالعنوان Database File الزر Name Browse وهو الزر الذي يمكنك منه تحديد ملف قاعدة البيانات خاصتك. إذا كنت قد أفلتت قاعدة البيانات بكلمة مرور واسم مستخدم عليك ان تحدها هنا في الخانتين التاليتين: User name-password وسوف نتحدث لاحقاً كيف يمكن إنشاء قاعدة بيانات في Access وكيف يمكنك حمايتها بكلمة مرور واسم مستخدم. عند تحديد ملف قاعدة البيانات يمكنك النقر على الزر Test Connection لاختبار الاتصال وإذا ظهرت هذه الرسالة فهذا يعني ان كل شيء على ما يرام.



بعدها انقر على Ok وانقر على Ok مرة أخرى لنعود إلى المكان الذي انطلقنا منه أول مرة:



لاحظ هنا مسار ملف قاعدة البيانات، وشفرة الاتصال كل شيء على ما يرام انقر على التالي لتطالعك الرسالة التالية:

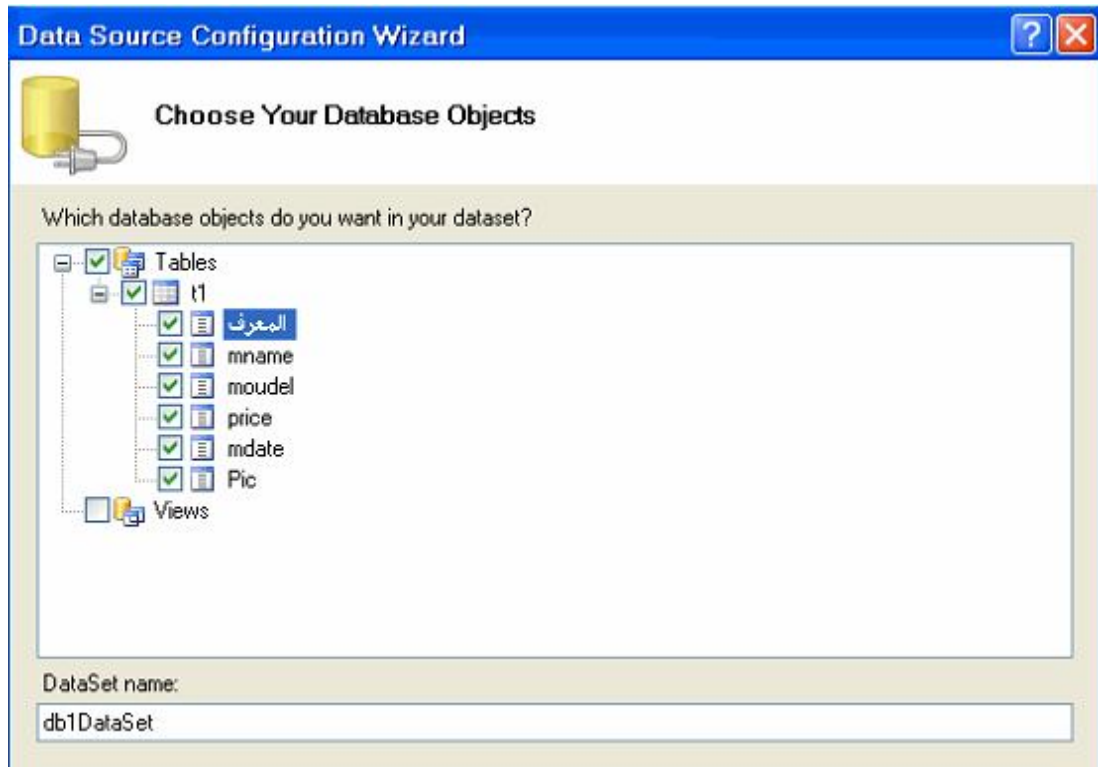


يخبرك VS.Net انك تستخدم ملف قاعدة بيانات ليس موجوداً بمجلد المشروع خاصتك فهل تريد نسخ الملف إلى المجلد، بالتأكيد انقر على الزر Yes لتطالعك النافذة التالية:



لا تغيير أي شيء فهنا سيتم تخزين شفرة الاتصال في ملف التهيئة للمشروع هذا ما يحدده مربع الاختيار، ولكن لحظة أريد أن اريك شيئاً ولذلك علينا العودة إلى الخلف انقر على الزر Previous
 Provider =microsoft.jet.oledb.4.0;data Source = e:\db1.mdb

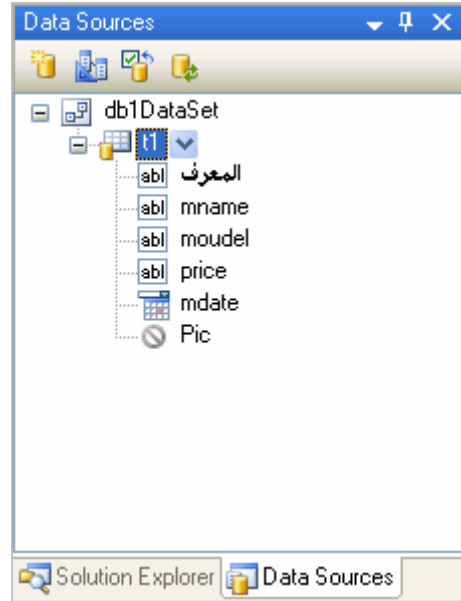
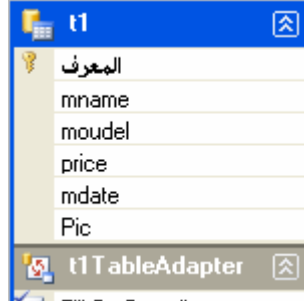
هل تلاحظ التغيير الذي حدث في شيفرة الاتصال String Connection انظر الى ما بعد الأمر Data Source لقد تم تغيير مسار قاعدة البيانات بالتالي:
 |DataDirectory| وهي تعني مجلد المشروع، وكنا نستخدمها في الفيچوال بيسك 6 App.path الأمر شبيهه بالتالي قد تحتاجها مستقبلاً او في إعدادات التقارير فلا تنساها.
 حسناً انقر Next تم مرة أخرى لتطالعك النافذة التالية:



هذه هي الجداول الموجودة في ملف قاعدة البيانات خاصتنا وقد حددت الجدول T1 تم انقر على الزر Finish وانظر إلى النافذة Data كما بالشكل:

الفصل الثامن قواعد البيانات

صارت مليئة بالحقول والجدول دعنا نجرب الزر الثاني من اليمين وسوف يقوم بإعادتك من حيث أتيت بحال الرغبة في تعديل أي شيء.
لننقر على الزر الثالث وهو المسئول عن التعديل بواسطة Data Designer كما بالشكل.

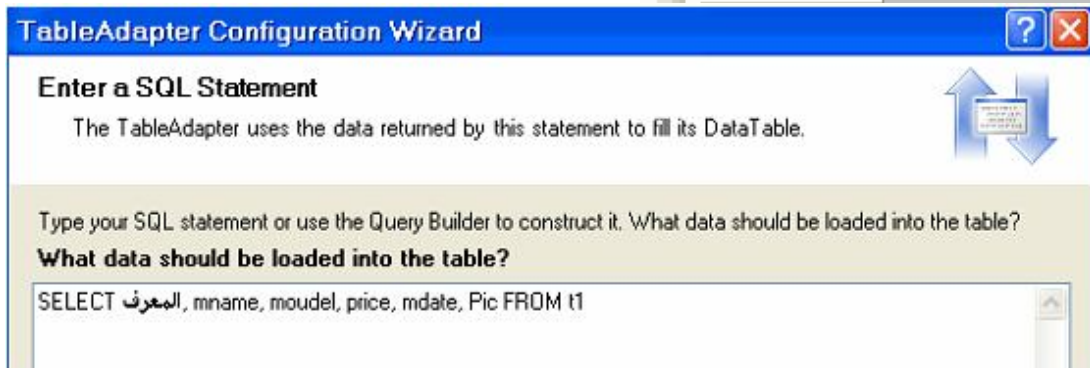
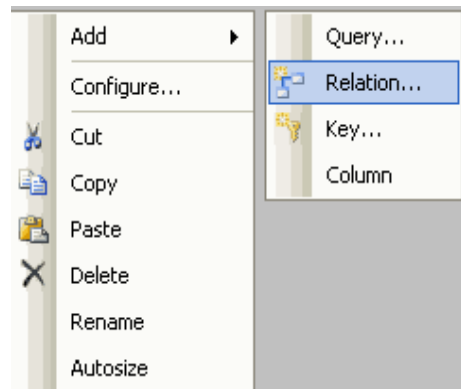


لاحظ أسم الجدول والحقول بقاعدة البيانات، اذا نقرت على احد الحقول او الجدول فيمكنك ان ترى خصائصه بنافذة الخصائص كالتالي:

لست مضطرا إلى القيام بأي تعديلات على الحقول من برنامج Access يمكنك إجراء التعديلات من هنا، نوعية الحقل أسمه، هل مسموح تركه فارغاً NULL الحجم الأقصى ، للقرءة فقط الخ..
أذا نقرت بزر الفأرة الأيمن فستطالعك القائمة التالية:

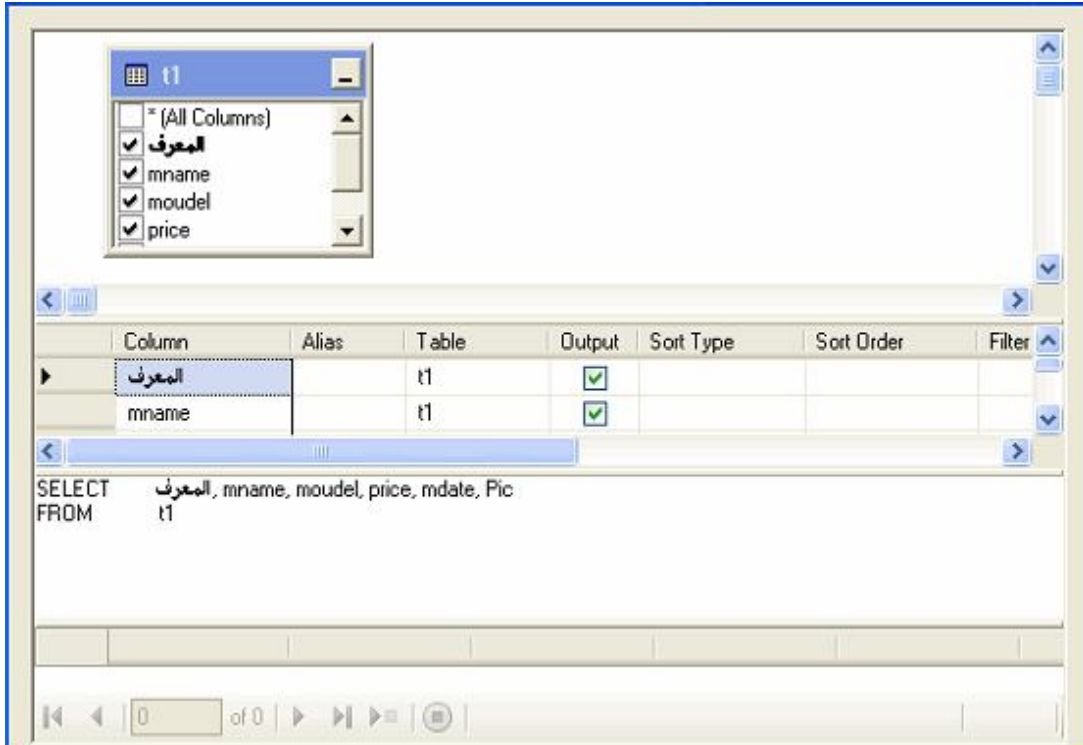
Caption	mname
DataType	System.String
DateTimeMode	UnspecifiedLocal
DefaultValue	<DBNull>
Expression	
MaxLength	255
Name	mname
NullValue	(Throw exception)
ReadOnly	False
Source	mname
Unique	False

هذه صورة للقائمة وصورة للقائمة الفرعية من Add والتي يمكنك من خلالها إضافة استعلام جديد ، أو إنشاء علاقة بين الجداول بحال كان هنا أكثر من جدول، أو مفتاح الخ. نفس برنامج Access تماماً اذا تعاملت معه.
الأمر Configure سوف يقودنا الى النافذة التالية:

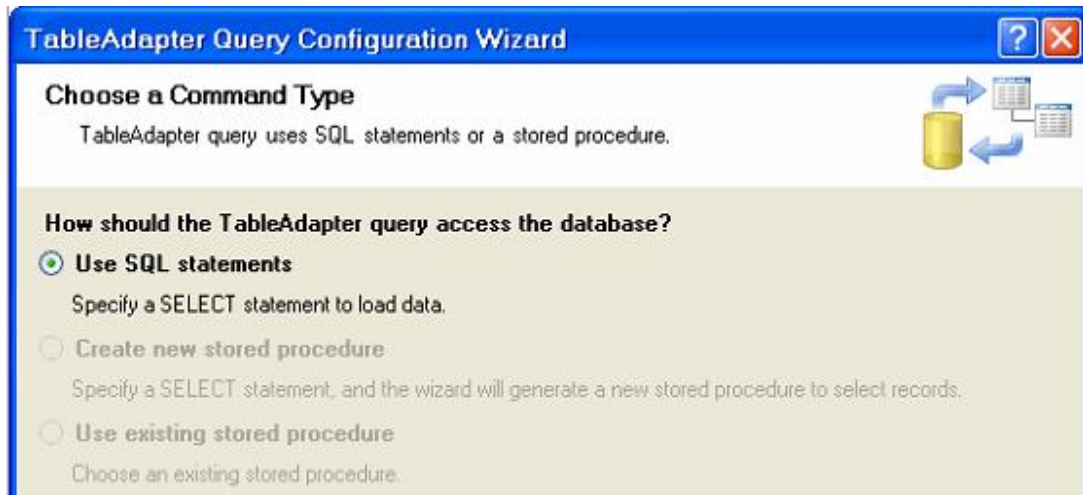


الفصل الثامن قواعد البيانات

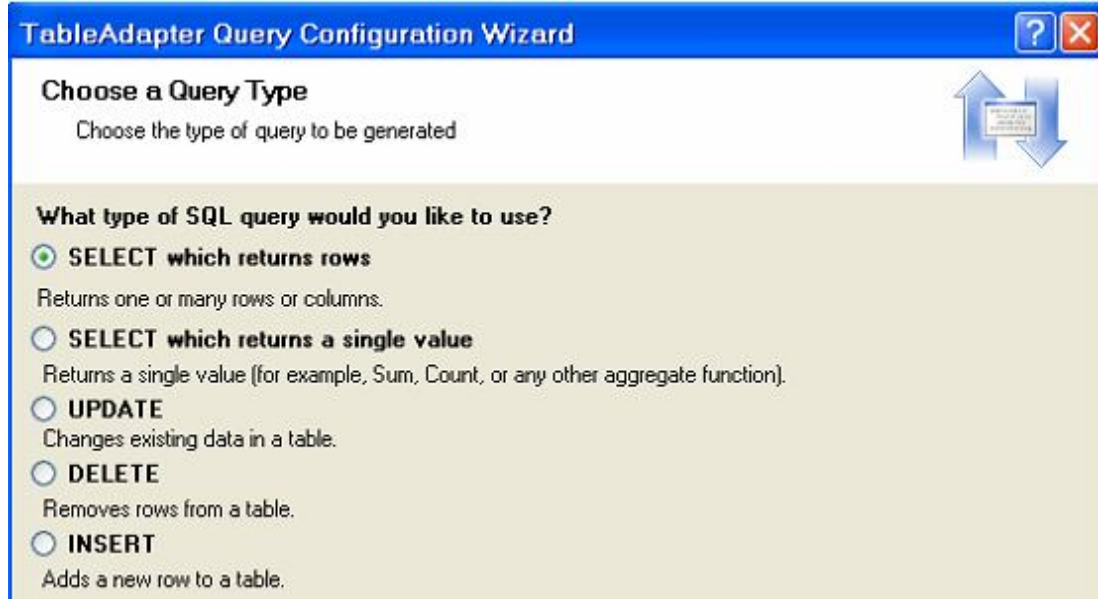
لاحظ جملة الاستعلام المحددة والافتراضية والتي كتبها المعالج عند جلب البيانات وموجودة بالاسم، Fill
GetData أسفل الجدول عد إلى الصورة لتشاهد ذلك.
يمكنك التعديل في جملة الاستعلام يدوياً أو اختيار Query Builder ليتم نقلك الى النافذة التالية:



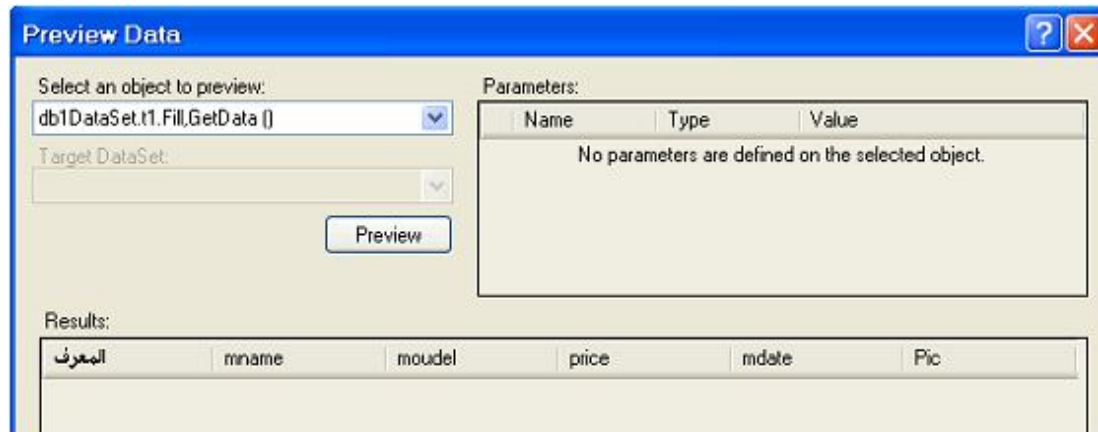
من هنا يمكنك تحديد الحقول ودع المعالج يقوم بالباقي، حقيقة عليك اكتشاف الباقي بنفسك، فانا لا أحبذ هذه الطريقة في التعامل مع البيانات ولك لإعطائك فكرة فقط.
الأوامر نسخ ولصق وقص وغيرها تقوم بنسخ الجدول او الحقل ولصقه يمكنك تجربة ذلك.
إذا أردت أن تضيف استعلاماً جديداً يمكنك اختيار Query لتطالعك النافذة التالية.



أنقر التالي Next



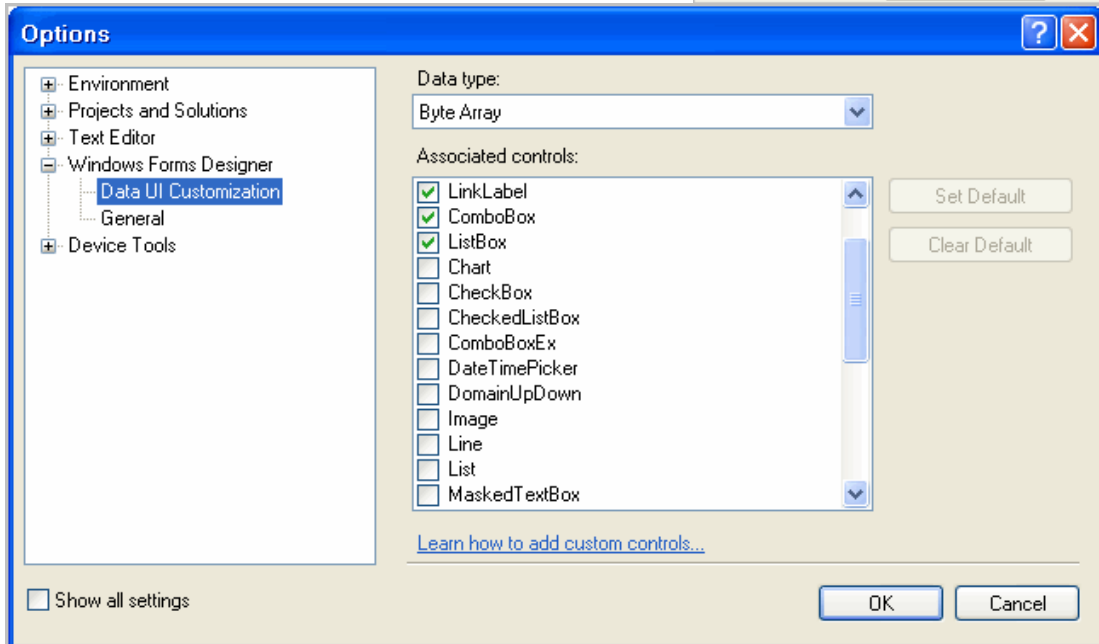
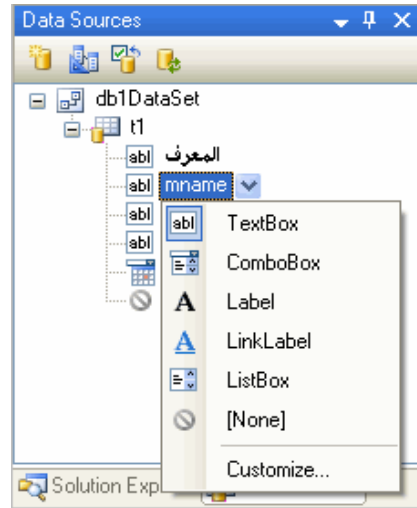
من هنا يمكنك تحديد نوع الاستعلام الذي تريده هل هو تحديث أم حذف أم إضافة وغيره تم أنقر Next فيتم نقلك إلى النافذة التي كنا فيها سابقاً لتحرير جملة الاستعلام حرر كما تشاء. الأمر Preview data يمكنك هذا الخيار من استعراض جملة الاستعلام بحال كانت قاعدة البيانات تحتوي على بيانات.



من القائمة المنسدلة Select an object to preview يمكنك تحديد الاستعلام المطلوب اذا قمت بإضافة العديد من الاستعلامات: وبعدها انقر على الزر Preview لتطالعك البيانات . حسناً دعنا ننهي هذه المهمة عد إلى النافذة Data source وانقر على احد الحقول ليتم عرض القائمة المنسدلة التابعة له كما بالشكل.

الفصل الثامن قواعد البيانات

من هنا يمكنك تحديد ما هي الأداة التي تريدها ان تتعامل مع هذا الحقل، هل تريد للحقل الارتباط بأداة النص مثلاً Textbox. وأحياناً لا يستطيع الفيجوال التعرف على ما هي الأداة المناسبة للتعامل مع حقل معين كما بالحقل الأخير عندنا هنا والذي هو بالاسم Pic . هذا الحقل هو عبارة عن بيانات ثنائية ونريده لوضع بيانات الصورة فيه. ولذلك حدد الخيار PictureBox لهذا الحقل. كما يمكنك تحديد أداة معينة والحصول على أدوات أخرى عن طريق الخيار Customize كما بالصورة السابقة. عندها ستحصل على هذه النافذة كما بالشكل التالي:



من تحت العنوان Associated Contrls يمكنك تحديد الأداة المناسبة ومن القائمة كما تشاهد. بعد ان تحدد لكل حقل الأداة الخاصة به قم بسحبها وإسقاطها على النموذج Form1 ليقوم البرنامج بباقي العمل كما بالشكل التالي

طبعا قمت بترتيب الأدوات بهذا الشكل .

تم إضافة شريط أدوات (T1BindingNavigator) ليتمكنك من الإضافة والتعديل والحذف، والتقدم الى الإمام والخلف وغيره.

وكذلك قارئ البيانات ومجموعة بيانات كذلك، ولزيادة فهم هذه الأدوات او الكائنات يمكنك مراجعة الدرس بالاسم Ado.net اما هنا فلننا بصدد الحديث عنها .

وإنما كيف نتعامل مع البيانات بشكل مرئي فقط وبقليل من الكود.

كما ستجد انه قد تم إضافة هذا الكود بنافذة محرر الكود.

```

Public Class Form1
    Private Sub T1BindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As
        Me.Validate()
        Me.T1BindingSource.EndEdit()
        Me.T1TableAdapter.Update(Me.Db1DataSet.t1)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Han
        |TODO: This line of code loads data into the 'Db1DataSet.t1' table. You can move,
        Me.T1TableAdapter.Fill(Me.Db1DataSet.t1)
    End Sub

```

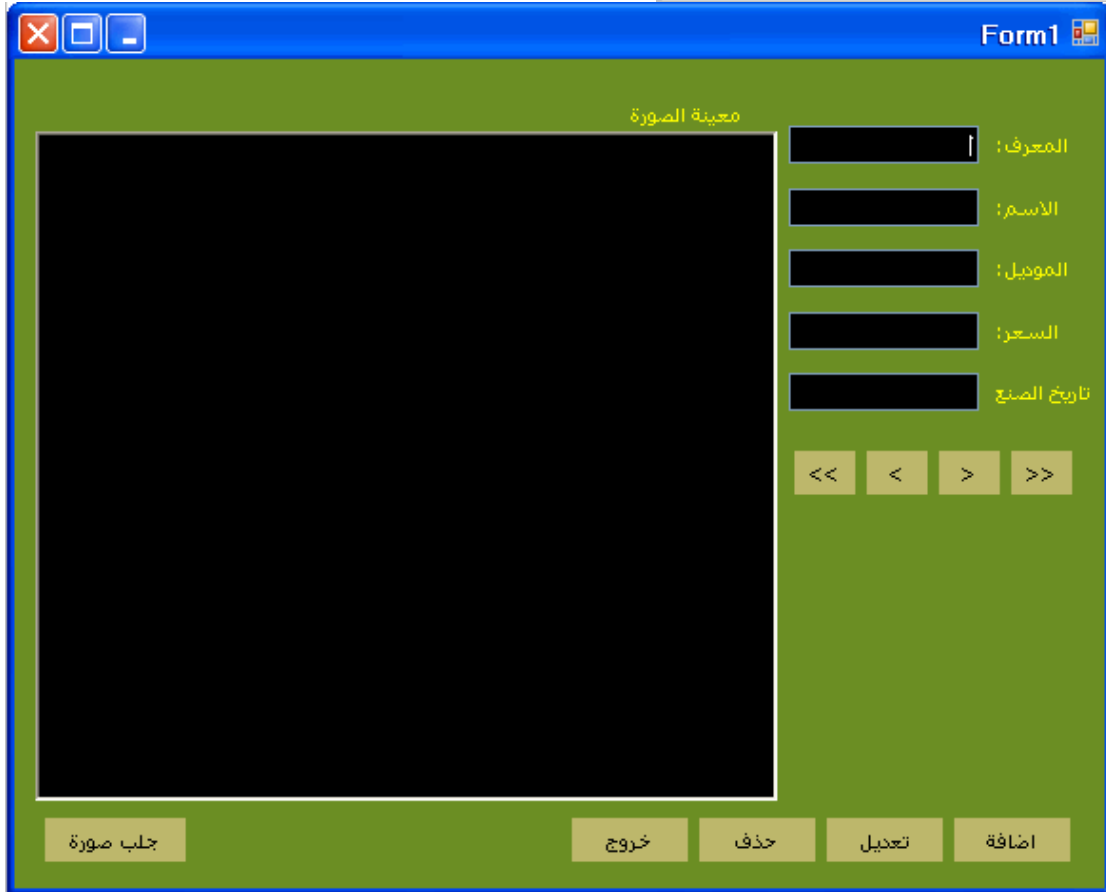
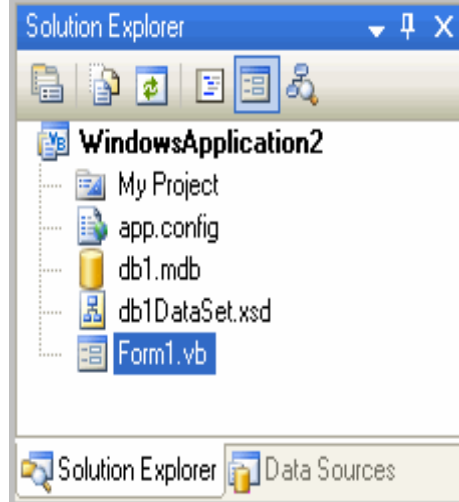
الفصل الثامن قواعد البيانات

إذا عدت إلى النافذة Solution explorer فسوف تلاحظ ظهور ملف قاعدة البيانات و الـ Dataset وهي Class للتعامل مع قاعدة البيانات كما بالشكل التالي:

يمكنك ان تختار عرض كل الملفات كما تعلمت سابقاً لتطالع النافذة كما بالشكل التالي أنقر على Dataset.designer لتتمكن من قراءة كامل الكود وتستفيد منه .

طبعاً اذا أردت وكانت لديك خلفية معقولة عن كيفية التعامل مع كائنات الاتصال .

حسناً الفيچوال إعطانا ما استطاع ان يقدمه ولكن الشكل السابق غير مقبول حقيقة ولا يمكنك ان تقدمه على انه برنامج لأي شخص .ويحتاج إلى تدخل طفيف منا حتى يصبح برنامجاً ذا طابع احترافي نوعاً ما أنظر للشكل التالي:



طبعاً لك كل الحرية في اختيار الألوان التي تناسبك وهذا ما حدده مزاجي انا الان وارجوا ان يعجبك. بداية وقبل كل شيء قم بإخفاء شريط الأدوات (T1BindingNavigator) اجعل الخاصية Visible له تساوي False فمظهره بشع جداً. ولا يمكن الاعتماد عليه بهذا المظهر. سوف نصلح الأمور.

الفصل الثامن قواعد البيانات

غير لون الخلفية للنموذج كما بالشكل واجعل لون الصورة وأدوات النصوص باللون الأسود ولون الخط باللون الأبيض.

ما تراه يمكنك ان تفعل مثله ولكن بخصوص الأزرار فيجب ان تبحث عن الخاصية Flat Appearance وستجد تحتها الخاصية FlatStyle اجعلها Flat .

طبعاً لكل الأزرار حتى يصبح شكلها مفلطح أنظر للشكل التالي :

غير هذه الخواص كما تراها بالشكل السابق .
واجعل الخلفية لكل الأزرار كما تراها بالصورة السابقة

FlatAppearance	
BorderColor	
BorderSize	0
MouseDownBackColor	Red
MouseOverBackColor	LightGray
FlatStyle	Flat

المهم هو الحصول على المظهر المناسب والغير اعتيادي.

الآن جاء دور الاكواد والتدخل اليدوي.

دعني أخبرك ان ما ستكتبه من كود هو شيء لا تبرمجه أنت وإنما هو شيء سبق برمجته من قبل الفيجوال وسوف نقوم باستدعائه فقط .

والآن عندنا أربعة أزرار تشير او تقوم بالعمل التالي:

الزر ورمزه	وظيفته
<<	ينقلنا الى أول السجل
<	السجل السابق
>	السجل التالي
>>	ينقلنا الى آخر سجل

أنقر على زر أول السجل واكتب فيه الكود التالي:

```
T1BindingNavigator.MoveFirstItem.PerformClick()
```

وفي زر السابق اكتب التالي:

```
T1BindingNavigator.MovePreviousItem.PerformClick()
```

وفي زر التالي اكتب التالي:

```
T1BindingNavigator.MoveNextItem.PerformClick()
```

وفي زر اخر السجل اكتب التالي:

```
T1BindingNavigator.MoveLastItem.PerformClick()
```

في زر الإضافة اكتب التالي:

```
T1BindingNavigator.AddNewItem.PerformClick()
```

وفي زر تعديل عليك بجلب الكود الموجود سابقاً كما بالشكل التالي :

الفصل الثامن قواعد البيانات

```
Public Class Form1
    Private Sub T1BindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As
        Me.Validate()
        Me.T1BindingSource.EndEdit()
        Me.T1TableAdapter.Update(Me.Db1DataSet.t1)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Han
        |'TODO: This line of code loads data into the 'Db1DataSet.t1' table. You can move,
        Me.T1TableAdapter.Fill(Me.Db1DataSet.t1)
    End Sub
End Class
```

الكود الموجود بالحدث T1BindingNavigatorSaveItem فيصبح كالتالي:

```
Me.Validate()
Me.T1BindingSource.EndEdit()
Me.T1TableAdapter.Update(Me.Db1DataSet.t1)
```

طبعاً لا داعي للكود الأول قم بقصه ولصقه بهذا المكان فقط.

وفي زر حذف اكتب التالي:

```
T1BindingNavigator.DeleteItem.PerformClick()
```

وفي زر خروج اكتب التالي:

```
End
```

وفي زر جلب صورة اكتب التالي:

```
Dim Dialog As New OpenFileDialog
Dialog.Filter = "All Support File;*.jpg;*.gif;*.png;*.bmp"
If Dialog.ShowDialog = Windows.Forms.DialogResult.OK Then
    Me.PicPictureBox.Image = Image.FromFile(Dialog.FileName)
End If
```

كل شيء يسير بشكل جيد ولكن ما زلنا بحاجة الى القليل من التعديلات فمثلاً يجب ان نخبر المستخدم ان التعديل قد تم على ما يرام كما بالشكل التالي :

```
Try
    Me.Validate()
    Me.T1BindingSource.EndEdit()
    Me.T1TableAdapter.Update(Me.Db1DataSet.t1)
    MsgBox("البيانات حفظ", MsgBoxStyle.Information, "بنجاح البيانات حفظ تم")
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

هناك أمر يجب ان تضعه بعين الاعتبار وهو بحال أردت ان تضيف سجلاً جديداً فسوف تنقر على الزر إضافة وتقوم بتعبئة البيانات وعندما تنتهي من ذلك يجب ان تنقر على الزر تعديل لحفظ البيانات.

الفصل الثامن قواعد البيانات

ولذلك لا يجب ان يكون الزر بالاسم تعديل وإنما بالاسم حفظ ليكون مقبولاً.
أيضاً الحقل رقم المعرف يجب إلغائه او جعله للقراءة فقط لأنه ترقيم تلقائي تقوم به قاعدة البيانات نيابة عنك.
كما بزر الحذف يجب إجراء هذا التغيير كما بالشكل التالي:

```
Dim Value = MsgBox("& بالاسم السجل حذف في رغبتك من متأكد انت هل  
Me.MnameTextBox.Text, MsgBoxStyle.OkCancel, "سجل حذف")  
If Value = 1 Then  
T1BindingNavigator.DeleteItem.PerformClick()  
End If
```

القيام بعملية التأكيد للمستخدم وإعطاء محاولة واحدة للتراجع عن عملية الحذف.
حسناً قمت بملء بعض البيانات كما بالشكل التالي:

المعينة الصورة

المعرف: 1

الاسم: BMW

الموديل: 528i

السعر: 6000

تاريخ البيع: 08/09/2002

جلب صورة

خروج

حذف

تعديل

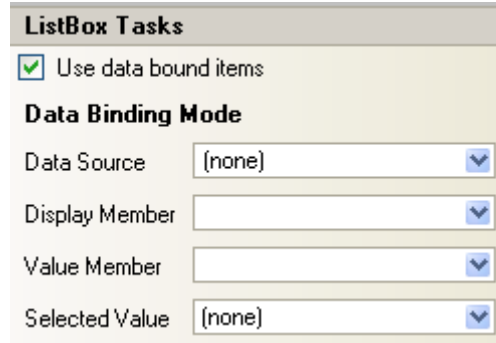
إضافة

والآن أريد أن أضيف قائمة يمكنها استعراض أسماء السيارات
هل ترى ذلك السهم الصغير أنقر عليه لتحصل على هذه النافذة

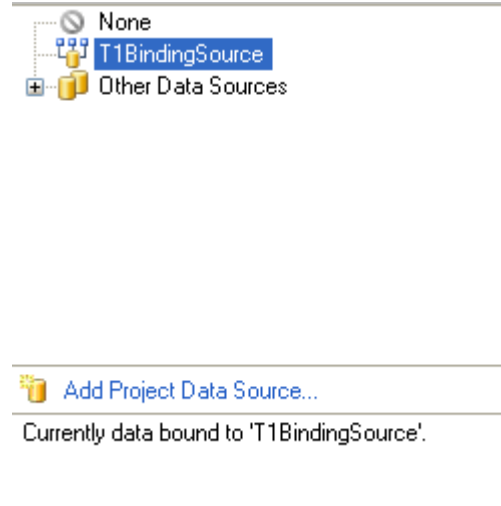
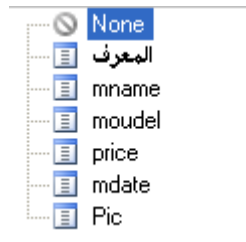
الفصل الثامن قواعد البيانات

حدد الخيار Use Data bound items كما بالشكل
لتحصل على المزيد من الخيارات.

من الخيار Data Source حدد
T1BindingSource كما بالشكل التالي:



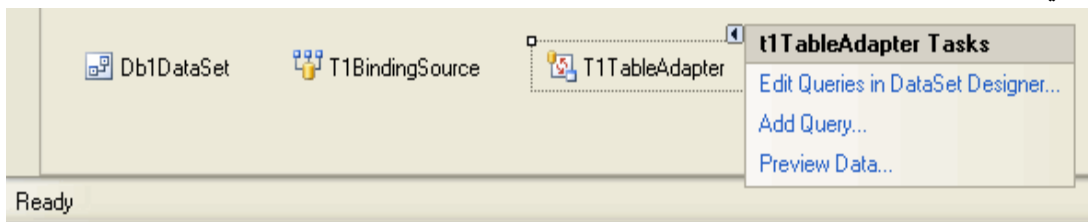
ومن الخيار Display Member حدد الحقل
mname كما بالشكل التالي



هذا كل شئ شغل البرنامج

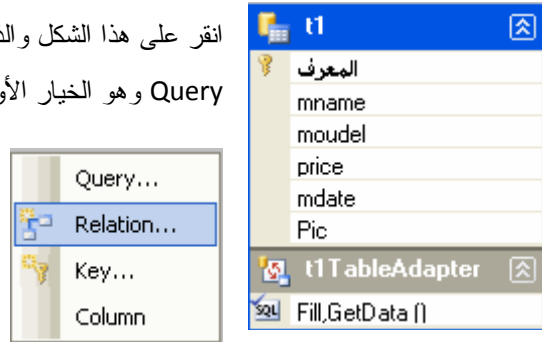
قام الفيچوال بتعبئة القائمة نيابة عنا انقر على أي اسم لاستعراض بياناته وصورته.
ولكن ماذا لو ان البيانات كثيرة ونريد ان نقوم بعملية بحث؟

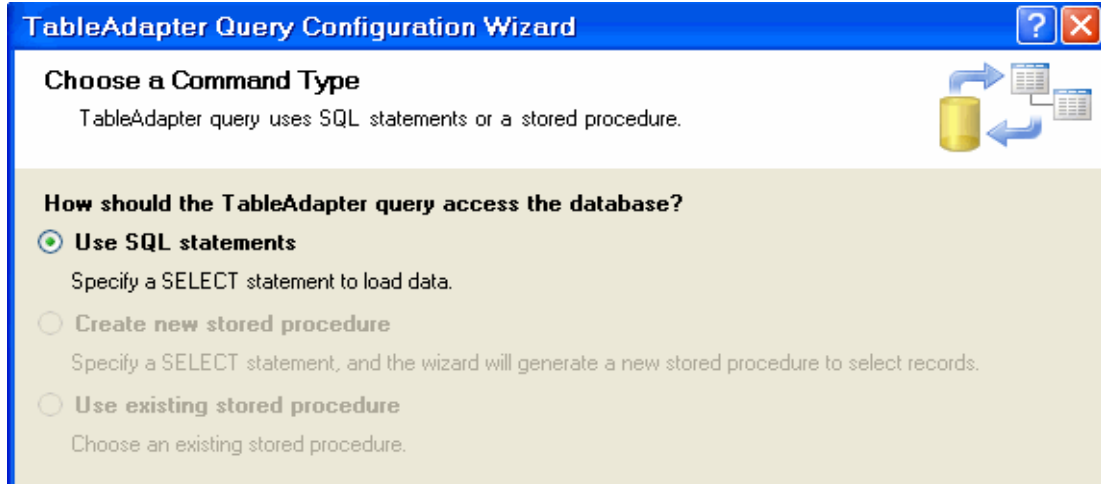
من الكائن T1TableAdapter اختر الخيار Edit Queriers in Dataset Designer كما بالشكل
التالي:



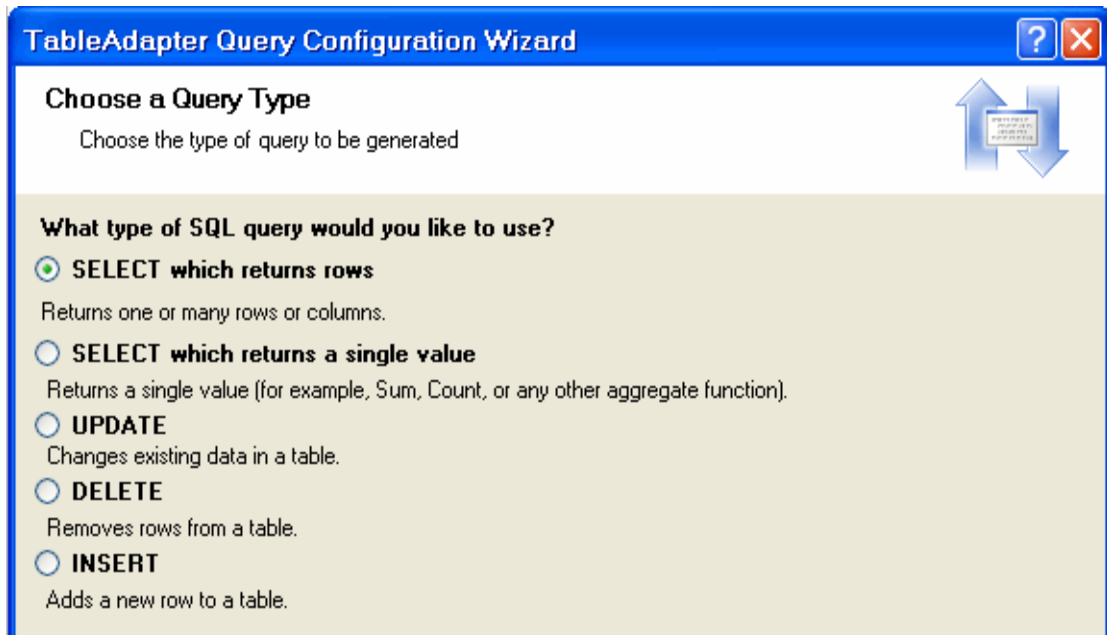
ستجد نفسك بهذا المكان الذي سبق وتحدثنا عنه

انقر على هذا الشكل والذي يمثل الجدول بزر الفأرة الأيمن ومن Add أختار
Query وهو الخيار الأول كما بالشكل التالي

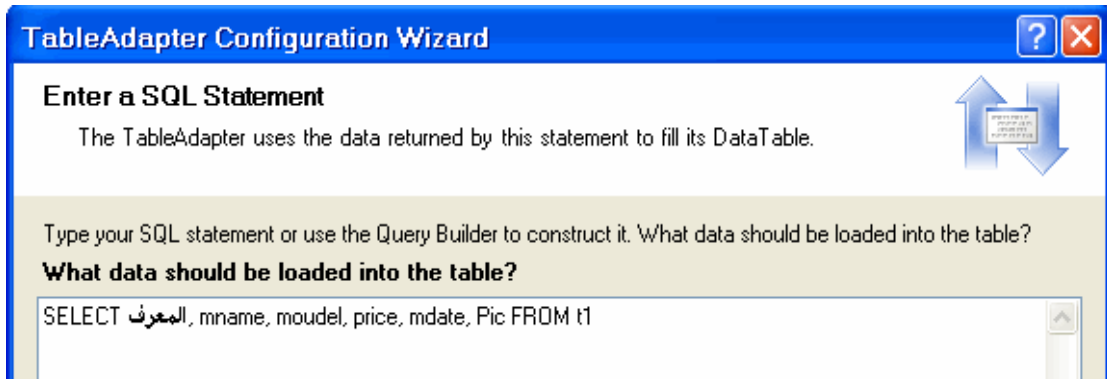




انقر على الزر Next لتنتقل الى النافذة التالية



انقر على الزر Next مرة ثانية لتنتقل الى الشكل التالي



كما تشاهد ان جملة الاستعلام مكتوبة ونريد ان نعدل فيها قليلاً .
لذلك غيرها الى الشكل التالي:

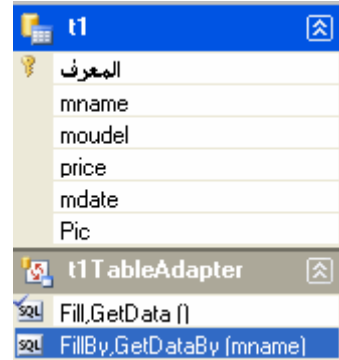
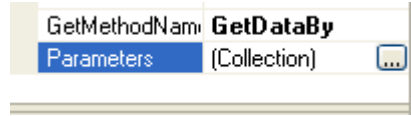
SELECT المعرف, mname, moudel, price, mdate, Pic FROM t1 **where mname=?**

أنظر الحروف المضللة هي ما قمنا بإضافته.

الفصل الثامن قواعد البيانات

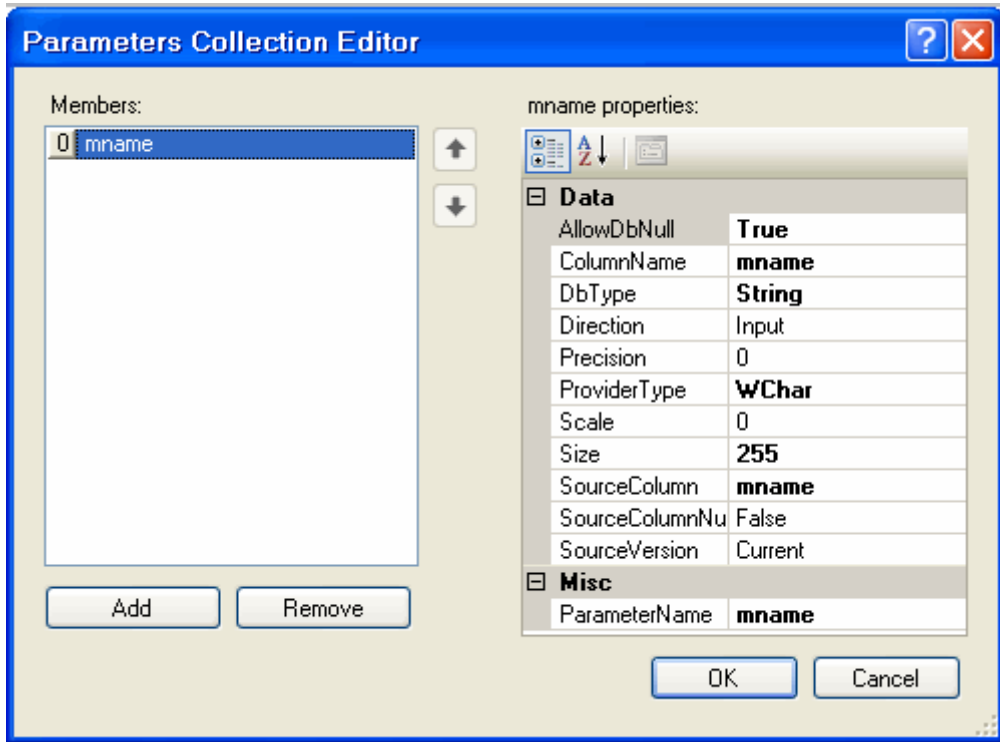
أنقر بعدها على الزر Finish لتعود للقائمة أو النافذة الرئيسية

حسناً إذا نظرت للأسفل لوجدت الاستعلام الجديد الذي قمنا بإضافته
انقر على هذا الاستعلام ومن نافذة الخصائص انتقل الى الخاصية
Parameters كما بالشكل التالي



إذا عدت إلى جملة الاستعلام لوجدت أننا حددنا القيمة بعلامة ؟ دلالة على أنها مجهولة والآن جاء الدور
لنحددها عن طريق بارامتر نقوم بتمريره فيما بعد.

أنقر على هذه الخاصية من خلال الصندوق المنقط كما بالشكل السابق لتطالعك النافذة التالية :



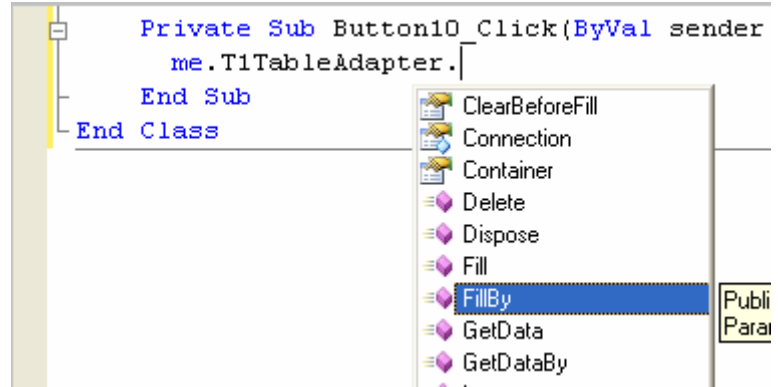
كما تتشاهد الفيچوال وجد الحقل المطلوب والذي يحتاج إلى بارامتر وهو هنا الحقل mname .
ووضع له الاسم mname كما تتشاهد .

ما يهمنا هو الخاصية DbType للبارامتر والتي تحدد نوع البيانات عادة الفيچوال يعلم ما هي البيانات على
حسب نوع الحقل، وهي نصية هنا.

أنقر الزر Ok .

اذهب إلى النموذج وأضف أداة نص جديدة وزراً جديداً اكتب فيه التالي:

الفصل الثامن قواعد البيانات



كما تشاهد عن طريق الكائن T1TableAdaprer ستجد الإجراء الذي أعدناه سابقاً وهو هنا بالاسم FillBy كما هو محدد بالصورة .

الإجراء يطالبنا باسم الجدول وقيمة البارامتر كما بالشكل التالي :

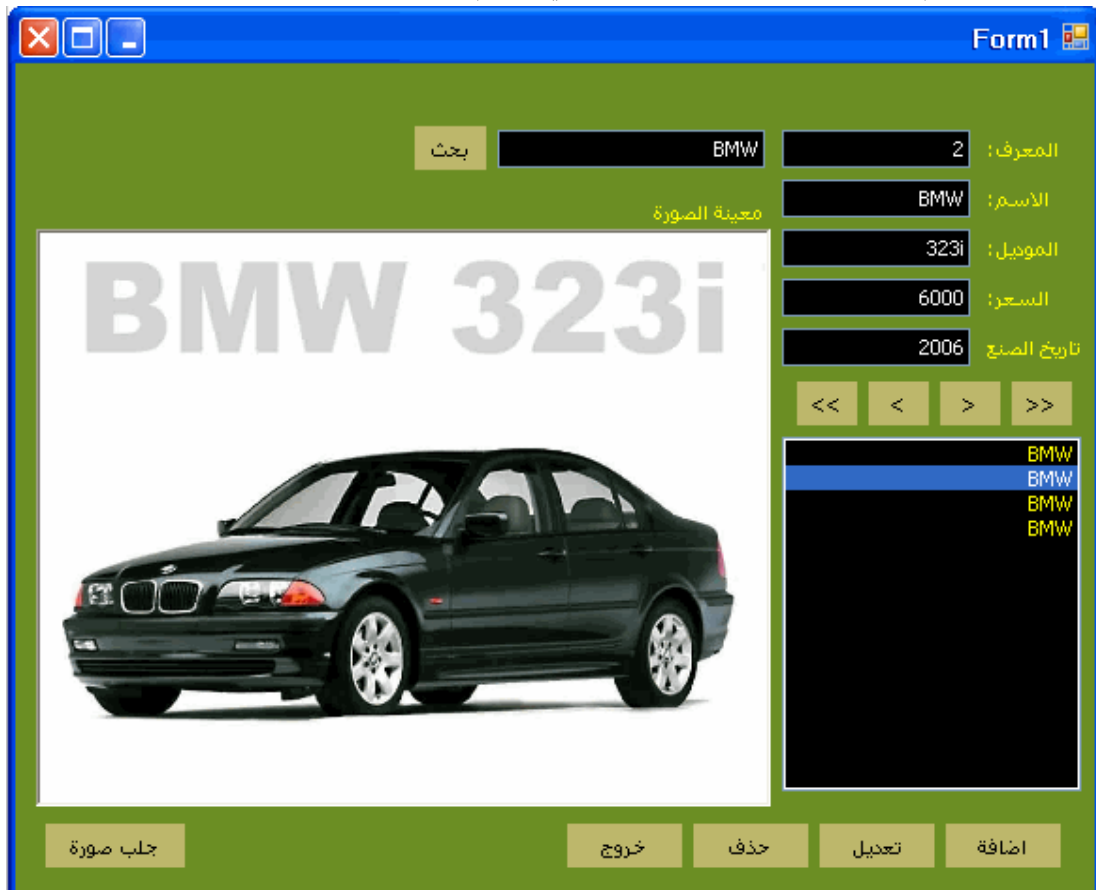
```
FillBy (dataTable As WindowsApplication1.db1DataSet.t1DataTable, mName As String) As Integer
```

وعليه سنكتب التالي :

