

[WWW.AleppoForce.vbcafe.net](http://WWW.AleppoForce.vbcafe.net)

# رائحة المسك



Visual Basic 6.0

## فيجوال بيسك

إعداد وتأليف :

فؤاد أصيل  
AlëppoForce

2008 - 1429 هـ



جميع الحقوق محفوظة للكاتب ...  
وعلى ذلك لا يحق الزيادة على هذا الكتاب أو التعديل عليه  
إلا من قبل من أعطاه المؤلف الحق بذلك

هذا الكتاب مجاني لا يجوز بيعه ..... تأليف : فؤاد أصيل

مقدمة الكتاب

بقلم الأستاذ الكبير RamSyria

منذ سنوات كان يندُر ما نرى مبدعاً شاباً .. يبحث عن طريقه لينشر علمه .. وهو صاحب سبعة عشرة ربيعاً ... والذي كان وما زال البعض يضع أصحاب الأبدعات الشابه بين أسوار كلمة (صغير) ،،،

ولقد أجبرتني إبداعات فؤاد أصيل ... أن أكتب هذه السطور البسيط عرفاناً مني له لما يقدم من مواضيع لفتت نظري .. وجعلني أبحث عن صاحبها

أما عن كتابه ( رائحة مسك ) فإن العنوان يدل على المحتوى ... بكل ما تعنيه كلمة محتوى من مكلمة ... وهناك نقطتان مهمتان يجب أن أوضحها لفتت نظري في هذا الكتاب :

\* هو ابتعاده عن الكلمات والمقدمات الروتينية....

\* ابتعاده عن عملية نقل .. فتجد شرحه للدروس نابعه عن إنسان فاهم تماماً ما يكتب ... بعكس غيره طبعاً ...

هذه النقطة المهمة التي جعلتني أحب أسلوب فؤاد أصيل في طريقة شرحه ... فهو يتجه إلى الناحية العملية أكثر من الناحية النظرية

مع خالص تمنياتي لك أخي أصيل بالتوفيق

رائحة المسك في تعلم الفيجوال بيسك ..... - ٣ -

وكن على ثقته إني أنتظرك ابداعتك الجديدة ، ونصيحتي لك أن تستمر على نفس  
الأسلوب بالشرح .. فهذا الأسلوب سوف يضعك في المقدمة عندما  
ترى المبدعين ... يسألون عن الفرق بينك وبينهم ... فهذا هو الفرق

وعلى هذا أنا أنصح الجميع أن يقرأ هذا الكتاب .. وجميع كتب والمواضيع فؤاد أصيل  
الماضية والمستقبلية

أخوكم Rami Bilal  
[www.RamSyria.com](http://www.RamSyria.com)

هذا الكتاب مجاني لا يجوز بيعه ..... تأليف : فؤاد أصيل

## • الباب الأول (بدء العمل مع البرنامج)

• تمهيد

• الشاشة الرئيسية للبرنامج

• أهم الأدوات و أهم الخصائص

• عن كذب مع الأكواد

• مثال عام على الباب الأول

• تدريب على الباب الأول



بسم الله و الصلاة والسلام على رسول الله ﷺ وعلى آله و صحبه ومن استن بسنته واقتفى أثره إلى يوم الدين ،  
أما بعد .....

بصراحة — أخي القارئ — لا يوجد أي داعٍ للبدء بمقدمة ، فأنت تتعلم لغة برمجة لا ريب أنك سمعت عنها كثيراً  
لكن سأبدأ بالتمهيد كي أجازي ما اعتاد المؤلفون عليه .

أولا : أعرف بنفسي

أخوكم : فؤاد بن محمد هاشم أصيل

العمر : ١٧ سنة وشوية

سوري مقيم في مدينة ينبع الصناعية بالمملكة العربية السعودية

وقد يتبادر سؤال إلى ذهنكم هاهنا يكون كما يلي : لماذا يؤلف طالب في المرحلة الثانوية وهو — حسب منهج الوزارة — لا يعلم شيئا عن البرمجة ولا الفيجوال بيسك والإجابة الصريحة هي كالتالي:

لقد درست البرمجة في مدارس الهيئة في الصف الثالث المتوسط — قبل عامين — وعشقت هذا التخصص وأبدعت فيه ( إلى حد ما ) — بشهادة المدرسين — غير أن المنهج البسيط لم يروي طموحي لتعلم هذه اللغة ، فبدأت بإحضار الكتب وتعلمت شيئا فشيئا حتى وصلت إلى ما أنا عليه الآن

فالشكر لله أولا ثم لمن درسي و علمني ولن أذكر أسماء لأن القائمة تطول وأنا أخشى من أن أنسى أحدهم.

والآن ندخل في الموضوع :

لماذا سميت البرمجة لغة ؟ — بغض النظر عن اسمها ( فجوال ، أوراكل ، ++C ..... ، إلخ )

بداية لا بد أننا نتفق على الآتي :

هذا الكتاب مجاني لا يجوز بيعه ..... تأليف : فؤاد أصيل

١) الكمبيوتر جهاز غبي ، ولا يتميز عن الإنسان بالذكاء بل يتميز عنه بالسرعة ، الدقة الحفظ ، إلخ ... لكن ليس بالذكاء مثال : لو سألت أي بني آدم أمي (لا يقرأ ولا يكتب ) السؤال التالي :-

هل تفضل ساعة متوقفة عن العمل ، أم ساعة متأخرة عن الوقت الرسمي بثانية واحدة فقط ؟

إجابة البني آدم (الذكي) ساعة متأخرة بثانية. ولا يوجد داعٍ للتفصيل في هذه الإجابة. إجابة الحاسوب (الغبي) الساعة الواقفة أفضل ، لأنها تكون صحيحة مرتين في اليوم بينما المتأخرة لا تكون صائبة أبدا

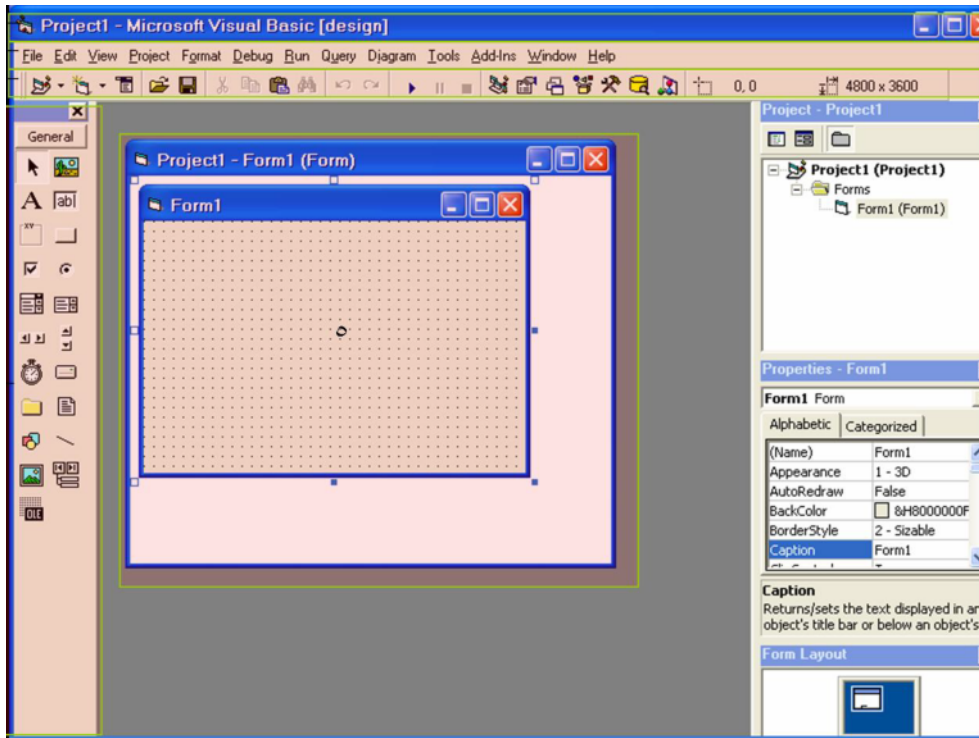
٢) الكمبيوتر (بسبب غبائه) لا يستطيع — على حد علمي — فعل أي شيء بدون إنسان (وهذا هو دور المبرمج)

٣) لا نستطيع أن نساوي بين ( مصمم فوتوشوب خبير ، بوربوينت ، فلاش ، ... إلخ ) ومبرمج محترف ، لأن المبرمج هو من يصنع البرنامج الذي يستخدمه هؤلاء ، وهنا وجب التنبيه على أن البرمجة سلاح ذو حدين ، ولن أفصل في هذا الموضوع.

٤) إذا أردت أن تكون مبرمج فعليك أن تكون ضليعا في مادتي ( الرياضيات ، والإنجليزي ) لأن الإنجليزي للمصطلحات والأحرف والرياضيات لأنها هي لب و قلب البرمجة بجميع أقسامها أي الرياضيات

والآن نعود لسؤالنا : لماذا سمينا البرمجة لغة ، الجواب : لأن اللغة هي أداة التواصل ، (تفهم الشخص الذي أمامك) والبرمجة لغة تواصل بينك وبين الكمبيوتر بل مهمتك الأولى هي أن تكون حلقة وصل بين الكمبيوتر ( نظام التشغيل ) والمستخدم (البني آدم). وستلاحظ أنه في لغة فجوال بيسك ٦ أغلب الأوامر عبارة عن معادلات ( مساواة بين طرفين ) وكيفينا مقدمات ..... ولنبدأ العمل

## الشاشة الرئيسية للفجوال بيسك ٦



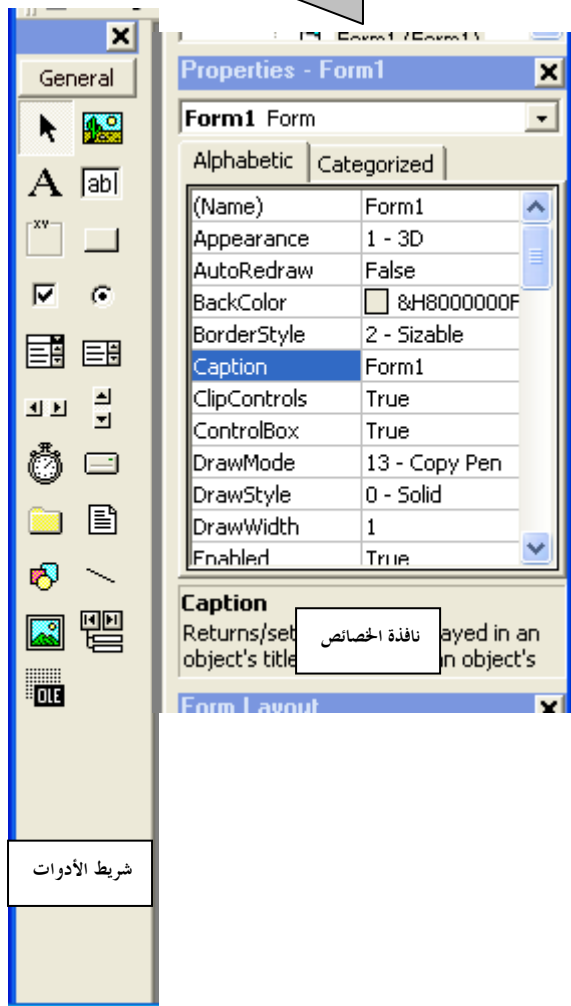
طبعاً أنا ماشرت كيفية الدخول إلى البرنامج و أن تختار standard exe لأن المفروض بك أن تعرف هذا لوحدهك  
الأسرطة والأشياء الموجودة في الصورة :

- (١) شريط العنوان (نفس البرامج الأخرى)
  - (٢) شريط القوائم (نفس البرامج الأخرى)
  - (٣) شريط الأدوات (وستكلم عنه بعدين)
  - (٤) شريط التنسيق (تقريباً يشبه البرامج الأخرى ونفس الكلام على الشريط القياسي)
  - (٥) منطقة التصميم (مكان الإبداعات التي يراها المستخدم)
  - (٦) نافذة الشيفرة (مكان التفاهم مع الكمبيوتر)
  - (٧) نافذة المشروع (بعدين)
- هذا يكفي بإذن الله



## الأدوات الهامة

شريط الأدوات



سيتم الآن شرح المهم من الأدوات  
والآن هل تذكر نافذة الخصائص  
في هذه النافذة ستستطيعون أن تعدلوا الخصائص  
الخاصة بالأدوات التي ستستخدمونها  
ولكن لا تظنوا أن هذه الخصائص هي للتعامل مع  
المستخدم فقط ! بل عليكم أن تعلموا أن بعض  
الخصائص هي للتعامل مع الكمبيوتر بواسطة الكود  
(الشفرة)

وسنبدأ الآن بذكر أهم الأدوات واحدا واحدا  
أخذين بخصائصها.