```
Me.T1TableAdapter.FillBy(Me.Db1DataSet.t1, Me.TextBox1.Text.Trim)
```

ما سيحدث هنا هو انه سيتم عرض كل أسماء السيارات التي نقوم بكتابتها وتعبئة كائن مجموعة البيانات بالبيانات الجديدة.

وكما تشاهد هنا تم فرز البيانات وعرض السيارات التي بالاسم BMW فقط .



الفصل الثامن قواعد البيانات

هذه بعض الأفكار التي أقدمها طبعاً يمكنك التعديل في الكثير وإضافة الكثير إليها لتصبح بشكل أفضل من ذلك. وكما لاحظت إنني لم اشرح الكود كثيراً وذلك لأننا هنا نتعلم البرمجة من غير كود تقريباً

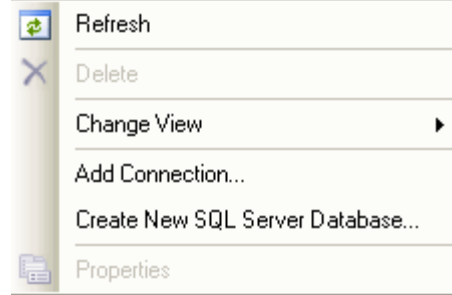
التعامل مع SQL Server 2005 من خلال النوافذ

طبعاً قواعد بيانات من هذا النوع تتميز بالسرعة وكذلك الأمان العالي بعكس قواعد بيانات Access . كيف ننشئ قاعدة بيانات....؟

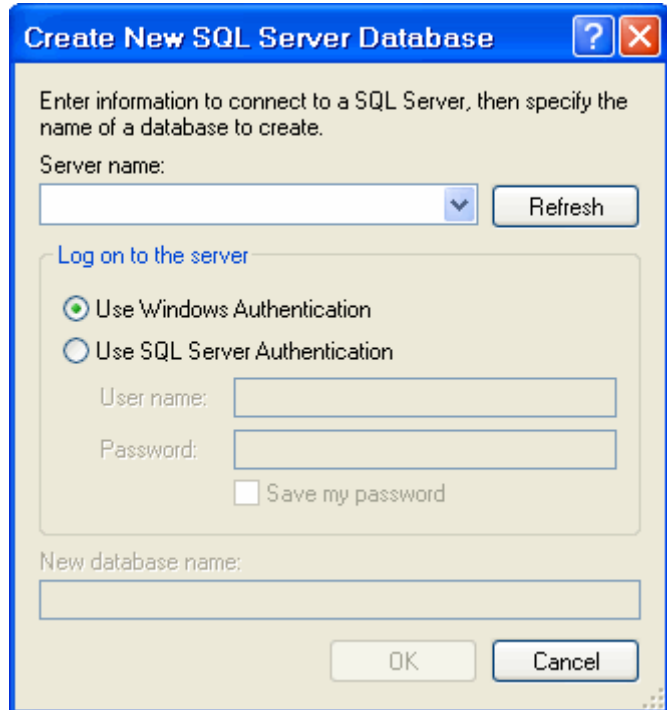
من الفيچوال أستوديو ومن النافذة Server Explorer نحدد بزر الفأرة الأيمن ونختار Create New SQL Server Database كما بالشكل التالي :



وهذه القائمة نتيجة نقر بزر الفأرة الأيمن كما بالشكل



إذا لم تجد نافذة Server Explorer فيمكنك إظهارها من القائمة View .

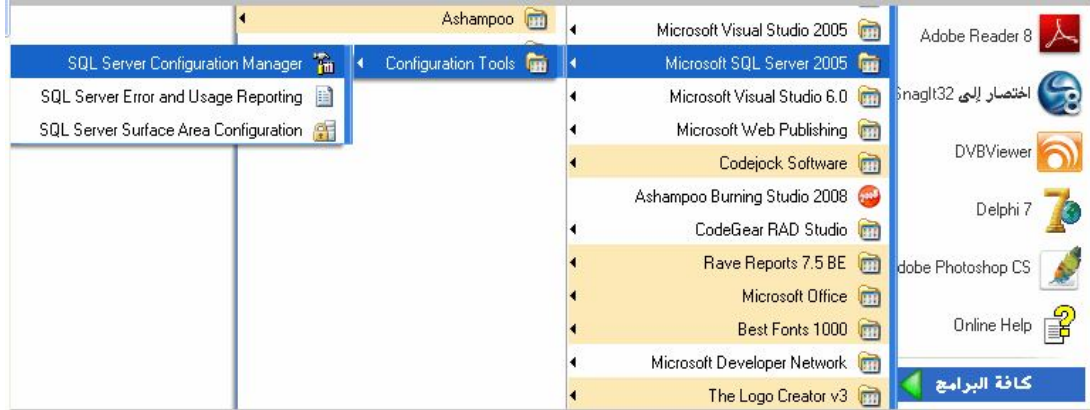


هذه النافذة تطالبنا باسم الـ Server وطبعاً قد يتغير من جهاز الى آخر على حسب إعدادات الشخص نفسه وبحسب ما أحدثه من تغييرات.

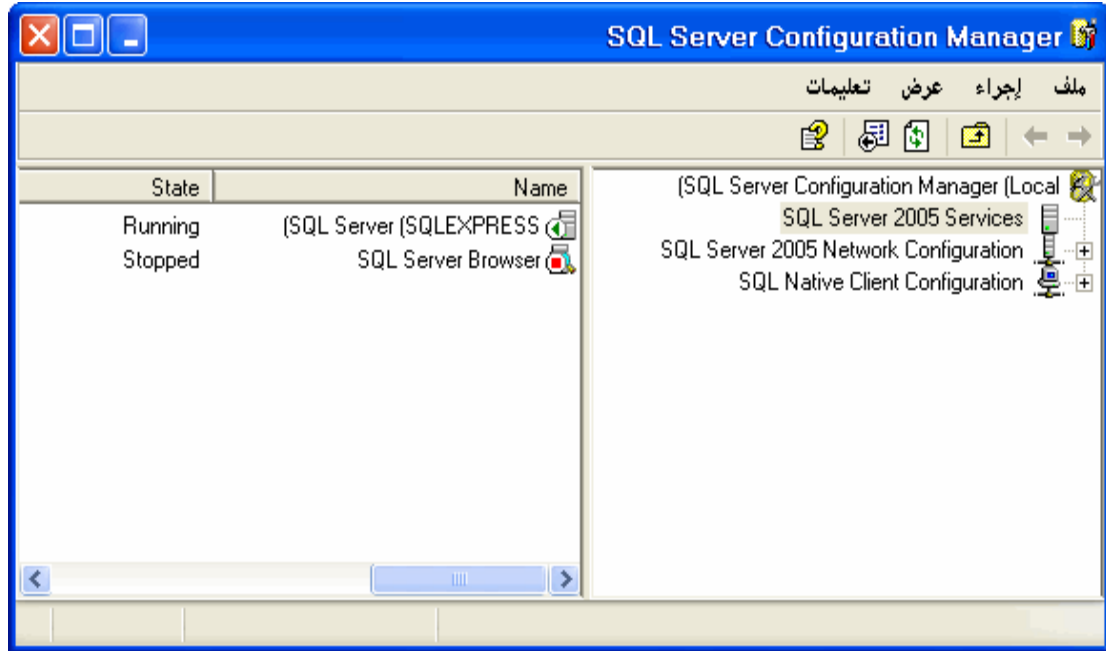
الفصل الثامن قواعد البيانات

ولكي تعلم ما هو اسم الـ Server لديك وبحال قمت بتثبيته مع إصداره الفيچوال و إلا فلن تستطيع العمل معه.

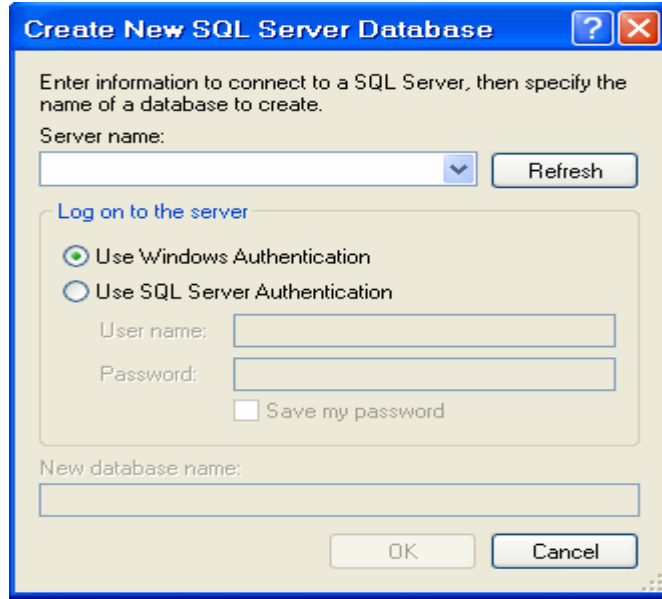
من قائمة أبدأ تم كافة البرامج ابحث عن Microsoft SQL Server 2005 كما تشاهد.
تتبع المسار كما بالصورة التالية :



وقم بتشغيل SQL Server Configuration Manager . ستظهر هذه النافذة كما بالشكل التالي :



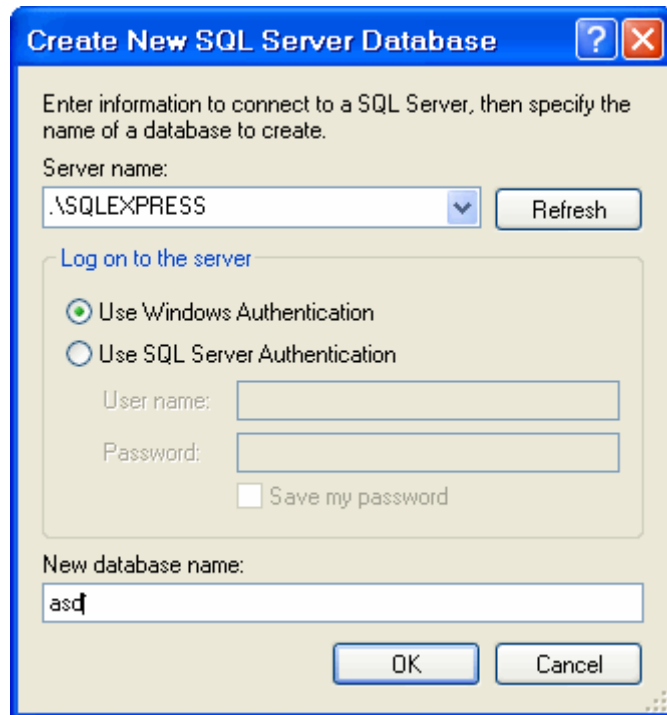
انظر إلى جهة اليسار تحت العنوان Name كما تشاهد أسم الـ Server هو (SQLEXPRESS) وتحت State توجد كلمة Running بمعنى أنه شغال.
وبالعودة إلى آخر نافذة توقفنا عندها وهي التالية :



إذا نكتب التالي للـ Server Name :

.\SQLEXPRESS

فتكون النتيجة كما بالشكل التالي :

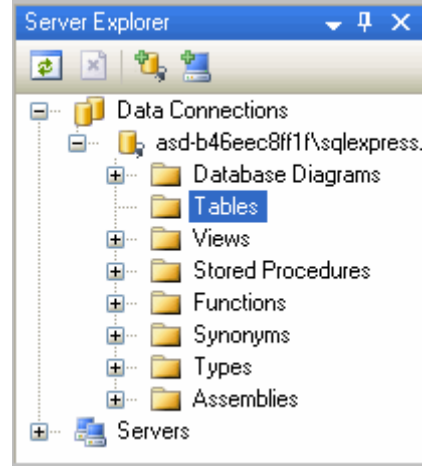
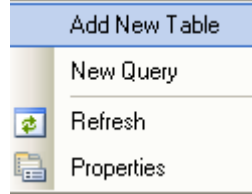


طبعاً تحت العنوان New database name مطلوب اسم لقاعدة البيانات وهو اختياري ضع ما تشاء فيه تم انقر Ok .

ستجد نفسك بنافذة Server Explorer من جديد ومع قاعدة البيانات التي أعدناها كما بالشكل :

الفصل الثامن قواعد البيانات

توجد العديد من المجلدات التي من الممكن ان نحفظ ونعد فيها مثل الجداول والإجراءات المخزنة والدوال وغيرها. ولكن ما يهمنا الآن هو إنشاء الجداول. لذلك عليك ان تنقر على المجلد المسمى Tables كما بالشكل السابق وتنقر بزر الفأرة الأيمن وتختار Add New Table كما بالشكل التالي



بعدها ستظهر هذه النافذة وكما تشاهد قمت بإضافة عمود جديد ونوعه نص، اما Allow Null فهي تحدد هل يقبل قيم فارغة ام لا.

وبالأسفل يمكنك أن تشاهد خصائص كل الأعمدة التي نضيفها. وحقيقة أجد الأمر مقبولاً أكثر عند إنشاء قاعدة بيانات وأقول او نقول إضافة عمود Column بدل حقل. وهو مصطلح يتماشى مع بيئة الفيجوال وكائنات الاتصال فيه.

فعمود (Column) تعني حقل وصف (Row) تعني سجل، سيواجهك هذان المصطلحان كثيراً

Column Name	Data Type	Allow Nulls
mname	text	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
(General)	
(Name)	mname
Allow Nulls	Yes
Data Type	text
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	

ما هي انواع البيانات التي يمكن التعامل معها في الـ SQL SERVER 2005

النوع	القيمة المخزنة	المدى	معلومات عن النوع
char	حرفي	1 إلى 8000 حرف	وهو عبارة عن نوع يجعل حفظ البيانات بتقل عدد مفس من الحروف , ويحجز نفس البيانات في الذاكرة , حتى وإن أدخل إليها أقل .
nchar	حرفي	1 إلى 4000 حرف	وهو أيضا يأخذ حروف ولكن حروف من النوع Unicode أي يمكنه تخزين حروف كل اللغات وليس الإنجليزية فقط , ولكن يحمل نفس عيب الـ Char .
varchar	حرفي	1 إلى 8000 حرف	ينطبق عليه نوع مطاقي بغير الحجم المخزن على حسب عدد الحروف المخزنة فيه , ولكن أقصى عدد حروف يأخذه هو ما حددته أنت .
varchar(max)	حرفي	1 إلى 8000 حرف	بهذه كآلة كنت بالشكل . , varchar(8000) . بس لو دخلت مثلا كلمة مكونة من 10 حروف فقط فسبحرنا في مكان بسبع لـ 10 حروف فقط وليس 8000 .
nvarchar	حرفي	1 إلى 4000 حرف	نفس قصة varchar بس يأخذ حروف Unicode (أي لغة) .
nvarchar(max)	حرفي	1 إلى 4000 حرف	نفس قصة varchar(max) بس يأخذ حروف Unicode .
text	حرفي	2 مليار حرف	ولكن تخبرني هذا الكم الهائل من البيانات داخل خلية بياناتك ومع استخدامه بكثرة يريد من بطه قاعدة بياناتك .. لذلك لو سبندخل بيانات بحجم كبير جدا فيصحني أن نخزنه في ملف خارجي أفضل .
ntext	حرفي	1 مليار حرف	مثل الـ النوع text بس يخزن حروف بالنظام Unicode .
image	نظام ثنائي	2 مليار بت	مثل النوع text بالصظ بس يخزن بيانات بالبيترى .. مثل الصور والصوت والأفلام ..
binary	نظام ثنائي	8000 بايت	هذا النوع يخزن بيانات بالبيترى .. زي الصور والصوت والأفلام .

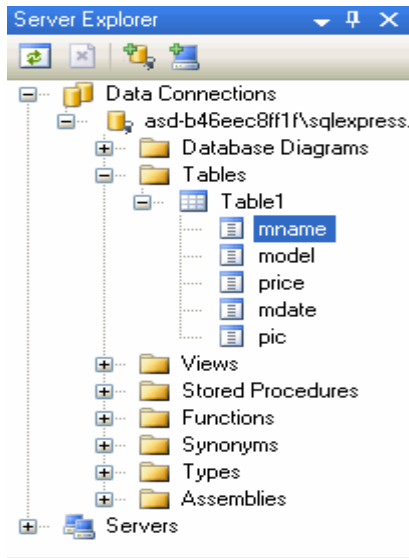
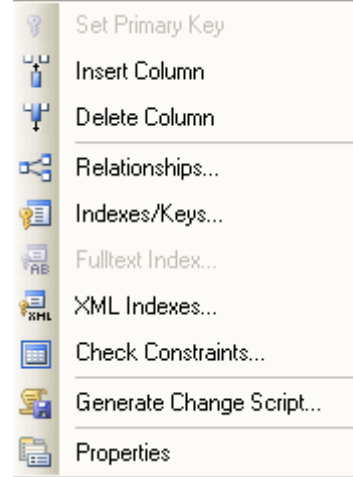
الفصل الثامن قواعد البيانات

وبتكملة إعداد كل الأعمدة أو الحقول يكون الجدول كما بالشكل التالي :

Column Name	Data Type	Allow Nulls
mname	text	<input checked="" type="checkbox"/>
model	text	<input checked="" type="checkbox"/>
price	int	<input checked="" type="checkbox"/>
mdate	datetime	<input checked="" type="checkbox"/>
pic	image	<input checked="" type="checkbox"/>

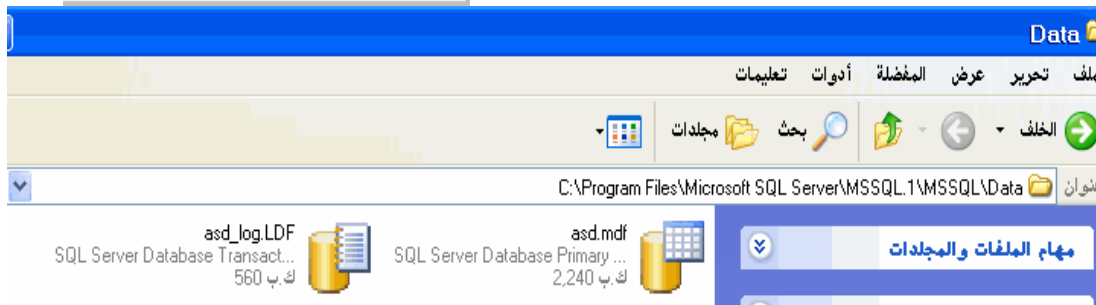
إذا نقرت على أي عمود أو حقل بزر الفأرة الأيمن فستحصل على القائمة التالية ما يهنا بداية وقبل كل شيء الخيارات التالية

الخيار	الوظيفة
Set Primary Key	لجعل العمود أو الحقل كمفتاح رئيسي وطبعاً من الضروري ان يكون من النوع Integer ليتمكنك فعل ذلك
Insert Column	حشر عمود أو حقل جديد بقاعدة البيانات
Delete Column	حذف عمود



والآن إذا حفظت المشروع أو قمت بإغلاق الجدول فسيطالبك الفيچوال باسم للجدول.

اكتب الاسم الذي تريد أو اتركه الافتراضي كما فعلت إنا هنا وسيكون هذا هو جدولنا والحقول أو الأعمدة التي فيه كما بالشكل التالي



تماماً لقد جعلنا أسمها asd كما يمكنك ان تلاحظ المسار فوق وهو المكان الذي ستجدها فيه.

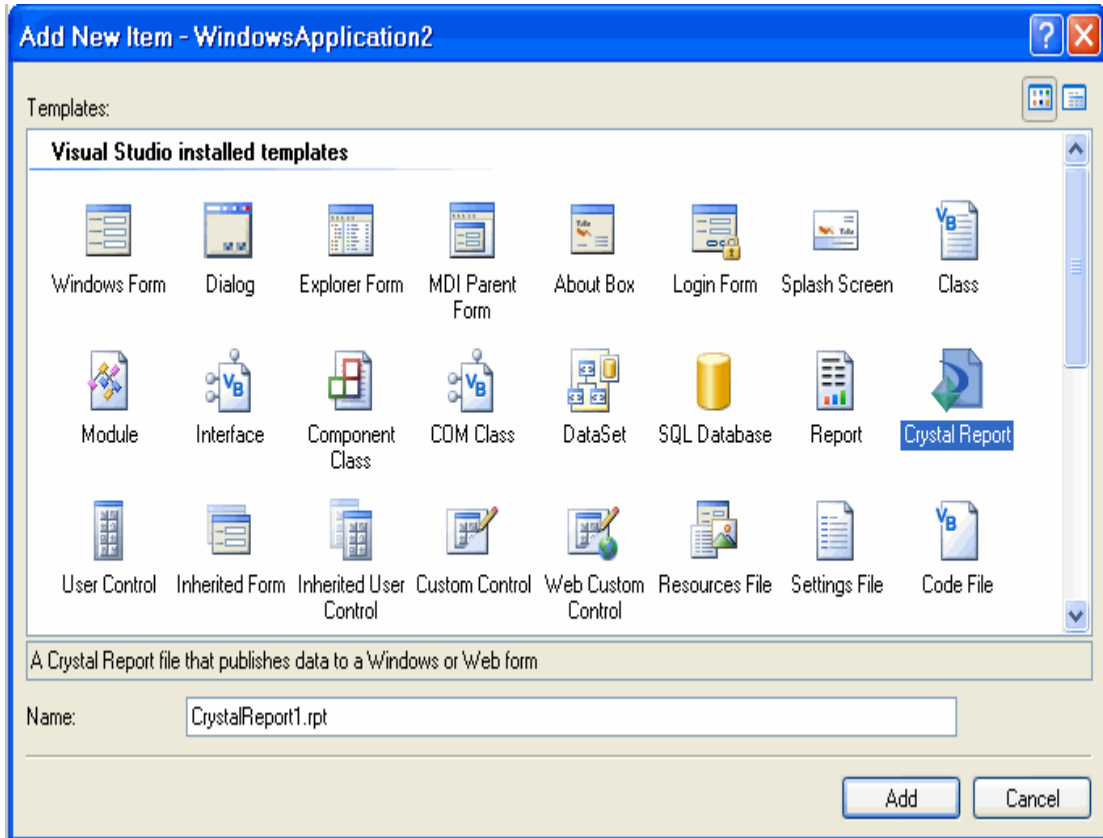
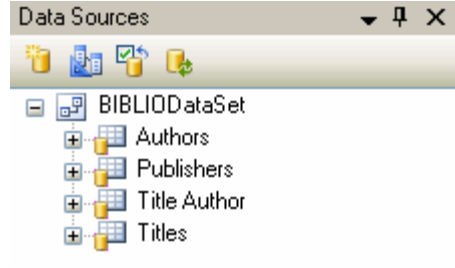
الفصل الثامن قواعد البيانات

بالنسبة لكائنات الاتصال بها نفس الدرس حول Ado.net ولكن الاختلاف بمزود الخدمة فقط. أما في الاتصال بها عن طريق الـ Data Source فنفس الطريقة التي اتبعناها بالدرس حول الـ Data Source لا اختلاف يذكر.

تقرير Crystal Report

أولاً ولكي ننشئ تقريراً نحتاج إلى الاتصال بقاعدة البيانات وجلبها إلى مشروعنا. من القائمة Data أختار Datasource Add New (راجع درس بناء قواعد البيانات من خلال النوافذ) عند إضافة قاعدة البيانات إلى البرنامج فسيمكننا أن نرى كل الجداول كما بالشكل التالي :

هذه هي الجداول و إنا استخدم قاعدة بيانات وهذا مسارها :
Files\Microsoft Visual Studio\VB98 C:\Program
وأسمها : BIBLIO
إذا جاء الآن دور إضافة تقرير جديد.
من القائمة Project وكما تعلمت سابقاً أختار Add New
Item لتطالعك النافذة التالي



حدد الخيار Crystal Report كما بالصورة زانقر الزر Add لتطالعك النافذة التالية

الفصل الثامن قواعد البيانات

الخيار الأول : Using The Report Wizard

وهو معالج سيأخذك في جولة قصيرة لإعداد تقريرك بالشكل المطلوب.

الخيار الثاني : As a Blank Report

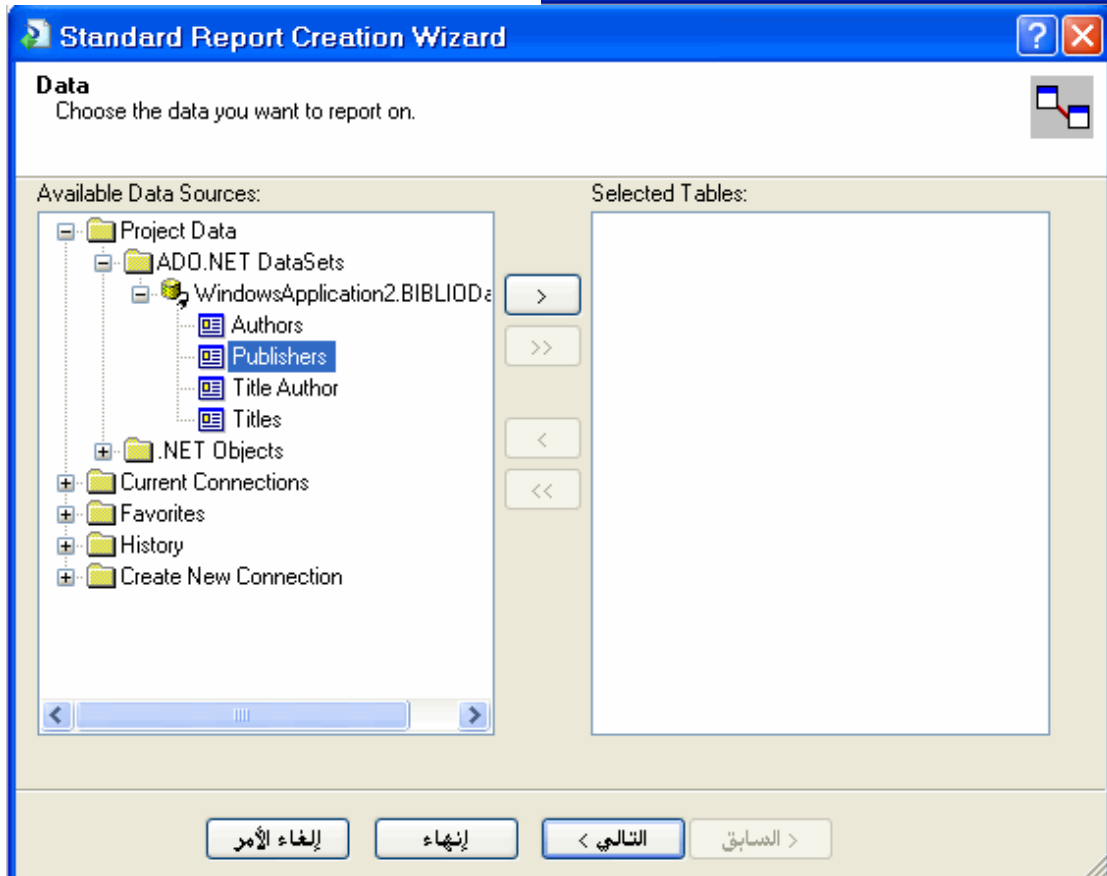
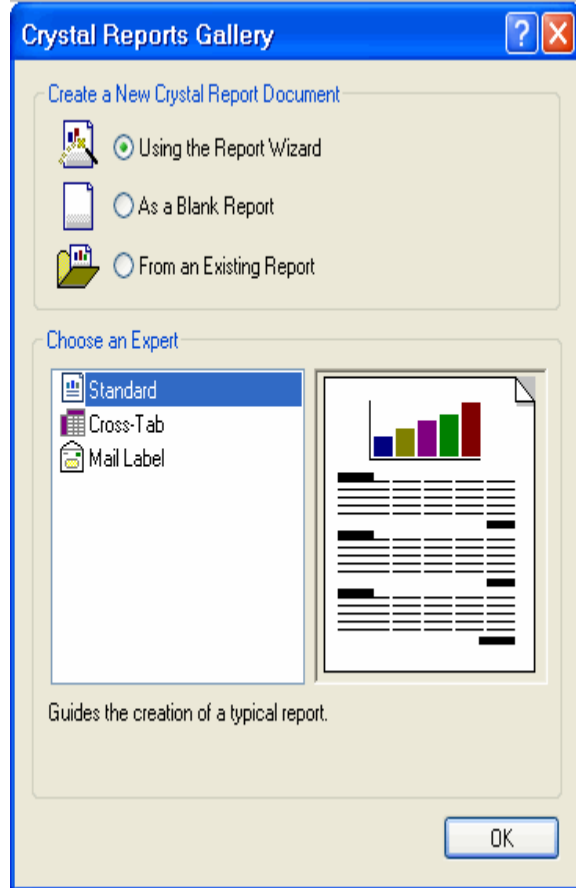
لإنشاء تقرير فارغ

الخيار الثالث : لإنشاء تقرير من تقرير موجود مسبقاً.

أما تحت العنوان Choose an Expert فتوجد القوالب التي من الممكن أن نتعامل معها كل حسب وظيفتها.

على العموم الخيار Standard هو خيارنا الآن . وكذلك Using The Report Wizard هو الخيار الأمثل الآن.

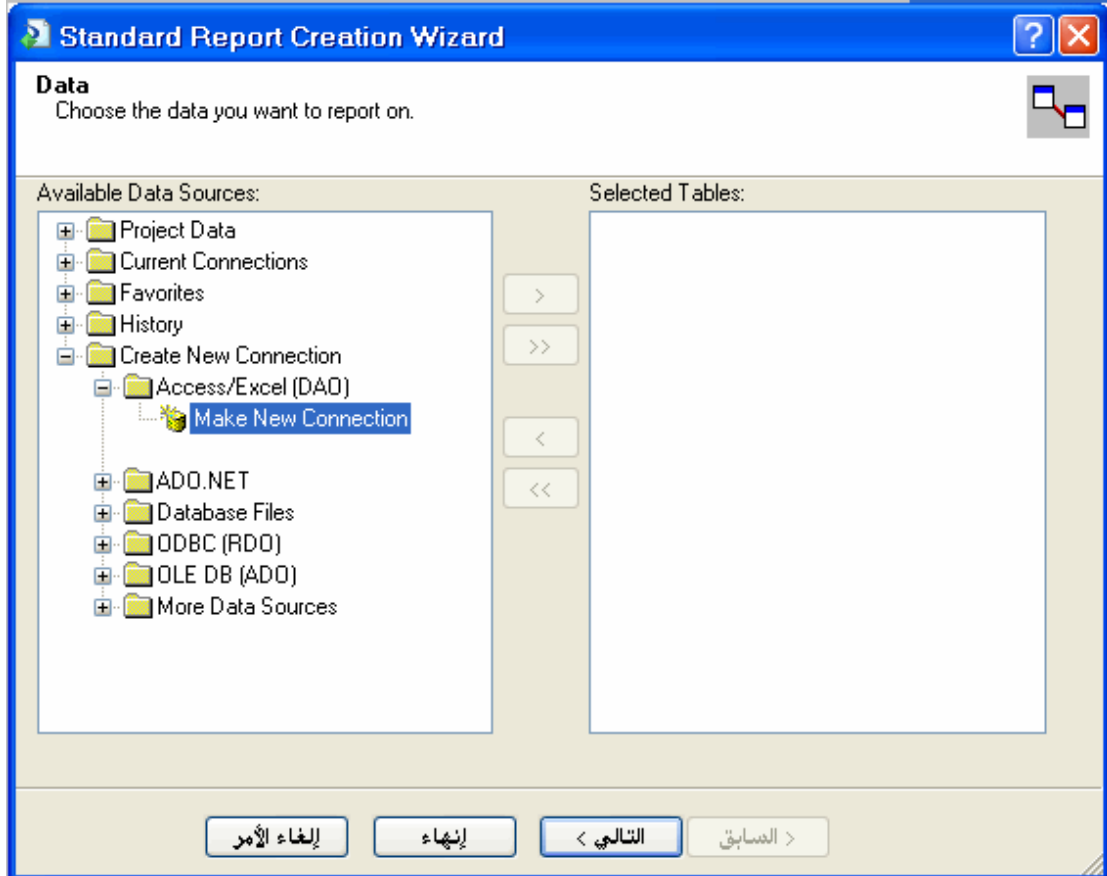
وبعدها انقر على الرز Ok فتطالعك النافذة التالية :



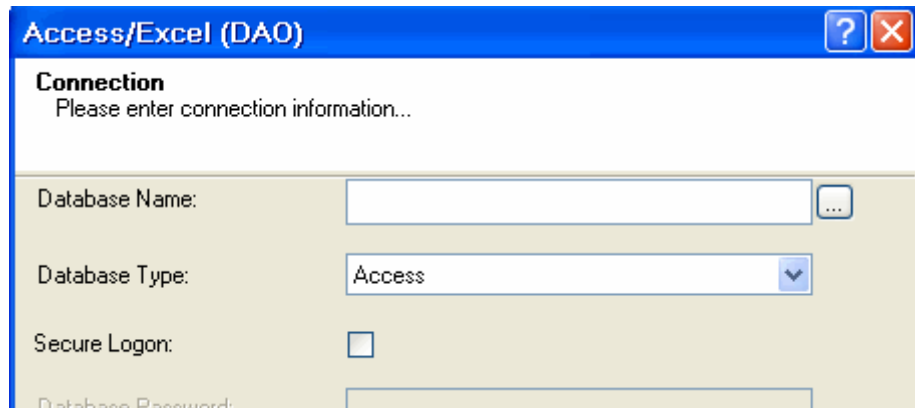
كما ترى يمكننا من هنا الان تحديد الجدول الذي نريد ان نتعامل معه.

الفصل الثامن قواعد البيانات

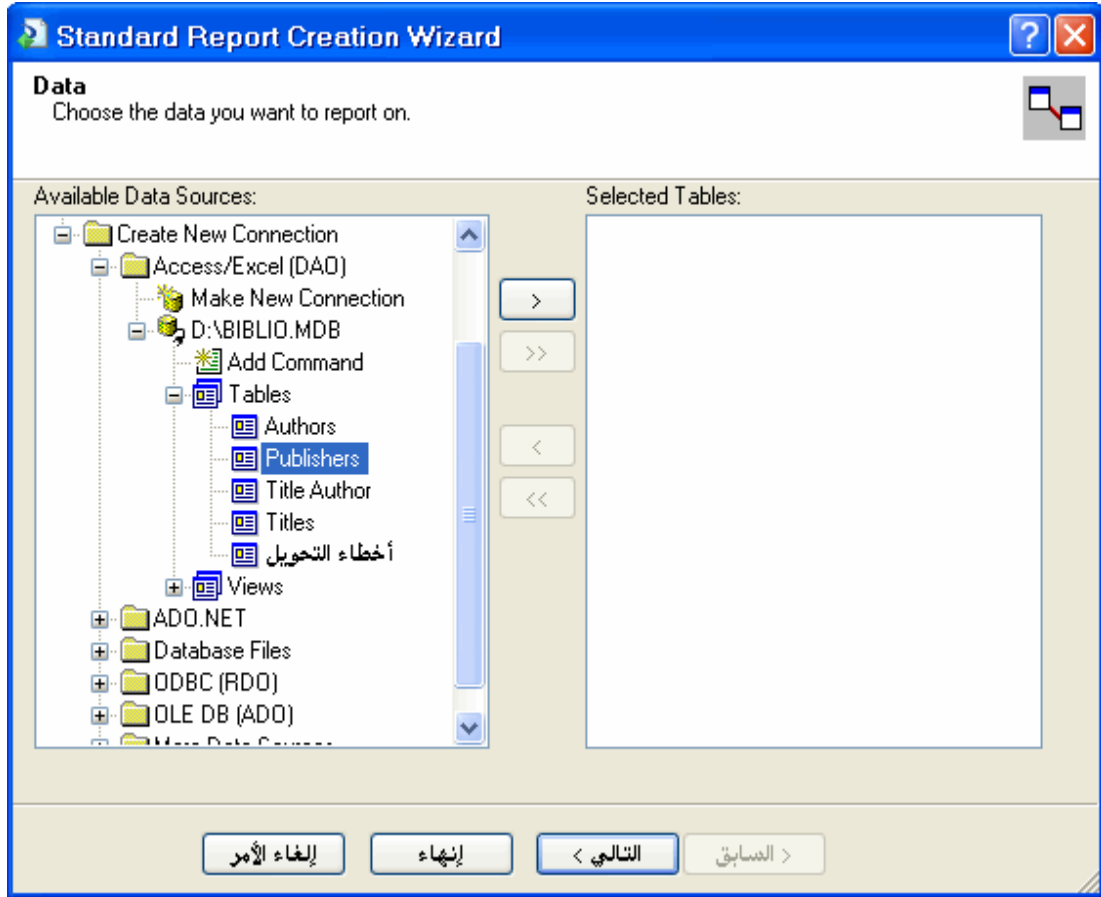
ولنعمل ذلك نقوم بتحديدته كما بالصورة تم النقر على الزر الى ناحية اليمين ويحمل الرمز > وهو سيقوم بنقل الجدول المطلوب. كما يمكنك نقل أكثر من جدول بحال كنت ستتعامل مع أكثر من جدول.
وإلا وقبل أن أكمل ماذا لو أننا لم نتصل بقاعدة البيانات كما فعلنا سابقاً...؟
عندها يمكننا من هنا تحديد الخيار Create New Connection والذي سيقوم بعمل اتصال جديد.
أنظر للشكل التالي :



كما ترى وبما اننا نتعامل مع قاعدة بيانات Access حددنا الخيار Access/Excel وبعدها أنقر على الخيار Make New Connection نقرتين.
لتطالعك النافذة التالية :



حدد مسار قاعدة البيانات من Database Name تم أنقر أنهاء لتشاهد الجداول كما بالشكل التالي

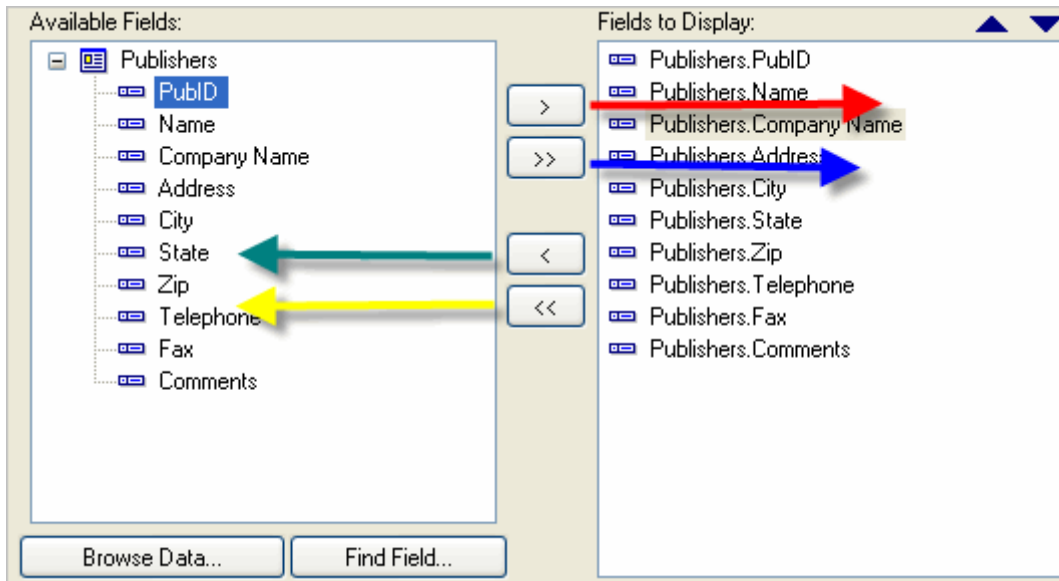


بنفس الطريقة يمكنك اضافة الجداول او الجداول التي تريد ان تتعامل معها.
والان بعد ان عدنا الى نفس المسار لنكمل الشرح .
طبعا سنحدد الجدول Publishers وننقر على التالي لتطالعنا النافذة التالية :

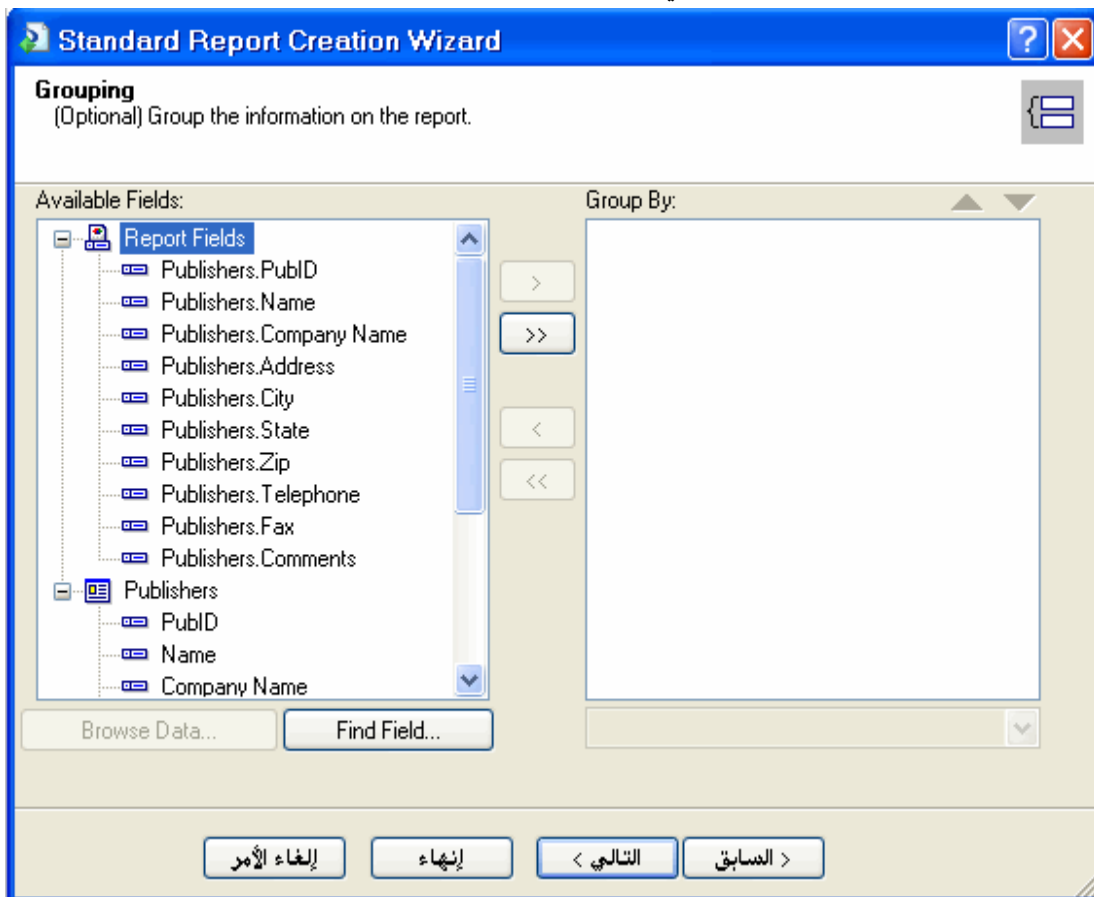


الفصل الثامن قواعد البيانات

المطلوب الآن تحديد الحقول التي نريدها أن تظهر بالتقرير عن طريق الأزرار التالية :



حدد الحقول المطلوبة وأنقر على الزر التالي لتطالعك هذه النافذة



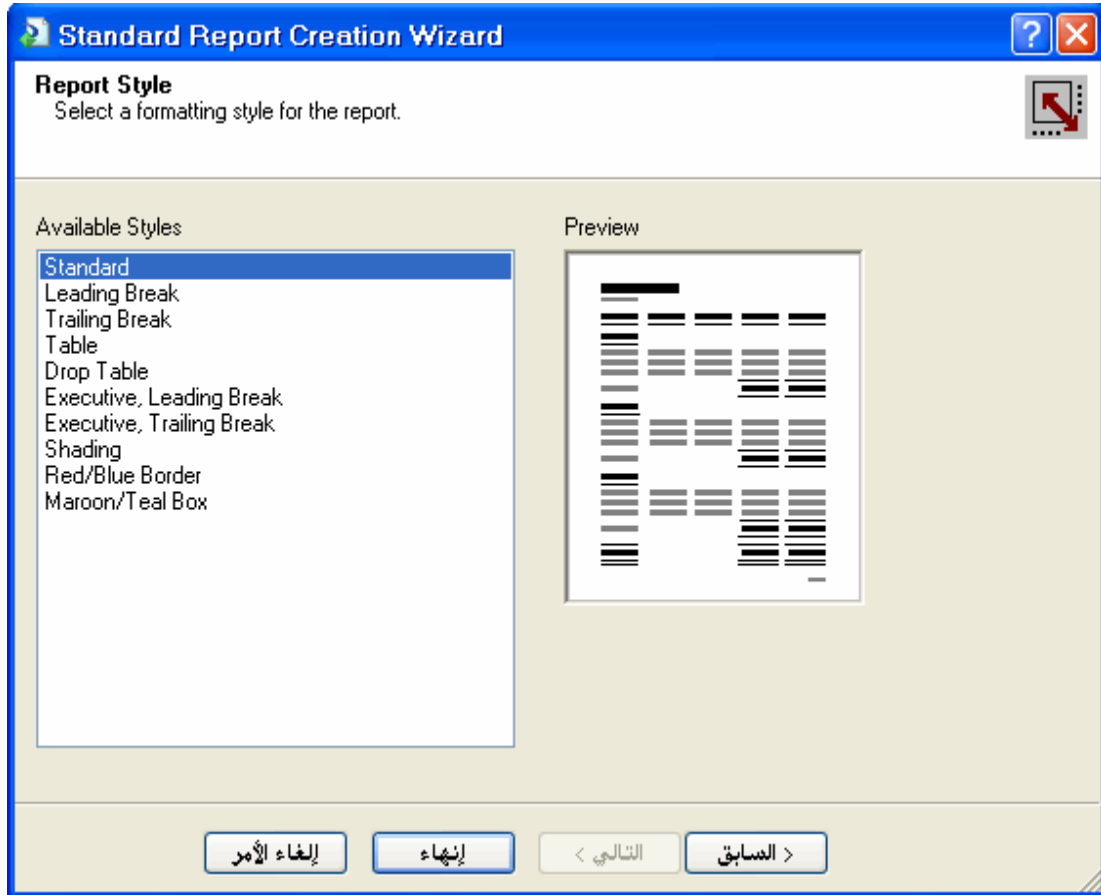
هذه النافذة تطالب بحقل ليتم استخدامه في تجميع البيانات وسوف تفهم لماذا لاحقاً .

على العموم هذا اختياري لك مطلق الحرية.

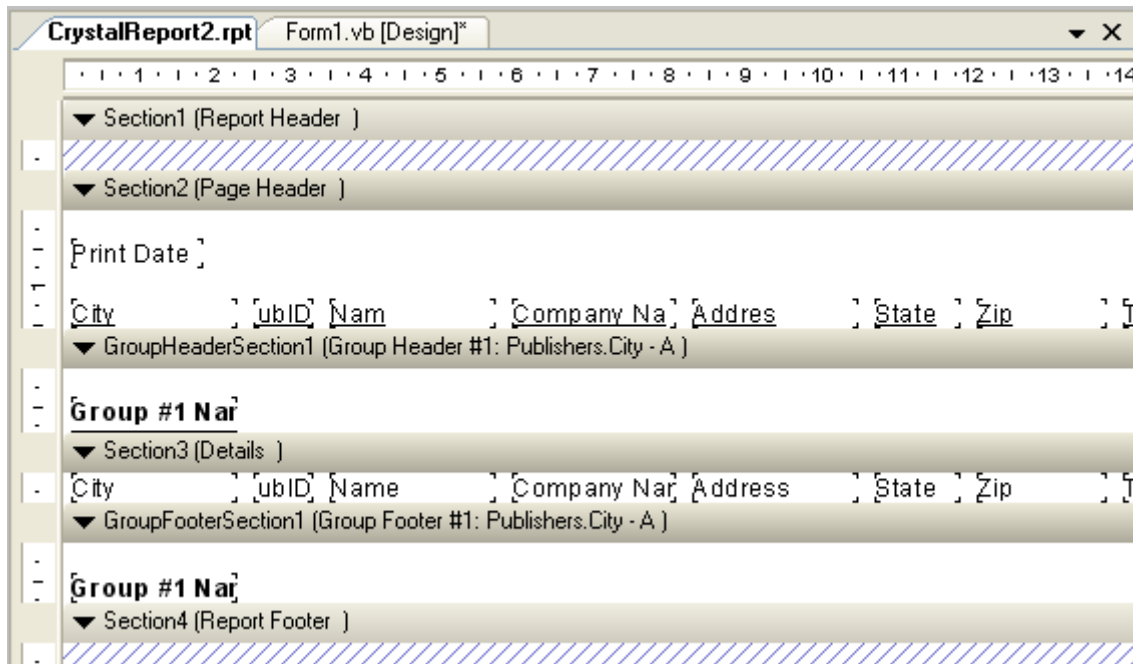
ونحن الآن سنحدد الحقل City وننقر على الزر التالي :

تم أنقر التالي مرة أخرى.

فتطالعك النافذة التالية :



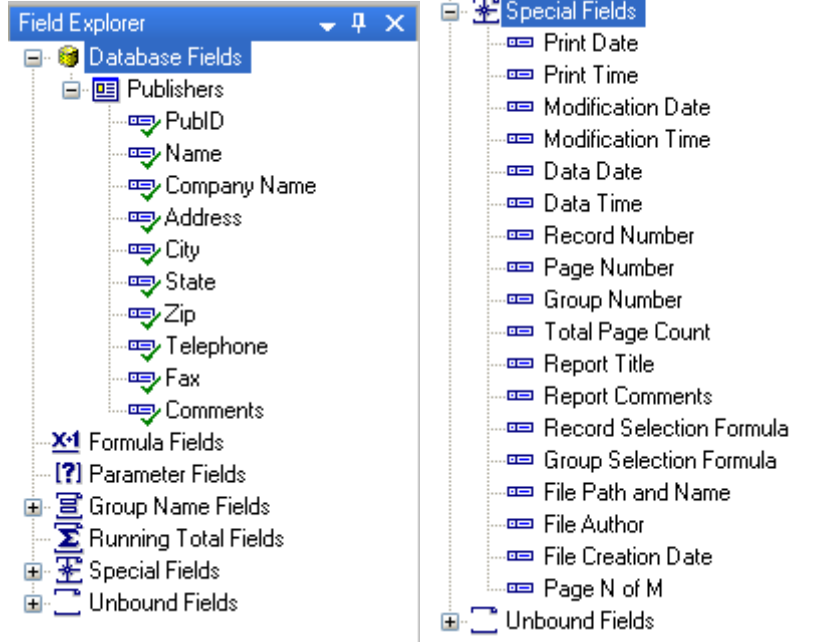
من هذه النافذة يمكنك تحديد شكل التقرير الذي تريده، وبعدها انقر إنهاء لتنتقل إلى مصمم التقارير كما بالشكل التالي :



الفصل الثامن قواعد البيانات

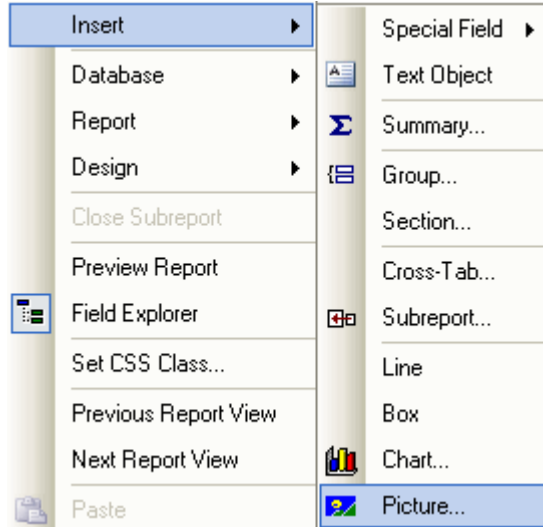
لاحظ الحقول والعناوين وكيفية ترتيبها وما الى ذلك، كل ما تحتاجه لتفهمها حركها واختبرها وانظر في خصائصها وسوف تجيد العمل معها.

إذا أردت إضافة أي حقل آخر ليس موجودا فيمكنك ذلك من النافذة Explorer Field كما بالشكل التالي :



كما تحت العنوان Special Fields يمكنك استخدام وأدراج حقول مخصصة في تقريرك كما بالشكل التالي :

كما إذا نقرت على التقرير بزر الفأرة الأيمن فستطالعك القائمة التالية



من الخيار Insert يمكنك ان تلاحظ ان هناك العديد من الأشياء التي من الممكن ان تستخدمها في تقريرك .

من Line وهو يسمح لك برسم الخطوط

Box لرسم مربعات.

Chart لإدراج واستخدام الرسومات البيانية.

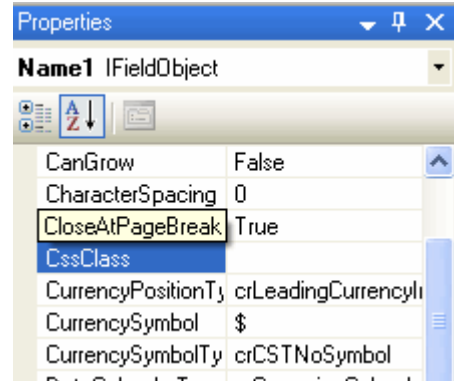
Picture لإدراج صورة في التقرير

طبعا الحديث عن التقارير بشكل مفصل يحتاج إلى كتاب لوحده.

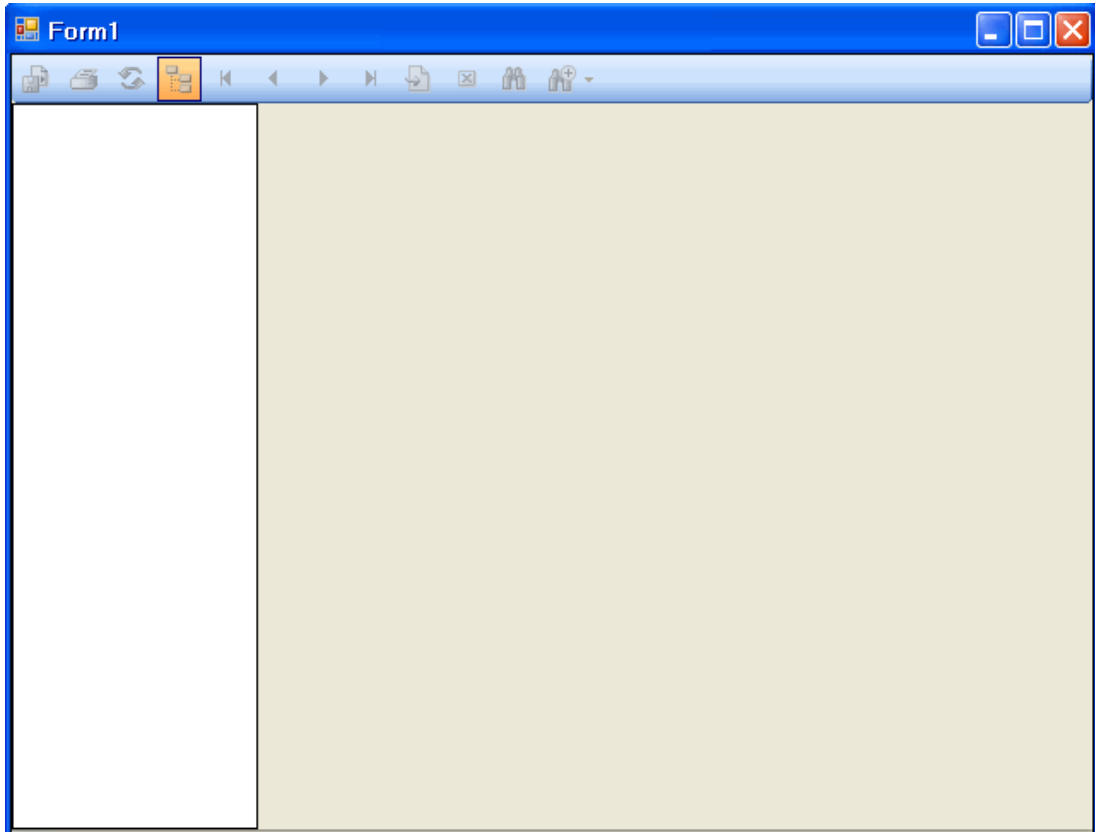
الفصل الثامن قواعد البيانات

ولكن على الأقل هذه بداية وإنشاء الله بعدد جديد سوف اطرح المزيد والمزيد من المعلومات والطرق والأساليب الأخرى.

كما توجد نافذة الخصائص والتي تتمكنك من ضبط وربط الكثير من الأدوات والحقول كما بالشكل التالي :



والآن تقريرنا جاهز للعمل وكل ما علينا هو الانتقال الى النموذج ومن شريط الأدوات نقوم بإسقاط Crystal Report Viewer على النموذج كما بالشكل التالي :



بعدها وحدث التحميل للنموذج نقوم بكتابة الكود التالي :

Code VB.Net

```
Imports System.Data.OleDb  
Public Class Form1
```

```
Private Sub Form1_Activated(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Activated  
Dim Cry As New CrystalReport2  
Dim Str As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
```

```
Source=BIBLIO.mdb"
Dim ds As New DataSet
Dim con As New OleDbConnection
Dim OleDbDataAdapter1 As New OleDbDataAdapter("select * from
Publishers", con)
Try
con.ConnectionString = Str
con.Open()
OleDbDataAdapter1.Fill(ds)
Cry.SetDataSource(ds.Tables(0))
CrystalReportViewer1.ReportSource = Cry
Catch ex As Exception
MsgBox(ex.Message)
Finally
con.Close()
End Try
End Sub
```

بداية إنشاء فئة جديدة من الكريستال بما انه مجرد كلاس عادي.

تم إجراء اتصال بقاعدة البيانات، ويمكنك مراجعة الدروس الخاصة بالـ Ado.net تم إسناد الجدول المحدد إلى التقرير بواسطة الخاصية Datasource وكذلك ربط عارض التقارير بالفئة الجديدة والتي تم إنشائها قبل قليل. بالنهاية عند تشغيل البرنامج أمكننا مطالعة التقرير كما بالشكل التالي :

City	PubID	Name	Company Narr	Address	State	Zip
	3	M & T	M & T BOOKS			
	4	MIT	MIT PR			
	6	HIGHTEXT PU	HIGHTEXT PU			
	7	SPRINGER VI	SPRINGER VI			
	11	SINGULAR	SINGULAR PL			
	12	Duke Press	Duke Press			
	13	Oxford Univers	Oxford Univers			
	14	Mit Press	Mit Press			
	15	CAMBRIDGE	CAMBRIDGE			
	16	Q E D	Q E D PUB C			
	17	Cambridge Un	Cambridge Un			
	18	WORLD SCIE	WORLD SCIE			
	19	IDG	IDG BOOKS V			
	20	GOWER PUB	GOWER PUB			
	22	MCGRAW HIL	MCGRAW HIL			
	23	ZIFF DAVIS	ZIFF DAVIS			
	24	ORACLE PR	ORACLE PR			
	25	CORIOLIS GR	CORIOLIS GR			
	26	Prentice-Hall	Prentice-Hall			
	28	INTL THOMSC	INTL THOMSC			
	30	MARCEL DEK	MARCEL DEK			

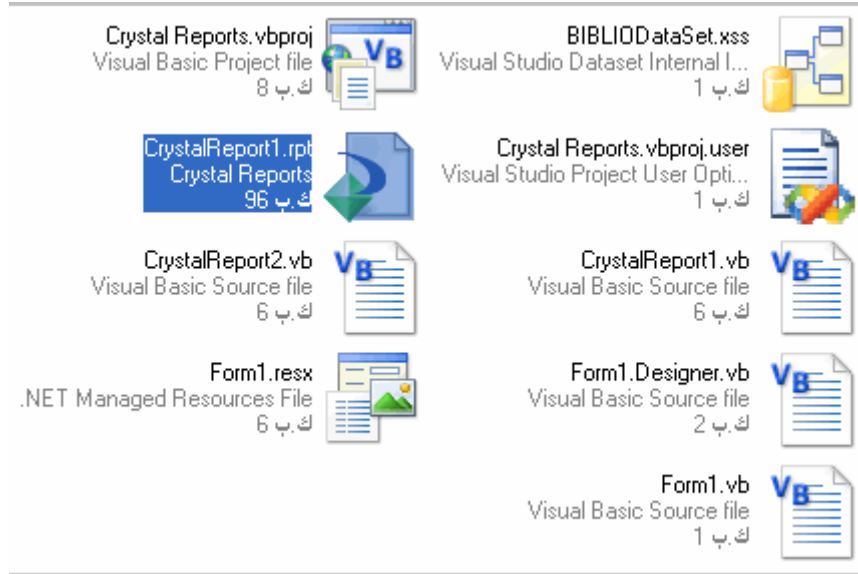
هل تتذكر عندما كنا نستخدم المعالج في إنشاء هذا التقرير...؟

طلب منا اختيار حقل ليتم إنشائه لتجميع البيانات.

وحددنا وقتها الحقل City ولهذا يمكنك ان تشاهد الى ناحية اليسار قائمة بها تجميع حسب المدينة.

الفصل الثامن قواعد البيانات

بمعنى آخر إذا اخترت مثلاً New York فسيتم عرض كل البيانات التي فيها المدينة تساوي New York طبعاً يمكنك التحكم في كم البيانات عن طريق جملة SQL من خلال الكود للقيام بفرز البيانات المطلوبة. وأيضاً إذا ذهبت إلى مجلد المشروع وجدت ملف التقرير كما بالشكل التالي :



أنه الملف المحدد ويحمل الاسم CrystalReport1 حقيقة هذا الملف يجب ان يكون بمجلد المشروع لا يمكن الاستغناء عنه.

Code VB.Net

عرض تقرير مباشرة

```
Dim cry As New RPover
CrystalReportViewer1.ReportSource = cry
CrystalReportViewer1.Zoom(50)
```

Code VB.Net

عرض تقرير مخصص

```
Dim cry As New RClasse
Dim Rst1 As New ADODB.Recordset
XClasse = InputBox("ادخل القسم")
If XClasse = "" Then Exit Sub
Rst1.CursorLocation = ADODB.CursorLocationEnum.adUseClient
Rst1.Open("Select * from Students where [Classe] = " & Xclasse & """, Con,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
cry.SetDataSource(Rst1)
CrystalReportViewer1.ReportSource = cry
CrystalReportViewer1.Zoom(50))
```

Code VB.Net

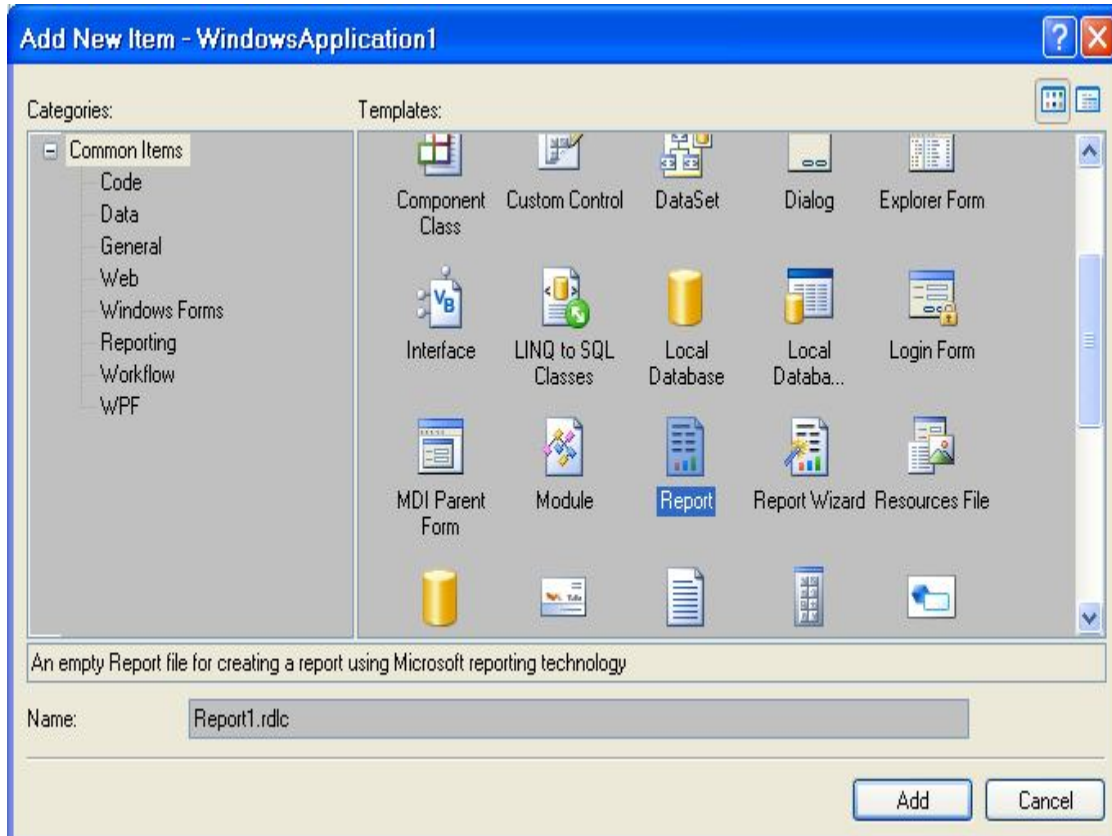
عرض تقرير مخصص

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
Dim rpt As New CrystalReport1
Dim X As String = InputBox("z")
If X = "" Then Exit Sub
```

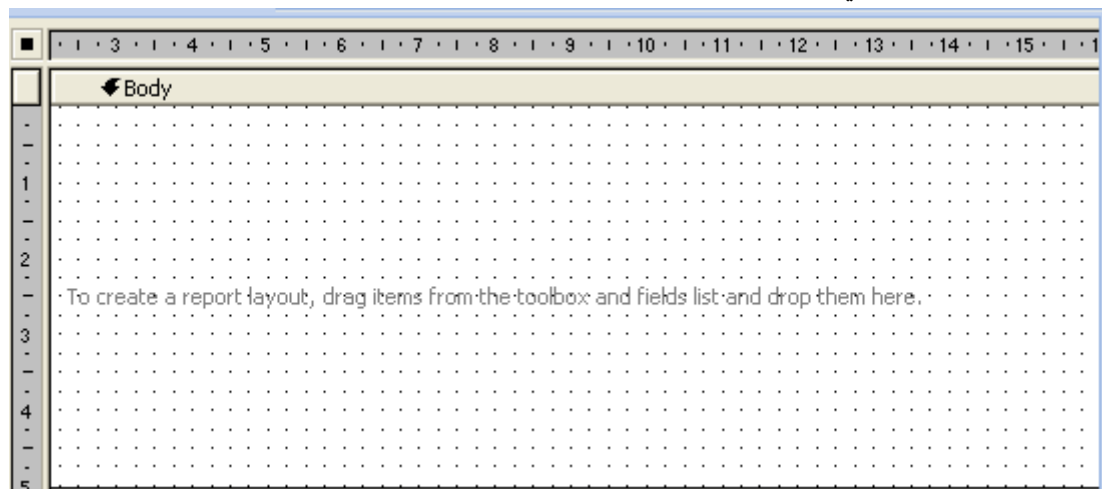
```
rpt.RecordSelectionFormula = (" {Students.اسم} = '" & X & "'")
CrystalReportViewer1.ReportSource = rpt
CrystalReportViewer1.Zoom(50)
CrystalReportViewer1.Refresh()
End Sub
```

تقرير Report

قم بإنشاء مشروع جديد ثم قم بإضافة زر و أداة التقرير Report و ذلك من قائمة Project ثم اختر Add New Item تظهر النافذة التالية اختر منها Report ثم اضغط Add



يظهر شكل التقرير كما يلي :



و يمكنك التعامل مع التقرير من خلال قائمة جديدة Report

الفصل الثامن قواعد البيانات

Report Properis : يمكنك من التحكم في شكل و

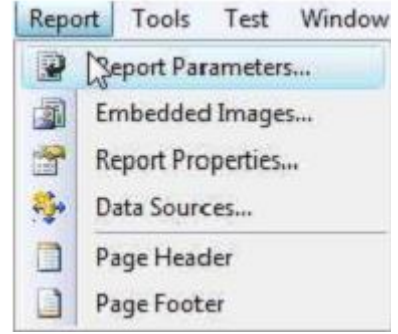
أبعاد التقرير

Embedded Image : لإضافة صورة

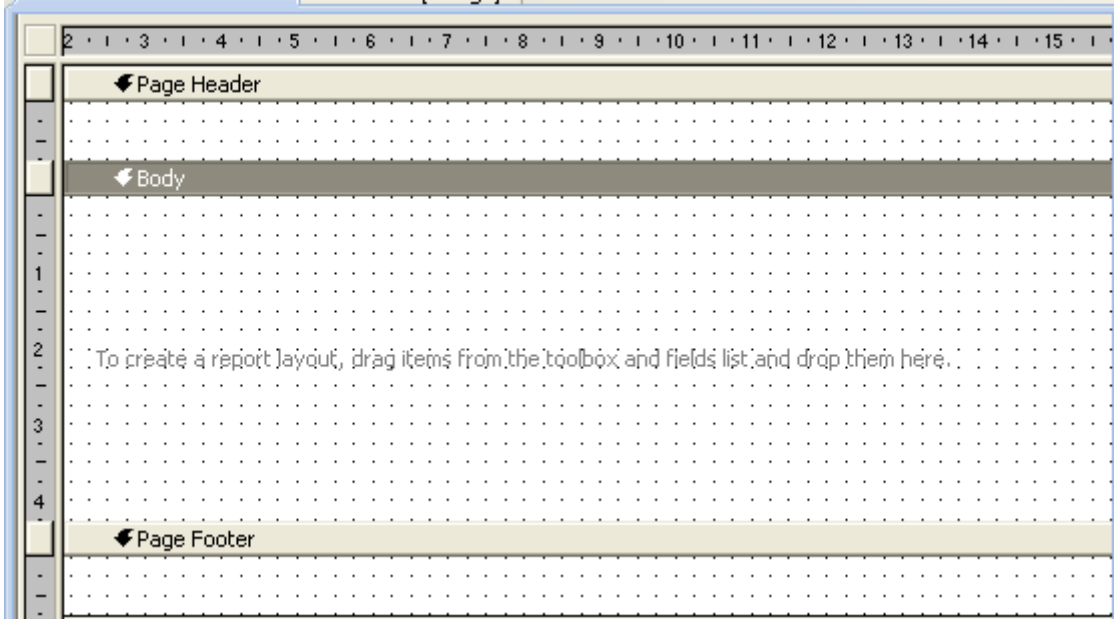
: Data Source

Page Header : إضافة رأس التقرير

Page Footer : إضافة تذييل التقرير

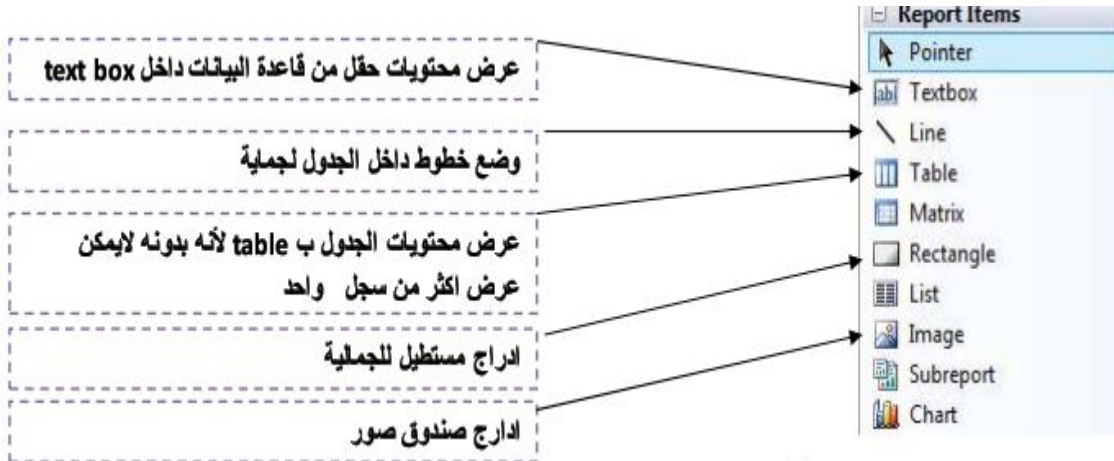


قم بإضافة رأس و تذييل التقرير ليبدو الشكل :



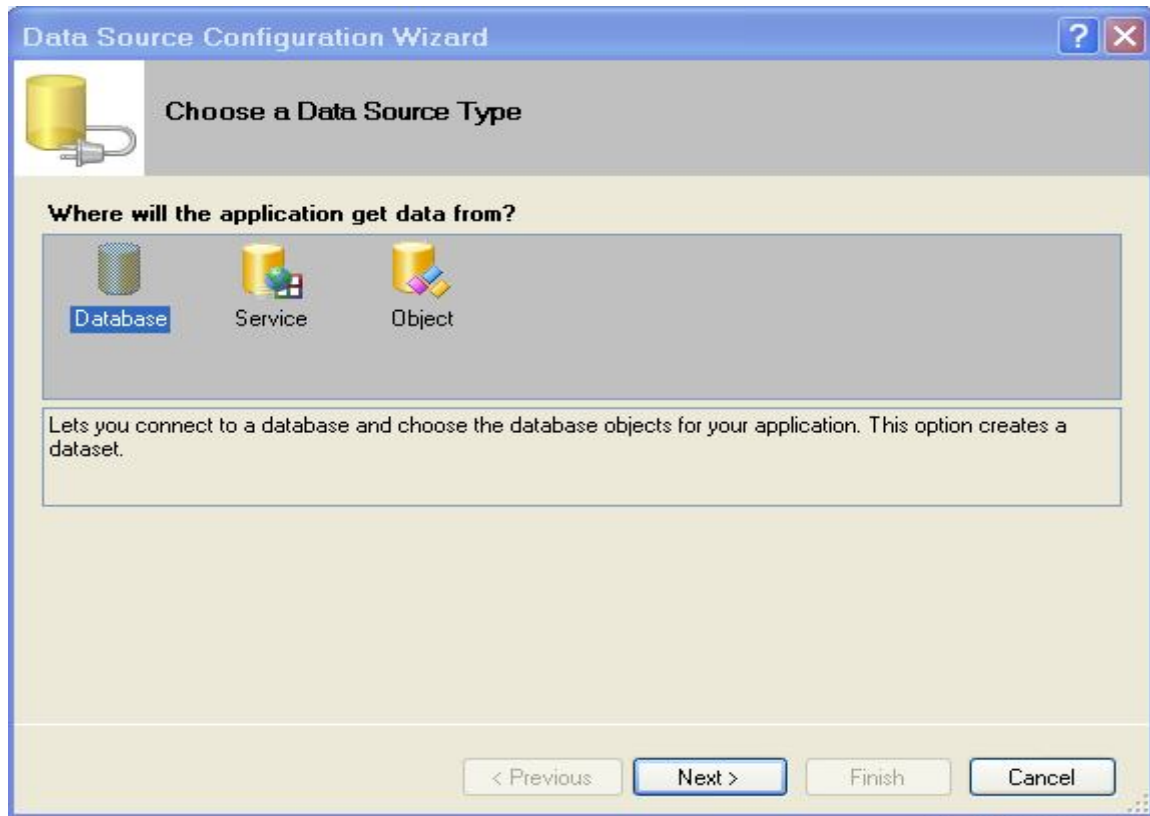
ملاحظة : يمكنك التحكم في أبعاد التقرير بسحب الحد الفاصل بين أجزاء التقرير للأعلى أو الأسفل

كما توجد أدوات خاصة بالتقرير تظهر تلقائياً بمجرد إضافة التقرير كما في الشكل

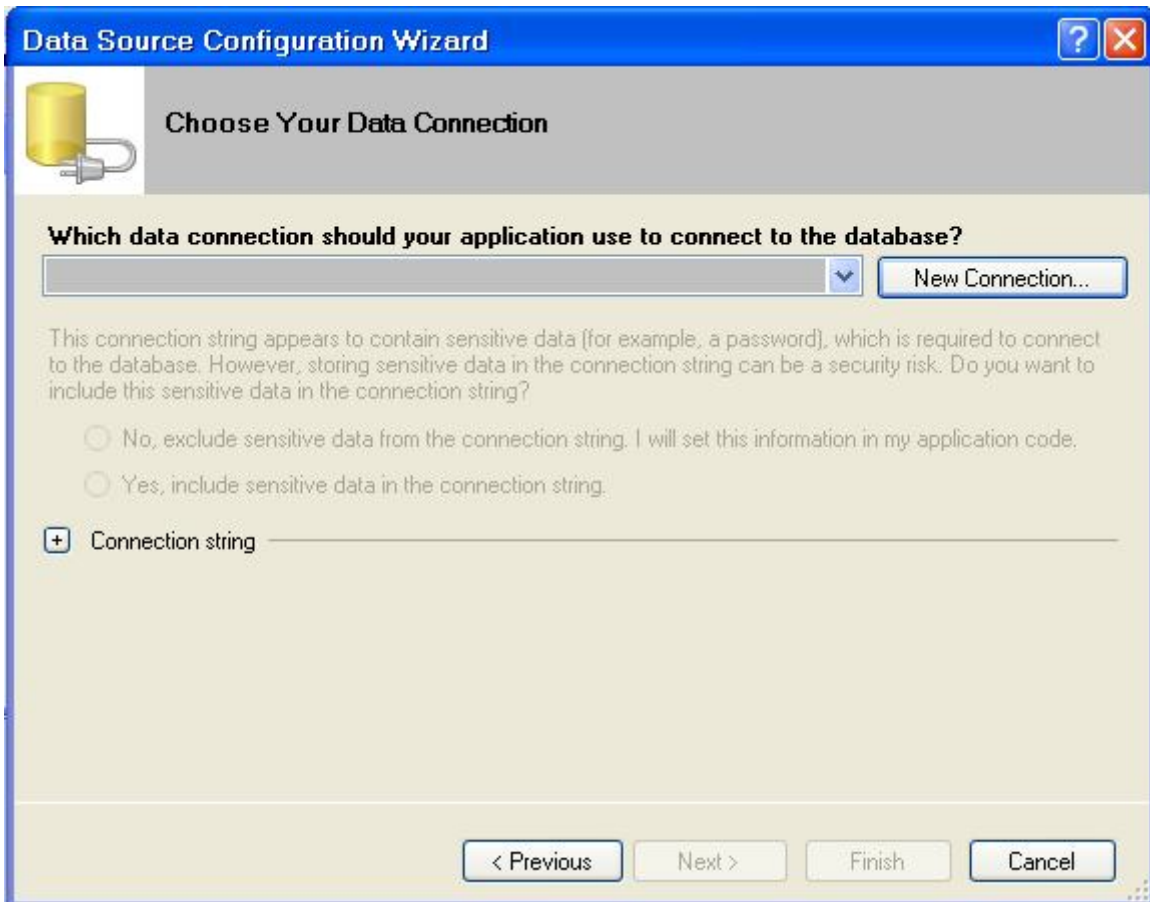


نقوم بإضافة قاعدة البيانات إلى المشروع و هذا من قائمة Data نختار Add New Data Source

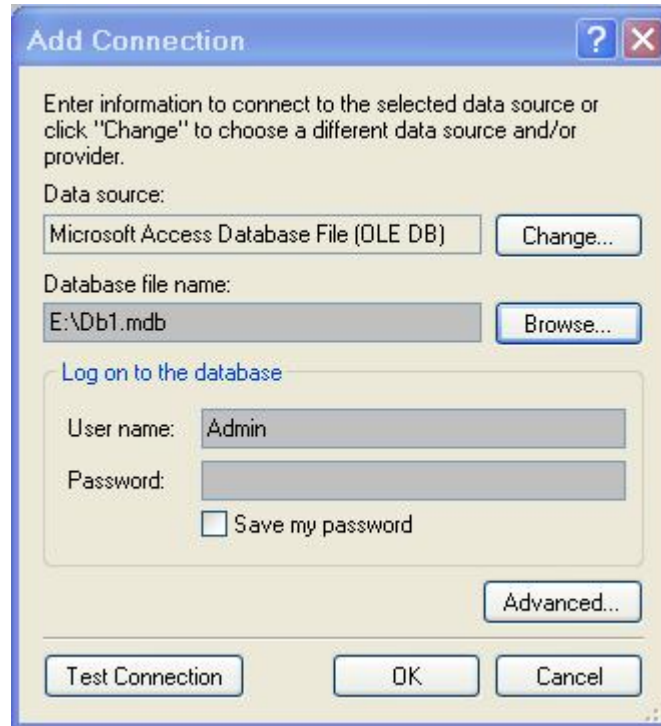
تظهر النافذة التالية :



انقر Next



انقر New Connection



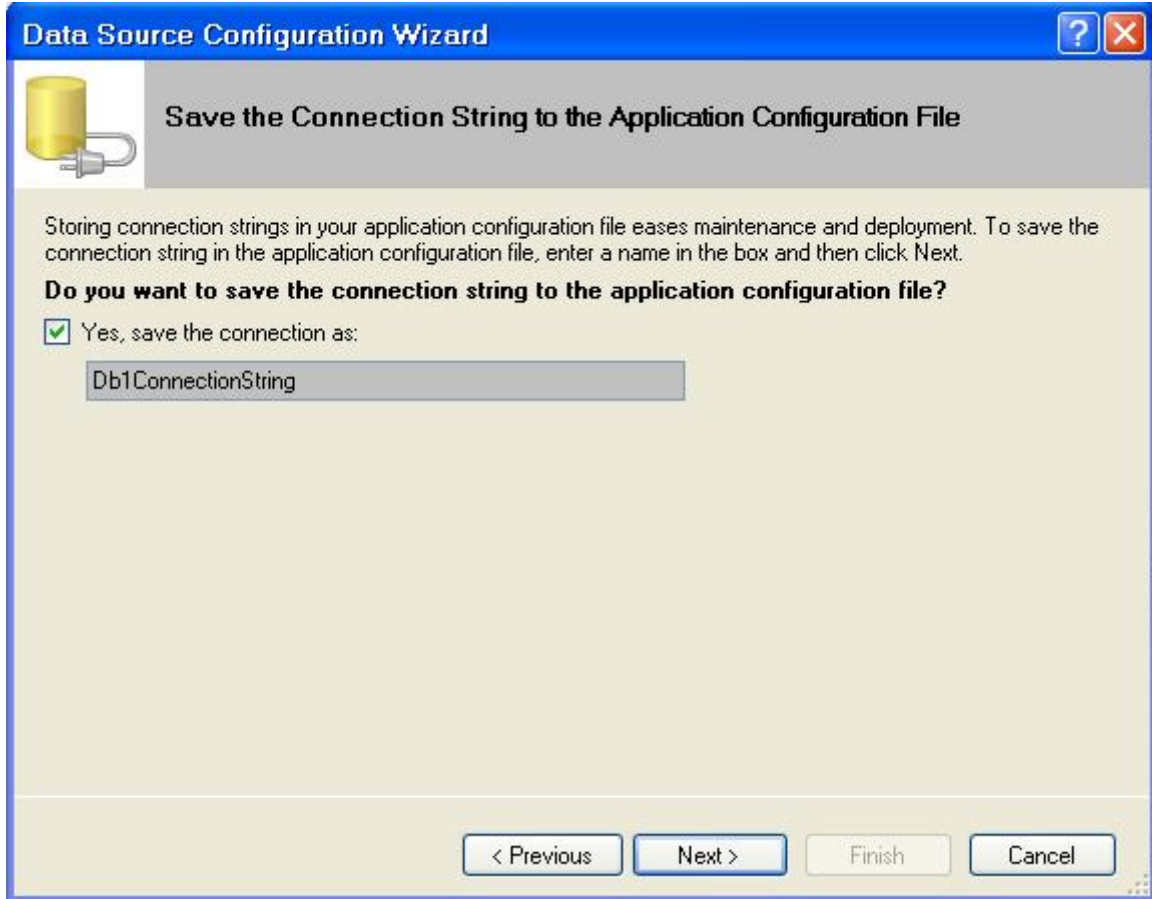
في خانة الاسم نكتب مسار القاعدة أو نختار Browse لتحديد مسار القاعدة بعد تحديد المسار نضغط Ok



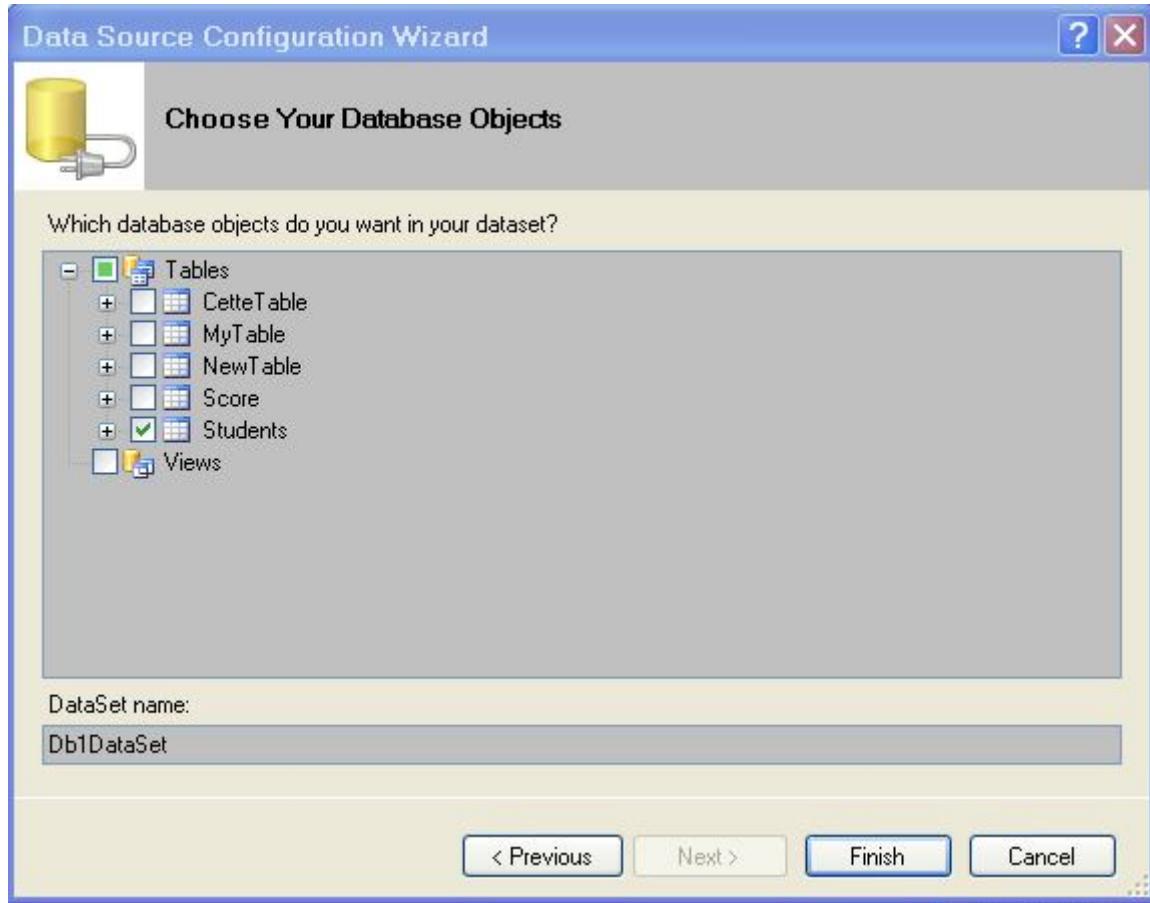
انقر Next



انقر نعم



انقر Next



اختر الجدول من القائمة وليكن Students ثم انقر Finish و لاحظ أنه تم إضافة القاعدة إلى المشروع و ظهورها في مستكشف الحل بعد ذلك نضيف أداة الجدول Table إلى التقرير

	Header	
	Detail	
	Footer	

و يتكون من الأجزاء التالية :

Header : رأس الجدول و يحتوي عناوين الأعمدة

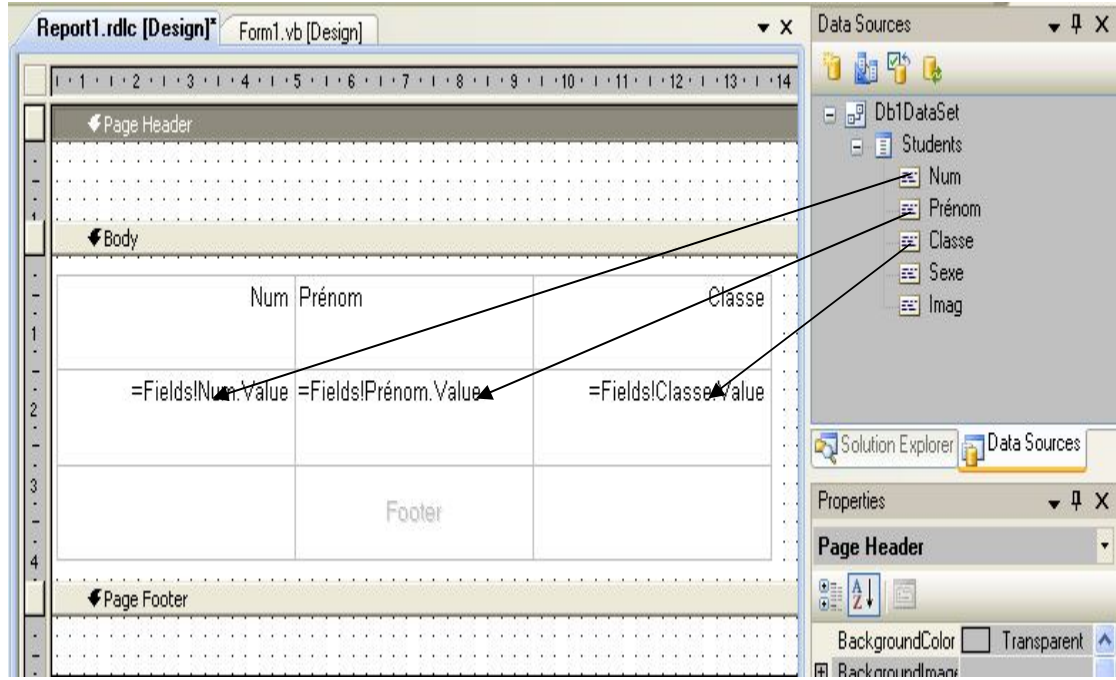
Detail : و هي الصفوف التي ستظهر في الجدول

Footer : تنليل الجدول

و تستطيع زيادة أو إنقاص أبعاد الأعمدة بسحب الفواصل بينها .