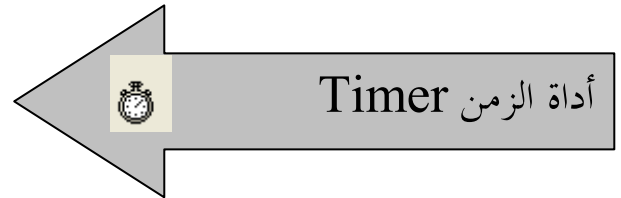
أداة الصورة picture box

الاسم ( إنجليزي )	العمل (التحكم بـ 😊)
Name	الاسم الذي تسمي به الأداة للكمبيوتر
Align	المحاذاة (يمين ، يسار ، ..... إلخ)
Appearance	نوع الإطار
Autosize	أخذ حجم تلقائي من الصورة
ToolTipText	النص الذي سيظهر عند إبقاء الماوس عليها
Picture	الصورة بداخلها
BackColor	لون الخلفية
Visible	مرئي أم مخفي

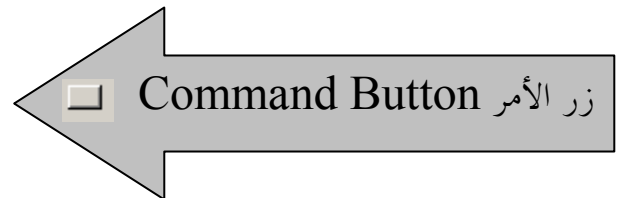




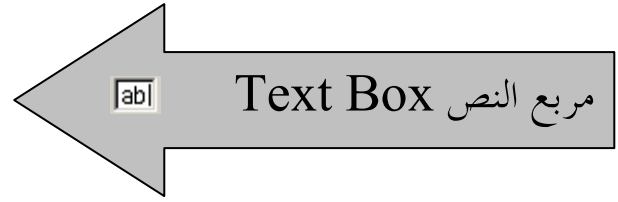
الاسم ( إنجليزي )	العمل (التحكم بـ😊)
Name	الاسم الذي تسمي به الأداة للكمبيوتر
Align	المحاذاة (يمين ، يسار ، ..... إلخ)
Appearance	إظهار
Caption	النص الذي سيظهر للمستخدم
ToolTipText	النص الذي سيظهر عند إبقاء الماوس عليها
Font	الخط
ForeColor	لون الخط
Visible	مرئي أم مخفي



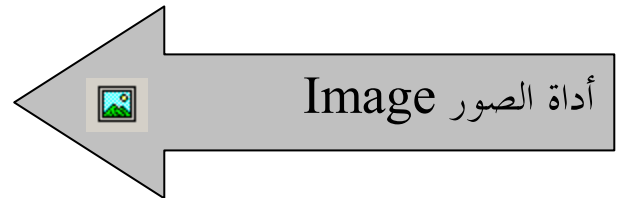
الاسم ( إنجليزي )	العمل (التحكم بـ😊)
Interval	التوقيت المطلوب ( الوحدة ثانية / ١٠٠٠ )
Enabled	تعمل / لا تعمل



ويعتبر من أهم الأدوات و سنفصل به إن شاء الله في الدروس



الاسم ( إنجليزي ) للخاصية	العمل (التحكم بـ😊)
Name	الاسم الذي تسمي به الأداة للكمبيوتر
Align	المحاذاة (يمين ، يسار ، ..... إلخ)
Appearance	إظهار
Text	النص الذي سيظهر للمستخدم
ToolTipText	النص الذي سيظهر عند إبقاء الماوس عليها
Font	الخط
ForeColor	لون الخط
Visible	مرئي أم مخفي
Lock	إمكانية الكتابة داخله



الاسم ( إنجليزي ) للخاصية	العمل (التحكم بـ😊)
هذه الأداة تشبه إلى حد كبير أداة Picture box لكن أهم فرق هو الخاصية:	
Streach	تعديل حجم الصورة على حسب حجم الأداة

وهكذا نكون قد عرفنا أهم الأدوات ، ولا تنسى أننا سنحتاج إلى فهم الخواص جيداً والفهم لا يأتي من الدراسة النظرية (الحفظ) بل تأتي بالتطبيق العملي.

## عن كُتب مع الأكواد

في هذا الجزء فُتِح دماغك تمام

❖ ما هو الكود : الكود — كما أسلفت في المقدمة — هو عبارة عن أوامر بترتيب معين تأمر بها الكمبيوتر وهو يقوم بتنفيذها كما أوردتها له بالترتيب. وهذا التعريف هو للكود بشكل عام ( من أيام نظام الدوس ) ويسمى بالبرمجة المسيرة بالأحداث ولكن الفيجوال يدعم أيضا البرمجة الموجهة بالكائنات ويمكنك البحث عن الفرق بينهما باستخدام الإنترنت وسأحاول شرح ذلك في كتاب آخر إن شاء الله

❖ من أشكال الكود:

○ معادلة : مساواة بين عبارتين | كذا = كذا

وعندما تعطي أمرا مشابها يقوم الكمبيوتر أو البرنامج بجعل ما قبل المساواة يأخذ نفس قيمة ما بعد المساواة .

مثال : اجعل النص في Text1 يساوي القيمة النصية ( أنا مبرمج )  
الحل :

(١) نكتب اسم الأداة ( الموجود في الخاصية Name )

(٢) نختار الخاصية المرغوبة عبر وضع Dot(.) ثم الخاصية وهي هنا

Text

(٣) وضع علامة المساواة (=)

(٤) وضع القيمة الجديدة وهي هنا نصية فنضعها بين علامتي

تنصيص "

سيبدو الكود كالتالي

Text1.Text = "أنا مبرمج"

## ○ تفعيل خاصية | control.action

هنالك خواص تحتاج فقط لإعطاء الأمر والمعلومات المطلوبة للتنفيذ

مثال : إجعل النموذج Form يختفي

الحل: Form1.hide

وهناك نوع يشبه هذا وهو مثل حذف ملف Kill FileName

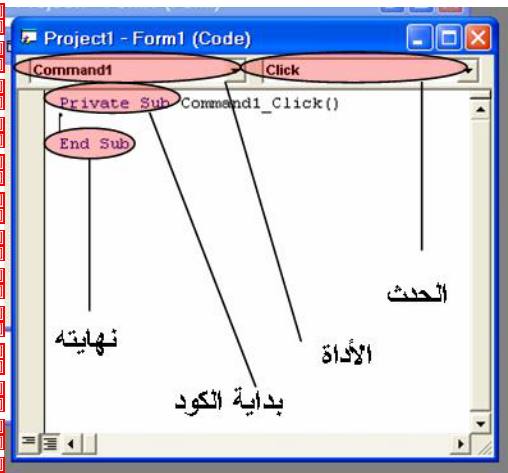
وطبعا هناك الكثير من الأنواع لكن ما سبق هو الأهم لك (كهاوي)

والآن إلى القسم العملي من "عن كذب مع الأكود"

أين أكتب الكود؟

لكتابة الكود أي عنصر في منطقة التصميم ما عليك إلى الضغط عليه مزدوجا وستظهر

نافذة الشيفرة :



الحدث : يعني متى يتم تنفيذ الكود.

الأداة : الأداة التي تستعمل من قبل المستخدم لتنفيذ الأمر

والأمر المكتوب هنا لن ينفذه الكمبيوتر إلا إذا استخدمت

هذه الأداة

ونلاحظ أن (General) يسمى قسم التصريحات يعني

ما تكتبه هنا سينطبق على جميع الأكود في الشاشة

بداية الكود : نلاحظ أنه مكتوب Private Sub وبعدها اسم الأداة ثم \_ ثم الحدث

نهايته : ينتهي الكمبيوتر من تنفيذ الأوامر المتتابة عندما يصل إلى هذا السطر.

---

هذه المنطقة ( شاشة الشيفرة أو الأكود ) هي منطقة تفاهمك مع الكمبيوتر بينما منطقة

التصميم هي منطقة التفاهم مع المستخدم.

وليكون برنامجك ناجح عليك بالإبداع مع الكمبيوتر ، كما عليك بالإبداع مع المستخدم

## مثال عام على الباب الأول

قبل أن أبدأ بالمثال أريد أن أوضح أموراً

❖ من الخصائص الهامة التي توجد في أغلب الأدوات الخصائص التالية :

- Height : وتعني الطول
- Width : وتعني العرض
- Top : وتعني البعد عن الطرف الأعلى للنموذج (Form)
- Left : وتعني البعد عن الطرف الأيسر للنموذج

انظر للرسم

❖ بعض الأدوات يمكن

أن يوضع داخلها

أدوات أخرى مثل :

الإطار Frame

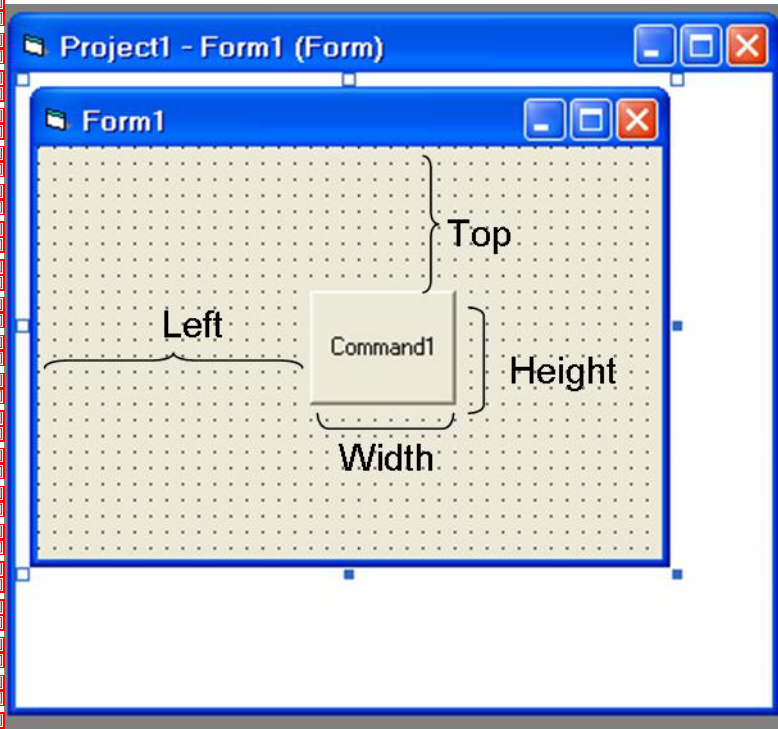
وأيضاً مربع الصورة

PictureBox

في جميع الأمثلة في هذا الكتاب لن

أتطرق لـ : كيف تفتح البرنامج بل

سأبدأ فوراً بالعمل



فَهَيَّا بِنَا نُبْرِمَجْ

رائحة المسك في تعلم الفيجوال بيسك

فكرة البرنامج (من نظرتك للتصميم)