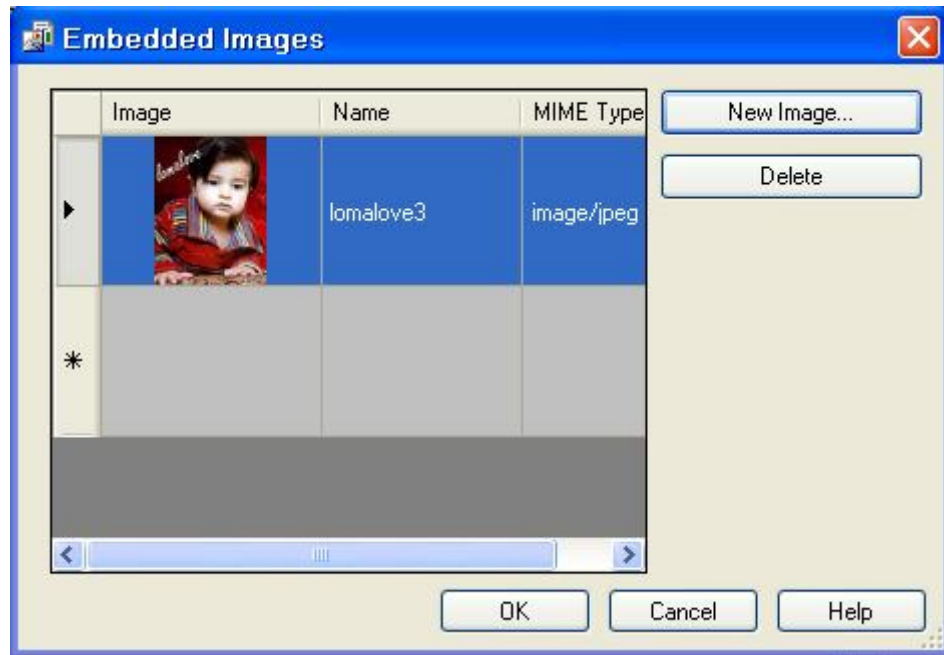
أما إذا أردت زيادة أو حذف أعمدة انقر على أحد الأعمدة في الصف الثابت في الأعلى بزر الفأرة الأيمن و الآن نضيف الحقول من قاعدة البيانات من نافذة مستكشف الحل إلى الجدول بالسحب و الإفلات انظر الشكل .

الفصل الثامن قواعد البيانات



لاحظ أنه بمجرد إضافة الحقول إلى الجدول في جزء Detail تضاف عناوين الأعمدة بعنوان الحقل و الآن نضيف عنوان التقرير في رأس التقرير كما يلي :

من قائمة Report اختر Embedded Image تظهر النافذة



اختر New Image يظهر مربع حوار فتح منه تختار الصورة المناسبة ثم اضغط Ok و الآن نضيف أداة Image من صندوق أدوات التقرير . و نغير خصائصها إلى :

الخاصية	القيمة
Source	Embedded
Value	اختر اسم الصورة

الفصل الثامن قواعد البيانات



نقوم بتغيير أبعاد الصورة و التقرير حتى نحصل على الشكل المناسب
و الآن نضيف عنوان التقرير في رأس التقرير وهذا بإضافة أداة TextBox إلى رأس التقرير ثم نكتب
العنوان كما يلي :

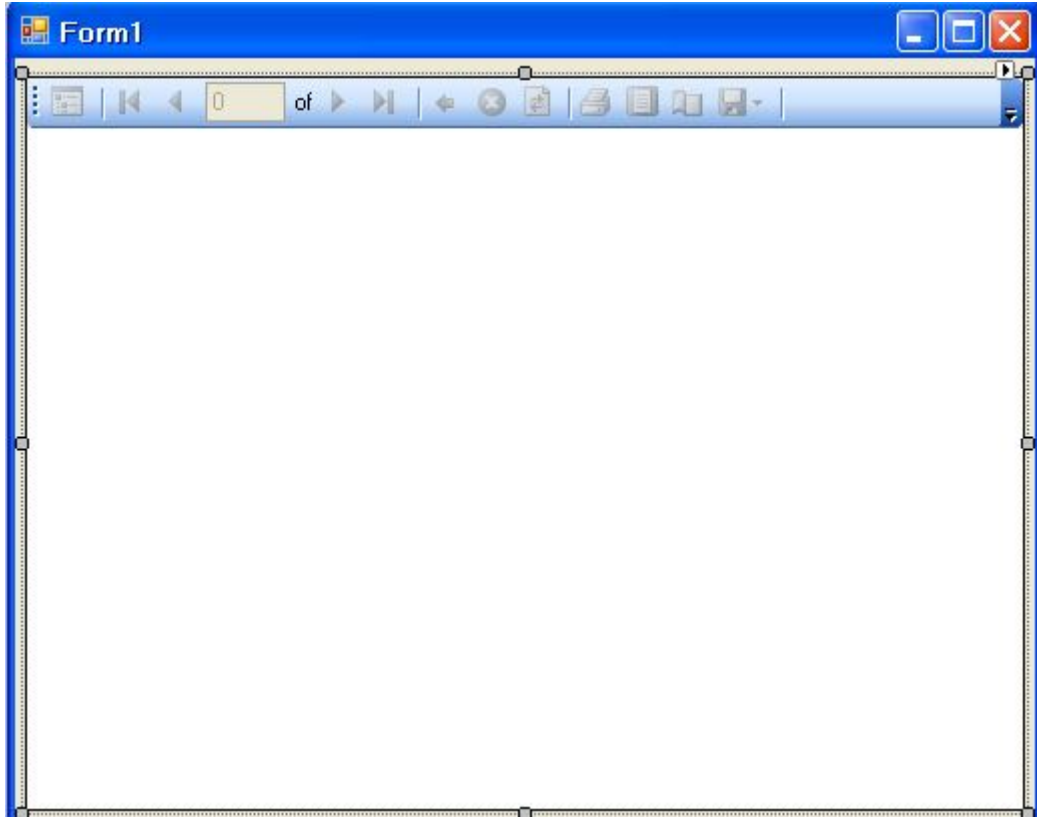


أما إذا أردت ربط مربع النص TextBox بحقل من حقول قاعدة البيانات من الخاصية Value نختار الحقل



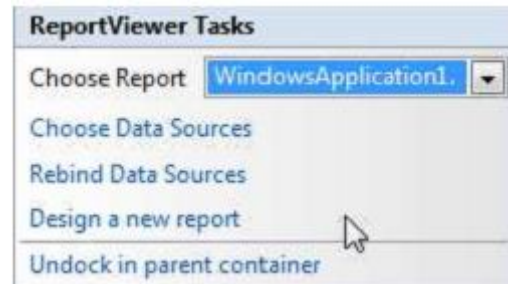
الفصل الثامن قواعد البيانات

بعد أن ربطنا أدوات التقرير بقاعدة البيانات يبقى أن نعرض التقرير .
نعود إلى مستكشف الحل و نضغط على نافذة التصميم Form ثم نقوم بإضافة أداة عرض التقارير Report Viewer إلى النموذج

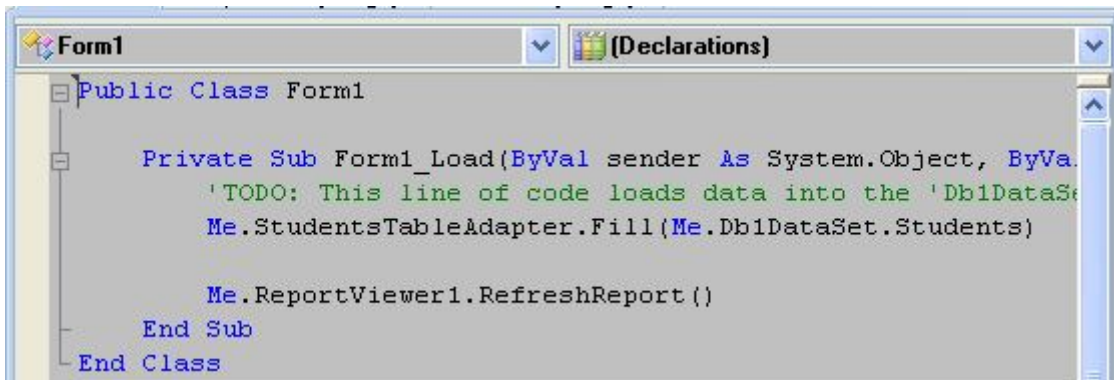


انقر السهم الصغير لأداة العرض بجوار علامة الإغلاق تظهر النافذة

اختر Choose Report لاختيار التقرير الذي
سيتم عرضه



سيقدم المترجم كودا تلقائيا يقوم بتحميل التقرير و تحديثه بالبيانات



الآن أصبح تقريرنا جاهزا للعرض . قم بتشغيل البرنامج و لاحظ الناتج ! ألف مبروك !



تصفية و البحث :

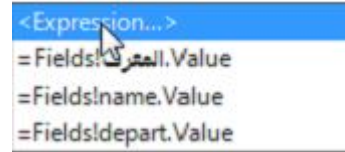
لعرض سجل أو مجموعة من السجلات من جدول موافقة لشرط أو مجموعة من الشروط لحقل أو لحقول نقوم بعملية تصفية الحقل RowFilter كما يلي :

Code VB.Net

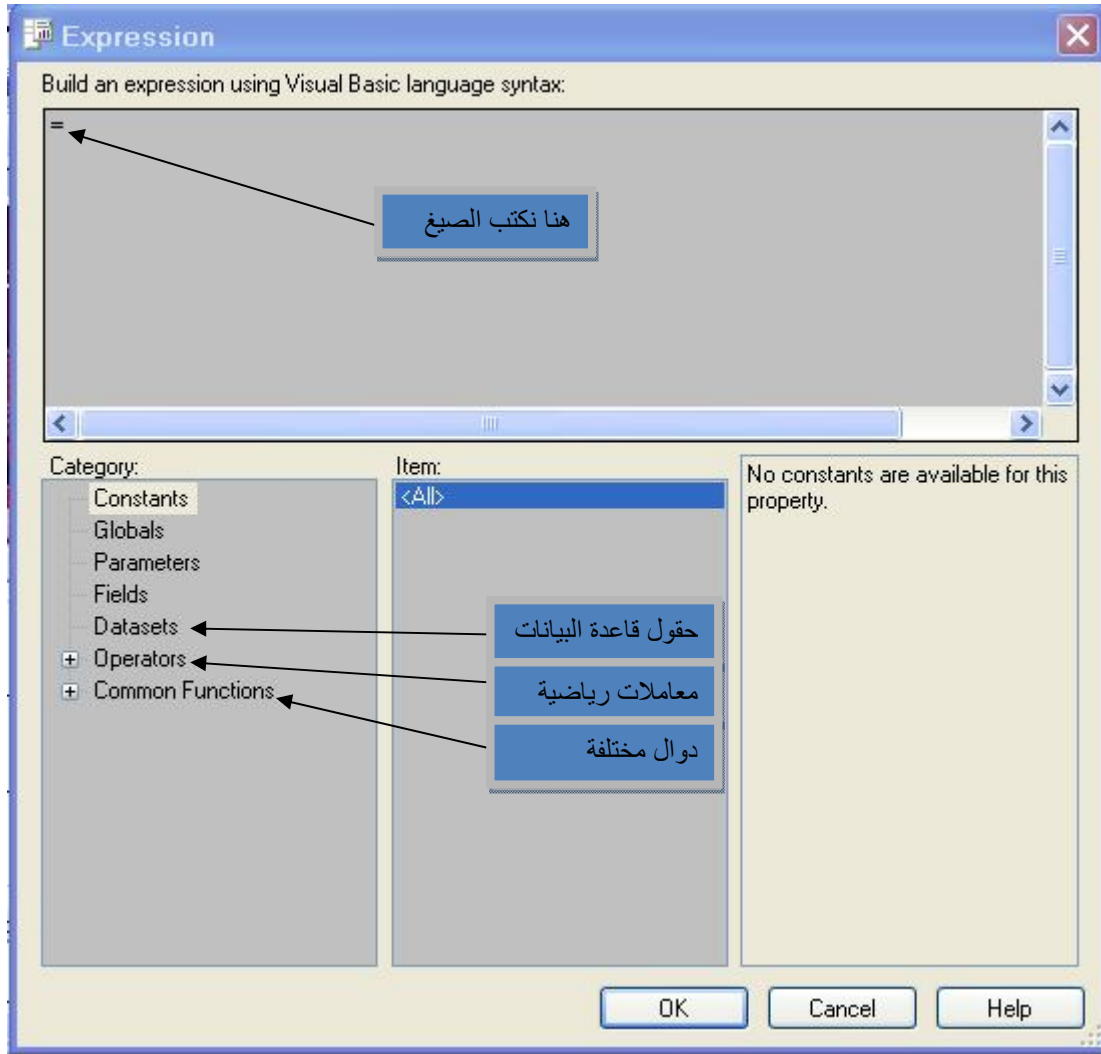
```
Me.StudentsTableAdapter.Fill(Me.Db1DataSet.Students)
Dim X As String = InputBox("الاسم ادخل")
Db1DataSet.Students.DefaultView.RowFilter = "Prénom =" & X & ""
StudentsBindingSource.DataSource = Db1DataSet.Students.DefaultView
Me.ReportViewer1.RefreshReport()
```

استعمال الدوال

قم بإضافة مربع نص textbox نحدد الخاصية Value تظهر قائمة اختر Expression



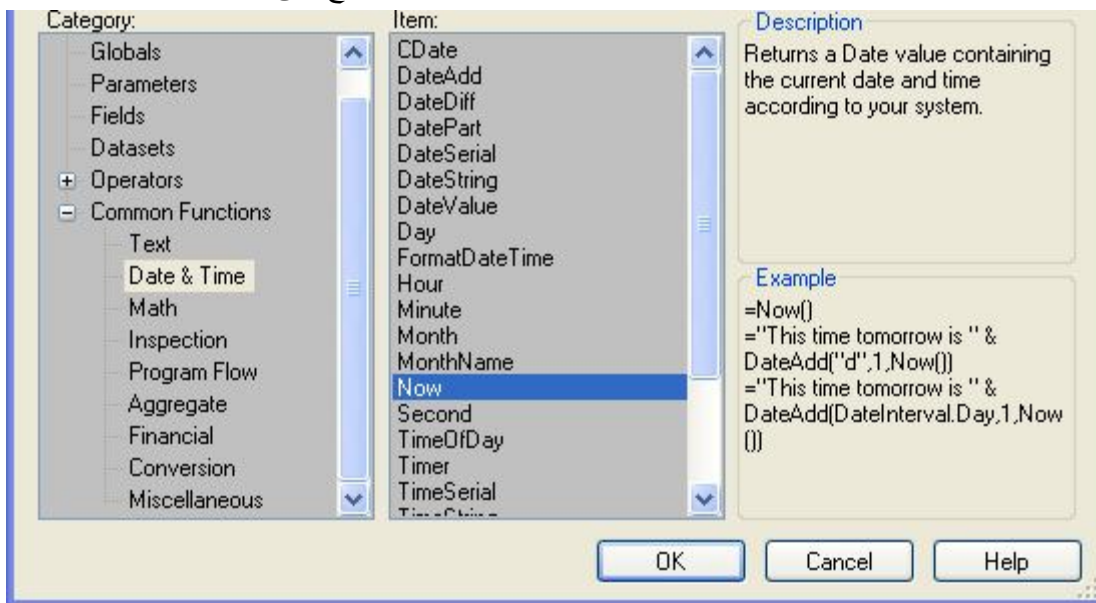
تظهر النافذة التالية



نكتب الصيغة التالية : =Count(Fields!Prénom.Value)

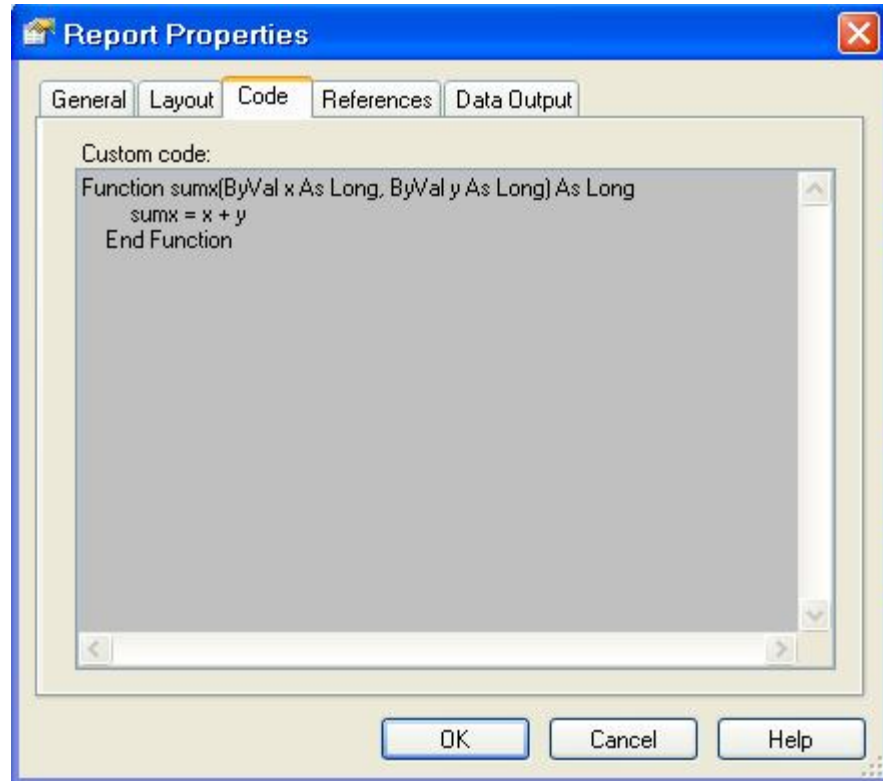
تقوم بحساب عدد أسماء الطلاب

حاول أن تضيف صيغا أخرى من Common Functions و لاحظ النموذج على يمين النافذة



دوال مخصصة

من القائمة Report نختر Report Properties ثم نختر التبويب Code و نكتب التعليمات



ولاستدعاء هذه الدالة نضيف Textbox من الخاصية value ثم Expression ثم نكتب الصيغة
=Code.sumx(1,2)

التعامل مع مكتبة (ActiveX Data Object) ADO

مع كل ما يمنحه لك الكائن ADO.Net من التقنيات و العمليات على قواعد البيانات .إلا أنه يمكن استعمال تقنية ADO القديمة و التي ظهرت مع الإصدار VB6

خطوات التعامل مع الكائن ADO

أولا : إنشاء قاعدة البيانات عن طريق Access أو SQL Server

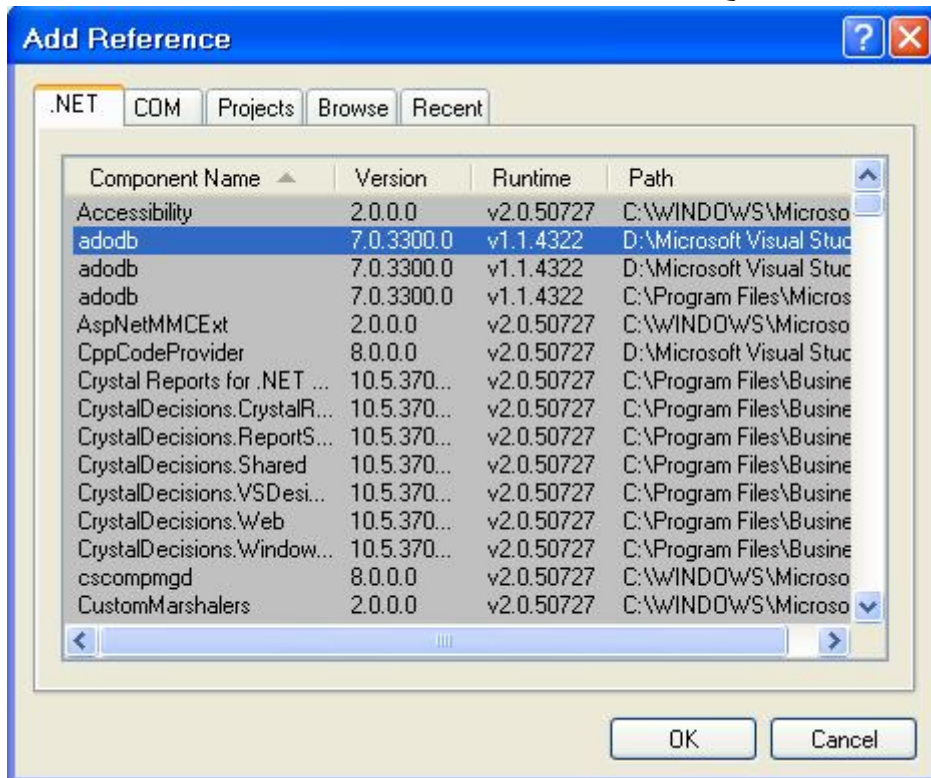
وهذا نعرفنا عنه مع الكائن ADO.Net

ثانيا : إنشاء واجهة التصميم

قم بتصميم النموذج على النحو التالي :



- ثالثاً : إضافة المرجع الخاص بكائن ADO لأنه من التقنيات القديمة COM كما يلي :
- من قائمة مشروع Project اختر Add References تظهر النافذة التالية :



- حدد الخيار adodb كما في الشكل ثم انقر Ok

الاتصال بقاعدة البيانات

نقوم بتعريف متغير عام يمثل كائن الاتصال في قسم التصريح العام للنموذج أو في الوحدة النمطية حتى تتمكن من استخدامه في كامل البرنامج يقوم بالاتصال بقاعدة البيانات

Code VB.Net

```
Dim con As New ADODB.Connection ' متغير خاص كائن الاتصال
```

الفصل الثامن قواعد البيانات

نحتاج لعملية الاتصال عنوان أو نص الاتصال `ConnectionString` الذي يتطلب تحديد اسم و مسار قاعدة البيانات ، المستخدم ، كلمة مرور...
و لكتابة نص الاتصال الخاص ب `ADO` و هو نفسه نص الاتصال الخاص ب `ADO.NET` (راجع الدرس)
ربط كائن الاتصال بنص الاتصال

Code VB.Net

```
Dim StrCon As String = "Provider =Microsoft.jet.oledb.4.0;" & _  
"Data source = E:/db1.mdb" ' نص الاتصال  
Con.ConnectionString = StrCon ' إرفاق نص الاتصال بكائن الاتصال
```

بعد الاتصال بقاعدة البيانات يبقى أن نفتح القاعدة

فتح وغلق القاعدة

لفتح القاعدة استعمل الطريقة `Open`

Code VB.Net

```
Try  
Con.Open() ' فتح القاعدة  
Catch ex As Exception  
MsgBox(ex.Message)  
End Try
```

ملاحظة : يمكنك إرفاق نص الاتصال بكائن الاتصال عند فتحه

`Con . Open (StrCon)`

و لغلق القاعدة استعمل الوسيلة `Close` أو `Dispose` لغلق الاتصال وتحرر أي مساحات كانت تشغلها في الذاكرة.

فتح القاعدة عن طريق مربع الحوار فتح

Code VB.Net

```
With CommonDialog1  
 . Title = "فتح قواعد البيانات"  
 . Filter = "All datbase (*.mdb)|*.mdb"  
 . ShowOpen  
 If .FileName = "" Then  
 Exit Sub  
 Else  
 con.Open ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source= " & .FileName)  
 End If  
  
 If con.State Then  
 MsgBox "تمت عملية الفتح بنجاح "  
 End If  
End With
```

فتح جدول

الفصل الثامن قواعد البيانات

بعد فتح القاعدة نقوم بفتح جدول أو عدة جداول
قبل كل شيء نقوم بتعريف متغير يحمل بيانات الجدول في قسم التصريح العام أو Module أسفل متغير
Con الخاص بالاتصال بقاعدة البيانات كما يلي :

 Code VB.Net

```
Dim Con As New ADODB.Connection
Dim Rst As New ADODB.Recordset 'متغير خاص بفتح الجدول'
```

الصيغة العامة لفتح جدول

 الصيغة العامة

`recordset.Open(Source, ActiveConnection, CursorType, LockType, Options)`

نوع التزامن , نوع المؤشر أو السجل , متغير قاعدة البيانات , "اسم الجدول" (`Rst.open`)
(خيارات , "المطلوب")

العامل الأول : اسم الجدول يشتمل على جملة استعلام مهيكلة SQL
متغير قاعدة البيانات : Con الذي تم تعريفه سابقا للاتصال بقاعدة البيانات
نوع المؤشر أو السجل : يشتمل على المرقم التالي :

العامل	الوظيفة
adOpenDynamic	تعامل مباشر من السجلات الحقيقية
adOpenForwardOnly	التحرك بين السجلات للأمام فقط
adOpenKeyset	للقراءة مع التعديل
adOpenStatic	للقراءة فقط

نوع التزامن المطلوب : و يشتمل على المرقم التالي :

العامل	الوظيفة
adLockOptimistic	غلق السجل على الآخرين أثناء التعديل
adLockPessimistic	غلق السجل أثناء التعديل فقط
adLockBatchOptimistic	غلق السجل عن الآخرين ثم إطلاع السرفور على التحديث

 Code VB.Net

```
Public Class Form1
Dim Con As New ADODB.Connection
Dim Rst As New ADODB.Recordset

Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" _
& "data Source =e:/db1.mdb"
Con.Open()
Rst.CursorLocation = ADODB.CursorLocationEnum.adUseClient
Rst.Open("Select * from Students", Con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
```

```
End Sub  
End Class
```

State - و للتأكد من نجاح عملية فتح الجدول نستعمل الطريقة

Close - و لغلقة نستعمل الطريقة

تفادي الاستثناءات

لتفادي الاستثناءات نستعمل الوسيلة Try كما يلي :

Code VB.Net

```
Try  
    Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" _  
        & "data Source =e:/db1.mdb"  
    Con.Open()  
    Rst.CursorLocation = ADODB.CursorLocationEnum.adUseClient  
    Rst.Open("Select * from Students", Con, ADODB.CursorTypeEnum.adOpenDynamic,  
        ADODB.LockTypeEnum.adLockOptimistic)  
Catch ex As Exception  
    MsgBox(ex.Message)  
End Try
```

بعد الاتصال و فتح القاعدة والجدول بقي الآن أن نعرض البيانات في مربعات النصوص
نكتب إجراء خاصا لعرض البيانات كما يلي

Code VB.Net

```
Sub LoadTxt()  
    TextBox1.Text = Rst.Fields(0).Value  
    TextBox2.Text = Rst.Fields("Prénom").Value  
    TextBox3.Text = Rst.Fields("Sexe").Value  
Position()  
End Sub
```

ثم نكتب إجراء لعرض موقع البيانات

Code VB.Net

```
Sub Position()  
    Me.Text = " رقم سجل " & Rst.AbsolutePosition & "/" & Rst.Fields.Count +1  
End Sub
```

ثم نضيف الإجراءين إلى حدث تحميل النموذج

Code VB.Net

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)  
Handles Me.Load  
    Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" _  
        & "data Source =e:/db1.mdb"  
    Con.Open()  
    Rst.CursorLocation = ADODB.CursorLocationEnum.adUseClient  
    Rst.Open("Select * from Students", Con,  
        ADODB.CursorTypeEnum.adOpenDynamic,  
        ADODB.LockTypeEnum.adLockOptimistic)
```

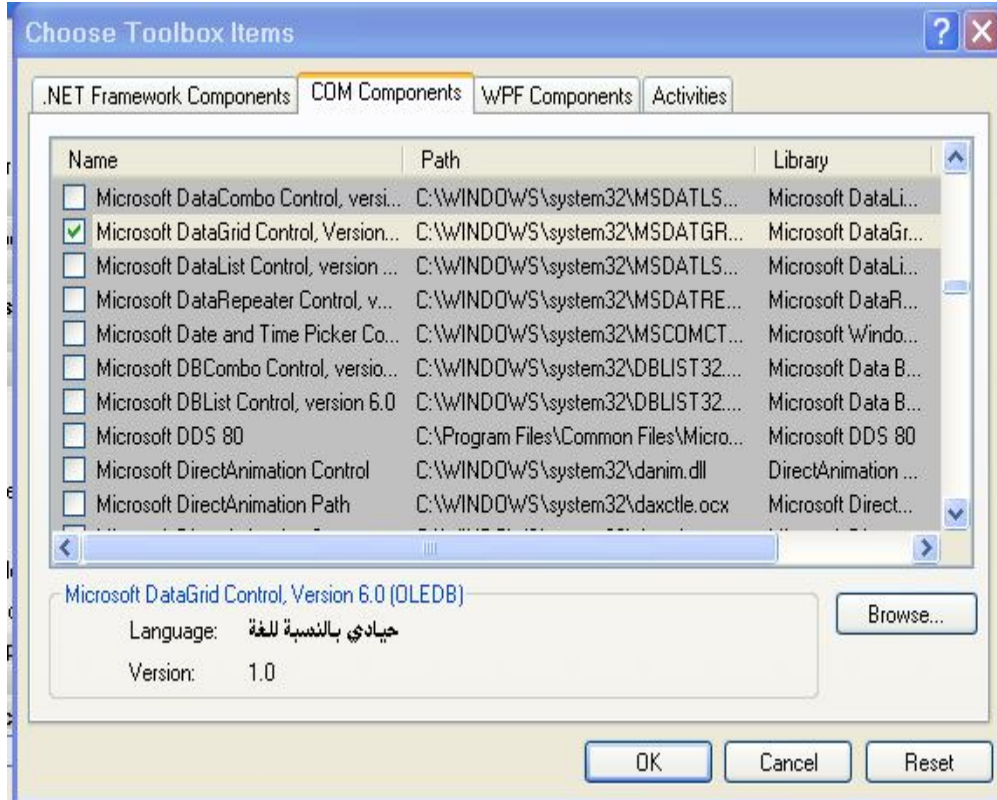
LoadTxt()

End Sub

عرض البيانات في جدول

نقوم بإدراج الجدول AxDataGrid بإتباع الخطوات التالية :

- انقر على صندوق الأدوات بزر الفأرة الأيمن أو Ctrl + T
- ثم اختر ما هو محدد في الشكل التالي :



- ثم اضغط Ok سيتم إضافة الأداة في صندوق الأدوات
- قم بإسقاطها على النموذج

ثم نكتب في حدث التحميل لتحميل الجدول البيانات

Code VB.Net

```
AxDataGrid1.DataSource = Rst.DataSource
```

عند تنفيذ البرنامج يظهر سجل واحد في مربعات النصوص نحتاج للتنقل بين السجلات .

أزرار التنقل

زر السجل التالي :

Code VB.Net

```
Rst.MoveNext() ' انتقل إلى السجل التالي
If Rst.EOF Then ' في حالة الوصول إلى آخر سجل
Rst.MoveLast() ' اعرض السجل الأخير
```

```
MsgBox("سجل آخر", MsgBoxStyle.Information) ' عرض رسالة  
End If  
LoadTxt() ' عرض البيانات
```

زر السابق

Code VB.Net

```
Rst.MovePrevious() ' انتقل إلى الخلف  
If Rst.BOF Then ' في حالة الوصول إلى السجل الأول  
Rst.MoveFirst() ' ابق في السجل الأول  
MsgBox("سجل أول", MsgBoxStyle.Information) ' يعرض رسالة  
End If  
LoadTxt() ' عرض البيانات
```

زر السجل الأول

Code VB.Net

```
Rst.MoveFirst()  
LoadTxt() ' عرض البيانات
```

زر السجل الأخير

Code VB.Net

```
Rst.MoveLast()  
LoadTxt() ' عرض البيانات
```

العمليات الأساسية

زر جديد (New) : يقوم بمسح محتوى مربعات النصوص للسماح بالكتابة و إضافة سجلات جديدة

Code VB.Net

```
Dim ctr As Control ' متغير يدل على أداة  
For Each ctr In Me.Controls ' مهما كانت الأداة من النموذج  
If TypeOf ctr Is TextBox Then ' اذ كانت هذه الأداة مربع نص  
ctr.Text = "" ' امسح هذه الأداة  
End If  
Next
```

زر إضافة سجل جديد AddNew

Code VB.Net

```
Try  
With (Rst)  
.AddNew() ' إمكانية إضافة سجل جديد  
.Fields(0).Value = TextBox1.Text ' السجل 1 = ما في مربع النص  
.Fields(1).Value = TextBox2.Text ' السجل 2 = ما في مربع النص  
.Fields("Sexe").Value = TextBox3.Text ' سجل الجنس = ما في مربع النص  
.Update() ' حفظ البيانات  
End With  
Catch ex As Exception
```

الفصل الثامن قواعد البيانات

MsgBox(ex.Message)
End Try

زر تعديل Update

Code VB.Net

```
With Rst
    .Fields(0).Value = TextBox1.Text
    .Fields(1).Value = TextBox2.Text
    .Fields("Sexe").Value = TextBox3.Text
    .Update()
    MsgBox("بنجاح التعديل عملية تمت", MsgBoxStyle.Information)
End With
```

زر حذف سجل Delete

Code VB.Net

```
If MsgBox("الحالي السجل حذف في ترغب هل", MsgBoxStyle.Question + MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then ' في حالة الموافقة
    Rst.Delete() ' احذف السجل الحالي
    Rst.MovePrevious() ' انتقل إلى الخلف
    LoadTxt() ' اعرض البيانات
End If
```

البحث Find

 الصيغة العامة

rst.Find(criteria ,SkipRecords,search direction,Start)

Rst.Find (" = Value = اسم الحقل " , نقطة بداية البحث , اتجاه البحث , صف بداية البحث)

تتكون الصيغة من المعاملات التالية

الشرح	الجدول
صيغة البحث	Criteria
نقطة بداية البحث	SkipRecords
اتجاه البحث	search direction
صف بداية البحث	Start

صيغة البحث

أولا : مع المعامل يساوي (=)

في حالة الأرقام	في حالة النصوص	في حالة التاريخ	
"Num = 5 "	"Prénom = 'Ali'"	"D = #1/5/95#"	في حالة التطابق التام
"Num =" & textbox1.text	"Prénom =" & textbox1.text & ""	"D =" & textbox1.text	في حالة أداة

الفصل الثامن قواعد البيانات

البحث بحقلين نصيين ' & TextBox1.text & "" & Nom = " & "Prénom & "" & ""

ثانيا : مع المعامل Like

الرمز	الدور	مثال
*	تحل محل مجموعة من الحروف	Rst.Filter = "Prénom Like 'a*'" يبدأ بحرف Rst.Filter = "Prénom Like '*a'" ينتهي بحرف Rst.Filter = "Prénom Like '*ali*'" في أي موضع
_ ?	تحل محل حرف واحد	Rst.Filter = "Prénom Like 'a? i'" Ali ,ami
[]	حرف واحد من مدى معين	Rst.Filter = "Prénom Like '[A,b]li'"
#	تحل محل رقم واحد	Rst.Filter = "Prénom Like 'c1#'" c10 , c15 ...
[!]	تستثني الحرف	Rst.Filter = "Prénom Like *c[!q]" عرض الأسماء التي تنتهي بالحرفين C مع أي حرف غير q



ملاحظة هامة جدا : الصيغ المستعملة في البحث و التصفية هي نفس الصيغ
نقطة بداية البحث SkipRecords :

توضع 0 نقطة البدء الافتراضية أو يتم البحث من نقطة التي توقف عنها المؤشر حاليا

اتجاه البحث search direction : وهو أحد الخيارين

ADODB.SearchDirectionEnum.adSearchForward : أي وصولا إلى آخر سجل

ADODB.SearchDirectionEnum.adSearchBackward : أي وصولا إلى أول سجل

البداية Start

و هنا نحدد الصف الذي نبدأ البحث منه
مثال :

Code VB.Net

```
Rst.Find("Prénom = 'سامح' ", 0, ADODB.SearchDirectionEnum.adSearchBackward, 1)
```

مثال آخر : نبحث عن الاسم حسب صندوق الإدخال InputBox

Code VB.Net

```
Dim XPrénom As String = InputBox("الاسم ادخل", "بحث")
If XPrénom = "" Then Exit Sub
Dim criter = "Prénom = " & XPrénom & ""
Rst.Find(criter)
```

البحث عن طريق الحلقة

Code VB.Net

```
Dim XPrénom As String = InputBox("الاسم ادخل", "بحث")
If XPrénom = "" Then Exit Sub
Do Until Rst.EOF
If Rst.Fields("Prénom").Value = XPrénom Then
Loadtxt()
```

الفصل الثامن قواعد البيانات

```
Exit Do
End If
Rst.MoveNext()
Loop
```

عرفنا أن عملية البحث ترجع سجلا وحيدا .
ماذا لو أردنا أن نبحث عن مجموعة من السجلات تشترك في بيانات حقل محدد . كأن نبحث عن تلاميذ القسم الأول . ما هو الحل ؟
الحل : هو استعمال عملية تصفية Filter لحقل القسم

 الصيغة العامة


```
RecordSet.Filter = criteria
```

Criteria : صيغة التصفية وهي نفسها صيغ البحث

مثال : تصفية تلاميذ القسم 1

 Code VB.Net

```
Rst.Filter = "Classe = 1"
AxDataGrid1.DataSource = Rst.DataSource
```

 لإلغاء عملية التصفية نكتب

عمليات أخرى

عرض سجلات عمود

 Code VB.Net

```
Do Until Rst.EOF
    ListBox1.Items.Add(Rst.Fields("Prénom").Value)
    Rst.MoveNext()
Loop
```

عرض حقول جدول

 Code VB.Net

```
Dim fld As ADODB.Field
For Each fld In Rst.Fields
    ListBox1.Items.Add(fld.Name)
Next
```

حفظ صورة في حقل

 Code VB.Net

```
My.Computer.FileSystem.WriteAllBytes("c:\Blue hills.jpg",
CType(RstStud.Fields("صورة").Value, Byte()), True)
PictureBox1.Image = Image.FromFile("c:\Blue hills.jpg")
```

التعرف على SQL Structured Query Language

هي لغة تستعمل للتعامل مع قواعد البيانات بأسلوب سهل و تعني حرفيا لغة الاستعلامات المهيكلة فهي تتعامل مع البرامج التالية :

.Ms-Access , Ms-SQL Server , DB2 , Oracle ,SQL Server 2000 OR 2005 , etc

و يقرأ "sequel"

أقسام جمل SQL

و تنقسم إلى قسمين رئيسيين :

أولا : قسم تعريف البيانات (DDL) Data Definition Language

يمكنك من إنشاء قاعدة البيانات و الجداول و الفهارس و القيود و العلاقات و كذا حذفها و تشتمل على :

- Create Table لإنشاء الجداول
- Creat Index لإنشاء الفهارس
- Alter Table لتعديل بنية الجدول
- Drop لحذف الجدول أو الفهرس

ثانيا : قسم معالجة البيانات (DML) Data Manipulation Language

تمكنك من عرض و استرجاع و تحديد بيانات قاعدة البيانات و تعديلها أو حذفها أو الإضافة و تشتمل على :

- جملة التحديد و الاسترجاع Select
- لإضافة سجلات Insert
- لتعديل و تحديث السجلات Update
- حذف السجلات Delete

ملاحظات و إرشادات حول كتابة جمل SQL

- 1- جمل SQL غير حساسة للحروف Not case sensitive يمكن كتابتها كبيرة أو صغيرة SLECT أو Select لا فرق
- 2- يفصل بين أسماء الحقول أو الجداول باستخدام الفاصلة (,)
- 3- لا يمكن فصل الكلمات المحجوزة KEYWORD أو اختصارها فمثلا : DistinctRow
- 4- يفضل كتابة الجملة على أسطر ليسهل قراءتها وفهمها و عند الرجوع إلى السطر
- 5- لكتابة حقول مركبة نستعمل [] مثل : [Last Name] أو نستعمل الرمز (_) كما يلي Last_Name
- 6- عند كتابة قيمة نصية نضعها ' ... ' أما القيمة العددية فنكتب عادية
- 7- اختر أسماء الحقول و الجداول بالانجليزية حتى يسهل قراءتها و كتابتها
- 8- غالبا ما تنتهي جمل SQL بفاصلة منقطة (;)

إنشاء الجدول Create Table

الصيغة العامة

```
CREATE TABLE table (field1 type [(size)] [NOT NULL] [index1] [, field2
type [(size)] [NOT NULL] [index2] [, ...] [, CONSTRAINT multifieldindex
[, ...]])
```



تستعمل لإنشاء الجداول بحقولها بخصائصها و كذا تخصيص حقول كفهارس دراسة مكونات الصيغة :

ملاحظة: لا يعتمد مشغل قاعدة بيانات Microsoft Access استخدام CREATE TABLE

أو أية جملة من جمل DDL

تتكون جملة CREATE TABLE من المعاملات التالية :

المعامل	الشرح
table	اسم الجدول المراد إنشاؤه.
field1, field2	اسم الحقل أو الحقول المراد إنشاؤها في الجدول الجديد. ينبغي إنشاء حقل واحد على الأقل.
type	نوع بيانات الحقل
size	حجم الحقل بالأحرف (الحقول النصية وحقول البيانات الثنائية فقط)
index1, index2	عبارة CONSTRAINT التي تقوم بتعريف الفهرس ذو الحقل الواحد .
multifieldindex	عبارة CONSTRAINT التي تقوم بتعريف فهرس متعدد الحقول .

ملاحظات: كل عبارة توضع بين [] فهي اختيارية.

استخدم جملة CREATE TABLE لتعريف جدول جديد وحقوله وقيود حقوله. في حالة تعيين NOT NULL

لأحد الحقول، يجب أن لا يترك فارغاً.

تقوم عبارة CONSTRAINT بإنشاء قيود متعددة على أحد الحقول، ويمكن استخدامها لإنشاء المفتاح الأساسي

Column constraints يشتمل على الخصائص التالية :

- NOT NULL مطلوب لا يقبل قيمة فارغة
- UNIQUE لا يقبل التكرار
- PRIMARY KEY مفتاح أساسي
- Default القيمة الافتراضية

إنشاء جداول Create Tables

نقوم الآن بإنشاء جدول الطلاب Students و يشتمل على الحقول التالية :

الحقل	المقابل له	نوعه
Num	الرقم	عدد صحيح
Prenom	الاسم	نص من 20 حرفا
Nom	اللقب	نص من 20 حرفا
Date_N	تاريخ الميلاد	تاريخ
Classe	القسم	عدد صحيح
Sexe	الجنس	نص من 10 حروف

مثال

```
create Table Students
(Num Int Primary Key ,
Prenom Varchar(20) Not Null ,
Nom Text(20) Not Null ,
Date_N Date ,
Sex Varchar(10) Default 'M' ,
Classe Int );
```

و نحصل على النتيجة :

Field	Type	Null	Key	Default	Extra
Num	int (11)	NO	PRI		
Prenom	varchar (20)	NO			
Nom	tinytext	NO			
Date_N	date	YES		NULL	
Sex	varchar (10)	YES		M	
Classe	int (11)	YES		NULL	

نقوم بتحميل الجدول ببعض السجلات كما يلي :

مثال

```
Insert Into Students Values
(1,'Salah','Meddour','1967,10,23','M',2);
```

وكذا السجل الثاني

مثال

```
Insert Into Students Values
(2,'Ali','Aksa','1995,02,25','M',1);
```

و كذا السجل الثالث

مثال

```
Insert Into Students Values
(3,'Nadia','Saada','1991,01,15','F',2);
```

و هكذا ...

والآن نعرض كل السجلات التي أضفناها إلى الجدول

```
mysql> Select * From Students ;
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 1967-10-23  | M   | 2      |
| 2   | Ali    | Aksa     | 1995-02-25  | M   | 1      |
| 3   | Nadia  | Saada    | 1991-01-15  | F   | 2      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

ثم ننشئ جدول النتائج Grade و يتكون من الحقول التالية :

```
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Num   | int(11)       | NO   | PRI |          |       |
| Prenom | varchar(20)   | NO   |     |          |       |
| Nom   | tinytext     | NO   |     |          |       |
| Arabe | double        | YES  |     | NULL    |       |
| Math  | double        | YES  |     | NULL    |       |
| Franc | double        | YES  |     | NULL    |       |
| Moyen | double        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

مثال

```
Create Table Grade
(Num Int Primary Key ,
Prenom Varchar(20) Not Null ,Nom Text(20) Not Null ,
Arabe Double ,Math Double , Franc Double ,Moyen Double);
```

والآن نقوم بتحميل الجدول ببعض السجلات كما يلي :

مثال

```
Insert Into Grade Values
(1,'Salah','Meddour',14,14,14,14);
```

وكذا السجل الثاني :

مثال

```
Insert into Grade Values
(2,'Aksa','Ali',11,11,11,11);
```

و كذا السجل الثالث :

مثال

```
Insert into Grade Values
(3,'Nadia','Saada',12,12,12,12);
```

و الآن نستعرض كل السجلات التي أضفناها

```
mysql> Select * From Grade ;
+-----+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 14    | 14   | 14    | 14    |
| 2   | Aksa   | Ali      | 11    | 11   | 11    | 11    |
| 3   | Nadia  | Saada    | 12    | 12   | 12    | 12    |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

و إذا أردنا إنشاء جدول فيه حقل لا يقبل إلا قيما مخصصة مثل حقل الجنس يقبل قيمتين 'M' أو 'F'

مثال

```
"create Table NewTable " _
& "(sex Varchar(6), Constraint bb Check(sex in('M','F')));"
```

الصيغة العامة لإنشاء الفهارس

الفهارس : ميزة تسهل عمليات البحث و السرد فهو كفهرس الكتاب

الصيغة العامة

```
CREATE [ UNIQUE ] INDEX index
ON table (field [ASC|DESC][, field [ASC|DESC], ...])
[WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }
```

تتكون جملة CREATE INDEX من المعاملات التالية:

المعامل	الوصف
<i>index</i>	اسم الفهرس المراد إنشاؤه.
<i>table</i>	اسم الجدول الموجود الذي سيحتوي على الفهرس.
<i>field</i>	اسم الحقل أو الحقول المراد فهرستها. لإنشاء فهرس لحقل واحد، قم بوضع اسم الحقل بين أقواس بعد اسم الجدول. لإنشاء فهرس لحقول متعددة، قم بسرد اسم كل حقل ليتم تضمينه في الفهرس..

استعمل :

Create Index : إنشاء فهرس

Unique : لمنع تكرار القيم في حقل أو حقول مفهرسة لسجلات مختلفة

DISALLOW NULL : لمنع القيم الفارغة للحقول المفهرسة

PRIMARY : لتعيين المفتاح الأساسي لحقل

أنواع الفهارس Type Of Index

فهارس فريدة لا تسمح بالتكرار UNIQUE INDEX

و لإنشائه استعمل الصيغة البسيطة التالية

الصيغة العامة

```
CREATE UNIQUE INDEX index_name ON tableName (FieldName)
```

توضيح: في جدول الطلاب Students يحتوي على أسماء و أرقام و حقول أخرى نختار حقل الرقم Num هو الفهرس لأن لكل طالب رقم و لا يمكن تكراره أما إذا اخترنا مثلا حقل الاسم فسنحصل على طلاب لهم نفس الاسم . و لنفعل ذلك نكتب

```
Create Unique Index Id_St On Students(Num) ;
```

فهارس غير فريدة تسمح بالتكرار

و لإنشائه استعمل الصيغة التالية :

الصيغة العامة

```
CREATE INDEX index_name ON tableName (FieldName)
```

اجعل حقلي الاسم و اللقب كمفاتيح عادية تقبل التكرار

```
Create Index Id_Name On Students(Prenom ,Nom);
```

و إذا أردنا أن نجعل حقلا مفتاحا أساسيا نكتب

مثال

```
Dim SQL As String = "CREATE UNIQUE INDEX [IndexName] " _
& "ON NameTable ([NameField]) " _
& "WITH PRIMARY;"
```

و يمكنك إنشاء فهرس بترتيب عكسي لحقل الاسم

مثال

```
Dim SQL As String = "CREATE INDEX Ind2 " _
& "ON Students (Prenom DESC)"
```

القيود Cinstaint

القيود : هو شرط يوضع لجعل الحقل :

- Unique : لا يقبل التكرار
- Not Null : مطلوبا لا يقبل قيما فارغة
- Primary Key : متاحا أساسيا

فهو يشبه الفهارس بالإضافة يمكنك استعماله في العلاقات

يمكنك استخدام عبارة CONSTRAINT في الحملتين ALTER TABLE و CREATE TABLE لإنشاء قيود أو حذفها

أولا : عند إنشاء الجدول Create Table

الصيغة العامة

```
CREATE [{GLOBAL | LOCAL}
TEMPORARY] TABLE <table_name> (
<column_name> [<domain_name> | <datatype>
[<size1>,<size2>] ] [<column_constraint>,...] [DEFAULT
<default_value>] [COLLATE <collation_name>],...
[<table_constraints>] [ON COMMIT {DELETE | PRESERVE} ROWS]
)
```

ثانيا : عند تعديل الجدول Alter Table

الصيغة العامة

```
ALTER TABLE table {ADD {COLUMN field type[(size)] [NOT NULL] [CONSTRAINT
index] |
ALTER COLUMN field type[(size)] |
CONSTRAINT multifieldindex} |
DROP {COLUMN field I CONSTRAINT indexname} }
```

الصيغة العامة

```
CONSTRAINT name {PRIMARY KEY | UNIQUE | NOT NULL |
REFERENCES foreigntable [(foreignfield1, foreignfield2)]
[ON UPDATE CASCADE | SET NULL]
[ON DELETE CASCADE | SET NULL]}
```

و تتكون من :

اسم القيد المراد إنشاؤه	<i>name</i>
اسم الجدول الخارجي	<i>foreigntable</i>
اسم الحقل أو الحقول من الجدول الخارجي	<i>foreignfield1,</i> <i>foreignfield2</i>

إنشاء قيد Primary Key عند إنشاء الجدول على مستوى الأعمدة

وهذا حسب المثال الذي رأينا عند إنشاء جدول الطلاب

مثال

```
Create Table Students
(Num Int Primary Key ,
Prenom Varchar(20) Not Null ,
Nom Text(20) Not Null ,
Date_N Date ,
Sex Varchar(10) Default 'M' ,
Classe Int );
```

إنشاء قيد Primary Key عند إنشاء الجدول على مستوى الجدول

```
[CONSTRAINT <constraint name>]
PRIMARY KEY [(<column>, <column>, ...)]
```

مثال

```
"CREATE TABLE NewTable " _
& "(Prénom TEXT, Nom TEXT, " _
& "SSN INTEGER CONSTRAINT MyFieldConstraint " _
& "PRIMARY KEY);"
```

إنشاء قيد Unique

إنشاء القيد عند إنشاء الجدول على مستوى الجدول

```
[CONSTRAINT <constraint name>]
UNIQUE [(<column>, <column>, ...)]
```

نفس المثال السابق فقط أغير Primary Key ب Unique

إنشاء القيد Foreign Key

يقوم بربط جدولين انظر هذا المثال

مثال

```
Create Table NewTable
(Fname Varchar(20) ,
Id int REFERENCES Students(Num));
```

حيث قمنا بربط الجدول الجديد NewTable مع جدول الطلاب Students بواسطة الحقل Num

إنشاء القيد Check

يستعمل لوضع شروط الصحة لحقل أي تحديد مجال أو شروط للقيم المدخلة لحقل

```
[CONSTRAINT <constraint name>] CHECK (<condition>)
```

مثال

```
"Create Table NewTable " _
& "(Sex Varchar(6) , " _
& "Constraint Sex_Check Check(Sex In ('M','F')))"
```

حيث قمنا بتخصيص قيما لحقل الجنس فلا يمكن للمستخدم لإدخال قيما غير القيم المحدد ذ (M)
أو أ (F)

إضافة قيود إلى جدول موجود Adding Constraint

يتم إضافة قيود إلى جدول موجود عن طريق الجملة Alter Table

Alter Table Table_Name

Add Constraint name_Constraint Constraint_Type

Primary Key إضافة مفتاح أساسي

مثال

```
Alter Table Students " _
& "Add Constraint pk Primary Key(Num)"
```

إضافة قيد Check

مثال

```
alter Table Students
Add Constraint b Check(Sex in('M','F'));
```

تعديل بنية الجدول Alter Table

يمكنك تعديل هيكل و بنية الجدول بواسطة الجملة Alter Table وهذا بـ :

- إضافة عمود أو فهرس أو مفتاح أو قيد باستعمال Add
- تعديل بيانات عمود موجود
- حذف عمود أو فهرس أو مفتاح أو قيد باستعمال Drop

الصيغة العامة

```
ALTER TABLE table {ADD {COLUMN field type[(size)] [NOT NULL]
[CONSTRAINT index] |
ALTER COLUMN field type[(size)] |
CONSTRAINT multifieldindex} |
DROP {COLUMN field I CONSTRAINT indexname} }
```

تتكون جملة التعديل من :

table	اسم الجدول الذي سيتم التعديل فيه
field	اسم الحقل الذي سيتم تعديله
index	الفهرس الذي سيتم إضلفته
type	نوع البيانات الجديدة
Size	حجم أو طول الأحرف التي يمكن أن يسعها الحقل / الجديدة
Multifieldindex	تعريف فهرس الحقول المتعددة التي سيتم إضافتها
indexname	اسم الفهرس المتعدد الحقول المراد إزالته

كل ما هو بين [] فهو اختياري

Add Column : لإضافة حقل جديد

Alter Column : لتعديل بيانات حقل

Constraint : لإضافة قيد

Drop Column : لحذف حقل

Drop Constraint : لحذف قيد

إضافة حقول جديدة Add Column

Alter Table Name_Table Add Column type_Field (size)

مثال

```
Alter Table Students Add Column Notes Varchar(40) Default '/' ;
```

لا حظ الناتج

إضافة مفتاح أساسي

مثال

```
Alter Table Students " _  
& "Add Constraint pk Primary Key(Num)"
```

إضافة قيد Add Constraint

مثال

```
alter Table Students  
Add Constraint b Check(Sex in('M','F'));
```

تعديل بيانات حقل موجود Alter Column

مثال

```
"Alter Table Students " _  
& "Alter Column Prenom Varchar(22) Not Null"
```

حذف حقل Drop Column

مثال

```
Alter Table Students Drop Notes
```

حذف قيد Drop Constraint

مثال

```
"Alter Table Students Drop Constraint b;"
```

الحذف النهائي Drop

تقوم بحذف الجدول نهائيا ببياناته و حقوله و كذا الفهارس

الصيغة العامة

DROP {TABLE *table* | INDEX *index* ON *table* | PROCEDURE *procedure* |
VIEW *view*}

و تتكون جملة Drop من :

<i>table</i>	اسم الجدول المراد حذفه
<i>index</i>	أو اسم الفهرس المراد حذفه
<i>procedure</i>	اسم الإجراء المراد حذفه
<i>view</i>	اسم طريقة العرض المراد حذفها

ينبغي أن يكون الجدول مغلقاً حتى تتمكن من حذفه.

حذف جدول Drop Table

مثال

Drop Table Table_Name

و لحذف حقل أو فهرس استعمل جملة Alter Table Drop

جملة التحديد و الاسترجاع Select

تقوم باسترجاع و تحديد و عرض البيانات و فق شروط معينة من جدول أو من عدة جداول

الصيغة العامة


```
SELECT [predicate] { * | table.* | [table.]field1 [AS alias1] [, [table.]field2 [AS alias2]
[, ...]]}
FROM tableexpression [, ...] [IN externaldatabase]
[WHERE... ]
[GROUP BY... ]
[HAVING... ]
[ORDER BY... ]
[WITH OWNERACCESS OPTION
```

و تتكون جملة التحديد Select من :

<i>predicate</i>	تمثل أحد المستندات : All أو DISTINCT أو DISTINCTROW أو TOP. و تمثل عدد السجلات التي يمكن عرضها
*	تعني تحديد كافة حقول الجدول و عرض سجلات كل حقل
<i>table</i>	اسم الجدول المراد استرجاع أو عرض البيانات منه
<i>field1 Field2</i>	أسماء الحقول المراد استرجاع أو عرض بياناتها
<i>alias</i>	الأسماء المستعارة للحقول ليفهما المستخدم
<i>tableexpression</i>	اسم الجدول المراد استرجاع أو عرض البيانات منه
<i>externaldatabase</i>	اسم قاعدة البيانات الخارجية

استعمل :

- Select في بداية الجملة الاستعلامية لتحديد الحقول المراد عرض سجلاتها
- From : لتعيين اسم الجدول الذي تسترجع البيانات منه
- Where : لتحديد الشرط
- Group By : لتجميع السجلات في مجموعات
- Having : لوضع شرط عند تجميع السجلات
- Order by : لترتيب السجلات

 كل عبارة بين [] فهي اختيارية

إذا كانت الحقول من جداول مختلفة نكتب اسم الجدول متبوعاً بنقطة ثم اسم الحقل

Table_Name.Field_Name

عند استخدام دالات التجميع يستحسن تسمية الحقول الناتجة بأسماء مستعارة

يجب الفصل بين الحقول و الجداول بفاصلة (,)

جملة Select تعرض فقط و لا تقوم بالتغيير

تحديد الحقول Select Fields

تحديد كافة الحقول Select All

لتحديد كافة الحقول نستعمل الرمز (*) كما يلي :

Select * From Students ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2

لتحديد حقول معينة

Select Num ,Prenom ,Classe
From Students ;

Num	Prenom	Classe
1	Salah	2
2	Ali	1
3	Nadia	2


تحديد حقول من جداول مختلفة

```
mysql> Select Students.Num ,Students.Prenom ,Grade.Moyen
<- From Students,Grade;
```

```

+-----+-----+-----+
|Num | Prenom| Moyen|
+-----+-----+-----+
| 1 | Salah | 14 |
| 2 | Ali | 14 |
| 3 | Nadia | 14 |
| 1 | Salah | 11 |
| 2 | Ali | 11 |
| 3 | Nadia | 11 |
| 1 | Salah | 12 |
| 2 | Ali | 12 |
| 3 | Nadia | 12 |
+-----+-----+-----+
9 rows in set (0.03 sec)

```

 تستطيع ترتيب الحقول كما تريد و هذا لن يؤثر على الجدول الأصلي

تلاحظ تكرار السجلات و لتفادي هذا التكرار نقوم بربط الجدول كما يلي

```

Select Students.Num ,Students.Prenom ,Grade.Moyen
From Students,Grade
Where Students.Num = Grade.Num;

```

```

+-----+-----+-----+
|Num | Prenom| Moyen|
+-----+-----+-----+
| 1 | Salah | 14 |
| 2 | Ali | 11 |
| 3 | Nadia | 12 |
+-----+-----+-----+
3 rows in set (0.02 sec)

```

الأسماء المستعارة Aliases


تستعمل الأسماء المستعارة بدلا عن الأسماء البرمجية لكي تكون مفهومة عند المستخدم

 الصيغة العامة

```

Select Field As Alias_Field From Table_Name
اسم الجدول As الاسم الجديد As اسم الحقل

```

 هذا لن يؤثر على أسماء الحقول في الجداول الأصلية

اكتب استعمالا يظهر حقل Num بالاسم المستعار الرقم و حقل Prénom بالاسم المستعار الاسم من جدول الطلاب.

 مثال

```

SELECT Num as الرقم ,Prénom as الاسم ,Nom as اللقب , Date_N as [التاريخ] , Sex as
الجنس ,Classe as القسم from Students

```

```

+-----+-----+-----+-----+-----+-----+
| الرقم | الاسم | اللقب | التاريخ | الجنس | القسم |
+-----+-----+-----+-----+-----+-----+
| 1 | Salah | Meddour | 1967-10-23 | M | 2 |
| 2 | Ali | Aksa | 1995-02-25 | M | 1 |
| 3 | Nadia | Saada | 1991-01-15 | F | 2 |
+-----+-----+-----+-----+-----+-----+

```

الأسماء المستعارة مع الجداول

الصيغة العامة

SELECT column FROM table as Table Alias

Select الاسم المستعار للجدول As اسم الجدول From اسم الحقل

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و حقل Prénom بالاسم المستعار الاسم و الجدول Students بالاسم المستعار الطلاب.

مثال

SELECT Num as الرقم, Prénom as الاسم, Nom as اللقب from Students

دمج حقلين في حقل باسم مستعار

الصيغة العامة

SELECT column1 & " " & Column2 As [Column_Alias] FROMy table

Select اسم الجدول [اسم الحقل المستعار] As اسم الحقل 2 & " " & اسم الحقل 1

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و يدمج الحقلين Prénom و Nom في حقل واحد باسم مستعار الاسم الكامل.

مثال

Select num as الرقم, Prénom & " " & Nom AS [الاسم الكامل] from Students

استعمال Alias للربط بين جدولين

مثال

SELECT Nom, Prénom
FROM Students AS S
INNER JOIN Resultant AS R ON S.Num = R.Num

الأسماء المستعارة Aliases

تستعمل الأسماء المستعارة بدلا عن الأسماء البرمجية لكي تكون مفهومة عند المستخدم

الصيغة العامة

Select Field As Alias_Field From Table_Name

Select اسم الجدول As الاسم الجديد اسم الحقل

هذا لن يؤثر على أسماء الحقول في الجداول الأصلية

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و حقل Prénom بالاسم المستعار الاسم من جدول الطلاب.

مثال

SELECT Num as الرقم ,Prénom as الاسم ,Nom as اللقب , Date_N as [التاريخ] , Sex as الجنس ,Classe as القسم from Students

الرقم	الاسم	اللقب	التاريخ	الجنس	القسم
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aksa	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2

الأسماء المستعارة مع الجداول

الصيغة العامة

SELECT column FROM table as Table Alias

Select الاسم المستعار للجدول As اسم الجدول From اسم الحقل

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و حقل Prénom بالاسم المستعار الاسم و الجدول Students بالاسم المستعار الطلاب.

مثال

SELECT Num as الرقم ,Prénom as الاسم ,Nom as اللقب from Students

دمج حقلين في حقل باسم مستعار

الصيغة العامة

SELECT column1 & " " & Column2 As [Column_Alias] FROMy table

Select اسم الجدول [اسم الحقل المستعار] As اسم الحقل 2 & " " & اسم الحقل 1

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و يدمج الحقلين Prénom و Nom في حقل واحد باسم مستعار الاسم الكامل.

مثال

Select num as الرقم ,Prénom & " " & Nom AS [الاسم الكامل] from Students

استعمال Alias للربط بين جدولين

مثال

SELECT Nom, Prénom
FROM Students AS S
INNER JOIN Resultant AS R ON S.Num = R.Num

أعلى عدد من السجلات TOP n

يقوم بإرجاع عدد معين من السجلات التي توجد في الجزء العلوي أو السفلي من نطاق معين من قبل

عبارة: ORDER BY

الصيغة العامة

```
Select Top N Field1 ,Field2 .....
From Table_Name
Where Condition
Order by Field1 ,Field2 .....
```

يستحسن الترتيب للحصول على النتائج الدقيقة

الرقم N يجب أن يكون موجبا

mysql> Select * From grade;

```
+-----+-----+-----+-----+-----+-----+-----+
|Num   |Prenom | Nom       | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
| 1    |Salah  | Meddour  | 14    | 14   | 14    | 14    |
| 2    |Aksa   | Ali      | 11    | 11   | 11    | 11    |
| 3    |Nadia  | Saada    | 12    | 12   | 12    | 12    |
| 4    |Said   | Meddour  | 15    | 15   | 15    | 15    |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

اكتب استعلاما يعيد الطالبين المتفوقين حسب المعدل

مثال

```
"Select Top 2 * From Grades "
& "Order By Moyen"
```

```
+-----+-----+-----+-----+-----+-----+-----+
|Num   |Prenom | Nom       | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
| 1    |Salah  | Meddour  | 14    | 14   | 14    | 14    |
| 4    |Said   | Meddour  | 15    | 15   | 15    | 15    |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

إظهار أعلى نسبة من السجلات Top N Percent

يقوم بإرجاع عدد معين من السجلات حسب النسبة المحددة التي توجد في الجزء العلوي أو السفلي

من نطاق معين من قبل عبارة: ORDER BY

الصيغة العامة

```
Select Top N Percent Field1 ,Field2 .....
From Table_Name
Where Condition
Order by Field1 ,Field2 .....
```

يستحسن الترتيب للحصول على النتائج الدقيقة

يكتب العدد بدون علامة النسبة المئوية

```
mysql> Select * From grade;
```

Num	Prenom	Nom	Arabe	Math	Franc	Moyen
1	Salah	Meddour	14	14	14	14
2	Aksa	Ali	11	11	11	11
3	Nadia	Saada	12	12	12	12
4	Said	Meddour	15	15	15	15

اكتب استعلاما يظهر الطالب الناجحين بنسبة 50%

مثال

```
"Select Top 50 Percent * From Grades " _
& "Order By Moyen Desc "
```

Num	Prenom	Nom	Arabe	Math	Franc	Moyen
1	Salah	Meddour	14	14	14	14
4	Said	Meddour	15	15	15	15

العمليات الحسابية

يمكن إجراء العمليات الحسابية على مستوى الحقول

الصيغة العامة

```
Select ..... , الحقل 2, عملية حسابية على الحقل
From Table_Name
```

لن يحدث تغيير على الحقول الأصلية بعد إجراء العملية الحسابية

يستحسن تسمية الحقول التي تمت على مستواها العمليات الحسابية باسم مستعار حتى تفهمه

العملية الحسابية تكون على مستوى الحقل و لكن النتيجة تكون على كافة السجلات

يمكن استخدام العمليات الحسابية على جميع مستويات جملة SELECT ماعدا مستوى الجدول

أي Form Clause

المعاملات الحسابية المستعملة في جملة SQL

- (+) عملية الجمع
- (-) عملية الطرح
- (*) عملية الضرب
- (/) عملية القسمة

أولويات العمليات الحسابية Operator Precedence

1- الأقواس

2- الضرب ثم القسمة ثم الجمع ثم الطرح

اكتب استعلاما يظهر نتائج الطلاب حسب المعامل أي المعدل = (الرياضيات * 2) + (العربية * 2) / 5 فرنسية

مثال

```
Select Num , Prenom , ((Arabe*2)+(Math*2)+Franc)/5 as Moyen_Gen
From Grade;
```

```
+-----+-----+-----+
|Num   |Prenom |Moyen_Gen|
+-----+-----+-----+
| 1    |Salah  | 14      |
| 2    |Aksa   | 11      |
| 3    |Nadia  | 12      |
| 4    |Said   | 15      |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

دمج الحقول Concat

يمكنك دمج حقلين أو أكثر في حقل واحد يفضل تسميته باسم مستعار تفهمه

اكتب استعلاما يظهر حقل Num بالاسم المستعار الرقم و يدمج حقل Prenom و Nom في حقل واحد باسم مستعار الطالب

مثال

```
"Select Num As الرقم ,Prenom & "" "" & Nom As [الطالب] _
& "From Students"
```

جملة الشرط Where

تقوم باسترجاع و تحديد السجلات وفق الشروط المحددة على مستوى الحقل

الصيغة العامة

```
SELECT fieldlist
FROM tableexpression
WHERE criteria
```

تتكون جملة SELECT التي تحتوي على جملة WHERE من الأجزاء التالية:

الجزء	الوصف
<i>fieldlist</i>	اسم الحقل أو الحقول المراد
<i>tableexpression</i>	اسم الجدول أو الجداول المراد استرداد بيانات منها.
<i>criteria</i>	التعبير أو الشرط الذي يجب أن تحققه السجلات ليتم تضمينها في نتائج الاستعلام.



تتعامل جملة الشرط مع عدة معاملات الخاصة بالشرط

Criteria = الحقل عامل المقارنة الشرط

مثل : Where Moyen >= 10

▪ عند استخدام الحقول النصية توضع قيمتها بين ' ' مثل 'Where Prenom = 'Salah'

▪ عند استخدام حقول تعبر عن التاريخ توضع قيمتها بين علامتي # # مع Vb أو ' ' مع MySQL

مثلا : Where Date_N = #01/01/2001#

▪ المعامل > في التواريخ يعني بعد هذا التاريخ

▪ المعامل < في التواريخ يعني قبل هذا التاريخ

استعمال معاملات المقارنة Comparison Operators

Operator	Description	الوصف
=	Equal	يساوي
<> or !=	Not equal	لا يساوي
<	Less than	أصغر من
>	Greater than	أكبر من
<=	Less than or equal	أصغر أو يساوي
>=	Greater than or equal	أكبر أو يساوي

mysql> Select * From Students ;

```

+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 1967-10-23 | M   | 2      |
| 2   | Ali    | Aksa     | 1995-02-25 | M   | 1      |
| 3   | Nadia  | Saada    | 1991-01-15 | F   | 2      |
| 4   | Said   | Meddour  | 1990-10-21 | M   | 1      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.11 sec)
    
```

اكتب استعلاما يظهر الطالب الصالح

مثال

```

Select * From Students
Where Prenom ='Salah' ;
    
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 1967-10-23 | M   | 2      |
+-----+-----+-----+-----+-----+-----+
1 rows in set (0.11 sec)
```

إليك جدول النتائج

mysql> Select * From Grade ;

```
+-----+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 14    | 14   | 14    | 14    |
|  2  | Aksa   | Ali      | 11    | 11   | 11    | 11    |
|  3  | Nadia  | Saada    | 12    | 12   | 12    | 12    |
|  4  | Said   | Meddour  | 15    | 15   | 15    | 15    |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

اكتب استعلاما يعرض الطلاب الذين تحصلوا على ومعدل أكبر من 12



Select * From Grade
where Moyen >= 12 ;

```
+-----+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 14    | 14   | 14    | 14    |
|  3  | Nadia  | Saada    | 12    | 12   | 12    | 12    |
|  4  | Said   | Meddour  | 15    | 15   | 15    | 15    |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

اكتب استعلاما يظهر الطالب المولود في 1991/01/05 من جدول الطلاب



Select * From Students
where Date_n =#1991,01,15# ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
|  3  | Nadia  | Saada    | 1991-01-15 | F   | 2      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

اكتب استعلاما يظهر اسم الطالب حسب ما كتب في صندوق النص Text Box



وهذا مع Vb


SELECT * from Students Where Nom ="" & text1.text & ""

اكتب استعلاما يظهر كل بيانات الطلاب الذين أسماؤهم قبل الحرف S في الترتيب الهجائي



Select * From Students
where Prenom < 'S' ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom   | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  2  | Ali    | Aks   | 1995-02-25 | M    | 1      |
|  3  | Nadia  | Saada | 1991-01-15 | F    | 2      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

اكتب استعلاما يظهر كل بيانات الطلاب المولودين قبل سنة 1995 

```
Select * From Students
where Year(Date_n) < 1995 ;
```


```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom   | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour | 1967-10-23 | M    | 2      |
|  3  | Nadia  | Saada  | 1991-01-15 | F    | 2      |
|  4  | Said   | Meddour | 1990-10-21 | M    | 1      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

استعمال معاملات المنطقية Logical Operators

المعامل	الوصف	المعامل	الوصف
And	و	XOR	إما - أو
OR	أو	EQV	يكافؤ
Not	النفى		


```
mysql> Select * From Students ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom   | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour | 1967-10-23 | M    | 2      |
|  2  | Ali    | Aksa   | 1995-02-25 | M    | 1      |
|  3  | Nadia  | Saada  | 1991-01-15 | F    | 2      |
|  4  | Said   | Meddour | 1990-10-21 | M    | 1      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.11 sec)
```

اكتب استعلاما يظهر كل بيانات الطالب الذي اسمه Ali و لقبه Aksa 

```
mysql> Select * From Students
-> where Prenom = 'Ali' And Nom ='Aks';
```


```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  2  | Ali    | Aksa | 1995-02-25 | M    | 1      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

اكتب استعلاما يظهر كل بيانات الطلاب الذين لقبهم Meddour أو Saada من جدول 

الطلاب الأول


Select * From Students
where Nom ='Meddour' Or 'Saada' ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 1967-10-23 | M   | 2     |
|  4  | Said   | Meddour  | 1990-10-21 | M   | 1     |
+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.02 sec)
```

اكتب استعلاما يظهر كل بيانات الطلاب الذين لا يدرسون في القسم 1 من جدول الطلاب 

Select * From Students
where Not Classe = 1 ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 1967-10-23 | M   | 2     |
|  3  | Nadia  | Saada    | 1991-01-15 | F   | 2     |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

اكتب استعلاما يظهر كل بيانات الطلاب المولودين في 1990-10-21 و يدرسون في 

القسم 1

mysql> Select * From Students
-> Where Date_n ='1990,10,21' And Classe = 1 ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  4  | Said   | Meddour  | 1990-10-21 | M   | 1     |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

معاملات أخرى Other Operators

المعامل	الوصف	الشرح
In	من	مثل أو Or لتحديد قيمة من مجموعة القيم
Like	مثل	حصر بيانات حسب مطابقتها للنص المعين
Between	بين	حصر بيانات بين قيمتين
Is Null	إنه فارغ	تحديد السجلات الفارغة

-1 مع المعامل IN

 توضع القيم المحددة بين قوسين و تفصل بين كل قيمة بفاصلة

mysql> Select * From Students ;


```

+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
| 1 | Salah  | Meddour  | 1967-10-23 | M   | 2      |
| 2 | Ali    | Aksa     | 1995-02-25 | M   | 1      |
| 3 | Nadia  | Saada    | 1991-01-15 | F   | 2      |
| 4 | Said   | Meddour  | 1990-10-21 | M   | 1      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.11 sec)

```

اكتب استعلاما يعرض كل بيانات الطلاب الذين ألقابهم إما Saada أو Meddour



```

Select * From Students
where Nom in('Meddour','Saada');

```

```

+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
| 1 | Salah  | Meddour  | 1967-10-23 | M   | 2      |
| 3 | Nadia  | Saada    | 1991-01-15 | F   | 2      |
| 4 | Said   | Meddour  | 1990-10-21 | M   | 1      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

```

اكتب استعلاما يظهر كل بيانات الطلاب الذين معدلاتهم إما 11 أو 12 أو 14



```

Select * From Grade
where Moyen in(11,12,14);

```

```

+-----+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Salah  | Meddour  | 14    | 14   | 14    | 14    |
| 2 | Aksa   | Ali      | 11    | 11   | 11    | 11    |
| 3 | Nadia  | Saada    | 12    | 12   | 12    | 12    |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

2- مع المعامل Between

اكتب استعلاما يعرض كل بيانات الطلاب الذين معدلاتهم محصورة بين 14 و 15 من

جدول النتائج

```

mysql> Select * From Grade
-> where Moyen Between 14 and 15 ;

```

```

+-----+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Arabe | Math | Franc | Moyen |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Salah  | Meddour  | 14    | 14   | 14    | 14    |
| 4 | Said   | Meddour  | 15    | 15   | 15    | 15    |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

اكتب استعلاما يظهر كل بيانات الطلاب المولودين بين 1990 و 2000



```

Select * From Students
-> where Year(Date_n) Between 1990 And 2000 ;

```

```

+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 2   | Ali    | Aks      | 1995-02-25 | M    | 1      |
| 3   | Nadia  | Saada    | 1991-01-15 | F    | 2      |
| 4   | Said   | Meddour  | 1990-10-21 | M    | 1      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

3- المعامل مثل Like

يقوم بعرض البيانات المطابقة أو المشابه لحرف أو نص أي يمكنك استخدام العامل **Like** للبحث عن قيم في حقل تتطابق مع النمط المعين

الصيغة العامة

```

Select Field1 , Field2 ....
From Table_Name
Where Field Like Exprition

```

الجدول التالي يبين الرموز المستعملة مع المعامل Like

الرمز	الدور	أمثلة
* %	تحل محل مجموعة من الحروف	A * تعرض كل الأسماء التي تبدأ ب A A* تعرض كل الأسماء التي تنتهي ب A *Ali* تعرض كل الأسماء التي تشمل على الحروف في موضع كانت
?	تحل محل حرف واحد	A?i تعرض مثلا Ali ، Ami ، Sai
[]	حرف واحد من مدى معين	AL[i,y] تعرض Ali ، ALy
#	تحل محل رقم واحد	F# تعرض F1 ، F2 ،
[!]	تستثني الحرف	*A[!q] تعرض كل الأسماء التي تنتهي بحرفين A و أي حرف آخر

يوضح الجدول التالي كيفية استخدام **Like** لاختبار تعبيرات لأنماط مختلفة.

نوع التطابق	النمط	التطابق	بلا تطابق
		(إرجاع " True صواب")	(إرجاع " False خطأ")
أحرف متعددة	a*a	aa, aBa, aBBBa	aBC
حرف خاص	a[*]a	a*a	aaa
أحرف متعددة	ab*	abc, AABBB, Xab	aZb, bac
حرف مفرد	a?a	aaa, a3a, aBa	aBBBa
رقم فردي	a#a	a0a, a1a, a2a	aaa, a10a
نطاق من الأحرف	[a-z]	f, p, j	2, &
خارج نطاق	[!a-z]	9, &, %	b, a

0, 1, 9	A, a, &, ~	[!0-9]	ليس رقمًا
abc, aj0	An9, az0, a99	a[!b-m]#	مدمج

إليك جدول الطلاب كاملاً :


```
mysql> Select * From Students ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 1967-10-23  | M    | 2      |
| 2   | Ali    | Aks      | 1995-02-25  | M    | 1      |
| 3   | Nadia  | Saada    | 1991-01-15  | F    | 2      |
| 4   | Said   | Meddour  | 1990-10-21  | M    | 1      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

اكتب استعلاماً يظهر كل بيانات الطلاب الذين أسماؤهم مثل Ali تماماً 


```
Select * From Students
where Prenom Like 'Ali' ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom | Date_N      | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 2   | Ali    | Aks | 1995-02-25  | M    | 1      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

اكتب استعلاماً يظهر كل بيانات الطلاب الذين أسماؤهم تبدأ ب S 

```
mysql> Select * From Students
-> where Prenom Like 'S%' ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 1967-10-23  | M    | 2      |
| 4   | Said   | Meddour  | 1990-10-21  | M    | 1      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

اكتب استعلاماً يظهر كل بيانات الطلاب الذين أسماؤهم تبدأ ب S و تنتهي ب h 

```
mysql> Select * From Students
-> where Prenom Like 'S%h' ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 1   | Salah  | Meddour  | 1967-10-23  | M    | 2      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

اكتب استعلاماً يظهر كل بيانات الطلاب الذين تنتهي أسماؤهم ب d 

```
mysql> Select * From Students
```

```
-> where Prenom Like '%d' ;
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
|  4 | Said   | Meddour  | 1990-10-21 | M   |      1 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

اكتب استعلاما يظهر كل الطلاب الذين تشتمل أسماؤهم على الحرفين ad مهما كان موضعهم

```
mysql> Select * From Students
-> where Prenom Like '%ad%' ;
```

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex | Classe |
+-----+-----+-----+-----+-----+-----+
|  3 | Nadia  | Saada   | 1991-01-15 | F   |      2 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

اكتب استعلاما يظهر كل بيانات الطلاب الذين أسماؤهم تبدأ ب Al و تنتهي بأحد الحروف

بين المدى i إلى y

```
SELECT * FROM Students Where Prenom Like 'Al[i-y]';
```

4- المعامل Is Null

اكتب استعلاما يظهر كل بيانات الطلاب الذين ليس لديهم تاريخ الميلاد أي فارغ

```
Select * From Students
where Date_n Is Null ;
Empty set (0.00 sec)
```

الناتج لا يوجد أي طالب

الترتيب و الفرز Order By

تستخدم لترتيب و فرز السجلات تصاعديا أو تنازليا وفق حقل أو حقول معينة

و تكتب في نهاية جملة Select

الصيغة العامة

```
SELECT fieldlist
FROM table
WHERE selectcriteria
[ORDER BY field1 [ASC | DESC ][, field2 [ASC | DESC ]][, ...]]
Select الحقل1 , الحقل2 , .....
From اسم الجدول
[Where الشرط]
Order by الحقل1, الحقل2, ..... [Asc/DEsc]
```

وتتكون من

fieldlist	أسماء الحقول
table	اسم الجدول

تحديد الشروط	selectcriteria
أسماء الحقول المراد ترتيبها أو فرزها	field1 , field2



- ASC تعني تصاعديا من أ إلى Z (إلى A)

- DESC تعني تنازلي

- نختار أحدهما و الافتراضي تصاعدي

- نستطيع تحديد نوع الترتيب لكل حقل على حده

- إذا كان الحقل الذي يتم عليه الترتيب Null فإنها تأتي في أول الاستعلام

mysql> Select * From Students ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1

4 rows in set (0.16 sec)

اكتب استعلاما يظهر كل بيانات الطلاب مرتبة تصاعديا حسب الاسم



mysql> Select * From Students Order by Prenom ;

Num	Prenom	Nom	Date_N	Sex	Classe
2	Ali	Aksa	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1
1	Salah	Meddour	1967-10-23	M	2

4 rows in set (0.03 sec)

و للترتيب العكسي نكتب

mysql> Select * From Students Order by Prenom Desc ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
4	Said	Meddour	1990-10-21	M	1
3	Nadia	Saada	1991-01-15	F	2
2	Ali	Aksa	1995-02-25	M	1

4 rows in set (0.00 sec)

و يمكن الترتيب بحقلين لكل حقل ترتيب على حده

mysql> Select * From Students Order by Prenom Asc ,Nom Desc ;

```

+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N   | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
| 2   | Ali    | Aks     | 1995-02-25 | M   | 1     |
| 3   | Nadia  | Saada   | 1991-01-15 | F   | 2     |
| 4   | Said   | Meddour | 1990-10-21 | M   | 1     |
| 1   | Salah  | Meddour | 1967-10-23 | M   | 2     |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```


جملة التجميع Group By

تقوم بتجميع سجلات الحقول في مجموعات حسب حقل أو حقول و حسب عامل التلخيص (Sum ,Count ,...)

الصيغة العامة
SELECT fieldlist FROM table WHERE criteria [GROUP BY groupfieldlist]
Select (حقل) دالة , حقل التجميع
From اسم الجدول
[Where الشروط على أي حقل ما عدا حقل التجميع]
Group by حقل التجميع
[Having الشرط على حقل التجميع]
Order by الحقل1, الحقل2,.....[Asc/DEsc]

تتكون جملة SELECT التي تحتوي على عبارة GROUP BY من الأجزاء التالية:

fieldlist	أسماء الحقول المراد تحديدها
table	اسم الجدول الذي يشتمل على هذه الحقول
criteria	تحديد الشروط
groupfieldlist]	أسماء الحقول المراد تجميعها و يجب ترتيبها من أعلى مستوى إلى أدنى

 يجب أن يكون الحقول المراد تجميعها مطابقة للحقول المحددة في أول الجملة


- ناتج الاستعلام قيمة رقمية و ليس سجلات
- ليس من المنطقي استعمال الترتيب Order By
- يستحسن كتابة اسم الحقل المقصود بناتج التجميع
- دالة العد إذا كتبت اسم الحقل تتجاهل الحقول الفارغة
- أما إذا كتبنا (*) Count فإن دالة العد تعد كل الحقول حتى و إن كانت فارغة
- التجميع يمكن أن يكون أكثر من حقل

دوال التجميع :

الوصف	العامل	الوصف	العامل
أكبر قيمة	Max	الجمع	Sum
حساب عدد	Count	أقل قيمة	Min
التباين	VAR	المعدل	AVG

Select * From Students ;

```
+-----+-----+-----+-----+-----+-----+
| Num | Prenom | Nom      | Date_N      | Sex  | Classe |
+-----+-----+-----+-----+-----+-----+
|  1  | Salah  | Meddour  | 1967-10-23  | M    | 2      |
|  2  | Ali    | Aks      | 1995-02-25  | M    | 1      |
|  3  | Nadia  | Saada    | 1991-01-15  | F    | 2      |
|  4  | Said   | Meddour  | 1990-10-21  | M    | 1      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```


اكتب استعلاما يحسب عدد الطلاب في كل القسم 

Select Classe , Count(Classe) As Total

From Students

Group by Classe ;

```
+-----+-----+
| Classe | Total |
+-----+-----+
| 1      | 2     |
| 2      | 2     |
+-----+-----+
2 rows in set (0.02 sec)
```

اكتب استعلاما يحسب عدد الطلاب في كل قسم حسب العمر و الجنس 

mysql> Select Classe ,Year(date_n),Sex ,

-> Count(Classe) as Total

-> From Students

-> Group by Classe ,Date_n ,Sex ;

```
+-----+-----+-----+-----+
| Classe | Year(date_n) | Sex  | Total |
+-----+-----+-----+-----+
| 1      | 1990         | M    | 1     |
| 1      | 1995         | M    | 1     |
| 2      | 1967         | M    | 1     |
| 2      | 1991         | F    | 1     |
+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

تقليص المجموع

تقوم بتقليص مجموع السجلات و فق الشروط المحددة فمثلا : إذا قمنا بتجميع الطلاب حسب الأقسام

و إذا أردنا تقليص و تخصيص المجموع نجمع السجلات فقط حسب القسم الأول

أنظر المثال :

1- باستخدام Where

هذا الجدول يعرض و يحسب عدد الطلاب في كل قسم حسب العمر و الجنس

```
mysql> Select Classe ,Year(date_n),Sex ,
-> Count(Classe) as Total
-> From Students
-> Group by Classe ,Date_n ,Sex ;
```

Classe	Year(date_n)	Sex	Total
1	1990	M	1
1	1995	M	1
2	1967	M	1
2	1991	F	1

4 rows in set (0.03 sec)

و إذا أردنا عرض نفس الجدول السابق و نخصص طلاب القسم الأول نكتب

```
mysql> Select Classe ,Year(date_n),Sex ,
-> Count(Classe) as Total
-> From Students
-> Where Classe = 1
-> Group by Classe ,Date_n ,Sex ;
```

Classe	Year(date_n)	Sex	Total
1	1990	M	1
1	1995	M	1

2 rows in set (0.05 sec)

2- باستخدام Having

```
Select Classe ,Year(date_n),Sex ,
Count(Classe) as Total
From Students
Group by Classe ,Date_n ,Sex
Having Classe = 1 ;
```

Classe	Year(date_n)	Sex	Total
1	1990	M	1
1	1995	M	1

2 rows in set (0.00 sec)

جملة INSERT INTO

تضيف سجلاً أو سجلات متعددة إلى جدول. يتم الإشارة إلى ذلك كاستعلام إلحاقى .

إليك جدول الطلاب

```
mysql> Select * From Students ;
```

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1

4 rows in set (0.00 sec)

اكتب استعلاما يضيف سجلا جديدا لجدول الطلاب 

Insert Into Students

(Num ,Prenom ,Nom ,Date_n ,Sex ,Classe)

Values

(5 ,'Aicha' ,'Meni' ,'22,10,15' ,'F' ,3) ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1
5	Aicha	Meni	2022-10-15	F	3

5 rows in set (0.00 sec)

يمكن الاستغناء عن كتابة أسماء الأعمدة و لكن يجب كتابة كل قيم الأعمدة الموجودة في جدول

الطلاب

Insert Into Students

Values

(6,'Farid','Aksa','1995,02,14' ,'M',3) ;

```
mysql> Select * From Students ;
```

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3

6 rows in set (0.00 sec)

كما يمكن إدخال قيم لبعض الحقول و لا تنسى يجب تحميل الأعمدة التي تقبل قيما فارغا

Insert Into Students

(Num ,Prenom ,Nom ,Sex)

Values

(7 ,'warda' ,'Sadaa' ,'F') ;

mysql> Select * from Students ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3
7	warda	Sadaa	NULL	F	NULL

7 rows in set (0.00 sec)

إضافة سجلات من جدول آخر

نقوم بإنشاء جدول جديد ننسخ إليه بعض سجلات جدول النتائج

Create Table StdBack
(N int Primary Key , Prenom Varchar(20) ,
Moyen Long);

Field	Type	Null	Key	Default	Extra
N	int(11)	NO	PRI		
Prenom	varchar(20)	YES		NULL	
Moyen	mediumtext	YES		NULL	

3 rows in set (0.02 sec)

وهذا جدول النتائج

Num	Prenom	Nom	Arabe	Math	Franc	Moyen
1	Salah	Meddour	14	14	14	14
2	Aksa	Ali	11	11	11	11
3	Nadia	Saada	12	12	12	12
4	Said	Meddour	15	15	15	15

4 rows in set (0.03 sec)

اكتب استعلاما ينسخ بيانات الطلاب من الرقم و الاسم و المعدل إلى الجدول الجديد



StdBack

Insert Into StdBack
(N , Prenom , Moyen)
Select Num ,Prenom ,Moyen
From Grade ;

```
mysql> Select * From StdBack ;
+----+-----+-----+
| N | Prenom | Moyen |
+----+-----+-----+
| 1 | Salah  | 14    |
| 2 | Aksa   | 11    |
| 3 | Nadia  | 12    |
| 4 | Said   | 15    |
+----+-----+-----+
4 rows in set (0.00 sec)
```

يمكن تجاهل كتابة الأعمدة في حالة ما إذا إدخال البيانات كل الأعمدة كما يلي :

```
Insert Into StdBack
Select Num ,Prenom ,Moyen
From Grade ;
```

و إذا أردنا نسخ جدول بكل بياناته إلى جدول آخر مماثل له تماما

```
" INSERT INTO table1 " _
  & "SELECT * " _
  & "FROM [table2];"
```

جملة التحديث و التعديل Update

تقوم بتعديل بيانات سجل أو سجلات و فق شرط ولا تغيير في بنية الجدول كالأعمدة

الصيغة العامة

```
UPDATE table
SET newvalue
WHERE criteria;
```

Update **اسم الجدول**

Set **القيمة الجديدة = الحقل**

القيمة الجديدة = الحقل 2

[Where **الشرط**]

و تتكون من

table	اسم الجدول الذي يتم فيه التعديل
newvalue	القيمة الجديدة للحقل أو الحقول التي يتم تعديلها
criteria	الشرط المحدد للتعديل للحقول



- في حالة عدم تحديد الشرط سيتم تعديل كل السجلات بالقيمة الجديدة

- يجب أن يكون نوع البيانات الجديدة من نفس نوع بيانات الأعمدة التي سيتم تعديلها

تعديل سجلات في جدول

إليك جدول الطلاب :

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Said	Meddour	1990-10-21	M	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3
7	warda	Sadaa	NULL	M	NULL

اكتب استعلاما يعدل كل الطلاب الذين أسماؤهم سعيد إلى سعيد و تغير الجنس



7 rows in set (0.00 sec)

Update Students

Set Prenom ='Saida' , Sex ='F'
where Prenom ='Said' ;

mysql> Select * From Students ;

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Saida	Meddour	1990-10-21	F	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3
7	warda	Sadaa	NULL	M	NULL

7 rows in set (0.00 sec)

أنواع الربط Type Of Joins

- الربط بالتساوي EquiJoin
- الربط بعدم التساوي Non_EquiJoin
- الربط الخارجي Outer Join
- الربط الداخلي في نفس الجدول Self Join

و يتم عمل هذه الأنواع في جملة Select في الجزء المخصص للشرط Where

ربط الجداول

الربط الطبيعي Where

الصيغة العامة

الحقل.الجدول2 , الحقل.الجدول1 Select

From الجدول2 , الجدول1

حقل الربط من الجدول2 = حقل الربط من الجدول1 Where

الربط بين أكثر من جدولين

Select 1 الجدول , الحقل. الجدول 2 , الحقل. الجدول 3 ,.....

From 1 الجدول , 2 الجدول, 3 الجدول,.....

Where 1 حقل الربط من الجدول = 2 حقل الربط من الجدول And

2 حقل الربط من الجدول = 3 حقل الربط من الجدول

- لا بد من استخدام الحقول التي تربط الجداول (حقول المفتاح الرئيسي أو المفتاح الخارجي) أو على الأقل حقول لها نفس المعنى و المدلول و إلا سيكون ناتج البط غير صحيح
- إذا كانت تود سجلات في أحد الجداول ليس لها قيمة مناظرة في الدول الآخر فإنها لن تطبع
- هام جدا : يجب أن يكتب الحقل كما يلي : اسم الحقل.اسم الجدول

اكتب استعلاما يظهر أرقام و أسماء الطلاب من جدول الطلاب و معدلاتهم من جدول النتائج

مثال

"Select Students.Num,Students.Prénom," _
& "Score.Moyen " _
& "from Students,Score" _
& " where Students.Num =Score.Num"

الربط بعدم التساوي

يتم الربط بهذا النوع عندما لا توجد علاقة مباشرة بين الجدولين و لا نستعمل = و نستعمل علاقة غير مباشرة مثل تطبيق شرط مناسب لربطهما
مثال : في جدول الموظفين فيه عمود الراتب Sal و جدول الفئات SalGrade وفيه عمود أعلى راتب و آخر لأدنى راتب و عمود الفئات يحدد حسب الراتب الأعلى و الأدنى فمثلا الراتب 3000 من الفئة 4 كما في الشكل

EMP table				SALGRADE table		
EMPNO	ENAME	JOB	SAL	GRADE	LOSAL	HISAL
7839	KING	PRESIDENT	5000	1	700	1200
7698	BLAKE	MANAGER	2850	2	1201	1400
7782	CLARK	MANAGER	2450	3	1401	2000
7566	JONES	MANAGER	2975	4	2001	3000
7654	MARTIN	SALESMAN	1250	5	3001	9999

SAL	GRADE
5000	5
2850	4
2450	4
2975	4
1250	2

البيانات التي توضح فئة كل راتب من رواتب الموظفين والتي حصلنا عليها من الجدولين

وهذا مثال يوضح ما قلناه