القدرة على تكبير وتصغير وتحريك صورة معينة

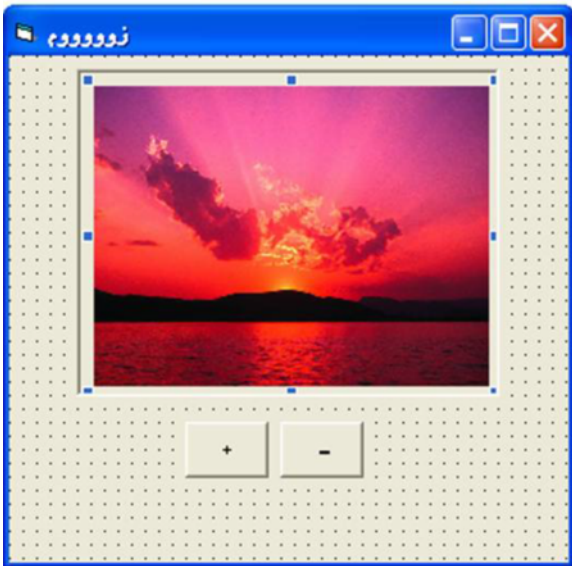
فكرة البرنامج (من نظرتك للكود)

وضع أداة Image داخل أداة PictureBox وبما أن ال Image لديها الخاصية Streach إذا سيزيد حجم الصورة كلما زدنا حجم الأداة Image ومهمة الأداة PictureBox هو جعل نظر المستخدم محدودا ، وستكون الفكرة أوضح بعد التطبيق.  
طور التصميم :

ضع أداة PictureBox ثم ارسـم داخلها أداة Image  
ضع ٢ من أزرار الأمر في مكان مناسب في النموذج  
أنشئ التغييرات التالية على خصائص الأدوات :

التعديل	الخاصية	الأداة/الفورم
زوووووم	Caption	Form
Maximized	WindowState	Form
True	Streach	Image1
اختر أي صورة	Picture	Image1
+	Caption	Command1
-	Caption	Command2

سيصبح شكل النموذج كالتالي:



طبعاً لا تبدأ بمقارنة هذه البرامج الصغيرة ببرامج  
الأفيس MS Office لأنك في طور النمو برمجياً  
لكن بالمواظبة والإصرار ستكون ذات يوم مبرمج  
نظم

نأخذ الكود حبة حبة

أولا الزر **Command1** الذي سيزيد حجم الصورة (Zoom In)  
اضغط عليه ضغطتين وأدخل الكود التالي :

```
Image1.Height = Image1.Height + 100  
Image1.Width = Image1.Width + 100
```

السطر الأول يعني اجعل طول **Image1** = طولها الحالي + ١٠٠  
السطر الثاني يعني اجعل عرض **Image1** = عرضها الحالي + ١٠٠

ثانيا الزر **Command2** الذي سينقص حجم الصورة (Zoom Out)  
اضغط عليه ضغطتين وأدخل الكود التالي :

```
Image1.Height = Image1.Height - 100  
Image1.Width = Image1.Width - 100
```

السطر الأول يعني اجعل طول **Image1** = طولها الحالي - ١٠٠  
السطر الثاني يعني اجعل عرض **Image1** = عرضها الحالي - ١٠٠

وهكذا انتهى الباب الأول ولكي تتأكد أنك فهمته تماما حل هذا السؤال:  
تدريب الباب الأول

أنت مبرمج وطلب منك أحد الزبائن تصميم برنامج يحوي نصاً **Label** و ٤ أزرار أمر هي  
واحد يحرك النص لليمين وواحد يحركه لليسار وواحد يحركه لفوق وواحد يحركه للأسفل  
فهل تستطيع أن تصنع هذا البرنامج؟

مساعدة : نفس فكرة المثال تقريبا ، لكن باستخدام **Left** و **Top** وإذا تمكنت من صنع هذا  
البرنامج فأنت مشروع ("مبرمج محترف") وأما إن لم تتمكن من ذلك فراسلني لكن بعد أن تستسلم



• الباب الثاني (التصميم و التزيين)  
• التواصل مع المستخدم

• تمهيد

• حيل في التصميم

• خلفية للنموذج

• استخدام (خط)

• تزيين زر الأمر

• تلوين نص بشكل لافت للانتباه

• إنشاء شريط أدوات في البرنامج

• إضافة أدوات أخرى

• وضع فلاش في البرنامج

• تغيير الواجهة للنموذج Skin



علمت مما سبق أنك - كمبرمج - عليك التعامل مع جبهتين مختلفتين ، هما  
أولا : مع المستخدم (البي آدم الذي يستخدم البرنامج) عن طريق التصميم ، فعليك أن تجعل  
استعماله للبرنامج سهل وواضح  
ثانيا : مع نظام التشغيل ( الكمبيوتر ) عن طريق الكود ، فتكون أوامرك واضحة العبارة ، صحيحة  
الصياغة كي ينفذ أوامرك على أكمل وجه.  
وستكلم في هذا الباب عن تصميم البرنامج وبعض الحيل في ذلك ، معتمدا على تجربتي الشخصية  
في هذا المجال.  
وأعيد وأذكر أن هذا الكتاب هو عمل بشري قابل للخطأ فعلى من يجد أي خطأ في أي جزء من  
الكتاب أن يبلغني به لتعم الفائدة بإذن الله  
لا تظنّ يا زميلي المبرمج أنك في وضع التصميم سوف تقوم بإضافة الأدوات وتعديل بعض  
خصائصها فقط ، لأن الواقع هو أنك ستحتاج أيضا لكتابة أكواد بسيطة ، سوف تزين لك  
البرنامج على أكمل وجه إن شاء الله.  
هذا الباب عملي أكثر ما أنه نظري ، إذا عليك أن تجرب ما تتعلمه أول بأول

ستبدأ مباشرة / فهيا بنا نبرمج

## حيل في التصميم

(١) خلفية للفورم :

يجب أن تكون مثلي قد سئمت من خلفية الفورم ذات اللون التعيس. المائل للرمادي الفاتح لكن يمكنك تغييرها بـ:


(أ) صورة :

عن طريق الخاصية Picture في الفورم

(ب) لون :

عن طريق الخاصية BackColor في الفورم

وهذا لا يحتاج لا إلى شرح ولا لتفصيل

(٢) استخدام Line أو الخط ، وهو أداة شكلها  وقد تعتقد أنه ليس لها داعٍ لكن راقب المثال:



بعد



قبل

لا شك أنك تلاحظ الفرق مع أنه فارق بسيط

(٣) تزيين زر الأمر

تعلم تماما أن الخاصية BackColor تغير لون خلفية الأداة ، طيب غير هذه الخاصية في زر أمر إلى الأحمر ، ماتغير؟؟؟ ، إذا هل هذا خطأ في الفيجوال بيسك؟ | قطعاً لا هناك خاصية يجب تغييرها لتتمكن من التغيير في الخواص العامة لزر الأمر ، فهي كالقفل الذي يمنعك من إظهار تغييراتك هذه الخاصية هي : Style غيرها من standard إلى Graphic وشاهد الفرق

(٤) تلوين نص بشكل مثير للانتباه

أحيانا يكون هناك نص معين تريد لفت الأنظار إليه ، إما اسمك أو دعاية أو معلومة مفيدة وعلى حد علمي فإن أفضل طريقة للفت الانتباه تغيير اللون باستمرار، فيبدو كأنه يلعب طيب ،،، لتنفيذ ذلك علينا معرفة كودين

الأول Rnd لتوليد رقم عشوائي

مثال 5 Rnd يولد رقم عشوائي أقل من ٥ (بكسور)

الثاني RGB لتوليد لون بمعرفة الألوان الأساسية الأحمر R والأخضر G والأزرق B

وهي كالتالي RGB ( red, green, blue)

طيب الفكرة هي توليد الرقم العشوائي للأحمر و الأخضر و الأزرق لينتج في النهاية لون عشوائي وطبعا تكرر هذه العملية كل ٠.٥ ثانية مثلا (باستخدام تايمر) طبعا الفكرة وضحت وللتنفيذ :

ضع تايمر وغير الخاصية InterVal إلى ٥٠٠ (نص ثانية) وضع فيه الكود التالي:

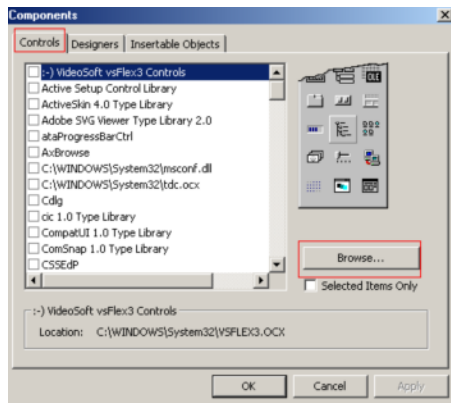
**Label1.forecolor = RGB( Rnd \* 255, Rnd \* 255, Rnd \* 255)**

يعني في كل نص ثانية يتغير لون النص إلى لون عشوائي . الفكرة واضحة بإذن الله

(٥) إضافة أدوات أخرى

أحيانا لا تكفيك الأدوات الموجودة في شريط الأدوات (زر الأمر ، مربع النص ... إلخ) يمكننا إضافة أدوات أخرى كالتالي :

بعد تشغيل البرنامج واختيار Standard EXE اضغط Ctrl + T فتظهر لك شاشة



كالتالي :

ضع علامة (صح) أمام الأداة المطلوبة لإدراجها مع الأدوات الأخرى .

طبعا تلاحظ الآن أن أمامك كم هائل من الأدوات إذا كانت لديك أداة في الكمبيوتر (تعرف مكانها) ولم تعثر عليها في هذه القائمة فاضغط زر الاستعراض أو

**Browser**

لا تكثر لعددها فلكل منها مهمة . وستطرق للحديث عن بعضها في هذا الكتاب.

رائحة المسك في تعلم الفيجوال بيسك

## ٦) إنشاء شريط أدوات

انظر إلى هذا الشرط :



هذا أحد أسطرة برنامج الورد ويسمى "شريط الأدوات (القياسي)"

وبالمثال يتضح المقال .

أعتقد أن فكرة شريط الأدوات أصبحت معروفة .

لكن هل يمكن إنشاء شريط أدوات بالفيجوال بيسك ؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟

طبعاً ممکن!

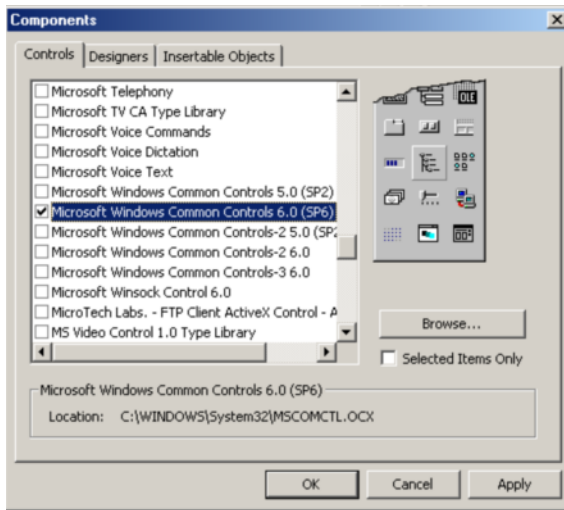
لكن يجب أن نتذكر الموضوع السابق "إضافة أدوات جديدة".

لإنشاء شريط أدوات علينا إضافة مجموعة أدوات .

تابع خطوات إضافة أداة .

اسم مجموعة الأدوات التي تلزمنا هو : Microsoft Windows Common Control 6

أضفها .



لعلك تلاحظ أنه تمت إضافة أكثر من أداة .

على كل سنحتاج لأداتين منهما فقط لإنشاء شريط

## أدوات.

## الأداتين هما:

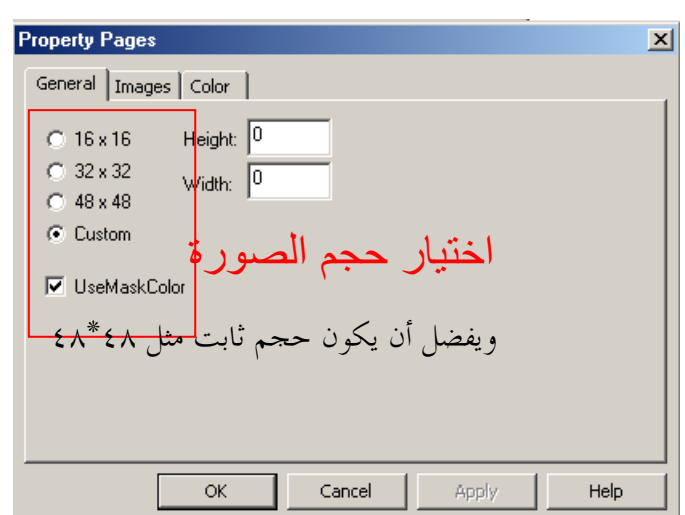
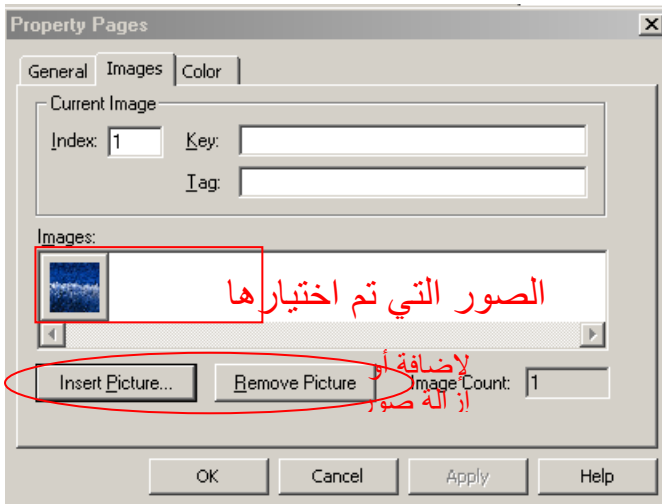
Image List (١) :وضع صور في الشريط

٢) Tool Bar : لإنشاء الشريط

## الخطوات :-

## وضع أداة ImageList و أداة ToolBar ثم

اضغط بزر الفأرة الأيمن على الأداة ImageList واختر properties تظهر لك الشاشة التالية :



الآن انقر بالزر الأيمن للماوس على الأداة ToolBar واختر properties تظهر لك الشاشة التالية :  
اختر ImageList واختر Imagelist1 كما في مثالنا :

الآن نحن ربطنا الToolBar بالImagelist1

ويمكننا استخدام الصور التي حملناها في ال

ImageList واستخدامها في الشريط

وإليك شرح عن الصورة التي في الأسفل والتي

تظهر عند اختيار التبويب Buttons

Index : رقم الزر

Caption : التسمية

Key : سنستخدمها في الكود فعليك أن تكتب

لكل زر Key مختلف عن الآخر

Image : رقم الصورة لهذا الزر وهذه الخاصية

يفضل أن يتم تحديدها لكل زر قبل البداية

بالكود .

أما بقية الخصائص فاكشفها بنفسك

الآن لدينا شريط أدوات فيه أزرار

عليها صور .

لكن إن شغلت البرنامج ولم يحصل

شيء عند الضغط على أحد الأزرار

فقط تستغرب!!!

لكن لا داعي لاستغرابك ، لأننا لم

نضع الأوامر في أزرار الأمر هذه

ولوضع الأوامر أحد نفسي مضطرا

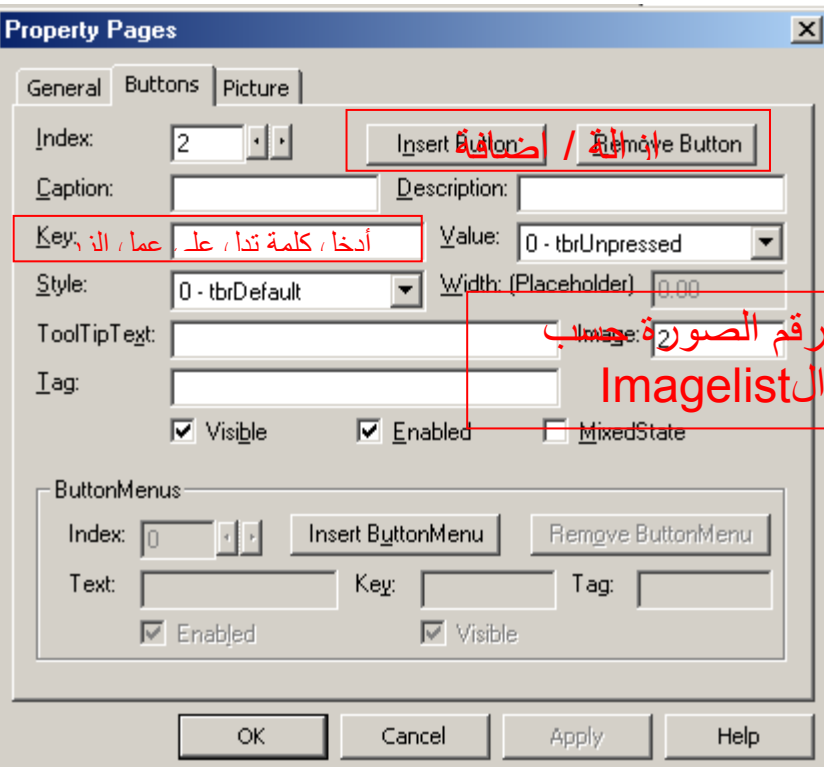
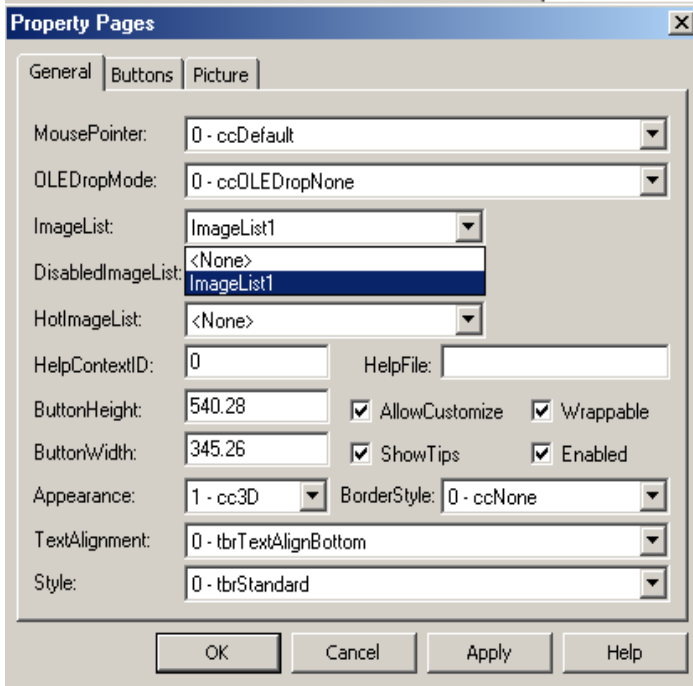
لشرح جمل Select Case

فهي التي سوف نستعملها في الأكواد

لذلك سوف أشرحها هنا شرحا بسيطا

وسوف يكون الشرح المفصل في باب

آخر من أبواب هذا الكتاب إن شاء الله.



• شرح بسيط عن جملة Select Case

سبق وذكرنا أنه عليك إعطاء الكمبيوتر الأوامر لينفذها ، وإحدى طرق إعطائه الأوامر هي جملة Select Case

ونستعمل هذه الجملة عندما نريد فحص قيمة ما ، ونريد مقارنتها بأكثر من قيمة ، وإذا تطابقت إحدى القيم ينفذ الكمبيوتر عملاً نعطيه إياه ،، أي لكل قيمة عمل مختلف وعلى الكمبيوتر أن يحدد القيمة ثم يقوم بعمله شكل الجملة :

```
Select case "القيمة المراد فحصها"
    Case is "عملية مقارنة"
        الأمر
    Case is "عملية مقارنة ٢"
        الأمر
    .....
    .....
    Case else "غير ذلك"
        الأمر في حال عدم انطباق أي مقارنة
Ens select
```

هذا وصف مختصر فقط

الكود في شريط الأدوات

سنعتمد على الخاصية Key لمقارنة أي زر قد ضُغَط من الأداة Toolbar

ولا تنسى أننا حددنا لكل زر Key مختلف

على سبيل المثال وضعت شريط أدوات فيه زرین فقط و ال Key هو "Exit" و "Msg" على التوالي  
والآن نضع الكود التالي :

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
    Case Is = "msg"
        MsgBox ""
    Case Is = "exit"
        End
End Select
End Sub
```

الآن شريط الأدوات جاهز ،،، جرب وحاول اكتشاف المزيد عنه عن طريق الملاحظة الذاتية



## رائحة المسك في تعلم الفيجوال بيسك

(٧) إضافة عرض فلاش للبرنامج :

إضافة عرض فلاش إلى برنامجك عليك أولا إضافة أداة ، وقد تعلمنا سابقا كيفية القيام بذلك.

لكن إعادة سريعة ستذكر بإذن الله

نضغط **Ctrl + T** لتظهر قائمة بالأدوات المتوفرة

نختار **ShockWave Flash** ويمكن إيجادها بسهولة إذ أن القائمة مرتبة أبجديا من **A** إلى **Z**

عند إضافتها ستظهر مع باقي الأدوات كما في الشكل :

ارسمها في مكان مناسب في النموذج (Form).

والخاصية التي نهمنا هي **Movie**

ويمكن إضافتها إما من خصائص الأداة أو من خلال الكود

فمثلا : إذا أردنا إضافة الفلم **1.swf** الموجود في **D:\** عن طريق الخصائص نكتب امتداد فلم الفلاش الذي

امتداده **\*.swf** في الخاصية **Movie** كما في الشكل

أو يمكن إضافته بالكود كالتالي :

```
Private Sub Form_Load()
```

```
ShockwaveFlash1.Movie = "D:\1.swf"
```

```
End Sub
```

ولاحظ أننا وضعناه في الحدث **Form\_Load** ليشتغل الفلم

عند بداية النموذج ، وطبعاً يمكن وضعه في زر أمر مثلا

ملاحظة : قد لا يعمل الفلم عند نقل الملف **"D:\1.swf"** إلى

موقع آخر أو عند تشغيل البرنامج على جهاز آخر .

والحل هو بوضع عرض الفلاش في نفس مجلد البرنامج ، عندها يمكن

استعمال الكود التالي :