```
SQL> SELECT e.ename , e.sal , s.grade
2 FROM emp e , salgrade s
3 WHERE e.sal BETWEEN s.losal AND s.hisal ;
```

شرط الربط بين الجدولين

ENAME	SAL	GRADE
SMITH	800	1
ADAMS	1100	1
JAMES	950	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
ALLEN	1600	3
TURNER	1500	3
JONES	2975	4
BLAKE	2850	4
CLARK	2450	4

عملية INNER JOIN

يمكنك استخدام عملية INNER JOIN في أية جملة FROM. تعتبر هذه الصلة هي أكثر أنواع الصلات شيوعاً وتقوم الصلات الداخلية بدمج سجلات من جدولين أينما وجدت قيم متطابقة في حقل مشترك بين الجدولين.

الصيغة العامة

FROM table1 INNER JOIN table2 ON table1.field1 compopr table2.field2

تتكون عملية INNER JOIN من الأجزاء التالية:

الجزء	الوصف
<i>table1</i> <i>table2</i>	أسماء الجداول التي يتم تجميع السجلات منها.
<i>sfield1</i> <i>sfield2</i>	أسماء الحقول المتصلة. إذا لم تكن الحقول رقمية، فينبغي أن تكون من نفس نوع البيانات كما ينبغي أن تحتوي على نفس نوع البيانات، ولكن لا ينبغي أن يكون لها نفس الاسم.
<i>compopr</i>	أي عامل مقارنة علائقي: "=" أو "<" أو ">" أو "<=" أو ">=" أو "<>".

اكتب استعلاماً يظهر أرقام و أسماء الطلاب من جدول الطلاب و معدلاتهم من جدول النتائج

مثال

"SELECT Students.Num,Students.Prénom ,Score.Moyen " _
& "FROM Students INNER JOIN Score ON Students.Num = Score.Num;"

Ni	Préno	Ar	Ma	Fra	Mo	Cla	N	Prénom	Cla	Sex	
+	1	سامح	12	14	13	13	+	1	سامح	1	ذ
+	2	سعاد	12	12	12	12	+	2	سعاد	2	أ
+	3	داوود	14	14	14	14	+	3	داوود	3	ذ
+	14	وردة	11,5	11,5	11,5	11,5	+	4	وردة	2	أ
+	15	وسيم	13	13	13	13	+	5	وسيم	1	ذ
*							*				

	Num	Prénom	Moyen
▶	1	سامح	13
	2	سعاد	12
	3	داوو	14
*			

لاحظ أنه تم استرجاع السجلات التي لها نفس أرقام التسجيل

الربط الخارجي Outer Join

و يستعمل للربط بين الجداول التي لا يوجد بينها سجلات مناظرة و في هذه الحالة يتم طباعة السجل الأصلي بينما القيم المناظرة له من الجدول الآخر - التي هي غير موجودة - تطبع قيما فارغة Null

الربط الخارجي اليساري Left Outer Join

السجلات الموجودة في الجدول الأول (الأيسر) و التي ليس لها سجلات مناظرة في الجدول الآخر (الأيمن) سيتم استرجاعها و ستكون الحقول المناظرة لها في الجدول - الأيمن - فارغة

الصيغة العامة

```
FROM table1 [ LEFT | RIGHT ] JOIN table2
ON table1.field1 compopr table2.field2
```

الحقول

الجدول 1 Left Outer Join 2 الجدول

On

حقل ربط الجدول 2 = حقل ربط الجدول 1

نفس المثال السابق فقط نغير نوع الربط

مثال

```
"SELECT Students.Num,Students.Prénom ,Score.Moyen " _
& "FROM Students Left Outer Join Score ON Students.Num = Score.Num;"
```

	N	Prénom	Cla:	Sex
+	1	سامح	1	ذ
+	2	سعاد	2	أ
+	3	داوو	3	ذ
+	4	وردة	2	أ
+	5	وسيم	1	ذ
*				

الجدول 1

	Ni	Préno	Ar	Ma	Fra	Mo	Cla:
+	1	سامح	12	14	13	13	1
+	2	سعاد	12	12	12	12	2
+	3	داوود	14	14	14	14	3
+	14	وردة	11,5	11,5	11,5	11,5	2
+	15	وسيم	13	13	13	13	1
*							

الجدول 2

	Num	Prénom	Moyen
▶	1	سامح	13
	2	سعاد	12
	3	داوو	14
	4	وردة	
	5	وسيم	
*			

الربط الخارجي اليميني : Right Outer Join

السجلات الموجودة في الجدول الأول (الأيمن) و التي ليس لها سجلات منازرة في الجدول الآخر (الأيسر) سيتم طباعتها و ستكون الحقول المناظرة لها في الجدول - الأيسر - فارغة

الصيغة العامة

FROM table1 [LEFT | RIGHT] JOIN table2
ON table1.field1 compopr table2.field2

Select **الحقول**

From **الجدول 1** Right Outer Join **الجدول 2**

On

حقل ربط الجدول 2 = حقل ربط الجدول 1

نفس المثال السابق فقط نغير نوع الربط

مثال

"SELECT Students.Num,Students.Prénom ,Score.Moyen " _
& "FROM Students Right Outer Join Score ON Students.Num = Score.Num;"

	N	Prénom	Clas	Sex		N	Préno	Ar	Ma	Fra	Mo	Clas
+	1	سامح	1	ذ	+	1	سامح	12	14	13	13	1
+	2	سعاد	2	أ	+	2	سعاد	12	12	12	12	2
+	3	داوو	3	ذ	+	3	داوود	14	14	14	14	3
+	4	وردة	2	أ	+	14	وردة	11,5	11,5	11,5	11,5	2
+	5	وسيم	1	ذ	+	15	وسيم	13	13	13	13	1
*					*							

الجدول 1

الجدول 2

	Num	Prénom	Moyen
▶	1	سامح	13
	2	سعاد	12
	3	داوو	14
			11,5
			13
*			

جملة الحذف Delete

تقوم بحذف سجلات من جدول وفق شرط و حافظ على بنية الجدول

الصيغة العامة

```
DELETE [table[*].
FROM table
WHERE criteria
```

اسم الجدول Delete
اسم الجدول From
[الشرط Where]

تتكون جملة DELETE من الأجزاء التالية:

الوصف	الجزء
اسم الجدول الاخير.	table
اسم الجدول الذي يتم حذف سجلات منه.	table
الشرط المحدد للحذف	criteria

قبل الحذف يجب مراعاة علاقة هذا الجدول بالجدول الأخرى



```
mysql> Select * From Students;
```

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Saida	Meddour	1990-10-21	F	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3
7	warda	Sadaa	NULL	M	NULL

7 rows in set (0.17 sec)

اكتب استعلاما يحذف بيانات الطلاب الذين ليس لهم قسم أي فارغ Null



```
"DELETE * FROM Students " _
& " WHERE Classe Is Null;"
```

```
mysql> Select * From Students;
```

Num	Prenom	Nom	Date_N	Sex	Classe
1	Salah	Meddour	1967-10-23	M	2
2	Ali	Aks	1995-02-25	M	1
3	Nadia	Saada	1991-01-15	F	2
4	Saida	Meddour	1990-10-21	F	1
5	Aicha	Meni	2022-10-15	F	3
6	Farid	Aksa	1995-02-14	M	3

إليك جدول التالي :

```
mysql> Select * From Stdback;
```

N	Prenom	Moyen
1	Salah	14
2	Aksa	11
3	Nadia	12
4	Said	15

4 rows in set (0.03 sec)

و لحذف كامل السجلات نكتب

```
Delete * From Stdback ;
```

N	Prenom	Moyen
---	--------	-------

0 rows in set (0.03 sec)

الاستعلامات الفرعية

الصيغة العامة

Select الأعمدة أو الحقول المطلوبة
 From اسم الجدول
 Where (اسم الجدول From الحقول المطلوبة Select) عامل المقارنة الحقل

اكتب استعلاما يظهر بيانات الطلاب الذين يدرسون مع الطالب الصالح من جدول النتائج

مثال

```
Dim SQL As String = "Select Num , Prenom , Nom From Students " _  
& "where Classe = (Select Classe from Grades where Prenom = 'Salah' )"
```

يبدأ في تنفيذ الاستعلام الفرعي لا بد أن تكون نتيجة فريدة واحدة أي صف واحد

في مثالنا هذا يعيد الاستعلام الفرعي الناتج 2 لأن الطالب الصالح يدرس في القسم 2

بعد ذلك يعرض بيانات الطلاب من الاستعلام الرئيسي الموافقة لطلاب القسم 2 الذي تحصلنا عليه من الاستعلام الفرعي

Num	Prenom	Nom	Date_N	sexe	Classe	moye
1	Salah	Meddour	05/04/2000	M	2	9
2	Ali	Aksa	12/08/1999	M	1	11,5
3	Salima	Sadaa	15/09/2001	F	2	8
4	Amina	Aksa	01/01/2001	F	1	12

Nur	Prenom	Nom	Aral	Ma	Frai	Moye	Class
1	Salah	Meddour	9	9	9	9	2
2	Ali	Aksa	11,5	11,5	11,5	11,5	1
3	Salima	Sadaa	8	8	8	8	2
4	Amina	Aksa	12	12	12	12	1

النتيجة

Num	Prenom	Nom
1	Salah	Meddour
3	Salima	Sadaa

عملية الدمج UNION استعمال التوحيد

تقوم بإنشاء استعلام التوحيد ، الذي يقوم بدمج نتائج اثنين أو أكثر من الجداول أو الاستعلامات المستقلة

الصيغة العامة

[TABLE] query1 UNION [ALL] [TABLE] query2 [UNION [ALL] [TABLE] queryN[[...]

Select **جدول 1** From **الحقول**

Union

Select **جدول 2** From **الحقول**

Union

Select **جدول 3** From **الحقول**

تتكون عملية UNION من الأجزاء التالية:

الوصف	الجزء
جملة SELECT أو اسم استعلام مخزن أو اسم جدول مخزن مسبق بالكلمة الأساسية. TABLE.	query1-n



- عدد الحقول لا بد أن تكون متساوية

- الحقول يجب أن تكون متطابقة في النوع و الترتيب (ناتج غير صحيح)

- يتم عدم إظهار السجلات المكررة من نتائج الاستعلام

اكتب استعلام توحيد يدمج حقول من جدول الطلاب و النتائج

- جدول الطلاب (الرقم ، الاسم ، القسم)
- جدول النتائج (المعدل)

مثال

```
Dim SQL As String = "SELECT Num ,Prenom ,Moyen" _
& " FROM Grades" _
& " WHERE Classe = 1 UNION" _
& " SELECT Num ,Prenom , Moyen FROM Students" _
& " WHERE Classe = 1;"
```

Num	Prenom	Nom	Date_N	sexe	Clas:	Moyen
1	Salah	Meddour	05/04/2000	M	2	12
2	Ali	Aksa	12/08/1999	M	1	11
3	Salima	Sadaa	15/09/2001	F	2	9
4	Amina	Aksa	01/01/2001	F	1	13
*				'M'		

جدول النتائج

Nur	Prenom	Nom	Aral	Ma	Frat	Moye
*						

النتيجة

Num	Prenom	Moyen
2	Ali	11
4	Amina	13
*		

جملة الدمج مع إظهار التكرار Union All

يستخدم لدمج السجلات من جدولين متشابهين في البناء أو أكثر و يظهر السجلات المتكررة

الصيغة العامة

```
[TABLE] query1 UNION [ALL] [TABLE] query2 [UNION [ALL] [TABLE] queryN[[ ... ]
```

Select **جدول 1** From **الحقول**

Union All

Select **جدول 2** From **الحقول**

Union All

Select **جدول 3** From **الحقول**

```

+-----+
| Std      |
+-----+
| SalahMeddour |
| AliAks   |
| NadiaSaada |
| SaidaMeddour |
| AichaMeni |
| FaridAksa |
| wardaSadaa |
+-----+
7 rows in set (0.19 sec)

```

اكتب استعلاما يظهر أسماء الطلاب بحروف كبيرة

```

mysql> Select Ucase(Prenom)
-> From Students ;
+-----+
| Ucase(Prenom) |
+-----+
| SALAH        |
| ALI          |
| NADIA        |
| SAIDA        |
| AICHA        |
| FARID        |
| WARDA        |
+-----+
7 rows in set (0.00 sec)

```

الفصل التاسع : الفئات و الكائنات

- الفئات Classes
 - ما هي الفئة
 - إنشاء فئة جديدة
 - بناء أعضاء الفئة
 - الحقول Fields
 - الطرق Methods
 - المشيدات Constructors
 - الخصائص Properties
 - دراسة الصيغة القياسية
- الكائنات Objects
 - تعريفها
 - تساوي كائنن
 - عبارات خاصة بالكائنات
 - الكائن My
 - الخاصية Application
 - الخاصية Computer
 - المصادر Recoures
 - المستخدم User
 - كائن الخط
 - كائن Agent
 - كائن Skin

ما هي الفئة ؟

الفئة هي مجموعة من الصفات و الخصائص و الوسائل المشتركة نجعلها في برنامج واحد و نتخذها كمنوال و نموذج لإنشاء كائنات و برامج جديدة تحمل و ترث صفات المنوال . كما يمكنك تطويره و إضافة أشياء أخرى . و لهذا فكل شيء في إطار العمل .Net FrameWork ما هو إلا مجموعة هائلة من الفئات التي يمكن وراثتها و استخدامها و كذا تطويرها .

كيف نعرف فئة جديدة ؟

نعرّف متغيّراً من نفس نوع الخليّة، تماما كما نعرّف متغيّرا من نفس نوع السجّل

إنشاء فئة جديدة Create New Class

1. من قائمة File أنشئ مشروعا جديدا Creat New Project

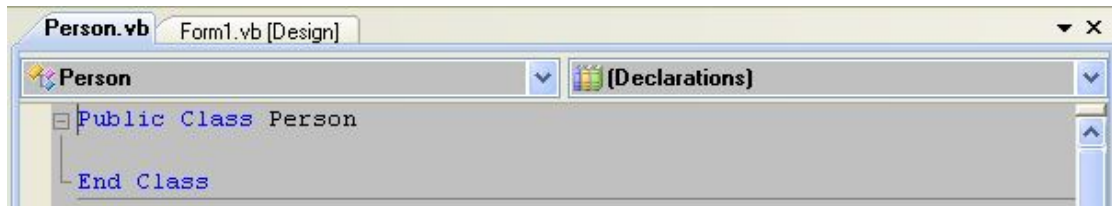
2. من قائمة Project ثم اختر Add New Item

يظهر مربع حوار إضافة عنصر جديد

3. اختر Add Class ثم اكتب في خانة Name اسما و ليكن Person



كما في الشكل : Class سيتم إنشاء فئة



كما قلنا تعرف الفئات كما تعرف السجلات Structures انظر المثال :


```
Dim pass_Wodd As String = 1
Dim User_Name As String = "Salah"
Sub New()
    ' وسائط بدون فارغا تركه '
    ' .....
End Sub
Sub New(ByVal Pw As String, ByVal Un As String)
    If pass_Wodd <> Pw And User_Name <> "Salah" Then End
End Sub
End Class
```

Public Class Form1

```
' الفئة من المشيد وسيطات إلى قيم ندخل لم أننا رغم الخطأ رسالة تظهر لا '
Dim p As New PassWord()
End Class
```

الخصائص Properties

هي إجراءات تتكون من مقطعين
 -Get: يتم استدعاؤه عندما تتم قراءة قيمة الخاصية.
 -Set: يتم استدعاؤه عندما يتم إسناد أو تغيير قيمة الخاصية.
 و يفضل استعمال متغير عام وسيط يحمل قيمة الخاصية كما في المثال التالي :

```
Class Persone
Dim m_BirthDate As Date
    ' متغير وسيط يحمل قيمة الخاصية '
Public Property BirthDate() As Date
    Get
        Return m_BirthDate
    End Get
    Set(ByVal value As Date)
        m_BirthDate = value
    End Set
End Property
End Class
```

ملاحظة : لا بد من تطابق نوع الوسيطة بالقيمة Value
 تستطيع التعامل مع الخصائص كما تتعامل مع الحقول بإسناد قيم لها و قراءتها

```
Dim Salah As New Persone
Salah.BirthDate = #10/23/1975#
Console.WriteLine(Salah.BirthDate)
    ' كما يمكنك التحكم في الخاصية عند إسناد القيم لها مع الإجراء Set هكذا :
Public Property BirthDate() As Date
    Get
        Return m_BirthDate
    End Get
    Set(ByVal value As Date)
        If value > Today Then ' إذا كان تاريخ الميلاد أكبر من تاريخ اليوم الحالي '

```


ملاحظة : الخصائص الافتراضية تكون من الخصائص التي تشمل وسيطات .
Parameters

الصيغة القياسية للفئات Classes

VB.Net Code

```
[attribute_list] [partial] [accessibility] [shadows] [inheritance] _  
Class name[(Of type_list)]  
[Inherits parent_class]  
[Implements interface]  
Statements()  
End Class
```

ملاحظة : العبارات بين قوسين [] اختيارية

يتضح لنا من الصيغة السابقة الشيء الوحيد الذي يجب كتابته الكلمة المحجوزة Class و End Class وبهذا يمكنك كتابة ما يلي :

VB.Net Code

```
Class Person
```

```
End Class
```

دعنا نتطرق بسرعة إلى المعاملات الاختيارية للصيغة القياسية

: Attribute_List

عبارة عن قائمة من الصفات و السمات تميز الفئة وتمنحها ميزة و سمة خاصة بالفئة .
مثلا : تريد أن تنشئ فئة (برنامج) يدعم خصائص السحب و الإفلات drag-and-drop
يجب عليك وصف الفئة على أنها serializable انظر المثال :

VB.Net Code

```
<Serializable> _  
Class clsPerson
```

```
End Class
```

Partial : تسمح الكلمة المحجوزة Partial بتقسيم الفئة إلى أجزاء

ما الفائدة من تجزئة الفئة ؟

لغرض تنظيم و تجميع التعليمات و وحدات الكود المرتبطة في حزمة منفردة لتسهيل فهم الكود.

VB.Net Code

```
Partial Class clsPerson
```

```
تعريف الشخص
```

```
Public FirstName As String
Public LastName As String
End Class
```

```
Partial Class clsPerson
```

تعريف بعنوان الشخص

```
Public city As Integer
Public country As String
End Class
```

Accessibility

هذا العامل يخص قابلية الوصول إلى الفئة.

: Public

قابلية الوصول إلى الفئة باستعمال الكلمة المحجوزة Public من داخل الفئة أو من خارجها أو من أي مشروع آخر. اعلم أنه يمكنك الوصول إلى فئة محتواة في فئة أخرى باستعمال الكلمة المحجوزة Protected تابع الكود التالي

VB.Net Code

```
Public Class Person
```

```
Public FirstName As String
Public LastName As String
Protected Address As PersonAddress
```

```
Protected Class PersonAddress
```

```
Public Street As String
Public City As String
End Class
End Class
```

Friend

قابلية الوصول إلى الفئة باستعمال الكلمة Friend من داخل الفئة أو خارجها و لكن داخل المشروع الحالي فقط

: Protected Friend

الفئة المصرحة بكلمة Protected Friend قابلة للوصول إليها من خلال فئة أخرى من نفس المشروع

Private

يمكن الوصول إلى الفئة المصرحة بكلمة Private من داخل الفئة نفسها ملاحظة : إذا لم تستعمل أيا من الكلمات المحجوزة للتصريح عن الفئة فإن الكلمة الافتراضية هي

Friend

Shadows

تستعمل لتظليل جزء من الكود و تصريحات بعض الكائنات المصرح عنها داخل فئة القاعدية

VB.Net Code

```
Class BaseClass
  Sub Method()
    ...
  End Sub
End Class
```

```
Class DerivedClass
  Inherits BaseClass
```

```
  Shadows Sub MyMethod()
    ...
  End Sub
End Class
```

:Inheritance

تستعمل لوراثة فئة أخرى تسمى الفئة القاعدية BaseClass أما الفئة الوارثة أو المشتقة Derived Class

VB.Net Code

```
Class Person ' الفئة القاعدية '
  Public Name As String
  Public Age As Integer
End Class
```

```
Class Employee ' الفئة المشتقة '
  Inherits Person ' Person وراثه الفئة
```

```
  Public Job As String
  Public Salary As Double
End Class
```

تعرفنا عن كيفية إنشاء الفئة والآن سنتعرف كيف نستخدم و نستدعي هذه الفئة أو أي فئة أخرى في أي مكان من مشروعك سيكون عليك استخدام العبارة New كما يلي:

VB.Net Code

```
Public Class Class1

End Class
```

```
Public Class Form1
    Dim Value As New Class1
    Private Sub Form1_Load(ByVal sender As System.Object,
        value,
        Equals
        GetHashCode
        GetType
        ReferenceEquals
        ToString
    End Sub
End Class
```

الأعضاء الافتراضية Default Members

تلاحظ بعد كتابة النقطة "." التي تلي اسم المتغير ظهور قائمة بأسماء جميع أعضاء الخلية Class1، وهي كالتالي:

"يساوي" Equals :

VB.Net Code

```
Public Class Form1
    Dim Value As New Class1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
        MsgBox(Value.Equals(Me))
    End Sub
End Class
```

عند فتحك للقوقس بعد العبارة Equals ستحصل على التلميح التالي:

```
▲ 1 of 2 ▾ Equals (obj As Object) As Boolean
obj: The System.Object to compare with the current System.Object.
```

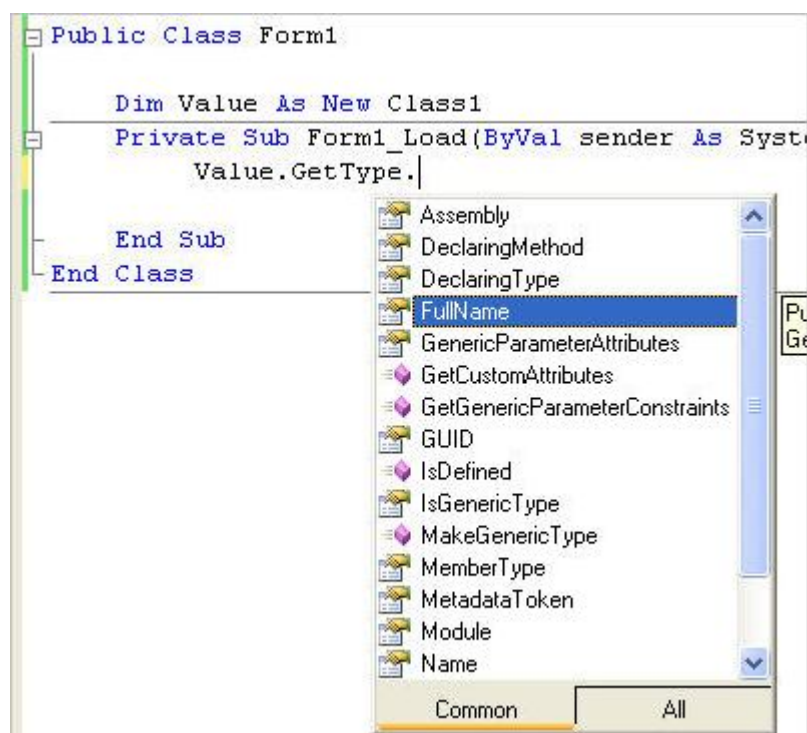
وتعني إجراء مقارنة للكائن مع كائن آخر لترى هل هو يشير إلى نفس الكائن أم لا، وبهذه الحالة طبعاً سنحصل على القيمة False والتي يمكن أن تعيدها لنا هذه الطريقة

"اقرأ الرمز" GetHashCode

تعيد عددا صحيحا، يميز الكائن عن باقي الكائنات

"اقرأ النوع" GetType:

ستعود لك بمعلومات عن الفئة مثل اسمه ونوعه وما إلى ذلك كما بالشكل التالي



فمثلا FullName ستعود لنا بالاسم الكامل بما فيه المرجع كما يلي :

VB.Net Code

```
Public Class Form1
    Dim Value As New Class1
    Dim Mybutton As New Button

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        MsgBox(Mybutton.GetType.FullName & vbCrLf &
Value.GetType.FullName)

    End Sub
End Class
```

"تساوي المرجع" **ReferenceEqual**:

(ويكافئ استخدامها استخدام المعامل IS لفحص تساوي كائنين مرجعيا).

- "حوّل إلى نص" **Tostring**:

ولتجربة هذه الوسائل، اكتب ما يلي في حادثة ضغط الزر

VB.Net Code

```
Dim obj1 As New Class1()
Console.WriteLine(obj1.ToString)
Console.WriteLine(obj1.GetType)
Console.WriteLine(obj1.GetHashCode)

Dim obj2 As New Class1()
Console.WriteLine(obj1.Equals(obj2))
Console.WriteLine(Class1.ReferenceEquals(obj1, obj2))
```

```
-----  
WindowsApplication1.Class1  
WindowsApplication1.Class1  
45592480  
False  
False
```

الكائنات Objects

كل شيء يقوم بوظيفة في Vb.Net فهو كائن و سندرس الكائنات من النوع المرجعي Reference Typ و تشمل الفئات Class أو المصفوفات Arrays أو النصوص String . و حتى تتمكن من التعامل معها نستعمل متغيرا معرفا منها و يسمى المؤشر .

```
Class Persone  
    Public Age As Integer  
End Class
```

نعرف مؤشر للوصول إلى الفئة Persone

```
Dim Salah As New Persone ' الفئة إلى اللوصل مؤشر  
Salah.Age = 45
```

تساوي كائنين

نفس الفئة السابقة Persone نضيف مؤشرا آخر Ali و نجعله مساويا للمؤشر Salah

```
Dim Salah As New Persone ' الفئة إلى اللوصل مؤشر  
Dim Ali As New Persone  
Ali.Age = 45  
Ali = Salah  
Salah.Age = 25  
Console.WriteLine(Ali.Age) ' 25
```

قمنا بتغيير قيمة المؤشر الصالح فتغيرت قيمة المؤشر Ali لتصبح القيمتين متساويتين و الآن تعال نضيف حقل آخر و نرى النتيجة

```
Dim Salah As New Persone ' الفئة إلى اللوصل مؤشر  
Dim Ali As New Persone  
Ali.Age = 45  
Ali = Salah  
Salah.Age = 25  
Salah.LastName = "Meddour"  
Console.WriteLine(Ali.Age) ' 25  
Console.WriteLine(Ali.LastName) 'Meddour
```

عبارات خاصة بالكائنات

الكلمة المجوزة New

الفصل التاسع الفئات و الكائنات

تستخدم لإنشاء كائن جديد و هذا عند التصريح عن المتغير (المؤشر) أو عند إسناد قيمة له بمعامل المساواة

استعمالها عند إنشاء الكائن '
Dim P As New Persone

استعمالها مع معامل المساواة'
Dim P1 As Persone
P1 = **New** Persone

التركيب With... End With

يستخدم للوصول السريع إلى أعضاء الكائن دون الحاجة لتكرار كتابة اسم الكائن

Dim P As New Persone
With P
 .Age = 45
 .LastName = "Salah"
End With

مثال : عن الأدوات

With OpenFileDialog1
 .Filter = "All text (*.txt)|*.txt| All File (*.*)|*.*"
 .ShowDialog()
 .If.FileName = "" **Then** Exit Sub
End With

القيمة Noting

يستخدم لإنهاء العلاقة بين المؤشر و الكائن الذي يشير إليه.

Dim P As New Persone
P.LastName = "Meddour"
P = **Nothing**
Console.WriteLine(P.LastName)' تظهر رسالة خطأ

المعامل Is

يستعمل للتحقق من تساوي كائنين بدلا من المساواة

Dim P1 As New Persone
Dim P2 As New Persone
Console.WriteLine(P1 **Is** P2) **'False**

يعيد المعامل Is قيمة منطقية فهو مناسب جدا مع الجمل الشرطية انظر المثال:

Dim P1 As New Persone
P1 = **Nothing**
If P1 **Is** **Nothing** **Then**
 P1 = **New** Persone

الفصل التاسع الفئات و الكائنات

End If

التركيب Typeof...Is

يمثل المعامل Is و لكنه يستخدم للتحقق من نوع الفئة التي يشير إليها المتغير انظر المثال

```
Dim P1 As New Persone
If TypeOf P1 Is Persone Then
    P1.Name = "Salah"
End If
```

مثال آخر عن الأدوات

```
For Each txt As Control In Me.Controls
    If TypeOf txt Is TextBox Then
        txt.Text = ""
    End If
Next
```

الكائن Me


يشير إلى الكائن الحالي أو إلى النسخة الحالية من الكائن

```
Class Persone
    Public Name As String
    Public Age As Integer
    Sub PrintPersone()
        Console.WriteLine(Me.Name)
    End Sub
End Class
```

الكائن My

يعبر عن الجهاز . فبمجرد أن تكتب في محرر الكود الكلمة المحجوزة My متبوعة بنقطة تظهر أمامك كل أعضاء الكائن بالشكل

معلومات عن المشروع الحالي	Applicaion	
للتعامل مع جهاز الكمبيوتر	Computer	Application
للتعامل مع النماذج	Forms	Computer
للتعامل مع ملفات المصادر	Resources	Forms
للتعامل مع الإعدادات	Setting	Resources
للتعامل مع اسم المستخدم	User	Settings
خدمات Web	WebServices	User
		WebServices



أولا : الخاصية Application (تطبيق)

تزودك بكل خصائص و طرق و أحداث التطبيق الحالي

لمعرفة كل المعلومات عن التطبيق الحالي نستعمل الفئة Info و تحتوي على الخصائص حسب

الجدول التالي

MsgBox(My.Application.Info.CompanyName)	اسم البلد
---	-----------

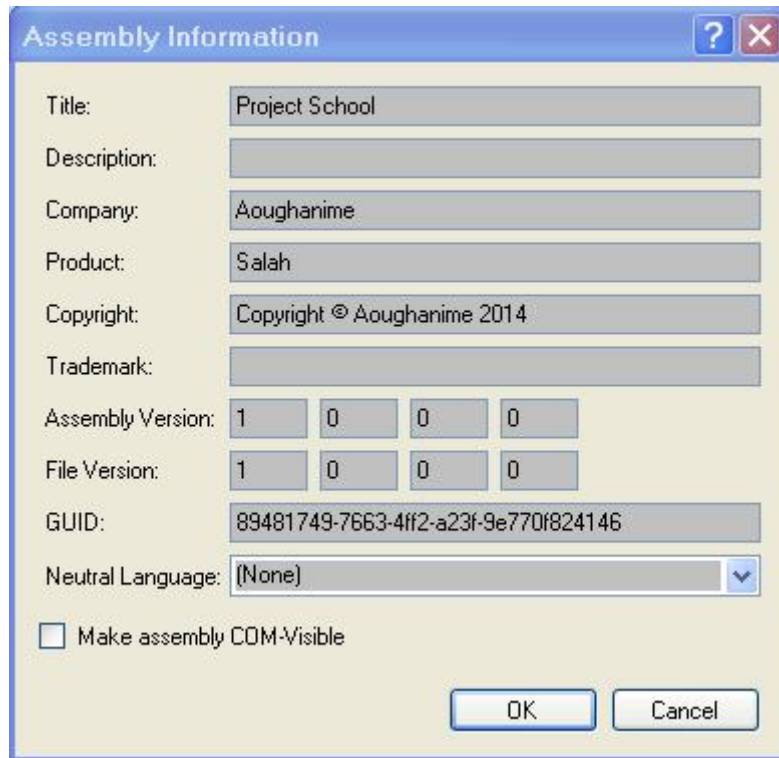
الفصل التاسع الفئات و الكائنات

MsgBox(My.Application.Info.Copyright)	حقوق النسخ
MsgBox(My.Application.Info.DirectoryPath)	مسار التطبيق
My.Application.Info.Version.ToString	إصدار التطبيق
MsgBox(My.Application.Info.Title)	عنوان التطبيق
MsgBox(My.Application.Info.Description)	معلومات

مثال :

```
Console.WriteLine(My.Application.Info.Title)
Console.WriteLine(My.Application.Info.ProductName)
Console.WriteLine(My.Application.Info.CompanyName)
Console.WriteLine(My.Application.Info.Copyright)
Console.WriteLine(My.Application.Info.AssemblyName)
Console.WriteLine(My.Application.Info.Version)
```

هذا المثال يعرض معلومات المشروع الحالي حسب ما ورد في نافذة خصائص المشروع ثم Assomby Information

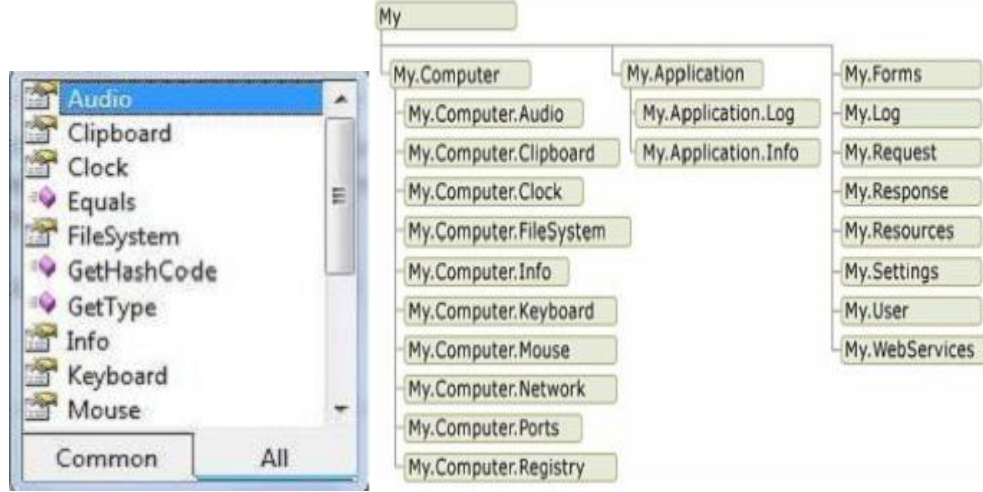


يكون الناتج كما يلي :

```
Project School
Salah
Aoughanime
Copyright © Aoughanime 2014
WindowsApplication1
1.0.0.0
```

ثانيا : الخاصية Computer

تزوّدك بمعلومات عن البرامج و الأجهزة المتعلقة بالكمبيوتر مثل : الصوت و الساعة و الحافظة و الماوس و الطابعة و الريجستري و الملفات... حسب الشكل التالي :



أهم خصائص الجهاز Compteur

الاسم Name

تعيد اسم الكمبيوتر : `MsgBox(My.Computer.Name)`

الصوت Audio

يمكن تشغيل الصوت من نوع Wav انظر المثال :

`My.Computer.Audio.Play("C:\Waterfall.wav")`

و تشمل خاصية الصوت الوسائل الهامة التالية

الوسيلة	الوظيفة	EXP
Play	تشغيل	<code>My.Computer.Audio.Play("C:\Waterfall.wav")</code>
PlaySystemSound	تشغيل نوع الصوت	<code>My.Computer.Audio.PlaySystemSound(_ System.Media.SystemSounds.Asterisk)</code>
Stop	وقف	<code>My.Computer.Audio.stop()</code>

الحافظة Clipboard

كل الأوامر المتعلقة بنسخ و قص و لصق ونقل البيانات في الجهاز نتعامل معها باستعمال كائن الحافظة

الوسيلة	الوظيفة	EXP
Clear	مسح	<code>My.Computer.Clipboard.Clear()</code>
ContainsAudio GetAudio	تحتوي صوت اقرأ الصوت	<code>If My.Computer.Clipboard.ContainsAudio Then My.Computer.Audio.Play(My.Computer.Clipboard.ContainsAudio) End If</code>

الفصل التاسع الفئات و الكائنات

<code>If My.Computer.Clipboard.ContainsImage Then PictureBox1.Image = My.Computer.Clipboard.GetImage End If</code>	يحتوي صورة اقرأ الصورة	ContainsImage GetImage
<code>If My.Computer.Clipboard.ContainsText Then TextBox1.Text = My.Computer.Clipboard.GetText End If</code>	يحتوي نص اقرأ النص	ContainsText GetText
<code>My.Computer.Clipboard.SetImage(coolPicture)</code>	نسخ صورة	SetImage
<code>My.Computer.Clipboard.SetText(TextBox1.Text)</code>	نسخ نص	SetText

الوقت Clock

EXP	الوظيفة	الوسيلة
<code>MsgBox(My.Computer.Clock.GmtTime)</code>	وقت Gmt	GmtTime
<code>Me.Text = My.Computer.Clock.LocalTime</code>	الوقت المحلي	LocalTime

التعامل مع الملفات FileSystem

سندرس هذا مع الفصل الخاص بالملفات

المصادر Recours

يمكن قراءة و استدعاء المصادر التي تم إضافتها إلى المشروع عن طريق نافذة خصائص المشروع

`Me.Text = My.Resources.String1`

`Me.PictureBox1.Image = My.Resources.Image1`

المستخدم User

هذا المثال يعرض اسم المستخدم

`MsgBox(My.UserName)`

كائن الخط Font Object

لإنشاء كائن الخط استعمل الصيغة البسيطة التالية :

`Dim MyFont As New Font(نوعه , حجمه, اسم الخط)`

Exp

`Dim MyFont As New Font("Arial", 14, FontStyle.Bold)`

المعامل الثالث نوع الخط ممكن أن يكون

توضيح	نوع الخط
غليظ	Bold
مائل	<i>Italic</i>
عادي	Regular
يعبره خط	Strikeout
تسطير	<u>Underline</u>

تنسيق الخط حسب مربع الحوار

VB.Net Code

```
If FontDialog1.ShowDialog = DialogResult.OK Then
    TextBox1.Font = FontDialog1.Font ' مربع حوار الخط
    TextBox1.ForeColor = FontDialog1.Color ' مربع حوار اللون
End If
```

عرض قائمة الخطوط

VB.Net Code

```
Dim f As New System.Drawing.Text.InstalledFontCollection
Dim fFamily As FontFamily
For Each fFamily In f.Families
    ListBox1.Items.Add(fFamily.Name)
Next
```

الكائن Agent

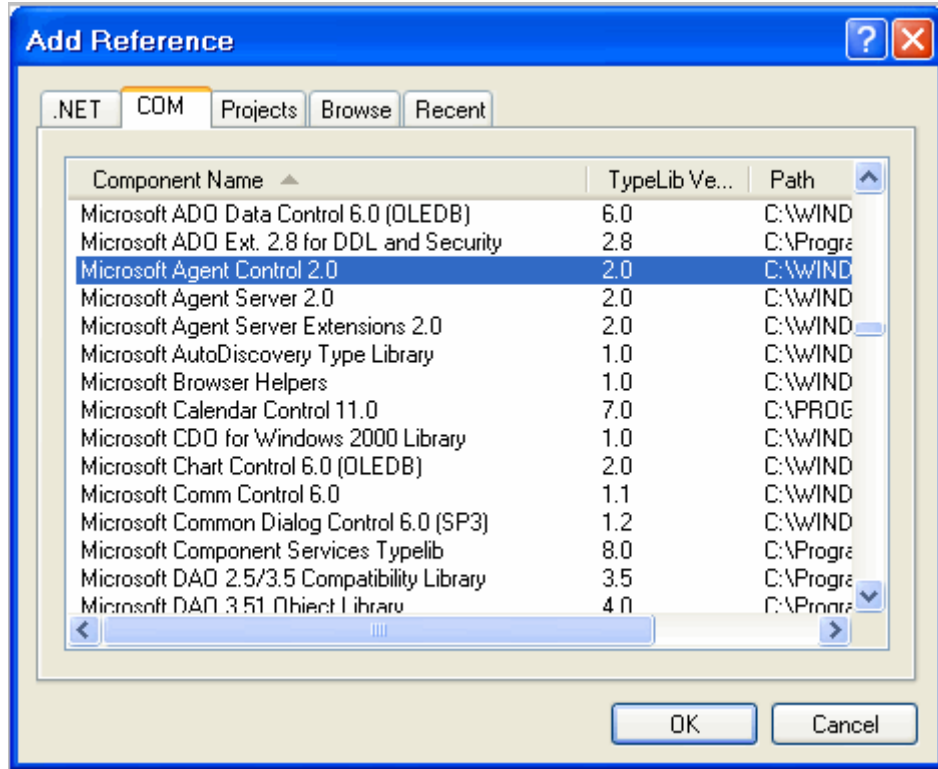
لعلك لاحظت أن برامج الأوفيس تستخدم بعض الشخصيات الكرتونية لتقديم بعض التعليمات أو التلميحات مثل الكائن الساحر Merlin . لعمل ذلك في برنامج اتبع الخطوات التالية :

هذه إحدى الشخصيات وتسمى Merlin وتعني الساحر او العراف، كما توجد العديد من الشخصيات الأخرى.



بداية أنشئ نموذجاً جديداً .

تم عليك باستيراد هذه المكتبة من الخيار Reference كما يظهر بالصورة.



المكتبة المطلوبة هي Microsoft Agent Control كما يظهر بالصورة حددها وأقر Ok
وبقسم التصاريح العامة يجب تعريف هذه المتغيرات كما بالشكل:

Agent بالكائن خاص متغير ' AgentObjects.Agent As New Public Agent

الكائن بحركات خاص متغير ' AgentObjects.IAgentCtlCharacter As New Public CrAgent

ضع زراً على النموذج وأكتب فيه التالي

```
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    With Agent
        .Connected = True ' بالكائن الاتصال
        .Characters.Load("Merlin", "Merlin.acs") ' Agent الكائن نوع تحديد
        CrAgent = .Characters("Merlin") ' بالكائن الحركات متغيرات إلحاق
    End With
    CrAgent.Show() ' بحركات الكائن عرض
End Sub
```

و الآن نقوم بإعطاء الأوامر لهذا الساحر Merlin بأن يقوم ببعض الحركات : مثل القراءة و الكتابة و التفكير الخ



الفصل التاسع الفئات و الكائنات

الآن ضع زراً آخر على النموذج وأكتب فيه التالي

```
Private Sub Button2_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button2.Click
    With CrAgent
        .MoveTo(200, 100, 150) ' محدد مكان إلى نقله '
        .Play("Writing") ' يكتب '
    End With
End Sub
```

طبعاً يوجد العديد من الأوامر وسوف اسردها هنا في قائمة وجربها

GestureDownReturn	DeclineReturn	Blink	Acknowledge
GestureLeft	DontRecognize	Confused	Alert
GestureLeftReturn	DontRecognizeReturn	ConfusedReturn	AlertReturn
GestureRight	Explain	Congratulate	Announce
GestureRightReturn	ExplainReturn	CongratulateReturn	Applaud
GestureUp	GestureDown	Decline	AnnounceReturn
Idle3_1	Hear_4	GlanceUp	GestureUpReturn
Idle3_1Return	Hide	Greet	GetAttention
Idle3_2	Idle1_1	GreetReturn	GetAttentionReturn
LookDown	Idle1_2	Hear_1	GlanceDown
LookDownReturn	Idle2_1	Hear_2	GlanceLeft
LookLeft	Idle2_2	Hear_3	GlanceRight
ProcessingReturn	MoveUp	MoveDownReturn	LookLeftReturn
Read	MoveUpReturn	MoveLeft	LookRight
ReadContinued	Pleased	MoveLeftReturn	LookRightReturn
ReadReturn	PleasedReturn	MoveRight	LookUp
Reading	Processing	MoveRightReturn	LookUpReturn
ReadingReturn	Uncertain	Suggest	MoveDown
RestPose	UncertainReturn	SuggestReturn	SadReturn
Sad	Wave	Surprised	Searching
WriteContinued	WaveReturn	SurprisedReturn	SearchingReturn
WriteReturn	Write	Think	Show
Writing		ThinkReturn	StartListening
WritingReturn			StartListeningReturn
			StopListening
			StopListeningReturn

مثال آخر

```
With CrAgent
    .Balloon.Style = 5 ' الإطار نوع '
    .Speak("لمساعدتك هنا انا مرحباً")
End With
```

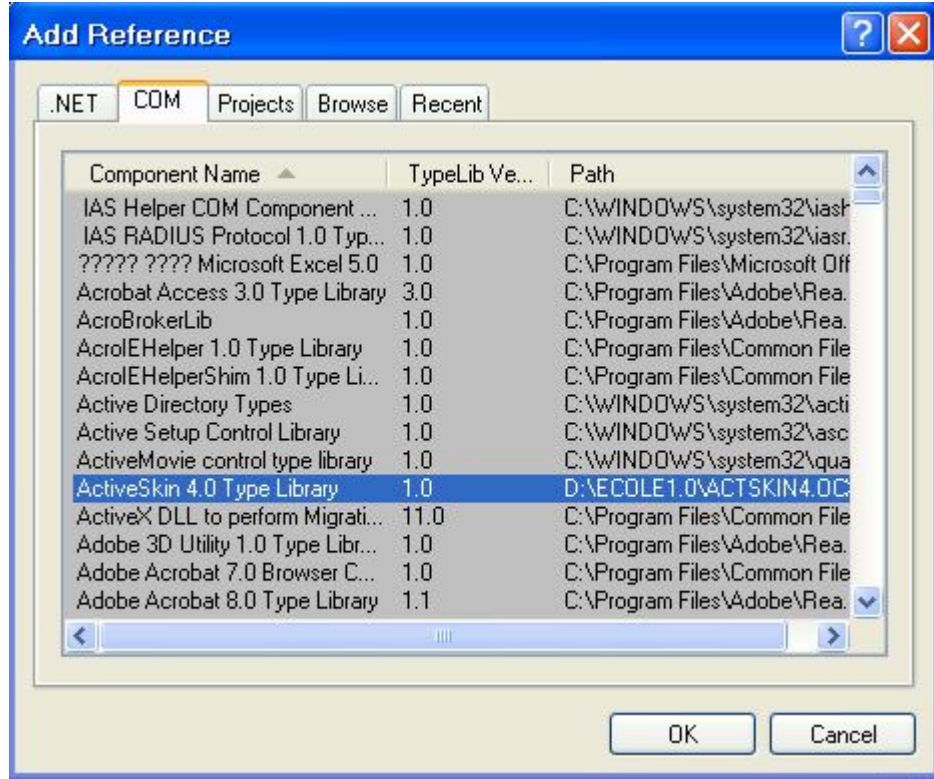


كائن Skin

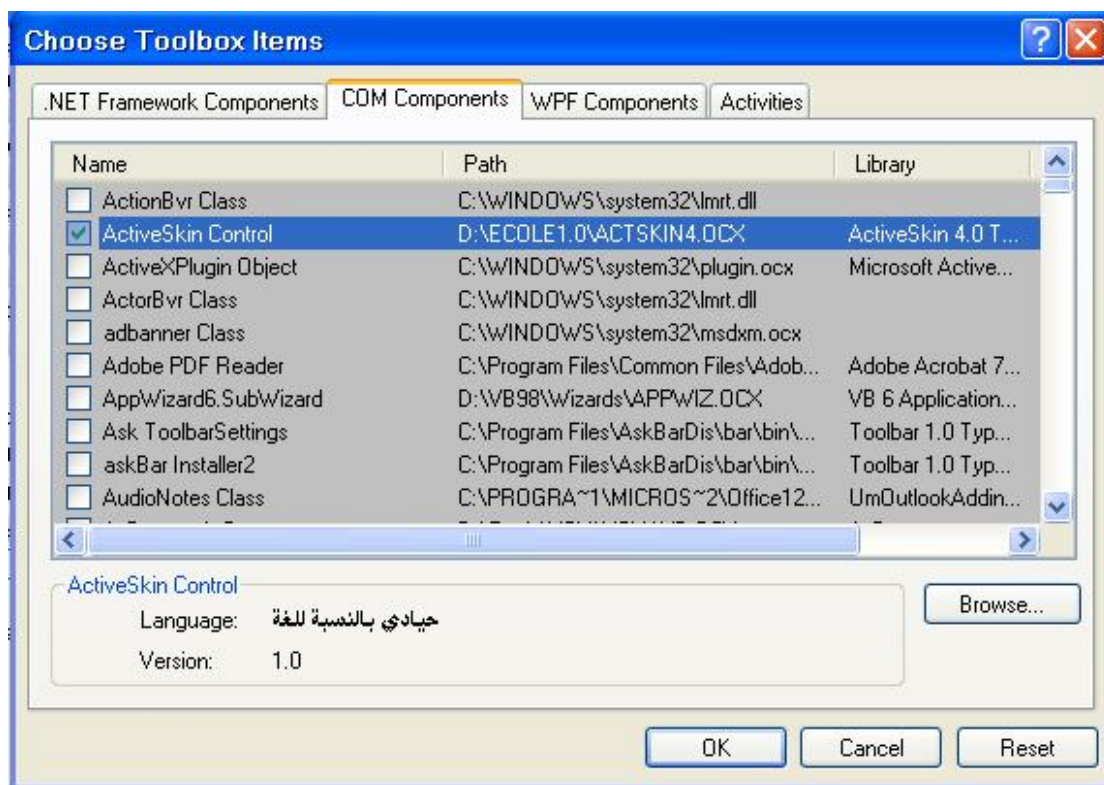
الفصل التاسع الفئات و الكائنات

يقوم بتنسيق النموذج فيجعله أكثر روعة و أناقة . و لعمل ذلك في برنامجك اتبع الخطوات التالية :
بداية أنشئ نمودجا جديداً .

تم عليك باستيراد هذه المكتبة من الخيار Reference كما يظهر بالصورة.



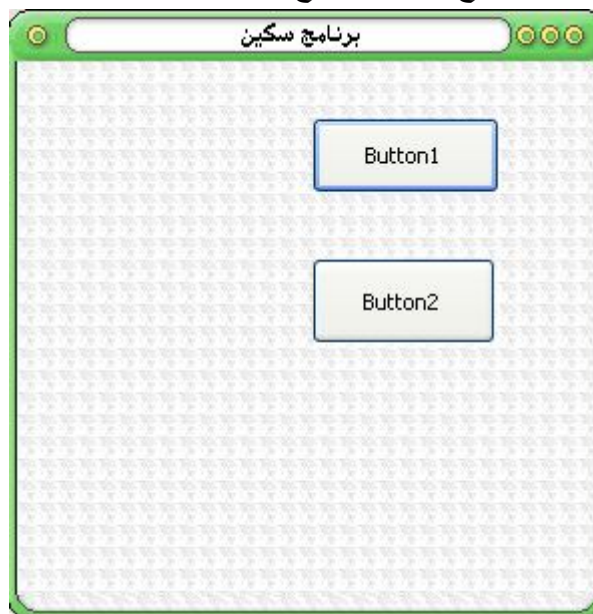
حدد المكتبة المطلوبة كما في الشكل ثم انقر Ok
ثم نضيف الأداة Axskin و هذا بالنقر بالزر الأيمن على صندوق الأدوات Tool box ثم اختر
الأمر Choose Item يظهر مربع الحوار



ثم حدد الخيار كما هو في الشكل و انقر Ok
 قم بإضافة الأداة إلى النموذج و في حدث تحميل النموذج نكتب

```
Private Sub Form1_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles Me.Load
    AxSkin1.LoadSkin("c:\9.SKN") 'سكين موقع تحديد
    AxSkin1.ApplySkin(0) 'عرض
End Sub
```

شغل البرنامج و لاحظ الناتج . ألف مبروك !



Inheritance الوراثة

تعتبر الوراثة من أهم مواضيع لغة البرمجة غرضية التوجه OOP . و تفيدك في وراثة و استثمار كائنات و فئات و أكواد موجودة دون الحاجة لإعادة إنشائها و كتابة أكوادها من جديد و يمكنك تطويرها و إضافة أشياء جديدة لها.

مصطلحات هامة :

كيف نبني خلية جديدة (تسمى الخلية الفرعية Subclass أو الخلية المشتقة Derived Class) من خلية قديمة (تسمى الخلية الأم Parent Class أو الخلية الأساسية Base Class)، بحيث ترث كل خصائص ووسائل الخلية القديمة، مع إمكانية احتوائها على خصائص جديدة، أو تعديل عمل بعض الخصائص القديمة باستبدالها بأخرى معدلة Overriding.

مثلا : عندنا فئة تمثل شخص Person بها خصائص كالاسم Name و العمر Age يمكننا اشتقاق فئة أخرى منها Employee لترث الخاصيتين من الفئة القاعدية الأم و إضافة خصائص أخرى جديدة.

```
Class Person ' القاعدة الفئة ' base
    Public Name As String
    Public Age As Integer
End Class

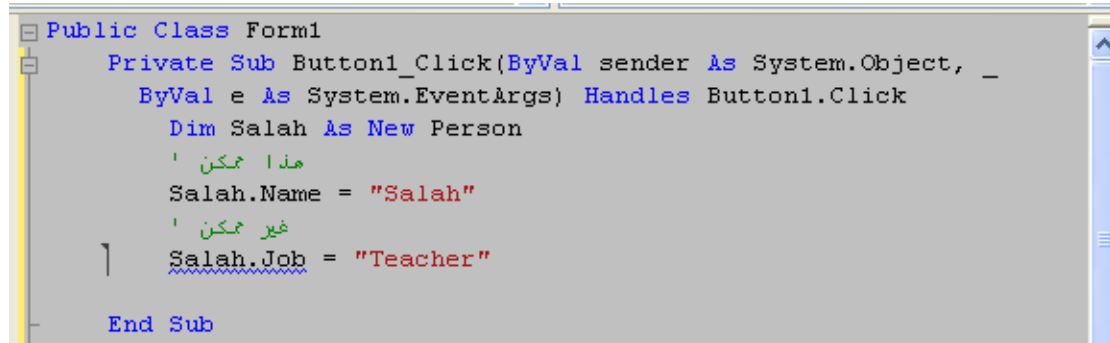
Class Employee ' المشتقة الفئة Derived
    Inherits Person ' القاعدية الفئة وراثة
    Public Address As String
    Public Job As String
End Class
```

و الآن ننشئ كائنا جديدا يمثل الفئة المشتقة لنصل إلى جميع الأعضاء في الفئتين.

```
Dim Salah As New Employee

Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    ' القاعدية الفئة من أعضاء Base
    Salah.Name = "Salah"
    Salah.Age = 45
    ' المشتقة الفئة من أعضاء
    Salah.Job = "Teacher"
    Salah.Address = "Alger"
End Sub
```

ملاحظة : لا يمكنك الوصول إلى أعضاء أو حقول الفئة المشتقة باستدعاء الفئة القاعدية .



```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim Salah As New Person
        ' هذا ممكن
        Salah.Name = "Salah"
        ' غير ممكن
        Salah.Job = "Teacher"
    End Sub
End Class
```

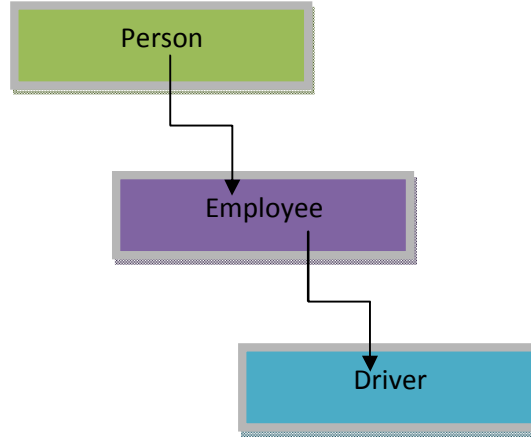
يمكنك اشتقاق أو وراثة الفئة Person و كذا الفئة المشتقة Employee

```
Class Driver
    Inherits Person

    Public CarModel As String
```

الفصل العاشر الوراثة

End Class



العلاقات الوراثة بين الفئات

لا تقتصر عملية الوراثة على اشتقاق الحقول فقط بل حتى الطرق و الخصائص و الأحداث

```
class Person ' القاعدة الفئة base
    Public Name As String ' حقل
    Public Age As Integer

    ' وراثة طريقة
    Sub SetValue(ByVal Name_Person As String, ByVal Age_Person As
Integer)
        Name = Name_Person
        Age = Age_Person
        Console.WriteLine("My Name is : {0} , I am {1} year Old ",
Name, Age)
    End Sub
    ' وراثة خاصية
    Public birthDay As Date
    Property birth_Day() As Date
        Get
            Return birthDay
        End Get
        Set(ByVal value As Date)
            birthDay = value
        End Set
    End Property
End Class
```

End Class

```
Dim Salah As New Person
    Salah.Name = "Salah"
    Salah.birth_Day = #10/23/1990#
    Console.WriteLine(Salah.birth_Day)
```

ليس هذا فقط بل حتى يمكن وراثة الحقول المشتركة

```
Class Person
    Public FirstName As String
    ' حقل مشتركة
    Public Shared LastName As String = "Meddour"
End Class
Class employee
    Inherits Person
    Public Job As String
End Class
```

الفصل العاشر الوراثة

```
Dim Salah As New employee

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    Salah.FirstName = "Salah"
    Console.WriteLine("My Name is {0} , {1}", Salah.FirstName,
employee.LastName)

End Sub
```

وراثة خلايا إطار العمل Framework

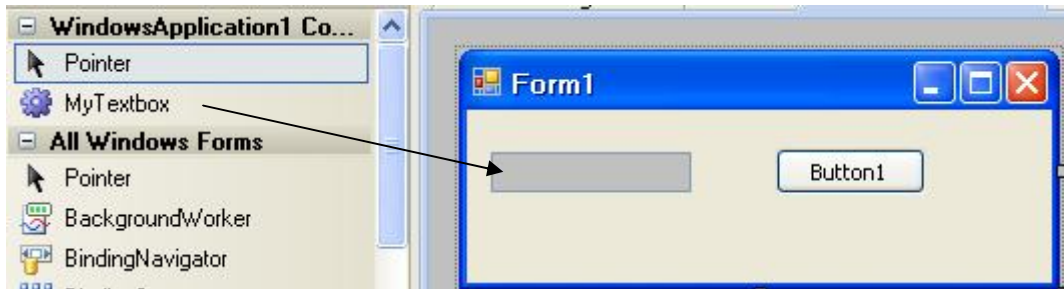
في مثالنا هذا نقوم بوراثة الأداة TextBox و تطويرها بإضافة خصائص جديدة لها
أولا : إضافة دالة تقوم بعكس حروف النص

```
Class MyTextbox
' وراثة الخلية القاعدية
Inherits TextBox
' إضافة دالة تقوم بعكس النص
Public Function Reve(ByVal x As String) As String
    Return StrReverse(x)
End Function

End Class
```

قم بحفظ المشروع ثم نشره Build من قائمة البناء build

ثم من صندوق الأدوات قم بإدراج أدايتك الجديدة باسم mytextbox إلى النموذج كما في الشكل



عد إلى فئة النموذج و اكتب في حدث نقر الزر

```
Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click
    Console.WriteLine(test.ReverseTxt("Salah")) ' halaS
End Sub
```

ثانيا : إضافة خاصية جديدة تعرض تاريخ إنشاء النص

```
Public myDate As Date
Property CreateDate() As Date
Get
    Return myDate
End Get
Set(ByVal value As Date)
    myDate = value
End Set
End Property
```

عد إلى فئة النموذج و اكتب في حدث نقر الزر

```
MyTextbox1.CreateDate = Now
Console.WriteLine(MyTextbox1.CreateDate)
```

ثالثا : إضافة حدث

لإضافة حديث الى فئتنا نضيف هذه العبارة

الفصل العاشر الوراثة

```
Event msEvent (ByVal ms As String)
```

الآن قم بعملية البناء مرة أخرى لتنفيذ التعديلات الجديدة

و انتقل إلى محرر الكود ستجد الحدث msEvents تم إدراجه لأداتنا



إعادة القيادة Overriding

كتبت إجراء يعرض الاسم ثم اللقب العائلي و لكن طلب مني عرض اللقب العائلي ثم الاسم ماذا افعل ؟
أفضل حل هو إعادة قيادة الإجراء و هذا الفئة المشتقة و هذا ما إذا أردت كتابة الاسم ثم اللقب أعود إلى الفئة
القاعدية .

```
Class Person
Public FirstName As String = "Salah"
Public LastName As String = "Meddour"
' القاعدة الفئة في الدالة قيادة بإعادة للسماح
' overriable نستعمل
Public Overridable Function CompleteName() As String
Return FirstName & " " & LastName
End Function

End Class
```

و الآن يمكنك إعادة قيادة هذه الدالة CompleteName في الفئات المشتقة

```
Class Employee
Inherits Person
Public job As String
' ممكن الدالة اعادة
Public Overrides Function CompleteName() As String
Return LastName & ", " & FirstName
End Function

End Class
```

عند استدعاء الدالة من الفئة القاعدية Person سيتم تنفيذ الاسم ثم اللقب و عند استدعائها من الفئة المشتقة Employee سيتم تنفيذ اللقب ثم الاسم

```
Dim Salah As New Person
Dim Salah2 As New Employee

Console.WriteLine(Salah.CompleteName()) ' Salah Meddour
Console.WriteLine(Salah2.CompleteName) ' Meddour Salah
```

ملاحظة : يمكنك استعمال إعادة التعريف OverLoading بدل من إعادة القيادة Overrides

```
Class BaseClass
Overloads Sub TestMethod()
Console.WriteLine("القاعدية الفئة")
End Sub
End Class

Class DerivedClass
Inherits BaseClass

Overloads Sub TestMethod(ByVal x As Integer)
Console.WriteLine("المشتقة الفئة")
End Sub
End Class
```

الفصل العاشر الوراثة

مصطلحات وراثية هامة :

استخدام MyBase

كلمة Me تشير إلى الكائن الحالي أو الخلية الحالية

أما كلمة My Base تشير إلى الخلية التي ترثها الخلية الحالية

```
Class BaseClass
    Sub MethodInBase ()
        ...
    End Sub
End Class
```

```
Class DerivedClass
    Inherits BaseClass

    Sub MethodInDerived()
        ' القاعدية الفئة في الطريقة استدعاء
        MyBase.MethodInBase ()

        ' أيضا القاعدية الفئة في الطريقة استدعاء
        Me.MethodInBase ()

    End Sub
End Class
```

صحيح أن المؤشر Me يمكننا من استدعاء أعضاء الفئة القاعدية وهذا فقط في حالة عدم وجود طريقة معادة

التعريف Over Loading

```
Class BaseClass
    ...
    ...
    Overridable Sub MyMethod ()
        ...
    End Sub
End Class
Class DerivedClass
    Inherits BaseClass
    ...
    ...
    Overrides Sub MyMethod ()
        ...
    End Sub

    Sub TestMethod ()
        ' القاعدية الفئة من الطريقة استدعاء
        MyBase.MyMethod ()

        ' المشتقة الفئة من الفئة استدعاء
        Me.MyMethod ()
    End Sub
End Class
```

من هنا يتضح لنا أن My Base موجهة للفئة القاعدية دوما.

استخدام MyClass

تستعمل لاستدعاء أعضاء الفئة الحالية التي تمت إعادة تعريفها

```
Class BaseClass
    Sub Test ()
```

الفصل العاشر الوراثة

```
' قیللخا اذھ متیسء اعدتسا ةقیرط ةئفلا  
MyClass.OverriddenMethod()  
End Sub  
...  
...  
End Class
```


إنشاء أداة جديدة

قم بإنشاء مشروع جديد من نوع Windows Control Library و افتح محرر الكود التابعة للأداة UserControl1 و امسح كل شيء و اكتب هذه الفئة التي تشتق أداة Text Box

VB.Net Code

```
Public Class TextBoxEx
    Inherits TextBox ' نرث خصائص أداة النص

    Sub New()
        MyBase.New()
    End Sub

    ننشئ خاصية تقوم بتحديد كل محتوى صندوق النص
    Private m_autoSelect As Boolean
    Public Property AutoSelect() As Boolean
        Get
            Return m_autoSelect
        End Get
        Set(ByVal Value As Boolean)
            m_autoSelect = Value
        End Set
    End Property

    تنفيذ الخاصية بمجرد انتقال التركيز إلى الأداة
    Private Sub TextBoxEx_Enter(ByVal sender As Object, _
        ByVal e As System.EventArgs) Handles MyBase.Enter

        تحديد الكل ' Me.SelectAll()
        تغيير اللون ' Me.BackColor = c
    End Sub

    ننشئ خاصية اللون

    Private c As Color
    Public Property colr() As Color
        Get
            Return c
        End Get
        Set(ByVal value As Color)
            c = value
        End Set
    End Property
End Class
```

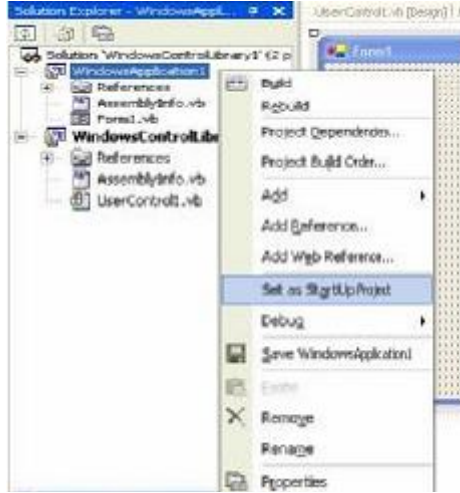
من قائمة Build قم بنشر المشروع و تحويله إلى ملف DLL لتتمكن من استخدامه في مشاريع أخرى
برمجة الأداة :

إنشاء أداة جديدة خاصة

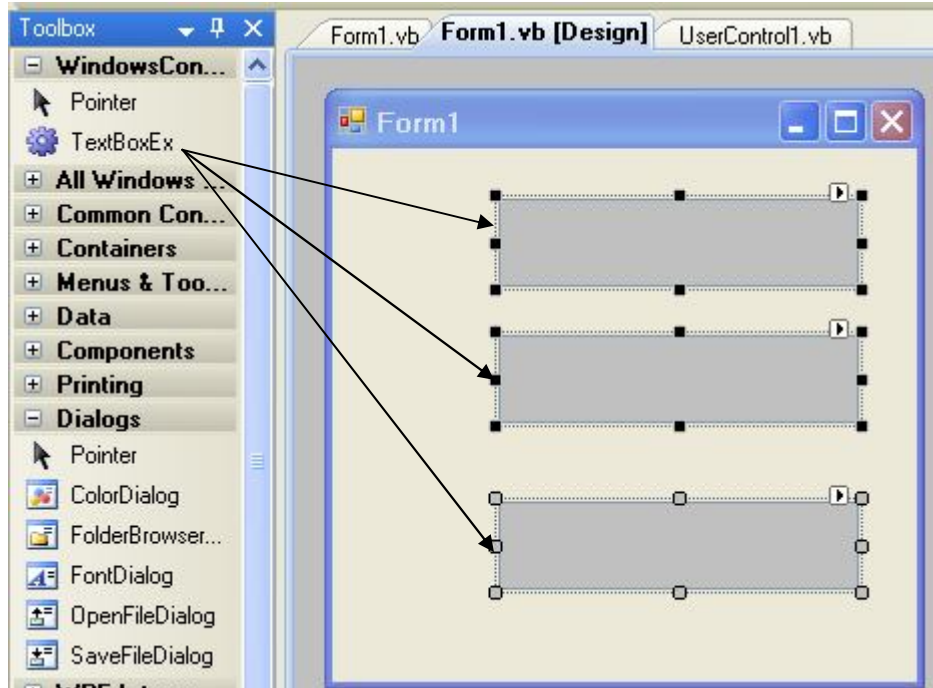
قم بإضافة مشروع جديد من نوع Windows Application في نفس المشروع السابق ثم اجعله المشروع الابتدائي

Startup Project و ذلك بنقر يمين على اسمه في نافذة مستكشف الحل Solution Explorer و اختر الأمر

Set as Startup Project

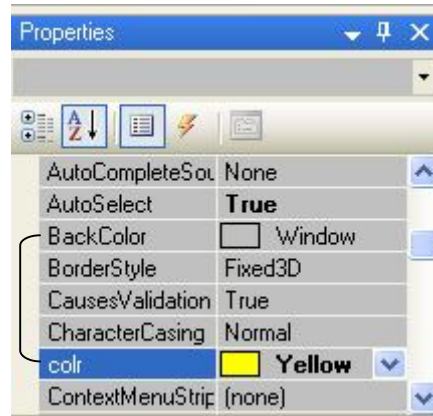


ستجد الأداة التي أنشأتها ضمن الأدوات في صندوق الأدوات Toolbox قم بإضافة 3 نسخ منها على النموذج كما يلي :



ثم قم بتحديد الأدوات ثم غير خاصية AutoSelect إلى True و خاصية Color إلى اللون المناسب ستجد الخاصيتين مضافتين إلى نافذة الخصائص

إنشاء أداة جديدة خاصة



ثم قم بتنفيذ البرنامج و لاحظ التغيير و الناتج

معالجة الأخطاء

يمكن تقسيم أنواع الأخطاء إلى :

ErrorType	نوع الخطأ
Design Time errors	أخطاء وقت التصميم
Run_Time Errors	أخطاء وقت التشغيل
Logic Error	أخطاء منطقية

أخطاء وقت التصميم :

وهي الأخطاء التي تحدث في بنية الجمل البرمجية مثل الأخطاء الإملائية و في استدعاء إجراء أو متغير لم يتم تعريفه أو كتابة صيغة بطريقة غير صحيحة يقوم محرر الكود بتلويين الخطأ بخط أزرق متعرج عند مكان حدوث الخطأ و بمجرد تحريك الفأرة فوق الخطأ يظهر تلميح يوضح لك سبب الخطأ.

```
Dim x As String = 256
```

Option Strict On disallows implicit conversions from 'Integer' to 'String'.

عند تنفيذ البرنامج تظهر أمامك رسالة خطأ تخبرك ما إذا كنت تريد تنفيذ البرنامج قبل حدوث الخطأ أم لا
طبعا تختار لا

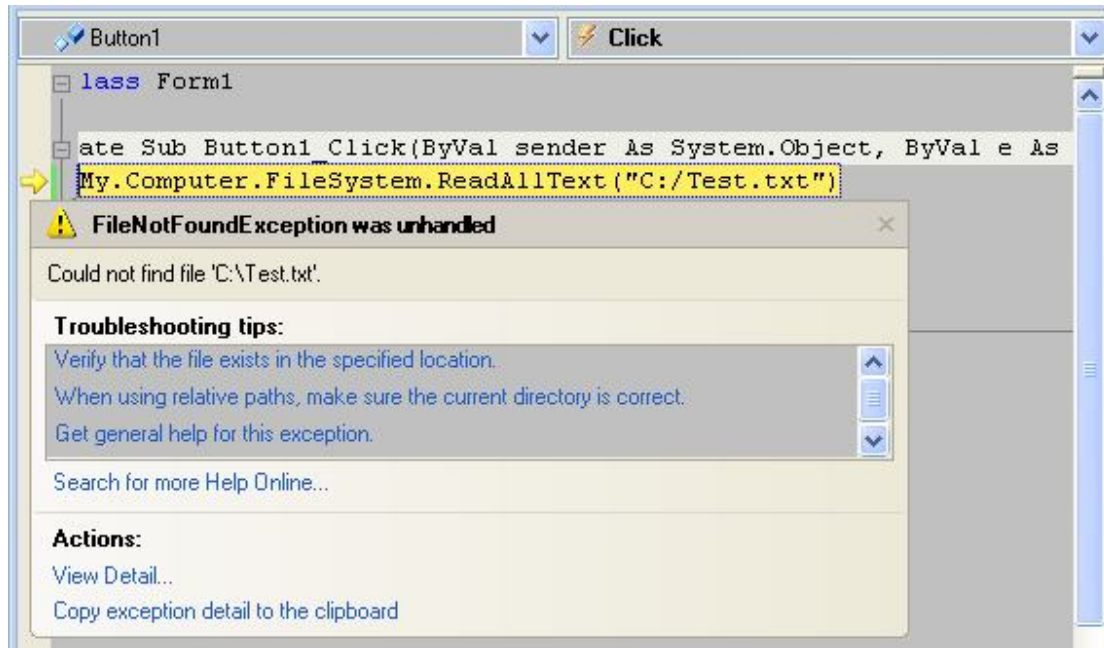


تظهر أمامك قائمة الأخطاء Error List بها كل الأخطاء . انقر مزدوجاً على الخطأ الذي تريد تنقيحه تنتقل مباشرة إليه

Error List					
1 Error 0 Warnings 0 Messages					
	Description	File	Line	Column	Project
1	Name 'loadtxt' is not declared.	Form1.vb	4	9	WindowsApplication1

أخطاء وقت تشغيل البرنامج Run_Time Errors

وهي الأخطاء التي تقع عند تنفيذ و تشغيل البرنامج تسمى بالاستثناءات Exceptions و تحدث عند الاتصال بمصادر أخرى كمسار الملفات و المجلدات و الصور
مثال : نقوم باستدعاء و قراءة محتوى ملف الجمل البرمجية صحيحة



و لكن عند تنفيذ البرنامج نحصل على استثناء كما هو واضح في الصورة يخبرنا عن عدم العثور

على الملف Could not Find File

لمعالجة الخطأ أو الاستثناء تحت العنوان Troubleshooting tips يقترح عليك البرنامج اختيار

احد الروابط لمزيد من التفصيل ومناقشة الحلول المقترحة.

اختر الأمر الأول تظهر صفحة التعليمات توضح لك سبب الاستثناء

Visual Studio Debugger

Troubleshooting Exceptions: System.IO.FileNotFoundException

[See Also](#)

Collapse All Language Filter: Visual Basic

A [FileNotFoundException](#) exception is thrown when an attempt is made to access a file or directory on disk.

Associated Tips

Verify that the file exists in the specified location.

If the file does not exist, this exception is thrown. For more information, see [FileExists](#).

When using relative paths, make sure the current directory is correct.

If you are assuming the current directory incorrectly, relative paths will be incorrect as well.

See Also

Tasks

[How to: Find Out More About an Exception with the Exception Assistant](#)

[How to: View a File](#)

Reference

[FileNotFoundException](#)

يقترح عليك لحل مثل هذه الأخطاء التحقق أولاً من وجود الملف باستعمال الوسيلة FileExists كما يلي :

```
If My.Computer.FileSystem.FileExists(_  
    "c:/test.txt") Then  
    My.Computer.FileSystem.ReadAllText("C:/Test.txt")  
Else  
    MsgBox("موجود غير الملف")  
End If
```

الأخطاء المنطقية Logic

وهي أخطاء يكون سببها المبرمج فيحصل على نتيجة غير متوقعة أو خاطئة بسبب ما

تحديد الأخطاء المنطقية Logic Error

 VB.Net Code

```
Dim Age As Single  
If Age > 13 And Age < 20 Then  
    TextBox2.Text = "المراهقة سن في أنت"  
Else  
    TextBox2.Text = "المراهقة سن في لست"  
End If
```

مراجعة الأخطاء

معالجة الأخطاء

من ضمن الطرق المخصصة لمعالجة الأخطاء المنطقية تنفيذ الكود سطرا سطرا فبذلك يمكن تحديد السطر الذي حدث فيه الخطأ
لنأخذ هذا المثال :

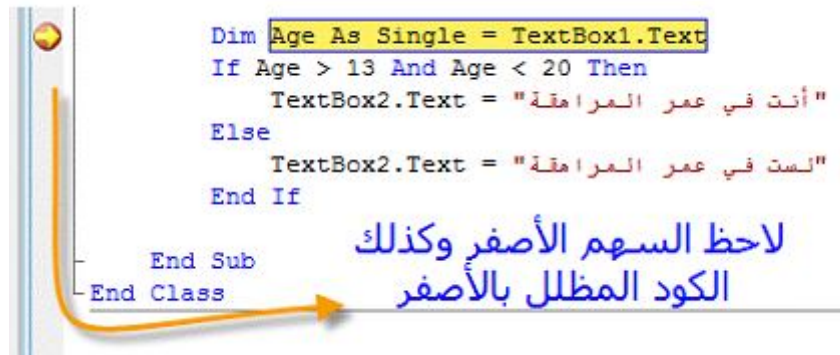
VB.Net Code

```
Dim Age As Single = TextBox1.Text
If Age > 13 And Age < 20 Then
    TextBox2.Text = "المراهقة عمر في أنت"
Else
    TextBox2.Text = "المراهقة عمر في لست"
End If
```

ماذا لو أدخلنا العمر 13 فستظهر النتيجة لست في عمر المراهقة رغم أنه في سن المراهقة
للتتبع الأخطاء نضيف نقطة توقف أو ما يسمى Break Point حسب الشكل التالي



عند تنفيذ البرنامج و كتابة العمر 16 في مربع النص Text Box1 و الضغط على الزر يقودنا
البرنامج إلى محرر الكود حسب الشكل :



وعند تمرير مؤشر الفأرة على TextBox1.text ستظهر القيمة المدخلة "16"
إما عند تمريره على المتغير Age يظهر (Age |0.0)
وفي السطر الثاني تصبح قيمة المتغير (Age |16.0)
ولمتابعة تتبع القيم المدخلة و المتغيرات ننتقل إلى الأدوات الخاصة بتتبع الأخطاء خطوة خطوة كما
في الشكل

معالجة الأخطاء



وللمتابعة نستخدم الزر Step Into فينتقل المؤشر الأصفر إلى السطر الموالي و في هذا المثال عند السطر الثالث ينتقل بعد الضغط Step Into إلى End If يقفز السطرين لان الشرط تحقق في السطر الثالث و للمزيد عن معالجة الأخطاء و معرفة قيم المتغيرات و المكونات المتغيرة و العديد من المعلومات فمن قائمة Debug نختار Windows ثم Autos تظهر النافذة التالية

Name	Value	Type
Age	16.0	Single
TextBox1	{Text = "16"}	System.Windows.Forms.TextBox
TextBox1.Text	"16"	String
TextBox2	{Text = ""}	System.Windows.Forms.TextBox
TextBox2.Text	""	String
e	{X = 54 Y = 16 Button = Left (1048576)}	System.EventArgs
sender	{Text = "قارن العمر"}	Object

لمعالجة الخطأ نقوم بتصحيح الجملة الشرطية

If Age >= 13 And Age < 20 Then

بالإضافة لوجود نافذة أخرى في قائمة Debug ثم نختار Windows ثم Watch واحدة من 4

تظهر النافذة التالية

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Age As Single = TextBox1.Text
        If Age >= 13 And Age < 20 Then
            TextBox2.Text = "أنت في عمر المراهقة"
        Else
            TextBox2.Text = "لست في عمر المراهقة"
        End If
    End Sub
End Class
```

Name	Value	Type
Age	16.0	Single

فيمكنك كتابة أي متغير أو كائن متغير القيمة في خانة Name سنظهر قيمته مباشرة فمثلا : قيمة Age = 16.0 عند السطر الثاني

النافذة السريعة Immediate Window

معالجة الأخطاء

و تستخدم لإدخال قيم تجريبه للمتغيرات أو للكائن في النافذة Immediate و متابعة النتيجة و التأثيرات الناجمة جراء هذا التغيير في نافذة Watch بعد إظهارها حسب الشكل التالي



كل ما تعلمناه يتعلق بنقطة التوقف و لحذفها نرجع إلى محرر الكود ونضغط مكان النقطة

صيد الأخطاء عن طريق Try ... Catch Statement

كثيرا ما تحدث أخطاء تسبب مشاكل للبرنامج قد تؤدي إلى وقفه أحيانا و غالبا ما تكون خارج البرنامج كاستدعاء ملف أو مسار صورة غير موجودة و لتحاكي مثل هذه الأخطاء نستعمل جمل صيد الأخطاء

كما يوضح هذا المثال

VB.Net Code

```
Try  
    الجمل التي يحتمل أن تحتوي على أخطاء  
Catch  
    الجملة التي نريد تنفيذها عند حدوث الخطأ  
Finally (اختيارية)  
    جملة أخيرة سيتم تنفيذها إجباريا سواء حدث خطأ أو لم يحدث  
End Try
```

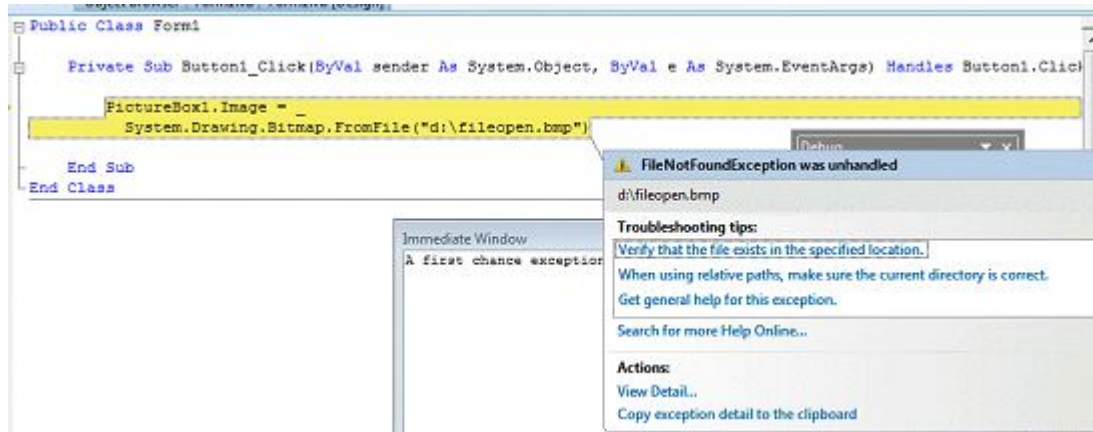
مثال تطبيقي :

نفترض أن مسار الصورة الحقيقي التي أردنا استظهارها : d:\fileopen.bmp
ولكن عند استدعاء الصورة أخطأت في كتابة المسار كما يلي :

```
PictureBox1.Image =  
System.Drawing.Bitmap.FromFile("C:\fileopen.bmp")
```

ستظهر رسالة خطأ كما يلي

معالجة الأخطاء



و لمعالجة الخطأ نستعمل جملة صيد الأخطاء Try

VB.Net Code

```
Try
    PictureBox1.Image _ =
        System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")
Catch
    MsgBox ("تحقق من مسار الصورة")
End Try
```

استخدام Finally لتحسين البرنامج

VB.Net Code

```
Try
    PictureBox1.Image _ =
        System.Drawing.Bitmap.FromFile("d:\fileopen.bmp ("
Cat    MsgBox ("تحقق من مسار الصورة")
Finally
    MsgBox ("تمت العملية")
End Try
```

نأخذ المثال السابق و نبين كيفية التعامل مع رقم الخطأ و عرض تقرير عن الخطأ

VB.Net Code

```
Try
    PictureBox1.Image _ =
        System.Drawing.Bitmap.FromFile("d:\fileopen.bmp ")
Catch When Err.Number = 53
    MsgBox ("تأكد من مسار واسم الملف")
Catch When Err.Number = 7
```

```
MsgBox ( "لا يمكن عرض الصورة " , Err.Description )  
Catch  
MsgBox ( "حدث خطأ عند تحميل الملف " , Err.Description)  
End Try
```

تحديد عدد مرات المحاولة بجملة Try

VB.Net Code

```
Try  
PictureBox1.Image = _  
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")  
Catch  
Retries += 1  
If Retries <= 2 Then  
MsgBox("الملف اسم و مسار من تأكد")  
Else  
MsgBox("أخرى مرة المحاولة يمكن لا")  
Button1.Enabled = False  
End If  
End Try
```

جمل Try...Catch المتداخلة

VB.Net Code

```
Try  
PictureBox1.Image = _  
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")  
Catch  
MsgBox("الملف اسم و مسار من تأكد")  
Try  
PictureBox1.Image = _  
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")  
Catch  
MsgBox("أخرى مرة المحاولة يمكن لا")  
Button1.Enabled = False  
End Try  
End Try
```

استخدام الخطوط الدفاعية .

بدل من صيد الأخطاء عن طريق جملة Try وعرض تقرير الخطأ - الملف غير موجود - نستعمل الطريقة File.Exists

تحت مجال الأسماء System.IO لمعرفة ما إذا كان الملف موجود أم لا كما يلي :

VB.Net Code

```
If File.Exists("d:\fileopen.bmp") Then
PictureBox1.Image = _
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")
Else
MsgBox("المسار من تحقق موجود غير الملف")
End If
```

متى نستخدم جمل صيد الأخطاء Try و الخطوط الدفاعية

عند الأخطاء المتوقعة المعلومة نستعمل خطوط الدفاع أما الأخطاء المتوقعة الغير معروفة نستعمل

جمل صيد الأخطاء

استخدام Exit...Try

VB.Net Code

```
Try
If PictureBox1.Enabled = False Then Exit Try
PictureBox1.Image = _
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")
Catch Retries += 1
If Retries <= 2 Then
MsgBox("الملف اسم و مسار من تأكد")
Else
MsgBox("الصورة عرض يمكنك لا")
Button1.Enabled = False
End If
End Try
```

في هذا المثال إذا كانت خاصية Enabled لصندوق الصورة مفعلة = False يعني أن صندوق

الصورة لا يمكنه عرض و استقبال الصورة وسيتم الخروج Exit Try

الجملة On Error ... Goto

في حالة حدوث خطأ اذهب إلى المكان المحدد انظر المثال

VB.Net Code

```
On Error GoTo A
PictureBox1.Image = _
System.Drawing.Bitmap.FromFile("d:\fileopen.bmp")
A:
MsgBox("موجود غير الملف")
```

الجملة On Error Resume Next

تستعمل لتجاهل الأخطاء فعند حدوث خطأ يتجاهله و ينتقل إلى السطر الموالي فلا يعيق البرنامج

التعامل مع الرسومات Graphics

لكي ترسم أي شيء يجب أن يتوافر لديك ما يلي:

سطح الرسم Surface:

يجب عليك قبل أن تبدأ الرسم، أن تختار السطح الذي سترسم عليه.. هذا السطح يمثله كائن الرسوم Graphics object، الذي يمدك بالوسائل التي ترسم الأشكال الأساسية وغير الأساسية. وللرسم على نموذج أو أداة، يجب أن نتعامل مع كائن الرسوم الخاص بها، والذي تحصل عليه من الوسيلة "إنشاء رسوم" CreateGraphics.. كما يمكن الرسم على كائن صورة نقطية Bitmap Object، ووضع الصورة الناتجة بعد ذلك على النموذج أو الأداة، كما سنرى فيما بعد.

VB.Net Code

```
Imports System.Drawing
Public Class Form1

    Private Sub Form1_Paint(ByVal sender As Object, _
        ByVal e As PaintEventArgs) Handles MyBase.Paint
        Dim gr As Graphics = e.Graphics
        ...
        ...
    End Sub
End Class
```

أداة الرسم:

لدينا أداتان رئيسيتان نرسم بهما:

1. القلم Pen:

ويمكنك استخدامه في رسم الأشكال المكوّنة من خطوط، مثل الخطوط lines، والمستطيلات rectangles والمنحنيات curves.. وأهم خصائص القلم: لونه وسمك خطّه.

2. الفرشاة Brush:

ويمكنك استخدامها في رسم المساحات الملونة.. وأهم خصائص الفرشاة: لونها والشكل الذي ستلوّنه وطريقة التلوين.

VB.Net Code

```
' تجهيز قلم أحمر سمك خطّه نقطتان '
Dim redPen As Pen = New Pen(Color.Red, 2)
' نقطة بداية الخط '
Dim point1 As Point = New Point(10,10)
' نقطة نهاية الخط '
Dim point2 As Point = New Point(120,180)
' رسم الخط على النموذج '
Me.CreateGraphics.DrawLine(redPen, point1, point2)
```

ويمكن دمج كل الخطوات الأربع السابقة في الخطوة الوحيدة التالية:

VB.Net Code

```
Me.CreateGraphics.DrawLine(New Pen(Color.Red, 2), _
    New Point(10,10), New Point(120,180))
```

وهناك صيغة أسهل للوسيلة DrawLine كالتالي

VB.Net Code

```
Me.CreateGraphics.DrawLine(New Pen(Color.Red, 2), 10, 10, 120, 180)
```

حيث أرسلنا إحداثيات الخطّ منفردة، بدلا من إرسالها كنقطتي بداية ونهاية، وإن كانت الصيغة الأولى أوضح عند قراءتها من الصيغة الثانية.

كما يمكنك تعريف كائن رسوم ثمّ ربطه بكائن رسوم النموذج كالتالي:

VB.Net Code

```
Dim G As Graphics = Me.CreateGraphics
G.DrawLine(redPen, point1, point2)
```

كائن النقطة Point Object:

يمثل هذه الكائن إحداثي نقطة: الإحداثي السيني X (موضع النقطة أفقيًا، بدءًا من أقصى يسار الشاشة)، والإحداثي الصادي Y (موضع النقطة رأسيًا، بدءًا من أعلى الشاشة). ولإنشاء نقطة جديدة، يجب أن تحدد إحداثيها (الأفقي ثمّ الرأسّي) كالتالي:

VB.Net Code

```
Dim P1 As New Point
P1.X = 34
P1.Y = 50
```

أو باختصار:

```
Dim P1 As Point
P1 = New Point(34, 50)
```

أو بشكل أكثر اختصارًا:

```
Dim P1 As New Point(34 50)
```

كائن الحجم Size Object:

مشابه لكائن النقطة، ولكنه يمتلك خاصيتين: العرض Width، والارتفاع Height، ويتمّ تعريفه كالتالي

VB.Net Code

```
Dim S As Size
S.Width = 100
S.Height = 20
```

أو بطريقة أخرى:

```
Dim S As New Size(100,20)
```

وبطريقة ثالثة:

```
Dim S As New Size(New Point(100,20))
```

كائن المستطيل Rectangle Object:

يتطلب تعريف المستطيل، موقع رأسه العلويّ الأيمن وعرضه وارتفاعه:

Dim box As New Rectangle(X, Y, العرض, الارتفاع)

وهناك صيغة أخرى باستخدام كائن النقطة وكائن الحجم:

 VB.Net Code

```
Dim P As New Point(1,1)
Dim S As New Size(100,20)
Dim box As New Rectangle(P, S)
```

كما يمكنك أن تنشئ مستطيلاً، ثم تضع قيم خصائصه كالتالي:

```
Dim box As New Rectangle()
box.X = 1
box.Y = 1
box.Width = 100
box.Height = 20
```

كائن القلم Pen Object:

ذكرنا من قبل أن تعريف هذا الكائن يتطلب اللون والسمك (ولو حذفنا السمك من التعريف فسيتم اعتباره 1):

 VB.Net Code

```
Dim P As Pen
P = New Pen(Color.Black, 3)
```

وهناك صيغة أخرى تسمح لك بتعريف اللون من كائن فرشاة كالتالي:

```
Dim patternPen As New Pen(brush, width)
```

وفي هذه الحالة يمكنك استخدام هذا القلم لتلوين الحد الخارجي لشكل بطريقة معينة (تحدها الفرشاة). ولديك مجموعة من الأقلام الجاهزة محددة الألوان وسمكها 1، يمنحها لك مرقم الأقلام Pens Structure.. مثل القلم الأزرق Pens.Blue.. ويمكنك أن تستخدم هذه الأقلام مباشرة بدون تعريف، كمعامل في أي وسيلة تتطلب قلماً.

كائن الفرشاة Brush Object:

تمكّنك الفرشاة Brush من ملء الأشكال بلون واحد أو بألوان متدرجة أو بصورة. ولا يمكنك تعريف فرشاة منفردة، فهذه الخليّة ترثها أنواع الفرش المختلفة، وهي:

الفرشاة الصلبة SolidBrush:

تملأ الشكل بلون واحد

```
Dim sBrush As New SolidBrush(لون الفرشاة)
```

فرشاة التظليل HatchBrush:

تملأ الشكل بتشكيلة معينة، من أكثر من 50 تشكيلة جاهزة، يمكنك اختيارها من خلال المرقم HatchStyle.. ويتم تعريف هذه الفرشاة كما يلي:

```
Dim hBrush As New HatchBrush (لون الخلفية، لون خطوط التظليل، طراز الفرشاة)
```

وهذه بعض تشكيلات التظليل، التي يمكنك استخدامها كقيم للمعامل الأول:

خطوط قطريّة من أعلى اليسار إلى أسفل اليمين.	ForwardDiagonal
خطوط قطريّة من أعلى اليمين إلى أسفل اليسار.	BackwardDiagonal
خطوط أفقيّة ورأسيّة متقاطعة.	Cross

خطوط قطريّة متقاطعة.	DiagonalCross
خطوط أفقيّة.	Horizontal
خطوط رأسيّة.	Vertical

كائن الرسوم Graphics Object:

أدوات كثيرة لا تتوقّعها تمنحك كائن رسوم، ممّا يمكّنك من الرسم عليها.. خذ مربع النصّ مثلاً! ولكي تحصل على كائن الرسوم من الأداة، استخدم وسيلتها CreateGraphics. ولكي تستخدم كائن الرسوم، استورد المكتبة Drawing2D في مشروعك أولاً:

```
Imports System.Drawing.Drawing2D
```

بعد ذلك يمكنك تعريف متغيّر من هذا الكائن كالتالي:

```
Dim G As Graphics
```

ولكي ترسم بهذا الكائن على مربع الصورة، استخدم الجملة التالية:

```
G = PictureBox1.CreateGraphics
```

رسم الأشكال بوسائل كائن الرسوم:

يمتلك كائن الرسوم العديد من الوسائل التي تمكّنك من رسم الأشكال.. ولكي نشرح هذه الوسائل، تعالَ نعرّف كائن رسوم يرسم على النموذج بقلم أزرق:

VB.Net Code

```
Dim G As Graphics = Me.CreateGraphics
Dim P As New Pen(Color.Blue)
```

إنّ احتياجنا للقلم يرجع إلى أنّ أول معامل في وسائل رسم الأشكال هو القلم.. وطبعاً باقي المعاملات هي إحدائيات الشكل.

دعنا نبدأ برسم مستطيل

VB.Net Code

```
G.DrawRectangle(P, 10, 10, 200, 150)
```

والآن سنرسم قطري المستطيل:

```
G.DrawLine(P, 10, 10, 210, 160)
G.DrawLine(P, 210, 10, 10, 160)
```

والجدول التالي يوضّح لك الوسائل التي ترسم الأشكال المختلفة (تلك التي تبدأ بالمقطع Draw)، والوسائل التي تملأ كل شكل منها بلون معيّن – إذا كان الشكل مغلقاً بالطبع – (وهي تبدأ بالمقطع Fill)، مع ملاحظة أنّ استخدام وسيلة الملء يملأ الشكل إذا كان موجوداً على النموذج، أو ترسمه ممثلثاً باللون إذا لم يكن موجوداً من قبل.