```
ShockwaveFlash1.Movie = App.Path + "\" + "1.swf"
```

طبعاً يجب أولاً حفظ المشروع ثم وضع فلم الفلاش في نفس المجلد حتى يعمل الفلم.

وشرح الكود السابق :

**App.Path** : مسار المجلد الذي يحتوي على البرنامج مثلا ، إذا كان البرنامج في **D:\myproject**

فسيكون هذا هو **App.path**

لكن هذا ليس المطلوب لأننا نحدد مسار فلم الفلاش .

وإذا كان بنفس المجلد ، فسيكون امتداده هو امتداد المجلد + "\" + اسم الملف

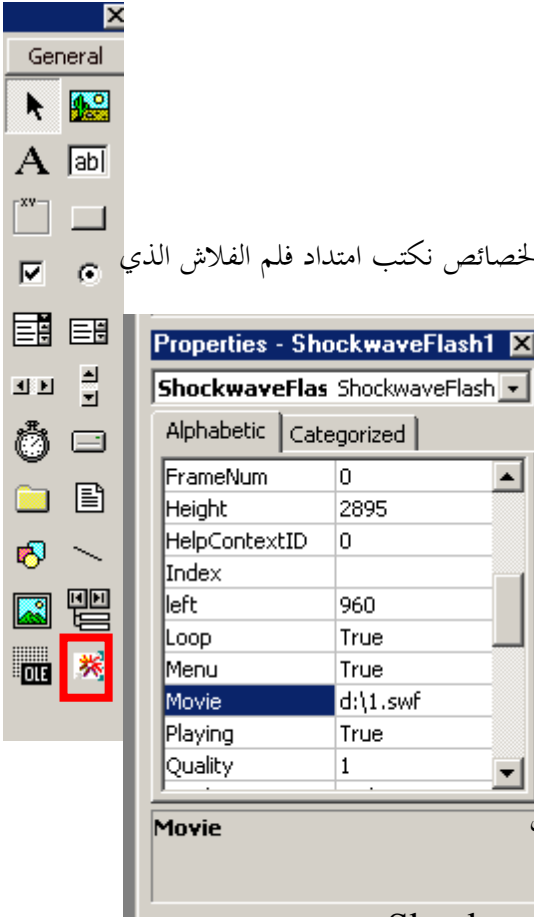
أتمنى أن يكون الأمر واضحا .

(٨) استخدام **Skin** :

لقد قمت بتأليف كتاب خاص عن هذه الجزئية وهو موجود على الرابط : [من هنا](#)

ولا أرى أن تكرار نفس الكتاب هنا ذو فائدة فأرجوا تحميل ذلك الكتاب من الرابط في الأعلى.

هذا الكتاب مجاني لا يجوز بيعه ..... تأليف : فؤاد أصيل



## رائحة المسك في تعلم الفيچوال بيسك

وفي ختام الباب الثاني لا توجد أسئلة يمكن وضعها ، فأنت إن طبقت كل ما سبق عمليا فستكون لك الفائدة بإذن الله في تنسيق برامجك بشكل أكبر .

وأعيد وأذكر بأن لغة البرمجة هذه بحر واسع وما ذكرته عن التواصل مع المستخدم ما هو إلا القليل ، ومع ازدياد خبرتك في هذا المجال لا بد أنك ستبدع طرقا ملفتة في التنسيق .

• الباب الثالث (قواعد كتابة الأكواد)  
• التعامل مع المعالج

• تمهيد

• جملة If الشرطية

• جملة If .....then....Else الشرطية

• جمل If المتداخلة

• جملة Select Case

• الحلقات التكرارية

• الحلقة التكرارية For ... Next

• الحلقة التكرارية Do

• تمارين



خلال هذا الباب : الباب الثالث سوف نحاول أن نتعلم جزءا مهما في كتابة الأكواد ،  
فلعلك لاحظت من الأبواب السابقة ماهية الأكواد وكيفية كتابتها.  
أحيانا يكون المشروع بسيط جدا كطباعة كلمة في TextBox فتكتب  
Text1.Text = "أنا مبرمج"  
هذا يبدو في منتهى البساطة .  
لكن أحيانا قد تحتاج للمقارنة ، وتشترط شرطا معيناً قد يغير مجرى الأحداث في برنامجك  
وذلك لا يتم إلا من خلال جمل الشرط التي سنشرحها بإذن الله .  
وفي بعض الأحيان تحتاج لتكرار الجمل أو العمليات فتقوم باستخدام حلقات التكرار التي  
تختصر عليك الجهد ، وتجعلك تشعر بالاحترافية .  
قد تكون في هذه اللحظة متشوق لتعلم هذه الأكواد ، لكن لا تتسرع فعليك تعلم بعض  
الأمور أولا :

• المقارنة : عليك أن تعرف أدوات المقارنة التالية في ال VB6

معامل المقارنة	معناه
=	يساوي
<>	لا يساوي
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوي
<=	أصغر من أو يساوي

لا حظ أنه عندما تستخدم المقارنة فإن الجواب هو (نعم أو لا) (Yes Or No)

وإليك جدول يوضح بعض الجمل الشرطية ونتائجها :

$$5 < 9$$

التعبير الشرطي	النتيجة
$15 < 20$	صح (١٥ لا تساوي ٢٠)
$X1 < 30$	صحيح إذا كان X1 أقل من ٣٠
$X1 = \text{Text1.Text}$	صحيح إذا كانت الخاصية Text للكائن Text1 هي نفس قيمة المتغير X1
$\text{Form1.Caption} = \text{"المشروع ١"}$	صحيح إذا كان عنوان النموذج الأول هو "المشروع ١"

كما ذكرت قبل قليل فهذه الجمل إجابتها نعم أو لا . ويمكن أن نضع جوابها في متغير من النوع Boolean عن طريق الأمر :

**Dim "اسم المتغير" As Boolean**

والآن نعود إلى فكرة تغيير مجرى الأحداث في برنامجك ولكي تغير مجرى الأحداث ستحتاج إلى أن تعرف أن التفرع نوعان هما (المشروط - سيأتي تفصيله- ، و غير المشروط)

- التفرع غير المشروط

ألا وهو جملة **Go To** وأجد نفسي مضطرا للتكلم عنها ،،، لكن اعلم أخي العزيز أ أبرز استخداماتها هو استخدامها مع الأخطاء (هذا من واقع خبرتي) و لذلك بعد دراسة هذه الجملة سنسقط ما درسناه على مثال واحد وهو (معالجة الأخطاء في البرنامج)

طبعا هنالك العديد من الاستخدامات لهذه الجملة ،،، لكننا سنأخذ معالجة الأخطاء على سبيل المثال لا الحصر.

إن كنت تعتقد أن هذه الجملة لن تفيدك أبدا ، فأنصحك بأن تراجع حساباتك !  
ملاحظة : في البرنامج نكتب الجملة هكذا (GoTo) بدون مسافة بين Go و To



سنبدأ بدراسة هذه الجملة ثم سنتقل إلى (الأخطاء ومعالجتها)

جملة Go TO:

صيغتها :

(سطر) GoTo [حدث]

سيكون المثال في دراستنا هو الخطأ ولذلك سيكون الحدث هو On Error أي : عند حدوث خطأ ،، وسنسمي السطر (١٠٠) على سبيل المثال فيصبح الكود :

On Error GoTo 100

ومعنى ذلك : عند حدوث خطأ اذهب إلى السطر ١٠٠ والحدث ليس ضروري لأنه بين أقواس من هذا الشكل [ ]

لكن أين هو السطر ١٠٠ ؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟

السطر ١٠٠ أنت تحدده ، لكن عليك توخي الحذر واستعمال الفطنة والذكاء في تسمية السطر فيجب أن يكون مكانه مناسباً !

حسنًا في مثالنا لن يحصل شيء إلا إذا حصل خطأ . لكن أليس هذا تفرعا مشروطا ؟؟  
إذ لن يتم التفرع إلا بتحقيق شرط هو (حدوث خطأ)

والجواب على هذا الإشكال أن الشرط ليس من الجملة GoTO بل من الجملة On Error

أرجو أن تكون هذه النقطة مفهومة .





وستتحدث عن الجملة On Error وكيفية التعامل مع الأخطاء من خلالها .  
جملة On Error تتحمل شكلين هما :

- 1) On Error Goto (سطر)
- 2) On Error Resume Next

• الصيغة On Error Resume Next

وهي الأبسط وتعني ببساطة : عند حدوث خطأ لا تكثر وتابع عملك  
وتستخدم كثيراً عندما لا تهتم بمعرفة المستخدم للخطأ ، فقط ضعها في أعلى الكود  
لأنها تؤثر على السطور التي تحتها حتى End Sub

• الصيغة On Error Goto (سطر)

وهي أكثر احترافية من وجهة نظري حيث يمكنك تبليغ المستخدم عن ماهية الخطأ ،  
ورقم الخطأ ، ..... إلخ  
ولكن هذا الكود (بخلاف سابقه) لن يكمل تنفيذ السطور متتبعاً ، بل سيترك عدة  
سطور ويذهب للسطر الذي تحدده له أنت  
ومن المهم أن تعرف ما معنى التالي

الأمر / القيمة	المعنى
Err.Number	رقم الخطأ
Err.Clear	إزالة الخطأ
Err.Description	وصف الخطأ

رقم الخطأ : أرقام ثابتة ، فلكل خطأ رقم خاص (لسهولة التمييز) وهذه الأرقام متوفرة على  
الإنترنت ، ولن أذكرها هنا لكثرتها

وصف الخطأ : وصف للخطأ يمكن أن تضعه في MsgBox كالتالي

MsgBox " حدث الخطأ التالي " & Err.Description, vbCritical



## رائحة المسك في تعلم الفيجوال بيسك

مثال على الخطأ :

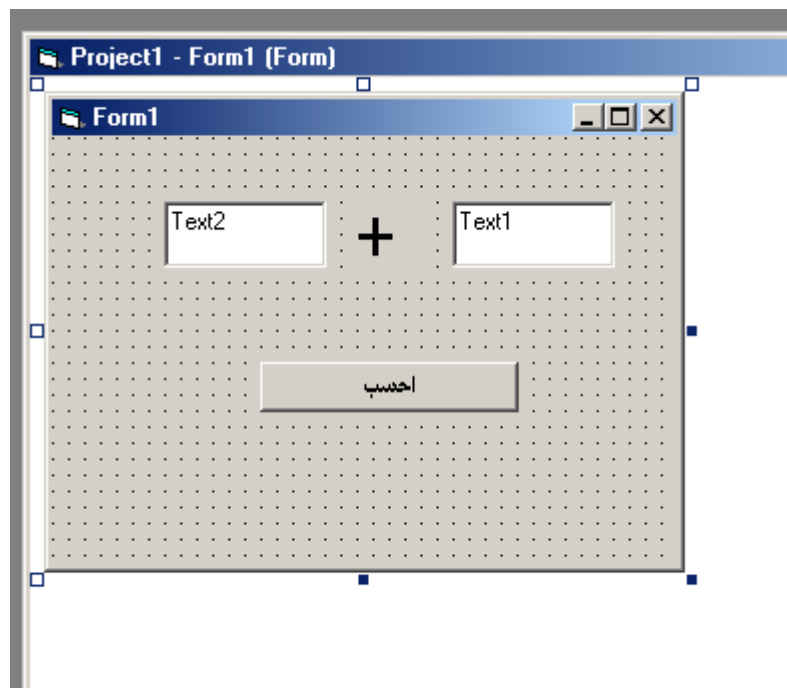
سنقوم بصناعة برنامج لجمع عددين (منتهى البساطة) ، وسيكون العددين مأخوذين من مربعي نص

Text1 و Text2

لكن سيظهر الخطأ عندما لا يقوم المستخدم بإدخال رقم (أي يدخل المستخدم أ مثلاً) وطبعاً لا يمكن أن نجمع أ + ١٥ (خاصة إذا كان المتغير من نوع (Integer) .

- أولاً : التصميم

- ضع في نموذج مربعي نص (لإدخال القيم)
- ضع أداة تسمية (Label) وغير الخاصية (Caption) إلى ("+" ) وكبر الخط
- ضع زر أمر وغير الخاصية (Caption) إلى ("احسب")
- سيبدو عملك كالتالي



:

طبعاً يمكنك التنسيق براحتك كما تعلمنا في الباب الثاني  
طبعاً قبل أن نبدأ بكتابة الكود يجب أن تكون الفكرة قد طُبخت في دماغك  
نريد أن نجمع قيمة ما في text1 و text2  
والآن ننتقل إلى الكود

- الكود

○ نضغط مزدوجاً على زر الأمر لنكتب الكود فيه

○ اكتب الكود التالي وسيأتي شرحه بإذن الله

```
On Error GoTo 100
Dim X1 As Integer, X2 As Integer, X3 As Integer
X1 = Text1.Text
X2 = Text2.Text
X3 = X1 + X2
MsgBox X3
Exit Sub
100:
MsgBox "حدث الخطأ التالي" & Err.Description, vbCritical _
+ vbMsgBoxRight + vbMsgBoxRtlReading
```

الشرح :

السطر ١ : إذا حصل خطأ توجه إلى السطر ١٠٠

السطر ٢ : نفرض ٣ متغيرات على شكل أرقام صحيحة

السطر ٣ : المتغير الأول يوضع فيه ما بداخل مربع النص الأول

السطر ٤ : المتغير الثاني يوضع فيه ما بداخل مربع النص الثاني

السطر ٥ : المتغير الثالث سوف توضع فيه قيمة المتغير الأول + قيمة المتغير الثاني

السطر ٦ : رسالة للمستخدم بقيمة المتغير الثالث (الناتج)

السطر ٧ : خروج من الإجراء (لكي لا يتابع ويظهر رسالة الخطأ)

السطر ٨ : هذا هو السطر ١٠٠

السطر ٩ : إرسال رسالة بوصف الخطأ عليها صورة (x) والمحاذاة فيها على اليمين

جرب البرنامج الآن .

أرجو أن تكون فكرة البرنامج واضحة



تكلّمنا عن التفرع الغير مشروط (goto) والآن سوف نتكلم عن التفرع المشروط باستخدام الجملة الشرطية IF وهي أنواع .  
سنتكلم عن أبسط الأنواع IF.....Then.....  
وهذه الجملة لها شكلان

- 1) جملة تنفيذية Then شرط If
- 2) If Then شرط

.....  
مجموعة جمل تنفيذية  
.....  
End If

سنتكلم الآن عن الشكل الأول  
ويستخدم إذا كانت الحالة هي (تنفيذ جملة واحدة عندما يكون الشرط نعم) وهي الجملة التي بعد  
Then أما إذا كان جواب الشرط "لا" فينتقل للجملة التي بعد If دون تنفيذ ما بعد Then  
على سبيل المثال

```
S = text1.text  
If s > 1000 then msgbox("لقد أدخلت قيمة أكبر من ١٠٠٠")  
Msgbox "شكرا لاستخدام البرنامج"
```

المثال واضح ، إذا أدخل المستخدم قيمة أصغر من أو يساوي ١٠٠٠ فستظهر رسالة واحدة  
أما إن أدخل عدد أكبر من ١٠٠٠ فستظهر رسالتين .  
أما إذا أردنا تنفيذ جملة إن لم يتحقق الشرط فنستخدم :

جملة تنفيذية أخرى Else جملة تنفيذية Then شرط If

وهذا الشكل سيتم تنفيذ ما بعد Then إن انطبق الشرط وإلا سيتم تنفيذ ما بعد Else  
أعتقد أن ذلك لا يحتاج إلى مثال . حاول ابتكار مثال بنفسك.

## الشكل الثاني ل If Then :

ويستخدم عندما تريد تنفيذ أكثر من جملة عند انطباق الشرط .  
شكلها العام

If شرط Then

.....

مجموعة جمل تنفيذية

.....

End If

مستعد على المثال السابق لتتضح الفكرة :

S = text1.text

If s > 1000 then

Msgbox "لقد أدخلت قيمة أكبر من ١٠٠٠"

PRINT "لقد أدخلت قيمة أكبر من ١٠٠٠"

END IF

Msgbox "شكرا لاستخدام البرنامج"

طبعا نفس الغرض في المثال السابق لكن سيكتب الجملة على النموذج أيضا .

أما إذا أردنا تنفيذ جملة إن لم يتحقق الشرط فنستخدم :

If شرط Then

.....

مجموعة جمل تنفيذية ١

.....

ELSE

مجموعة جمل تنفيذية ٢

End If

وسينفذ مجموعة الجمل التنفيذية ١ إذا انطبق الشرط

أما إن لم ينطبق فسينفذ مجموعة الجمل التنفيذية ٢

واعتقد أن هذا ليس بحاجة إلى مثال . اصنع مثالا بنفسك



الجمل السابقة كانت تختبر شرطا واحدا ، لكن ماذا إن كنت بحاجة لاختبار أكثر من شرط ؟  
لديك حلان ، الأول : جمل If المتداخلة والثاني Select case وسنتكلم الآن عن الأول  
لاحظ الشكل التالي

If شرط ١ then

شروط عند تحقق ١

If شرط ٢ then

شروط عند تحقق ٢

If شرط ٣ Then

الجمل التنفيذية عند تحقق ٣

Else

الجمل التنفيذية عند عدم تحقق ٣

End If

Else

شروط عند عدم تحقق ٢

End If

Else

شروط عند عدم تحقق ١

End If

نلاحظ هنا أن الشرط ٣ يتحقق عند تحقق الشرط ٢ الذي يتحقق عند تحقق الشرط ١

شكل آخر :

If شرط ١ then

الجملة الشرطية ١

Else If شرط ٢

الجملة الشرطية ٢

Else

End If

الآن لدينا مثال عن برنامج سنكتب كوده بكلتا الطريقتين :  
الفكرة :

السن	الرسالة
أكبر من ٦٥	أنت كبير بالسن
أكبر من ٣٥ وأقل من ٦٥	عمرك مناسب
أكب من ٢٥ وأقل من ٣٥	أنت شاب
أقل من أو يساوي ٢٥	أنت شاب صغير

الشكل الأول

```
If age >= 25 then
  If age > 35 then
    If age > 65 then
      Print "أنت كبير بالسن"
    Else
      print "عمرك مناسب"
    End If
  Else
    Print "أنت شاب"
  End If
Else
  Print "أنت شاب صغير"
End If
```

يبدوا هذا الكود معقدا . لكن انظر إلى الكود بتأنٍ ، سوف يوضح بإذن الله  
بقي أن تعلم أن هذا الشكل لا يستخدم كثيرا فهنالك أشكال أسهل  
كما سنرى بالشكل الثاني



## الشكل الثاني :

```
If age < 25 Then
    Print "أنت شاب صغير"
Else If age < 25 then
    Print "أنت شاب"
Else If age < 65 then
    Print "عمرك مناسب"
Else
    Print "أنت كبير بالسن"
End If
```

طبعاً يبدو هذا الشكل أسهل وأبسط بكثير من الشكل السابق لكن اعلم أنني شخصياً لكتابة مثل هذه الأكواد أستعمل الجملة **Select Case** لأنها أبسط و عملية أكثر .

وبالنهاية يبقى الخيار لك في كتابة الكود.

ولعلك استفدت أنه يمكن كتابة الكود بأكثر من طريقة ، لكن المبرمج المحترف يبحث عن الطريقة الأبسط والأقل خطأ والأسرع

وقبل الانتقال لـ **Select case** لابد من دراسة موضوع مهم ألا وهو : استعمال المعاملات المنطقية في الجمل الشرطية

- المعاملات المنطقية :

المعامل المنطقي	معناه
And	إذا كان التعبيران الشرطيان صحيحان معا كانت النتيجة صح
Or	إذا كان أحد التعبيرين الشرطيين صح فإن النتيجة صح
Not	إذا كان التعبير الشرطي صح كانت النتيجة خطأ ، وإذا كان خطأ كانت صح
Xor	إذا كان أحد التعبيرين فقط صح تكون النتيجة صح ، وإذا كان كلاهما صح أو خطأ فالنتيجة خطأ ، (متشابهان خطأ ، مختلفان صح)

لمزيد من التوضيح :



الجدول التالي يبين بعض الأمثلة عن المعاملات المنطقية حيث  $A = 200$  و  $B = "VB6"$

التعبير المنطقي	النتيجة
$B = "VB6" \text{ And } A < 500$	صح ، لأن كلا التعبيرين صحيح
$B = "C++" \text{ Or } A < 300$	صح ، لأن أحد الشرطين صحيح
$\text{Not } A < 7$	صح ، لأن الشرط خطأ
$B = "VB6" \text{ Xor } A = 200$	خطأ ، لأن كلا الشرطين صحيح

- مثال لاستعمال المعامل المنطقي مع جمل الشرط IF

فكرة المثال : طلب اسم المستخدم وكلمة المرور (ولا يوجد إلا عميل واحد)

التصميم : أضف مربعي نص (واحد لاسم المستخدم والآخر لكلمة المرور) و أضف زر أمر

(أنا لن أهتم بالتنسيق لكن أتوقع منك أن تهتم به)

الكود