ترسم قوساً.	DrawArc
ترسم منحنى بيزير، وهو منحنى له نقطتا بداية ونهاية، ويمرّ بنقطتين أخريين تتحكّمان في انحنائه	DrawBezier
ترسم مجموعة من منحنيات بيزير معاً.	DrawBeziers
ترسم منحنى مغلق.	DrawClosedCurve FillClosedCurve
ترسم منحنى يمرّ بنقاط معيّنّة (على الأقلّ 4 نقاط، بداية ونهاية	DrawCurve

<p>ونقطتان على المنحنى): Graphics.DrawCurve(القلم, _ (درجة الانحناء, مصفوفة من النقاط التي يمرّ بها المنحنى ولو لم ترسل المعامل الأخير، فسيتمّ رسم المنحنى بدرجة انحناء 1. والشكل التالي يبيّن مجموعة من المنحنيات التي تمرّ بنفس النقاط، ولكن لكلّ منها درجة انحناء مختلفة:</p>	
<p>ترسم قطعاً ناقصاً، بمعلومية المستطيل الذي يحتويه.. والقطع الناقص هو شكل بيضاويّ له قطران غير متساويين (مساويان لضلعي المستطيل الذي يحتويه). ونظراً لأنّه لا توجد وسيلة مستقلة لرسم الدائرة، فيجب عليك أن تستخدم هذه الوسيلة لرسم الدوائر، وذلك برسم القطع الناقص داخل مربع، وفي هذه الحالة سيكون طول ضلع المربع مساوياً لقطر الدائرة.. مثال: الجملة التالية ترسم دائرة زرقاء نصف قطرها 50: Graphics.DrawEllipse(Pens.Blue, _ New Rectangle(50,20,100,100)) ويقع مركز هذه الدائرة في مركز المربع الذي هو (50 + [100] ÷ 2 ، 20 + [100] ÷ 2)، أي النقطة (100 ، 70). وكقاعدة: يقع مركز الدائرة عند النقطة (س + نصف طول ضلع المربع ، ص + نصف طول ضلع المربع).</p>	<p>DrawEllipse FillEllipse</p>
<p>تحمّل أيقونة داخل كائن الرسوم.. هذه الأيقونة سيتمّ مطّها لتملأ مساحة كائن الرسوم.</p>	<p>DrawIcon</p>
<p>تحمّل أيقونة داخل كائن الرسوم، ولكن دون مطّها.</p>	<p>DrawIconUnstretched</p>
<p>تحمّل صورة داخل كائن الرسم، مع مطّها لتملأ مساحة كائن الرسوم.</p>	<p>DrawImage</p>
<p>تحمّل صورة داخل كائن الرسم، ولكن دون تغيير حجمها.</p>	<p>DrawImageUnscaled</p>
<p>ترسم خطاً يصل بين نقطتين.</p>	<p>DrawLine</p>
<p>ترسم مجموعة من الخطوط.</p>	<p>DrawLines</p>
<p>ترسم جميع الأشكال الموجودة في كائن مسار الرسوم GraphicsPath object.. وستنعرّف عليه لاحقاً.</p>	<p>DrawPath FillPath</p>
<p>ترسم شريحة من الدائرة (قوس من قطع ناقص (أو دائرة) وضلعين يصلان نهايتي القوس بمركز الدائرة).. ولرسم هذا القوس أرسل لهذه الوسيلة المستطيل الذي يحتوي على القطع الناقص، مع زاوية بداية القوس، وزاوية انفراج ضلعي القوس.. مثال: Graphics.DrawEllipse(Pens.Blue, _ New Rectangle(50,20,100,300), 50, 30)</p>	<p>DrawPie FillPie</p>

الرسومات

وأهم استخدام لهذا الشكل هو الرسوم البيانية التي توضح النسب كشرائح ملونة من دائرة (مثل الرسم الذي يوضح لك المساحة المشغولة والمساحة الفارغة من القرص الصلب في الويندوز).	
ترسم مضلعًا مغلقًا له أي عدد من الرؤوس.. هذه الرؤوس ترسلها لهذه الوسيلة في صورة مصفوفة من النقاط.	DrawPolygon FillPolygon
ترسم مستطيلًا.	DrawRectangle FillRectangle
ترسم مجموعة من المستطيلات.	DrawRectangles FillRectangles
ترسم نصًا بخط معين.	DrawString
لملء كائن من النوع "منطقة" Region بلون معين.	FillRegion

ولن تجد صعوبة في استخدام هذه الوسائل.. فقط لاحظ ما يلي:

- أول معامل في دوال رسم الأشكال هو كائن القلم Pen Object، وذلك لتحديد لون الخطّ وسمكه.
- أول معامل في دوال ملء وتلوين الأشكال المغلقة هو كائن الفرشاة Prush Object، وذلك لتحديد لون التلوين وطرازه.
- باقي معاملات وسائل الرسم والتلوين توضح موضع الشكل ومقاييسه.
- الوسائل التي ترسم أكثر من شكل في نفس الوقت، تتطلب - كمعامل - مصفوفة تعرف مواضع ومقاييس هذه الأشكال.. فمثلا DrawRectangles تقبل مصفوفة تحتوي على كائنات من النوع "مستطيل" Rectangle objects.

تطبيقات :

```
VB.Net Code رسم خط مستقيم
Private Sub Form1_Paint(ByVal sender As Object, _
    ByVal e As PaintEventArgs) Handles MyBase.Paint
    Dim gr As Graphics = e.Graphics
    gr.DrawLine(Pens.Black, 0, 0, 200, 200)
End Sub
```

```
VB.Net Code رسم مستطيل
Private Sub Form1_Paint(ByVal sender As Object, _
    ByVal e As PaintEventArgs) Handles MyBase.Paint
    Dim gr As Graphics = e.Graphics
    gr.DrawRectangle(Pens.Red, New Rectangle(10, 10, 200, 200))
End Sub
```

```
VB.Net Code رسم دائرة
Private Sub Form1_Paint(ByVal sender As Object, _
    ByVal e As PaintEventArgs) Handles MyBase.Paint
    Dim gr As Graphics = e.Graphics
```

```
gr.DrawEllipse(Pens.Blue, 0, 0, 200, 200)
End Sub
```

 VB.Net Code رسم دائرة مظلة

```
Dim circle_brush As New SolidBrush(Color.Orange)
e.Graphics.FillEllipse(circle_brush, 10, 10, 100, 100)
Dim circle_pen As New Pen(Color.Blue, 10)
e.Graphics.DrawEllipse(circle_pen, 10, 10, 100, 100)
```

 VB.Net Code رسم صورة

```
Using G As Graphics = Me.CreateGraphics
Dim bitmap As New Bitmap("E:\lomalove3.JPG")
G.DrawImage(bitmap, 60, 10)
End Using
```

 VB.Net Code رسم صورة

```
Dim gr As Graphics = e.Graphics
Dim points() As Point = {New Point(10, 10), _
New Point(200, 200), _
New Point(10, 100), _
New Point(10, 10)}
gr.DrawLine(Pens.Green, points)
```

 VB.Net Code رسم خطوط منقطة

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As _
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
Dim myPen As New Pen(Color.Black, 4)
myPen.DashStyle = Drawing2D.DashStyle.DashDot
gr.DrawLine(myPen, 0, 0, 200, 200)
myPen.DashStyle = Drawing2D.DashStyle.Dash
gr.DrawLine(myPen, 0, 20, 200, 220)
myPen.Dispose()
End Sub
```

 VB.Net Code رسم خطوط منقطة

```
Dim gr As Graphics = e.Graphics
Dim myPen As New Pen(Color.Black, 7)
Dim customDash() As Single = {2, 1, 5, 1}
myPen.DashPattern = customDash
gr.DrawLine(myPen, 10, 10, 300, 10)
```

 VB.Net Code رسم خطوط ذات أسهم

```
Dim gr As Graphics = e.Graphics
Dim myPen As New Pen(Color.Black, 7)
myPen.StartCap = Drawing2D.LineCap.ArrowAnchor
myPen.EndCap = Drawing2D.LineCap.ArrowAnchor
gr.DrawLine(myPen, 10, 40, 300, 40)
```

 VB.Net Code رسم مستطيلات حدودها مظلة

```
Dim gr As Graphics = e.Graphics
Dim myPen As New Pen(Color.LightGray, 8)
myPen.Alignment = Drawing2D.PenAlignment.Center
gr.DrawRectangle(myPen, New Rectangle(10, 10, 100, 100))
gr.DrawRectangle(Pens.Black, New Rectangle(10, 10, 100, 100))
myPen.Alignment = Drawing2D.PenAlignment.Inset
gr.DrawRectangle(myPen, New Rectangle(150, 10, 100, 100))
gr.DrawRectangle(Pens.Black, New Rectangle(150, 10, 100, 100))
myPen.Dispose()
```

 VB.Net Code

تظليل دائرة

```
Dim gr As Graphics = e.Graphics
Dim myBrush As New SolidBrush(Color.Red)
gr.FillEllipse(myBrush, 0, 0, 200, 200)
myBrush.Dispose()
```

 VB.Net Code

نموذج رائق

```
Imports System.Drawing.Drawing2D
Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As _
System.Windows.Forms.PaintEventArgs) Handles Me.Paint

Dim gr As Graphics = e.Graphics
Dim myBrush As New LinearGradientBrush(New Rectangle(0, _
0, 200, 200), Color.Black, Color.Blue, _
LinearGradientMode.BackwardDiagonal)
gr.FillRectangle(myBrush, 0, 0, 300, 300)
myBrush.Dispose()
End Sub
End Class
```

 VB.Net Code

مربع مخطط

```
Imports System.Drawing.Drawing2D
Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e _
As System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
Dim myBrush As New HatchBrush(HatchStyle.BackwardDiagonal _
, Color.Green, Color.White)
gr.FillRectangle(myBrush, 0, 0, 100, 100)
myBrush.Dispose()
End Sub
End Class
```

 VB.Net Code

مربعات متداخلة

```
Imports System.Drawing.Drawing2D
Public Class Form1
Private Sub Form1_Paint(ByVal sender As Object, ByVal e _
As System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
gr.TranslateTransform(Me.ClientRectangle.Width \ 2, _
```

```

Me.ClientRectangle.Height \ 2)
gr.DrawRectangle(Pens.Black, -50, -50, 100, 100)
gr.RotateTransform(-45)
gr.DrawRectangle(Pens.Black, -50, -50, 100, 100)
End Sub
End Class

```

VB.Net Code

رسم نص

```

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As _
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
Dim myFont As New Font("Tahoma", 14)
Dim sf As New StringFormat()
Dim text As String = "أكبر الله"
sf.Alignment = StringAlignment.Far
gr.DrawString(text, myFont, Brushes.Black, _
New RectangleF(10, 10, 200, 200), sf)
gr.DrawRectangle(Pens.Black, New Rectangle(10, 10, _
200, 200))
End Sub

```

VB.Net Code

حروف هندية

```

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As _
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
Dim sf As New StringFormat()
Dim myFont As New Font("Tahoma", 13)
Dim myText As String = "1234567890"
gr.DrawString(myText, myFont, Brushes.Black, _
New RectangleF(10, 10, 280, 200), sf)
sf.SetDigitSubstitution(&H401, StringDigitSubstitute.Traditional)
gr.DrawString(myText, myFont, Brushes.Black, _
New RectangleF(10, 40, 280, 200), sf)
End Sub

```

VB.Net Code

نص 3D

```

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As _
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
Dim gr As Graphics = e.Graphics
Dim sf As New StringFormat()
Dim myFont As New Font("Tahoma", 13)
Dim myText As String = "1234567890"
gr.DrawString(myText, myFont, Brushes.Black, _
New RectangleF(10, 10, 280, 200), sf)
sf.SetDigitSubstitution(&H401, StringDigitSubstitute.Traditional)
gr.DrawString(myText, myFont, Brushes.Black, _
New RectangleF(10, 40, 280, 200), sf)
End Sub

```

VB.Net Code

معينات

```
Imports System.Drawing.Drawing2D
Public Class Form1
Private Sub DrawDiamond(ByVal gr As Graphics)
    Dim pts() As Point = {
        New Point(75, 125), _
        New Point(125, 75), _
        New Point(175, 125), _
        New Point(125, 175) _
    }
    gr.DrawPolygon(Pens.Black, pts)
End Sub

Private Sub Form1_Paint(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.PaintEventArgs) _
    Handles Me.Paint
    ' Draw the original diamond.
    DrawDiamond(e.Graphics)
    ' Translate to center at the origin.
    e.Graphics.TranslateTransform(-125, -125, _
        MatrixOrder.Append)
    ' Scale by a factor of 2.
    e.Graphics.ScaleTransform(2, 2, MatrixOrder.Append)
    ' Translate the center back to where it was.
    e.Graphics.TranslateTransform(125, 125, _
        MatrixOrder.Append)
    DrawDiamond(e.Graphics)
End Sub
End Class
```

VB.Net Code

دائرة مزخرفة

```
e.Graphics.ScaleTransform(90, 90, _
    MatrixOrder.Append)
    ' Translate to center on the form.
    e.Graphics.TranslateTransform( _
        Me.ClientRectangle.Width \ 2, _
        Me.ClientRectangle.Height \ 2, _
        MatrixOrder.Append)
    For i As Integer = 5 To 90 Step 5
        ' Save the state.
        Dim graphics_state As GraphicsState = _
            e.Graphics.Save()
        ' Rotate i degrees.
        e.Graphics.RotateTransform(i, _
            MatrixOrder.Prepend)
        ' Draw a rectangle.
        e.Graphics.DrawRectangle(New Pen(Color.Black, _
            0), -1, -1, 2, 2)
        ' Restore the saved state.
        e.Graphics.Restore(graphics_state)
    Next i
```

VB.Net Code

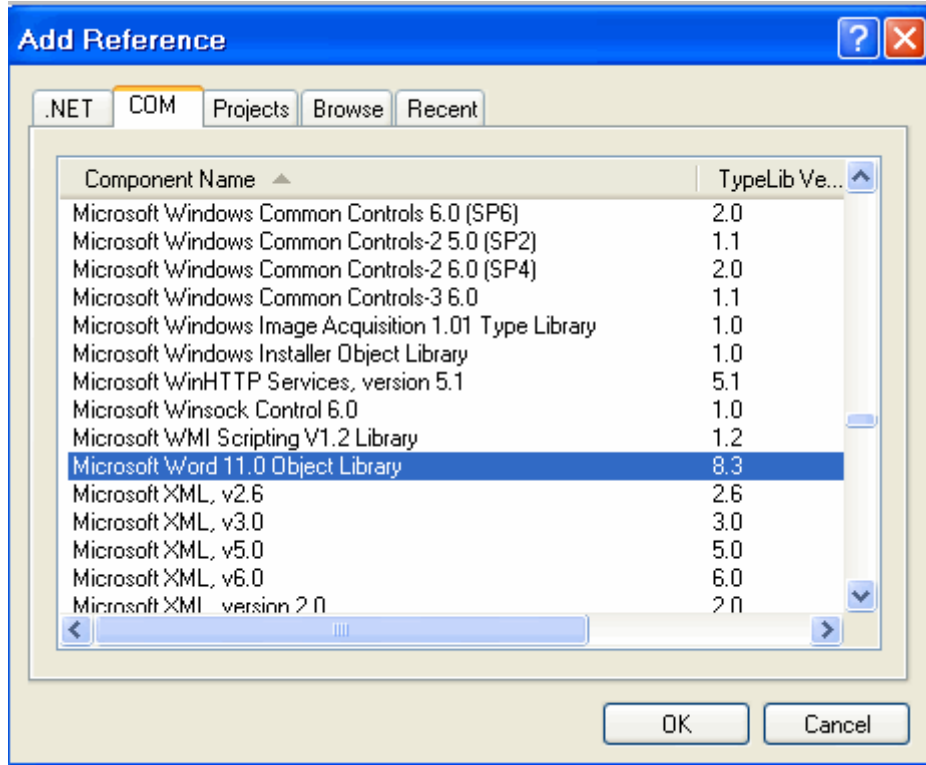
نموذج على شكل نص

```
Dim txt As String = "Path"  
Dim graphics_path As New GraphicsPath  
graphics_path.AddString(txt, _  
New FontFamily("Times New Roman"), _  
FontStyle.Bold, 150, _  
New Point(0, 0), _  
New StringFormat)  
' Set the form's region to the path.  
Me.Region = New Region(graphics_path)  
' Fill the path with white and outline it in black.  
Dim origin As Point = Me.PointToScreen(New Point(0, 0))  
  
e.Graphics.TranslateTransform(Me.Left - origin.X, _  
Me.Top - origin.Y)  
e.Graphics.SmoothingMode = SmoothingMode.AntiAlias  
e.Graphics.FillPath(Brushes.White, graphics_path)  
e.Graphics.DrawPath(New Pen(Color.Black, 5), graphics_path)
```


التعامل مع Microsoft Office

أولاً : مع برنامج Word

للتعامل مع برنامج الورد (Word) نحتاج إلى إضافة المكتبة ولفعل ذلك. أنشئ مشروعاً جديداً ومن القائمة Project أختَر Reference لتطالعك النافذة التالية



حدد المكتبة المحددة بالصورة وأنقر الزر Ok :
بعدها علينا ان نضيف مجال الأسماء التالي

Imports Microsoft.Office.Interop

وننشئ متغير جديد يشير الى الـ Application Word بالشكل التالي :

VB.Net Code

```
Imports Microsoft.Office.Interop
```

```
Public Class Form1
```

```
    Dim word As New Word.Application
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
    End Sub
```

```
End Class
```

كما ترى بالشكل المضلل هناك، ان برنامج الورد الان تحت تصرفك في كلمة واحدة Word وهي تمثل البرنامج ككل وبشكل عام .

الآن نريد ان ننشئ مشروعاً جديداً او مستنداً جديداً أنظر للشكل التالي :

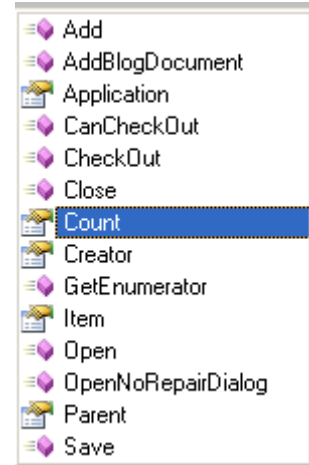
التعامل مع Office

VB.Net Code

```
word.Documents.Add()  
word.Visible = True
```

ان الخاصية Documents والتي هي للقراءة فقط تمثل مستند الو ورد.
وعن طريق الوظيفة Add يمكننا إضافة مستند جديد.
ان الخاصية Visible هي المسؤولة عن إظهار البرنامج وإلا فانك لن تراه فعلياً أمامك حتى وان كان يعمل
بخلفية النظام.

كما أن Documents لها العديد من الطرق والخصائص كما بالشكل التالي :



Add تحدثنا عنه .

Count يعود بعدد يمثل عدد المستندات المفتوحة.

Open لفتح مستند، و Save لحفظ المستند المفتوح حالياً، و Close
تستخدم لإغلاق المستند المفتوح حالياً.

كما يفضل للتعامل مع المستند المفتوح حالياً استخدام هذه الطريقة:

VB.Net Code

```
Dim doc As New Word.Document  
doc = word.Documents.Add()  
With doc  
    .Content.Text = "Qnefo Word 2007"  
    .Content.Underline =  
Microsoft.Office.Interop.Word.WdUnderline.wdUnderlineDotDash  
End With  
word.Visible = True
```

من المفيد جدا أن تعرف متغير جديد على انه المستند الذي تتعامل معه الآن.
وكما تلاحظ عرفنا متغير Doc على انه من النوع Document تم أسندناه إلى المستند الذي قمنا بفتحه.
والآن المستند تحت تصرفنا لنفعل به ما نشاء.
وأنا هنا كتبت جملة تم طبقت عليها بعض الوسوم .
يمكنك فتح ملف موجود بالشكل التالي :

VB.Net Code

```
word.Documents.Open("c:\dc.docx ")
```

أو حفظ الملف المفتوح حالياً بالشكل التالي

التعامل مع Office

VB.Net Code

```
word.Documents("المسار").save
```

أو لحفظ كافة المستندات المفتوحة حالياً كما بالشكل التالي

VB.Net Code

```
word.Documents.save
```

حيث أن Documents يشير إلى كل المستندات وليس مستند واحد فقط.
أيضاً يمكن حفظ المستند باسم آخر كما بالشكل التالي :

VB.Net Code

```
word.ActiveDocument.SaveAs(FileName:="C:\Qnefo.docx")
```

أن ActiveDocument تشير إلى المستند النشط حالياً أو الذي نتعامل معه الآن
إذا أردت أن تغلق أي وثيقة فيمكنك ذلك كما بالشكل التالي :

VB.Net Code

```
word.Documents("Qnefo").Close()
```

بحال أغلقت أي مستند بهذه الطريقة سيسألك البرنامج إذا كنت تريد حفظ التغييرات.
ولتجاهل ذلك أو لتتأكد من ذلك أنظر للشكل التالي :

VB.Net Code

```
word.Documents("Qnefo").Close(SaveChanges:=WdSaveOptions.wdDoNotSaveChanges)
```

أنظر ما الذي كتبناه بين القوسين، اننا نقول للبرنامج لا نريد حفظ المستند
يمكن إدراج النصوص بالشكل التالي

VB.Net Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
word.ActiveDocument.Content.InsertAfter("After")  
word.ActiveDocument.Content.InsertBefore("Before")  
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load  
Dim doc As New Word.Document  
doc = word.Documents.Add()  
word.Visible = True  
With doc  
.Content.Text = "Qnefo Word 2007"  
.Content.Underline =  
Microsoft.Office.Interop.Word.WdUnderline.wdUnderlineDotDash  
End With  
End Sub
```

لاحظ السطران المضللان الأول يقوم بإدراج نص بعد النص الموجود والثاني يضيفه قبله
أو بهذا الشكل :

التعامل مع Office

```
Word.Application.Selection.TypeText("نص أدرج")
```

تحويل المستند للمعاينة قبل الطباعة

```
Word.Application.ActiveDocument.PrintPreview()
```

تغيير عنوان المستند :

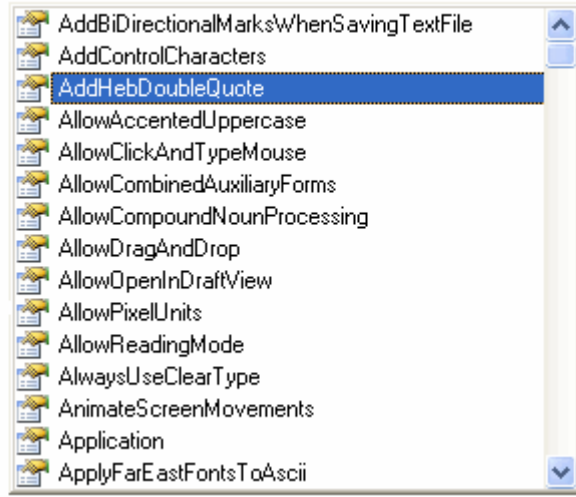
```
Word.Application.Caption = "العنوان"
```

فتح مستند في نافذة جديدة :

```
Word.Application.NewWindow()
```

كما تحت الـ Option يمكنك إيجاد الكثير من الخصائص التي يمكنك من التحكم في طريقة عرض المستند أنظر للشكل التالي :

```
Word.Application.Options.
```



تحديد نوعية الورق المستخدم أنظر للشكل التالي :

```
Word.Application.ActiveDocument.PageSetup.PaperSize = WdPaperSize.wdPaperA4
```

كما اذا كنت تحتاج الى تحديد الصفحة هل هي افقية ام عمودية أكتب التالي :

```
Word.Application.ActiveDocument.PageSetup.Orientation =  
WdOrientation.wdOrientLandscape
```

وهذه مجموعة من الخيارات أنظر للتالي :

```
Word.Application.ActiveWindow.DisplayRulers = False  
Word.Application.DisplayRecentFiles = False  
Word.Application.DisplayScreenTips = False
```

الخيار الاول هل اعرض المسطرة ام لا.

الخيار الثاني هل اعرض المستندات الأخيرة ام لا.

الخيار الثالث هل اعرض تلميح اليوم ام لا

وهذه تعني إغلاق الكل بدون حفظ أي شيء أنظر للتالي :

```
Word.ActiveWindow.Close(False)
```

ايضاً طباعة المستند الحالي :

```
Word.Documents.Application.PrintOut()
```

وبهذا الامر يمكننا معرفة الطابعة الافتراضية الآن أنظر للتالي:

```
MsgBox(Word.Documents.Application.ActivePrinter)
```

أيضاً يمكننا ادراج جدول كما بالشكل التالي :

```
Imports Microsoft.Office.Interop.Word
```

```
Public Class Form1
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Dim Word As New Microsoft.Office.Interop.Word.Application  
Word.Application.Documents.Add()
```

```
Dim rng As Range
```

```
rng = Word.Application.Selection.Range
```

```
Dim tbl As Table = Word.Application.Selection.Tables.Add(rng, 4, 4)
```

```
Word.Application.Visible = True
```

```
End Sub
```

```
End Class
```

إذا شغلت البرنامج بهذا الشكل فسيتم إنشاء الجدول ولكن لن تراه لان حدوده مخفية، وكما ترى اخبرنا البرنامج إننا نريد جدول من أربعة أعمدة و أربعة صفوف.
إذا اردنا تحديد سنابل أو شكل الحدود فيمكننا أن نضيف التالي :

```
 VB.Net Code
```

```
Dim Word As New Microsoft.Office.Interop.Word.Application  
Word.Application.Documents.Add()
```

```
Dim rng As Range
```

```
rng = Word.Application.Selection.Range
```

```
Dim tbl As Table = Word.Application.Selection.Tables.Add(rng, 4, 4)
```

```
tbl.Borders.InsideLineStyle = WdLineStyle.wdLineStyleDot
```

```
tbl.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleDot
```

```
Word.Application.Visible = True
```

وستكون النتيجة كما بالشكل التالي

التعامل مع Office

تحتاج أحيانا إلى تطبيق بعض الأشياء على الجدول ولتقوم بذلك تحتاج الى تحديد الجدول ولتفعل ذلك اكتب التالي :

```
Word.ActiveDocument.Tables(1).Select()
```

طبعاً رقم الفهرس للجدول وربما كان بالمستند اكثر من جدول.
طبعاً تلك المربعات هي عبارة عن حقول او خلايا يمكن التعامل معها كلا على حدة.
لحذف خلية نكتب التالي :

```
tbl.Cell(1, 4).Delete()
```

أنظر لهذه النتيجة لتفهم المقصود :

لقد تم حذف الخلية المحددة.
أو القيام بعملية دمج انظر للشكل التالي :

```
tbl.Cell(1, 4).Merge(tbl.Cell(6, 3))
```

وهذه هي النتيجة :

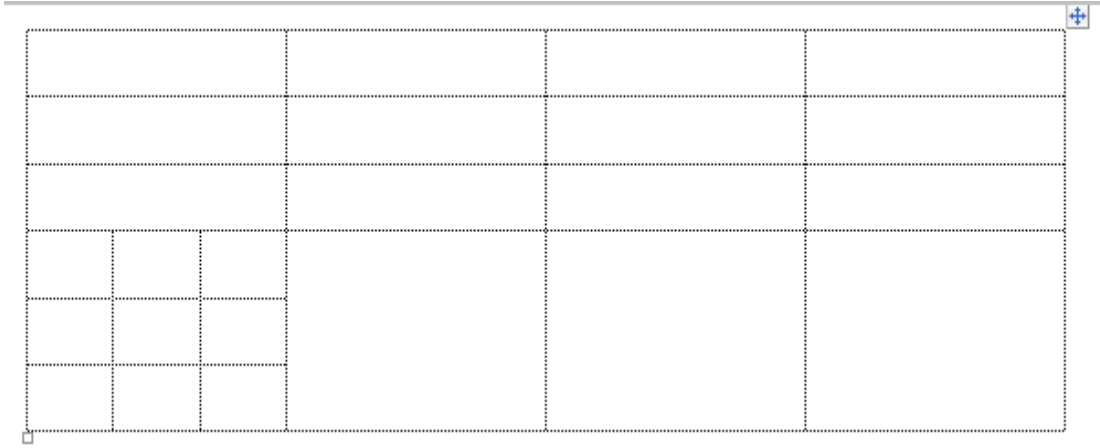
طبعاً العدد 6 خارج حدود تعريفنا للجدول، والبرنامج أعطانا حل تقريبي.

أو تقسيم الخلية أنظر للشكل التالي :

```
tbl.Cell(4, 4).Split(3, 3)
```

وهذه هي النتيجة :

التعامل مع Office



وأيضاً هل نجعل النص يلتف أم لا بالخلية المحددة أنظر للشكل التالي :

```
tbl.Cell(2, 4).WordWrap = True
```

ويمكننا محاذاة النص بالحقل كما بالشكل التالي :

```
tbl.Cell(3, 4).VerticalAlignment = WdCellVerticalAlignment.wdCellAlignVerticalCenter
```

كما إذا أردنا ملئ هذا الجدول بالبيانات نكتب التالي :

VB.Net Code

```
Dim Word As New Microsoft.Office.Interop.Word.Application
Word.Application.Documents.Add()

Dim rng As Range
rng = Word.Application.Selection.Range
Dim tbl As Table = Word.Application.Selection.Tables.Add(rng, 4, 4)

tbl.Borders.InsideLineStyle = WdLineStyle.wdLineStyleDot
tbl.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleDot

tbl.Cell(1, 1).Select()
Word.Application.Selection.Text = "متسلسل رقم"

tbl.Cell(1, 2).Select()
Word.Application.Selection.Text = "الملف أسم"

tbl.Cell(1, 3).Select()
Word.Application.Selection.Text = "الملف حجم"

tbl.Cell(1, 4).Select()
Word.Application.Selection.Text = "ملاحظات"

Word.Application.Visible = True
```

وستكون النتيجة كما بالشكل التالي

التعامل مع Office

رقم متسلسل	اسم الملف	حجم الملف	ملاحظات

يمكن استرجاع النصوص من المستند بالطريقة التالية

VB.Net Code

```
Dim Word As New Microsoft.Office.Interop.Word.Application
Word.Application.Documents.Add()

Dim rng As Range
rng = Word.Application.Selection.Range
Dim tbl As Table = Word.Application.Selection.Tables.Add(rng, 4, 4)

tbl.Borders.InsideLineStyle = WdLineStyle.wdLineStyleDot
tbl.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleDot

tbl.Cell(1, 1).Select()
Word.Application.Selection.Text = "متسلسل رقم"

tbl.Cell(1, 2).Select()
Word.Application.Selection.Text = "الملف أسم"

tbl.Cell(1, 3).Select()
Word.Application.Selection.Text = "الملف حجم"

tbl.Cell(1, 4).Select()
Word.Application.Selection.Text = "ملاحظات"

MsgBox(Word.ActiveDocument.Words(1).Text)

Word.Application.Visible = True
```

لاحظ السطر الذي أضفناه إذا شغلت البرنامج ستحصل على التالي :



ثانيا : التعامل مع EXCEL

خاص ب Excel 2003

بعث البيانات إلى Excel

 VB.Net Code

```
Public Class Form1
    Dim Exl As New Microsoft.Office.Interop.Excel.Application
    Dim Book As Microsoft.Office.Interop.Excel.Workbook
    Dim Sheet As Microsoft.Office.Interop.Excel.Worksheet

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Book = Exl.Workbooks.Add()
        Sheet = Book.ActiveSheet()
        Exl.Cells(1, 1) = "A"
        Exl.Visible = True
    End Sub
End Class
```

إرسال بيانات قاعدة إلى Excel

 VB.Net Code

```
Public Class Form1
    Dim Exl As New Microsoft.Office.Interop.Excel.Application
    Dim Book As Microsoft.Office.Interop.Excel.Workbook
    Dim Sheet As Microsoft.Office.Interop.Excel.Worksheet
    Dim StrCon As String = "Provider = microsoft.jet.oledb.4.0;" & _
"data source = d:\dbecole3.mdb"
    Dim con As New ADODB.Connection
    Dim rst As New ADODB.Recordset

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Book = Exl.Workbooks.Add()
        Sheet = Book.ActiveSheet()
        Try
            con.Open(StrCon)

            rst.CursorLocation = ADODB.CursorLocationEnum.adUseClient
            rst.Open("select * from attestation", con,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        Do Until rst.EOF
```

```

Dim I As Integer
i = i + 1
Exl.Application.Cells(I, 1) = rst.Fields("الاسم").Value
Exl.Application.Cells(I, 2) = rst.Fields("اللقب").Value
rst.MoveNext()
Loop
Exl.Visible = True
End Sub
End Class

```

من Excel إلى vb

 VB.Net Code

```

Public Class Form1
    Dim Exl As New Microsoft.Office.Interop.Excel.Application
    Dim Book As Microsoft.Office.Interop.Excel.Workbook
    Dim Sheet As Microsoft.Office.Interop.Excel.Worksheet
    Dim StrCon As String = "Provider = microsoft.jet.oledb.4.0;" & _
        "data source = d:\dbecole3.mdb"
    Dim con As New ADODB.Connection
    Dim rst As New ADODB.Recordset

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Book = Exl.Workbooks.Open("d:\book1.xls")
        Sheet = Book.ActiveSheet()

        Try
            con.Open(StrCon)
            rst.CursorLocation = ADODB.CursorLocationEnum.adUseClient
            rst.Open("select * from Tm3", con,
ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
            Do Until rst.EOF
                Dim I As Integer
                i = i + 1
                If I = 11 Then Exit Do
                rst.Fields("رقم التسجيل").Value = Exl.Application.Cells(I, 1)
                rst.Update()
                rst.MoveNext()
            Loop
            Exl.Visible = True
        End Sub
End Class

```

قاموس مصطلحات GLOSSARY

Visual Basic.Net 2005

مصطلحات: لغت البيزيك المرئي دوت نت

المصطلح	المعني أو الوظيفة
Activate	تنشيط الكائن (حدث لبعض الكائنات)
Add Form	إضافة نموذج جاهز
Add Project	إضافة مشروع جاهز
ADO .NET	مكتبة التعامل مع قواعد البيانات
Alignment	محاذاة (خاصية لبعض الكائنات)
Alignment - Center	محاذاة إلى الوسط أو التوسط (خاصية لبعض الكائنات)
Alignment - Left	محاذاة الكتابة إلى اليسار (خاصية لبعض الكائنات)
Alignment - Right	محاذاة الكتابة إلى اليمين (خاصية لبعض الكائنات)
Apostrophe	الرمز (') الفاصلة العنوية
Applications	التطبيقات
Arithmetic	حساب
Arithmetic Expressions	التعبيرات الحسابية
Arithmetic Operator	المعاملات الحسابية
Arithmetical	حسابي
Array	المصفوفات
Asc ()	دالة تستخدم في الحصول على كود أي حرف أو رمز أو كود أول حرف في سلسلة حرفية
Auto Size	تجسيم تلقائي طبقاً لمحتوي الكائن (خاصية لبعض الكائنات)
Back Color	لون الخلفية (خاصية لبعض الكائنات)
Back Style	نمط الخلفية (خاصية لبعض الكائنات)
BackGroundImage	التحكم في صورة الخلفية للنموذج (خاصية للنموذج)
BASIC	كود التعليمات الرمزية لجميع أغراض المبتدئين وهو اختصار <u>Beginner's All purpose Symbolic Instruction Code</u>
Basic Concepts	مفاهيم أساسية
Boolean	منطقي (نوع البيان)
Border Style	نمط حدود الكائن (خاصية لبعض الكائنات)
Break	إيقاف تشغيل البرنامج مؤقتاً
Byte	عدد صحيح (نوع البيان)
Change	تغيير في الكائن (خاصية لبعض الكائنات)
CharacterCasing	معرفة أو تحديد ما إذا كانت الأداة ستتدخل في تعديل الحروف
Check Box	صندوق اختبار (أداة من صندوق الأدوات)
Chr ()	دالة تستخدم في الحصول على الحرف أو الرمز المناظر لكود معين
Circle	رسم دائرة (وسيلة للنموذج ، كود برمجي)
Classes	التصنيف أو الفئة
Click	نقر منفرد بالزر الأيسر للفأرة (خاصية لبعض الكائنات)
CLS	مسح الكتابة والرسم من داخل الكائن (كود برمجي)
Code Window	نافذة الأكواد البرمجة
Comma	الفاصلة (,)
Command Button	زر أوامر (أداة من صندوق الأدوات)
Comparison Operator	المعاملات المقارنة
Compiler	المرجم
Computer Languages	لغات الكمبيوتر
Concatenation - String	سلسلة حرفية
Concatenation Operator	المعاملات الربط

المصطلح	المعني أو الوظيفة
Condition	الشرط
Conditional Branching	التفرع المشروط
Const (Constant)	ثابت - تعريف المخزن الثابت
Control Box	صندوق تحكم ، صندوق الأدوات
CStr()	دالة تستخدم في الحصول على السلسلة الحرفية المكافئة لعدد معين أو لمتغير عددي
Current X	البعد الحالي للكتابة أو الرسم عن الحافة اليسرى للكان
Current Y	البعد الحالي للكتابة أو الرسم عن الحافة اليمنى للكان
Customize	تخصيص
Data Type	نوع البيان (حرفي - عددي -)
DB Click (Double Click)	نقر مزدوج بالزر الأيسر للفأرة (حدث)
Defined Functions	الدوال المعرفية
Definition	تعريف الدالة
Design Mode	نمط التصميم
Dim (Dimension)	إعلان ، تعريف المخزن المتغير أو المصفوفة (كود برمجي)
Draw Width	سمك خط الرسم داخل الكائن (خاصية لبعض الكائنات)
Edit	تحرير (قائمة في شريط القوائم)
Encapsulation	التغليف
End	إغلاق أو إنهاء البرنامج (كود برمجي)
End Sub	إنهاء الإجراء (كود برمجي)
Enterprise Edition	النسخة المشروحات لبرنامج البيزيك المرني
Enum	البناءات
Event	حدث (يقع علي كائن ويتأثر به)
Event Driven Programs	البرامج الموجهة بالأحداث
Event Driven	الاستجابة للأحداث التي تقع عليه
Event Handler	معالج الأحداث
EXE File	ملف تنفيذي (البرنامج النهائي)
Expression	تعبير (قد يكون نصاً أو حساباً أو)
False	خطأ
File	ملف (قائمة في شريط القوائم)
Fill Color	لون تظليل الرسومات أو لون التعبئة (خاصية لبعض الكائنات)
Fill Style	نمط تظليل الرسومات أو لون التعبئة (خاصية لبعض الكائنات)
Fix ()	دالة تستخدم في الحصول على الجزء الصحيح الموجود في عدد حقيقي معين
Font	خط الكتابة (خاصية لبعض الكائنات)
For..... Next	حلقة تكرارية (كود برمجي)
Fore Color	لون الأمامية ، لون خط الكتابة (خاصية لبعض الكائنات)
Form	نموذج (الكائن الرئيسي في البيزيك المرني)
Form Layout Window	نافذة التخطيط للنموذج
Form Window	نافذة النموذج
Form1.Top	بُعد نافذة النموذج عند حافة الشاشة العليا (خاصية للنموذج)
FormBorderStyle	التحكم في الأطار (خاصية للنموذج)
Function	دالة - للحصول على النتيجة قد تكون (حرفية ، عددية ،)
GroupBox	وعاء تجميع (أداة من صندوق الأدوات)
Hardware	الأجهزة المادية
Height	ارتفاع (خاصية لبعض الكائنات)
Help	مساعدة
High Level Language	اللغة عالية المستوى (لغة يفهمها الإنسان)
Hscrollbar	شريط التمرير الأفقي (أداة من صندوق الأدوات)

المصطلح	المعنى أو الوظيفة
HScrollbar -LargeChange	أكبر تغيير في شريط التمرير الأفقي (خاصية لشريط التمرير)
HScrollbar -SmallChange	أقل تغيير في شريط التمرير الأفقي (خاصية لشريط التمرير)
Icon	رمز أو أيقونة (خاصية للنموذج)
If....Then Else.... Endif	جملة شرطية (كود برمجي)
PictureBox	أداة رسم (أداة من صندوق الأدوات)
Image	أضافة صورة الى كائن (خاصية للنموذج)
Inheritance	التوريث (OOP)
InputBox ()	دالة تعرض رسالة داخل صندوق حوار، ثم ينتظر من المستخدم إدخال نص أو النقر على أي زر من أزرار صندوق الحوار.
Instr ()	دالة تستخدم للحصول على مكان حرف أو مكان كلمة مطلوب البحث عنها داخل سلسلة حرفية معينة.
Integer	عدد صحيح (نوع البيان)
IntelliSense list	أستشعار ذكي (هي القائمة التي تظهر عند كتابة الكود)
Interpreter	المفسر
Interval	فترة زمنية مستغرقة (تفاوت على فترات ، خاصة لميقاتي)
Label Control	أداة العنوان (أداة من صندوق الأدوات)
Language	لغة
LCase ()	دالة تستخدم لتحويل الحروف الإنجليزية الكبيرة إلى الصغيرة
Left	يسار (خاصية لبعض الكائنات)
Len ()	دالة تستخدم في الحصول على طول أي سلسلة حرفية
Limited looping	التكرار المحدد
Line	وسيلة رسم خط (من وسائل النموذج)
Load	تحميل النموذج (إجراءات الحدث)
Logic - logical	منطقي - منطقي
Logical Operator	المعاملات المنطقية
Long Long Integer	عدد صحيح طويل (نوع البيان)
Low Level Language	اللغة المتدنية - منخفضة - المستوى (لغة يفهمها الكمبيوتر)
Macros	الماكرو
Main Window	النافذة الرئيسية
MaxButton	زر تكبير النافذة (من خصائص النموذج)
MDI	نموذج واجهة المستخدم المتعددة Multiple Document Interface Form
Menu Bar	شريط القوائم
MessageBox.Show	دالة عرض الرسائل الثابتة بأي طرق
Method	وسيلة
Mid ()	دالة تستخدم للحصول على جزء من سلسلة حرفية معينة بدءاً من حرف معين داخل هذه السلسلة.
Min Button	زر تصغير النافذة (من خصائص النموذج)
Module	وحدة نمطية (جزء البرمجة)
Mouse Down	زر الفأرة للضغط عليه (حدث لبعض الكائنات)
Mouse Move	حركة زر الفأرة (حدث لبعض الكائنات)
Mouse Up	ترك زر الفأرة عند الضغط عليه (حدث لبعض الكائنات)
MsgBox ()	دالة تعرض رسالة في صندوق حوار
MultiLine	الكتابة داخل صندوق النصوص في عدة أسطر (خاصية للنص)
Multimedia	الوسائط المتعددة
Multiple Branching	التفرع المتعدد

المصطلح	المعنى أو الوظيفة
Name	الاسم البرمجي للكان (خاصية في جميع الكائنات)
Name Space	مجالات الأسماء
New Project	مشروع جديد
New Value	القيمة الجديدة التي يتم إضافتها للخواص
Object	كائن
Object Behavior	سلوك الكائن
Object Name	اسم الكائن
Object Oriented Programming	البرمجة الموجهة بالأحداث
OOP	البرمجة الكائنية التوجيهية - البنائية - Object Oriented Programming
Open Project	فتح مشروع
Operator	معامل
Options	خيارات
Paint	حدث رسم النموذج على الشاشة (كود برمجي)
Picture	صورة (خاصية لبعض الكائنات)
Pointer	المؤشر
Print	وسيلة طباعة (من وسائل النموذج)
Procedure	إجراء
Professional Edition	النسخة المتخصصة لبرنامج البيزيك المرني
Project Explorer Window	نافذة مستكشف المشروع
Project Window	نافذة المشروع
Properties Window	نافذة الخصائص
Puse Button	ضغط الزر (من لوحة المفاتيح ، الفارة)
Radio Button	زر اختيار (أداة من صندوق الأدوات)
Randomize	وسيلة لتوزيع البيانات عشوائياً
ReadOnly	جعل الكائن للقراءة فقط (خاصية لبعض الكائنات)
Rem	جملة لوضع تعليق على الاكواد البرمجية في نافذة البرمجة
Replace ()	دالة تستخدم في لاستبدال حرف أو كلمة أو جزء من سلسلة حرفية بحرف آخر أو كلمة أو سلسلة أخرى.
Reserved Functions	الدوال المحفوظة
RightToLeft	التحويل الاتجاه من اليمين إلى اليسار (خاصية لبعض الكائنات)
Rnd ()	دالة تستخدم في توليد عدد حقيقي عشوائي قيمته أكبر من أو تساوي الصفر وأقل من الواحد الصحيح.
Run	تنفيذ (قائمة في شريط القوائم)
Run Mode	نمط التشغيل
Save Form	حفظ النموذج
Save Project	حفظ المشروع
Scale Height	عدد الوحدات الرأسية للكان و هي المستخدمة مع الرسومات
Scale Mode	نمط وحدة القياس في الكائن (خاصية للنموذج)
Scale Width	عدد الوحدات الأفقية للكان و هي المستخدمة مع الرسومات
Scrollbar – Max Value (Maximum Value)	أكبر قيمة لشريط التمرير
Scrollbar – Min Value (Minimum Value)	أقل قيمة لشريط التمرير
SDLC	دورة حياة تطوير النظام System Development Life Cycle
Select a control	أختيار الزر
Semi Colon	الفاصلة المنقوطة (;)
Setup Project	إعداد المشروع للنشر
Single	عدي حقيقي (نوع البيان)

المصطلح	المعنى أو الوظيفة
Single Precision	أحادى الدقة
Size Mode	نمط عرض الصورة فى الكائن (خاصية لصندوق الصور)
Snap Lines	خطوط الزرقاء التى تظهر عند ضبط المجاذاة الكائنات
Snippets	الأوامر الجاهزة فى الفيجوال بيسك ٢٠٠٥
Software	البرمجيات
Solution	الحل
Solution Explorer	نافذة الحل
Source Program	البرنامج المصدر
Special Symbols	الرموز الخاصة
SQL	لغة الاستعلامات الهيكلية <u>Structured Query Language</u>
Stander	قياسي
Stander Edition	النسخة القياسية لبرنامج البيزيك المرني
Statement	الجمل البرمجية
Stretch	تجديم الصورة الموجودة داخل صندوق الرسم
String	بيان لسلسلة نصية
String Expressions	التعابير النصية
Structured Programming	البرمجة الهيكلية
Sub	بداية الإجراء
Symbol	رمز
Text Box Control	أداة صندوق النص
TextBox	صندوق النص (أداة من صندوق الأدوات)
TextBox – Text	خاصية الكتابة داخل صندوق النص
TimeOfDay ()	دالة تستخدم فى الحصول على الوقت فى اليوم
Timer	التوقيت – الميقاتي (أداة من صندوق الأدوات)
Title Bar	شريط العنوان
Toolbar	شريط الأدوات
Toolbox	صندوق أدوات
Tools	أدوات (قائمة فى شريط القوائم)
True	صواب
Try Catch	لتفادى الأخطاء المنطقية والبرمجية فى البرنامج (قنص الأخطاء)
UCase ()	دالة تستخدم لتحويل الحروف الإنجليزية الصغيرة إلى الكبيرة
Unconditional Branching	التفرع الغير المشروط
User Interface	واجهة المستخدم
V.ScrollBar	شريط التمرير راسي
Val ()	دالة تستخدم فى الحصول على القيمة العددية لسلسلة حرفية معينة أو متغير حرفي
Value	قيمة (خاصية لبعض الكائنات)
Variable	متغير
View	عرض (قائمة فى شريط القوائم من خلالها اظهار النوافذ)
Visible	ظهور الكائن أو إخفاؤه (خاصية لبعض الكائنات)
Visual Basic	البيزيك المرني
Visual Studio.Net	حزمة البرامج التى تحتوى على برنامج البيزيك المرني
Width	عرض (خاصية لبعض الكائنات لقياس أو لضبط عرضها)
Working Area	نافذة منطقة عمل