```
If Text1.Text = "Name" And Text2.Text = "Pass" Then
MsgBox "True"
Me.Caption = "Hi Name"
Else
MsgBox "False"
End If
```

الكود واضح طبعاً وأنا متأكد أنك لو كنت مركز بالدروس الماضية لن يصعب الكود عليك بإذن الله.

أما الآن فسننتقل بإذن الله للحديث عن جملة Select Case



جملة select case تشبه جملة If ..... Then لكنها أكثر فاعلية عندما يعتمد التفرع على متغير رئيسي وأكثر من شرط ، أو عندما تكثر الاحتمالات .  
شكلها العام

المتغير أو الخاصية أو التعبير Select Case

قيمة ١ Case ١

جملة تنفيذية ١

قيمة ٢ Case ٢

جملة تنفيذية ٢

Case Else

جملة تنفيذية في حالة عدم انطباق أي قيمة معطاة

End Select

إذا طابقت إحدى القيم المتغير ، سيتم تنفيذ الجمل الموجودة تحت تعبيرها ويتابع الفيجوال - بعد

ذلك - تنفيذ البرنامج بعد الجملة End Select

إذا لم يطابق المتغير أي قيمة فسوف ينفذ الجمل الموجودة تحت Case Else ويتابع الفيجوال

- بعد ذلك - تنفيذ البرنامج بعد الجملة End Select

تمرين : هل تذكر جمل If المتداخلة ، لقد حلت المثال بطريقتين ، أريد منك أن تحله بواسطة

Select Case

استعمال معاملات المقارنة مع Select Case

المعاملات التي يمكنك استخدامها هي : = ، < ، > ، <= ، >= ، = .

ولا استعمالها عليك إضافة Is أو To في التعبير

عند استخدام IS فأنت تقارن مع قيمة واحدة أما عند استعمال To فأنت تقارن مع نطاق من

القيم

وإليك مثالا شاملا عن Select Case

مثال : عبر المتغير age سوف نحدد إمكانيات للشخص  
الكود

Select Case age

Case is < 13

Msgbox "أصبحت شابا"

Case 13 to 19

Msgbox "أصبحت مراهقا"

Case 18

Msgbox "يمكنك إصدار رخصة قيادة"

Case IS >= 100

Msgbox "ما شاء الله ، أنت من المعمرين"

Case Else

Msgbox "جميل أن تكون بهذا السن"

End Select

إذا كانت القيمة أصغر من ١٣ تظهر الرسالة (أصبحت شابا)  
إذا كانت القيمة بين ١٣ و ١٩ (تنتمي للنطاق) تظهر الرسالة (أصبحت مراهقا)  
إذا كانت القيمة ١٨ تظهر الرسالة (يمكنك إصدار رخصة القيادة)  
إذا كانت القيمة أكبر من أو تساوي ١٠٠ تظهر الرسالة "أنت من المعمرين"  
أما إن كانت القيمة غير ما ذكر مثل ٢٠ مثلا تظهر الرسالة الأخيرة

وبهذا نكون قد أنهينا الكلام عن التفرع . وسنتكلم فيما بعد عن حلقات التكرار

## الحلقات التكرارية

إحدى أهم ميزات الحاسب الآلي هي قدرته على التكرار بسرعة وبدون أخطاء .  
فمثلا تصور أن تقوم مدرسة ما أو جامعة بإحصاء مجموع جميع درجات الطلاب ، إن القيام بذلك  
من قبل ١٠٠ موظف سيأخذ وقتا ، ولا بد من وجود أخطاء .  
لكن الحاسب الآلي يستطيع القيام بذلك بسرعة هائلة وبدون أخطاء . وذلك عبر الحلقات التكرارية

لا بد أنك تستوعب الآن أهمية هذه الحلقات التكرارية .  
ولعل من المفيد أن تعلم أن الحلقات التكرارية تنقسم إلى قسمين :  
(١) الحلقة التي تنفذ جملة أو مجموعة من الجمل عددا معينا من المرات  
(٢) الحلقة التي تنفذ جملة أو مجموعة من الجمل عددا غير معلوم من المرات لكنها مرتبطة بشرط  
معين .

واعلم أخي العزيز أن استعمال حلقات التكرار خطر للغاية فقد تقع فيما يسمى "الدائرة المفرغة"  
والتي تتسبب في "تعليق الجهاز"

لا تظن أن حلقات التكرار صعبة ، إنها سهلة لكن تحتاج لبعض التركيز في دراستها



تسمح هذه الحلقة بتنفيذ مجموعة من الأوامر عددا محددا من المرات في الإجراء .

وتستعمل في الكثير من الحالات مثل

(١) إذا كنت تقوم بإجراء عمليات حسابية متصلة ببعضها .

(٢) الرسم مثلا .

(٣) أو كنت تعالج عدة أجزاء من النص

(٤) إلخ ....

إذا فهذه الحلقة إنما هي اختصار للائحة طويلة من الجمل .

-الصورة العامة :

[قيمة التخطي Step] النهاية To البداية = للمتغير For

جمل تنفيذية

المتغير Next

For و To و Next و = | هي مطلوبة ولا يمكن التخلي عنها.

\* يجب أن يكون المتغير رقمي الذي سيحمل القيمة التي وصلت إليها الحلقة

\*تستبدل البداية والنهاية وقيمة التخطي بمتغيرات رقمية .

طريقة العمل :

مثلا إذا أردنا طباعة الرقم ١٠ على الفورم ٤ مرات فإننا نكتب الكود التالي

For I = 1 to 4

Print "أنا مبرمج"

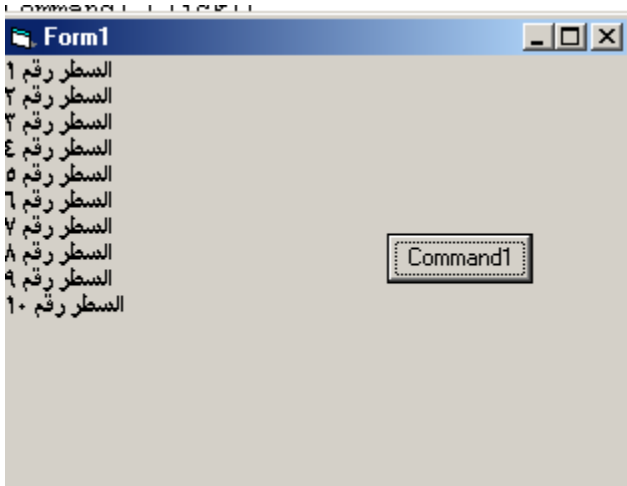
Next I

نلاحظ أننا لم نكتب Step لأنها ١

• إن جمل التكرار صعبة الفهم دون وجود أمثلة ، لذلك إليك بعض الأمثلة :

## رائحة المسك في تعلم الفيجوال بيسك

مثال : نطبع على الشاشة رقم السطر ل ١٠ أسطر كما يظهر بالصورة :



طبعا يمكن القيام بذلك بالطريقة الصعبة :

Print "السطر رقم ١"

Print "السطر رقم ٢"

.....

Print "السطر رقم ١٠"

لا بد أنك تلاحظ أن الأمر صعب هكذا مع أنك تكرر  
للعشرة فقط ، فما بالك لو للمئة أو للألف ؟؟؟؟

أما بطريقة جملة التكرار

```
For i = 1 To 10  
    Print "السطر رقم " & i  
Next i
```

سنتبع الكود :

المتغير I يبدأ من ١ وتتوالى عليه القيم حتى يصبح ١٠

في البداية القيمة ١ ، وطبع "السطر رقم " & قيمته (وهي ١)

ثم وصل إلى Next I ليعود للبداية لكن بقيمته هي (قيمته الحالية + الرقم بعد Step) ونلاحظ

عدم وجود Step إذا ١ فتصبح قيمته (١ + ١) = ٢ ثم يصل إلى Next ويعود إلى البداية ...  
إلى أن تصبح قيمته ١٠ .

أعتقد أن هذا المثال واضح

ولاحظ أنه عند عدم وجود Step فإن ما بعد Step سيكون ١

هذا أسهل مثال في حلقات التكرار .

لكن ماذا لو كان هنالك Step وبعدها عدد ما ؟؟؟؟

سندرس هذه الحالة عبر المثال التالي :-

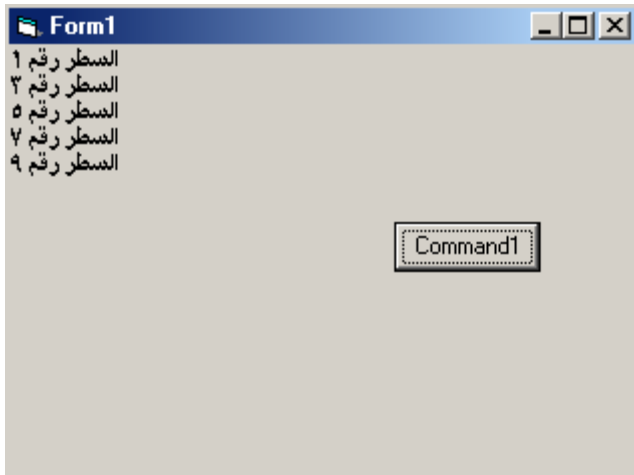


مثال / بنفس المثال السابق غير الكود إلى

```
For i = 1 To 10 Step 2
    Print "السطر رقم" & i
Next i
```

لقد جعلنا العدد بعد Step ٢

نفذ البرنامج فنلاحظ الطباعة على الفورم (كما في الشكل التالي):



تعالوا نتتبع خطوات البرنامج :

تبدأ القيمة ب ١ لأنها بين الـ (=) و الـ (TO)

الآن يطبع السطر رقم (١) وهي قيمة I

ثم يصل إلى I Next

فتصبح قيمته (القيمة الحالية + ما بعد Step)

أي (١ + ٢) ويساوي ٣ .

فيطبع السطر رقم ٣

وهكذا وهكذا

أرجو أن تكون فائدة Step أصبحت واضحة .

مثال / تغيير خاصية الـ FontSize عبر الحلقة التكرارية .

في نفس الكود السابق سوف نجعل الخاصية FontSize تساوي المتغير I

ولكن سنغير في الأرقام قليلا

إذا يصبح الكود :

```
For i = 1 To 7
    Me.FontSize = 10 + i
    Print "السطر رقم" & i
Next i
```

لاحظ أننا كتبنا ١٠ + I حتى يبان الفرق

وهذا تتضح فائدة حلقات التكرار



مثال :

```
For I = 1 To 2.5 Step 0.5  
Print I  
Next I
```

الكود التالي سيطبع على الفورم :

١  
١.٥  
٢  
٢.٥

بعد دراستك لهذه الحلقة التكرارية يمكنك تشغيل خيالك وابتكار الكثير بها .

لا تستهن بحلقات التكرار فهي توفر الكثير من العناء



حلقة Do بشكل عام تستمر إلى أن يتحقق شرط

الفروق بين الحلقة For والحلقة Do:

الحلقة For نعلم عدد مرات تكررها بينما لا نعلم ذلك بالنسبة ل Do

سنتكلم عن الحلقة التكرارية Do While (أي عندما)

صيغتها العامة :

الشرط Do While

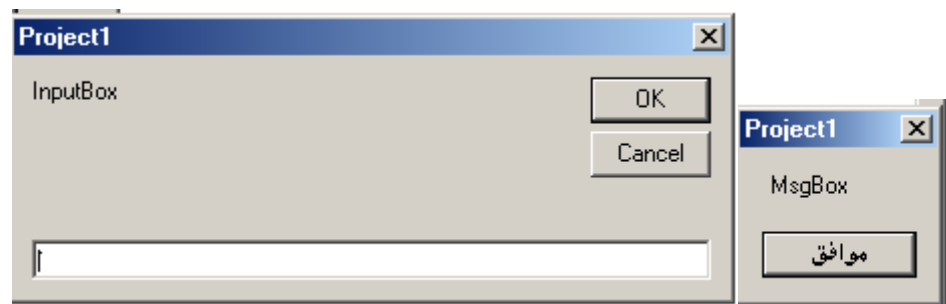
مجموعة جمل تنفيذية

Loop

ستستمر الحلقة إلى أن لا يتحقق الشرط.

لا أظن أن هذه الجملة تحتاج إلى المزيد من الشرح . ولن أطيل الكلام عنها . مثال واحد على الطائر ويمشي الحال إن شاء الله .

لكن دعنا نعرف بال InputBox لأننا سوف نستعمله بالمثال  
ال InputBox شبيه بال MsgBox إلا أن المستخدم يدخل قيمة فيه



وستعرف على طريقة أخذ القيمة من ال InputBox من خلال المثال التالي .

مثال / اكتب برنامجا يقوم بإظهار InputBox بشكل متكرر واحدا تلو الآخر حتى يكتب  
الكلمة End Inputbox استخدم داخل ال  
الكود ببساطة :

```
Do While aaa <> "End"  
aaa = InputBox(" ")  
Loop
```

لاحظ الكود وافهمه .

سنتكلم الآن عن Do Until  
الصيغة العامة

Do Until شرط  
جمل تنفيذية  
Loop

سيبقى البرنامج داخل حلقة التكرار إلى أن (يتحقق الشرط)

لاحظ الفرق بين Do While و Do Until  
سنحاول حل المثال السابق بواسطة Do until

```
Do Until aaa = "End"  
aaa = InputBox(" ")  
Loop
```

لاحظ الفرق بين الكودين .

أرجوا أن تكون حلقات التكرار واضحة ، وأن تكون الجمل الشرطية واضحة كذلك  
هذه (ما قرأته) بعض قواعد ال VB6 الذي هو بحر . لكن يكفي أن تعرف هذه الأساسيات  
لتمكن من التغلب على أي مشاكل برمجة تواجهك في حياتك العملية بإذن الله

وهذا نكون أنهينا الباب الثالث

لكن بقي التمارين . ☺

## تمارين الباب الثالث

وستكون تمارين هذا الباب غاية بالسهولة إن شاء الله

تمرين (١) وضح الفرق بين Do while و Do until

تمرين (٢) قم بكتابة برنامج يطبع الأرقام الزوجية فقط من ١ إلى ١٠٠ ، ثم قم بكتابة برنامج يكتب الفردية

تمرين (٣) باستعمال الدالة Rnd قم بكتابة برنامج يختار من الأرقام من ١ إلى ١٠٠ عشوائيا حتى يظهر الرقم ٥٥ ، و يقوم بطباعة عدد المرات الخاطئة (التي لم يظهر فيها العدد)

يكفي تمارينا ، هذا الباب الثالث فما أصبت فمن الله وحده لا شريك له ولا حول لي بذلك ولا قوة وما أخطأت فمن الشيطان ونفسي والله ورسوله بريثان من خطئي .

## •الباب الرابع (قبل أن تبرمج)

• تمهيد

• المتغيرات والثوابت

• استخدام ال Module

• نماذج لبرامج



قبل أن تطلق على نفسك كلمة مبرمج اقرأ هذه السطور :  
إن كنت تظن أنه ولأنك قرأت كتابا عن أساسيات ال VB6 أنك أصبحت مبرمجا محترفا فأنا أقول لك تواضع ، لأنك لم تر شيئا بعد ، فهذا الكتاب يتكلم عن الأساسيات ، أما إن أردت التعمق ، ومتابعة التعلم لهذه اللغة فستجد الكثير ، لكنني لا أنصحك بذلك ! .  
أنصحك بأن تنتقل إلى ال VB.net فهي تحمل المستقبل .  
والحديث عن مقارنة هاتين اللغتين طويل ، لكن يكفيك أن الأجيال القادمة من أنظمة التشغيل لن تدعم البرامج المكتوبة بال VB6 .

إذا ، حاول تعلم ال VB.net لكن لا تعتقد أن ما قرأته بهذا الكتاب ذهب هباءً ، فالآن أنت تعرف أساسيات ال VB6 وهذا سيسهل عليك كثيرا فهم لغة ال VB.net  
المهم ،،،، يتكلم هذا الباب الأخير عن أمور مهمة يجب أن تعرفها قبل أن تبدأ مسيرتك البرمجية .  
فأرجو منك أن تولي هذا الباب اهتمامك ، وألا تستخف بأي شيء .  
لكن لا مانع من أن تعلم أن هذا الباب سهل جدا وصغير أيضا (إلا جزئية نماذج برامج) لأننا سنحاول أخذ بعض البرامج البسيطة بالتفصيل .





المتغير : مساحة (حيز) يتم حجزها في ذاكرة الحاسب RAM لتخزين البيانات مؤقتا في فترة تشغيل برنامجك ويمكن أن يتغير أثناء سير البرنامج.

الثابت : نفس الشيء لكنه لا يتغير .

شروط تسمية المتغيرات :

(١) أن يبدأ اسم المتغير بحرف إنجليزي

(٢) ألا يحتوي على مسافات أو نقط

(٣) ألا يكون محجوزا (إما لمتغير آخر أو كلمة محجوزة مثل For)

(٤) ألا يزيد طوله عن ٢٥٥ حرف

أنواع المتغيرات

نوع المتغير	معناه
Integer	عدد صحيح (صغير نسبيا)
Long	عدد صحيح (طويل نسبيا)
Single	عدد عشري (مع فواصل)
Double	عدد عشري مع فواصل لكن طويل نسبيا
String	نص
Boolean	منطقي (نعم/لا)
Date\Time	وقت/تاريخ
Variant	متنوع (يمكن يكون أي شيء)

الإعلان عن متغير

الشكل العام :

نوع المتغير As اسم المتغير Dim

و يمكن تغيير Dim لتوسيع مدى المتغير



## مدى المتغير

(١) متغير عام ، يمكن استعماله في أي مكان من البرنامج (ويكون داخل موديل)

نوع المتغير As اسم المتغير Public

(٢) متغير على مستوى النموذج (داخل النموذج الذي عرف منه) ويتم تعريفها في ال General

نوع المتغير As اسم المتغير Dim

(٣) متغير على مستوى الإجراء (يتم تعريفه داخل الإجراء)

نوع المتغير As اسم المتغير Dim

العمليات على المتغيرات (القيم)

أولا /الأولية الحسابية .

(١) الأسس والأقواس

(٢) الضرب و القسمة

(٣) الجمع و الطرح

يعني الكمبيوتر يبدأ بالبحث عن الأقواس و الأسس ويحلها قبل أي شيء ثم الضرب ثم الجمع .

ثانيا / رموز العمليات الحسابية في ال VB6

العملية	الرمز
الجمع	+
الطرح	-
الضرب	*
القسمة	/
باقي القسمة	Mod
الأس	^
وهذه ليست للأرقام بل للنصوص (ضم سلاسل نصية مع بعض)	&

## رائحة المسك في تعلم الفيجوال بيسك

بعد أن أخذنا الرموز سنحاول حل هذه المسألة كما يحلها الكمبيوتر (الأولويات)

$$10 + 15 * 2 / 4 ^ 2$$

$$= 10 + 15 * 2 / 16 \text{ (نفذنا الأس أولا)}$$

$$= 10 + 30 / 16 \text{ (وحتى لو سويننا القسمة أول لن يختلف الحل ، لكن نأخذها من اليسار لليمين)}$$

$$= 10 + 1.875$$

$$= 11.875$$

أرجو أن تكون المتغيرات واضحة للجميع .

الآن نتقل للشوابت

طريقة تعريف النوابت

الصيغة العامة :

Const قيمته = اسم الثابت

مثال : نعرف ثابت الجاذبية الأرضية وهو يساوي ٩.٨

Const G = 9.8

ويمكنك استعماله في البرنامج

يكفينا حديثا عن المتغيرات والشوابت .

هيا بنا ننطلق للحديث عن الموديل

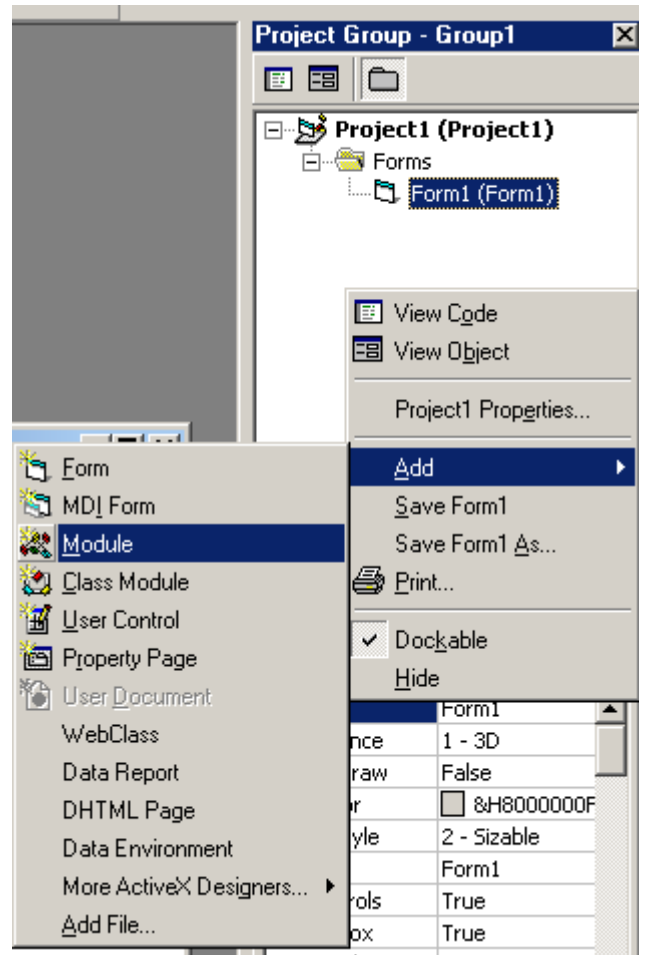
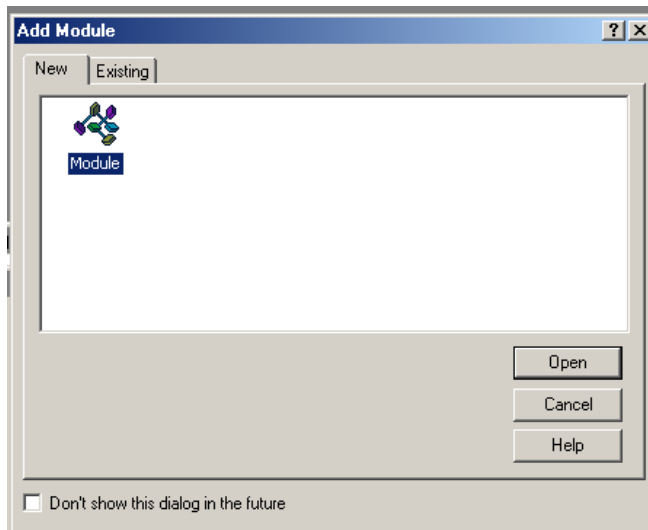
## الموديل (Module)

نستخدم الموديل عدة استخدامات أهمها تعريف متغير للبرنامج بالكامل .  
ويمكن استخدامها كـ Sub Main (لكن هذا ليس محور الحديث)

• كيفية إضافة موديل

(١) شغل الفيجوال

(٢) اتبع الصور



تجد عند ذلك أن الموديل قد أضيفت  
هذه المعلومات كافية عن الموديل بإذن الله .



نبدأ على بركة الله

سوف نطبق ما تعلمناه على صناعة برامج (بسيطة) لكي نتعود على ذلك !

(١) برنامج حساب مساحة المستطيل (علما أن مساحة المستطيل = الطول × العرض)

أنشئ مشروع جديد

ضع الأدوات التالية :

زر أمر (ليقوم بالحساب)

مربعي نص (واحد للطول وواحد للعرض)

تسمية Label لكتابة الناتج فيه .

سوي التنسيق على كيفك .

الكود :

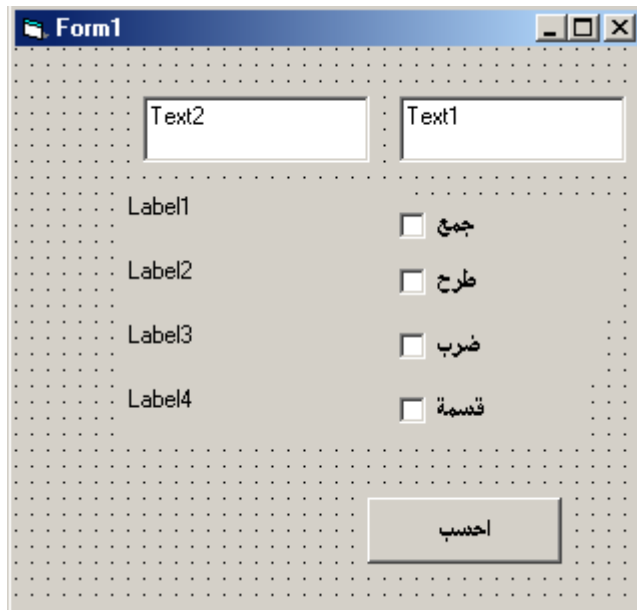
```
Private Sub Command1_Click()  
On Error GoTo 100  
Dim Toll As Single  
Dim ard As Single  
Dim msa7a As Single  
Toll = Text1.Text  
ard = Text2.Text  
msa7a = Text1.Text * Text2.Text  
Label1.Caption = " المساحة هي " & msa7a  
Exit Sub  
100:  
MsgBox "حدث الخطأ التالي" & Err.Description, vbCritical  
End Sub
```

لاحظ أننا استخدمنا : المتغيرات ، تلافي الأخطاء ، العمليات الحسابية ، إسناد القيم ، التعديل

في خصائص الأدوات . (الأكواد كلها مكررة وما في داعي نشرح)

٢) آلة حاسبة بسيطة (منقول من كتاب الحاسب للصف الثالث الثانوي في السعودية (بتصرف)

الأدوات : كما في الصورة :



الكود

```
Private Sub Command1_Click()  
Dim a As Single, b As Single  
a = Text1.Text  
b = Text2.Text  
If Check1.Value = 1 Then Label1.Caption = a + b  
If Check2.Value = 1 Then Label2.Caption = a - b  
If Check3.Value = 1 Then Label3.Caption = a * b  
If Check4.Value = 1 Then Label4.Caption = a / b  
End Sub
```

أيضا الكود واضح وما فيه مشاكل بإذن الله

حاول أن تحل الأمثلة السابقة بنفسك لتستفيد أكثر

عند وجود أي استفسار يرجى زيارة منتدى فيجوال بيسك العرب VB4ARAB.com

و اسأل هناك وبإذن الله ستجد الإجابة

مع تحيات أخوكم Aleppo Force

رائحة المسك في تعلم الفيض واليسك

تم بحمد الله