PHP & MySQL

كتاب مفصّل لتعليم لغة الـ بي إتش بي
و ربطها مع الـ ماي إس كيو إل

إعداد وتدقيق:
معتز بالله حاكمي

محتويات الكتاب:

الدروس المتقدمة

الدرس 06 : الجلساتSessions تستخدم متغيرات الجلسات session لحفظ المعلومات حول أو تغيير الضبط للمستخدم . تحمل متغيرات session معلومات حول مستخدم واحد كما أنها تكون متوفرة في جميع الصفحات وبتطبي واحد.

الدرس 07 : إرسال البريد الالكتروني تسمح PHP بإرسال بريد الكتروني مباشرة من خلال كود معين.

الدرس 08 : البريد الالكتروني المحمي هناك نقطة ضعف في انشاء البريد الالكتروني لكن سيتم حل هذه المشكلة في هذا الدرس .

الدرس 09 : معالجة الأخطاء معالجة الأخطاء الافتراضي في PHP بسيط جداً . يتم ارسال رسالة الخطأ بإسم الملف أو رقم السطر أو شرح الخطأ في رسالة الى المتصفح.

الدرس 10 : فلاتر (تصفيات)PHP لفلترة و جعل البيانات شرعية التي تأتي من جهات غير محمية مثل PHP تستخدم فلاتر مدخلات المستخدم .

استخدام PHP معMySQL

الدرس 01 : مقدمةMySQL تعتبر لغة MySQL أشهر نظام قواعد بيانات مفتوح المصدر.

الدرس 02 : الاتصال مع قاعدة البيانات MySQL تستخدم عادة MySQL مع لغة PHP.

الدرس 03 : إنشاء قاعدة بيانات و الجداول تحتوي قاعدة البيانات على جدول أو اكثر.

الدرس 04 : إدراج قيمة الى قاعدة البيانات تستخدم INSERT INTO لإدراج صف جديد في الجدول.

الدرس 05 : الاختيار SELECT يستخدم التعبير SELECT لاختيار البيانات من قاعدة البيانات

الدرس 06 : استخدام Where تستخدم العبارة WHERE لفلترة الصفوف في جداول قاعدة البيانات.

الدرس 07 : الترتيبOrder By تستخدم الترتيب ORDER BY لترتيب البيانات المعروضة من قاعدة البيانات.

الدرس 08 : تحديث القاعدةUpdate تستخدم خاصية التحديث UPDATE للتعديل على البيانات في الجدول.

الدرس 09 : الحذفDelete يستخدم الحذف DELETE لحذف بيانات في صفوف الجدول.

أساسيات PHP

الخطوة الأولى

تعتبر لغة PHP أداة مميزة لإنشاء صفحات انترنت ديناميكية وأكثر تفاعلية .

يتم استخدام لغة PHP بشكل واسع و هي مجانية و لغة بديلة فعالة ومنافسة للغات برمجية أخرى مثل Microsoft ASP .

في هذا الدورة ستتعلم الكثير عن PHP وكيفية إنشاء الأكواد على السيرفر الشخصي لديك .

ننصح وبشدة لكتابة و تجريب الأكواد استخدام برامج تحرير نصوص PHP مثل برنامج نوت باد بلاس بلاس Notepad++ و هو برنامج مجاني و جميل وأيضاً برنامج أدوبي دريم ويفر Adobe Dreamweaver

مقدمة عامة

تعتبر لغة PHP لغة برمجية تتعامل مع السيرفر .

تاريخ لغة PHP

بي إتش بي (PHP: Hypertext Preprocessor، 'الصفحة الرئيسية الشخصية': "المعالج المسبق للنصوص الفائقة") هي لغة برمجة نصية صممت أساسا من أجل استخدامها لتطوير وبرمجة تطبيقات الويب. كما يمكن استخدامها لإنتاج برامج قائمة بذاتها وليس لها علاقة بالويب فقط. بي إتش بي لغة مفتوحة المصدر ويطورها فريق من المتطوعين تحت رخصة PHP، تدعم البرمجة كائنية التوجه وتركيبها النحوي يشبه كثيرا التركيب النحوي للغة السي هذا بالإضافة إلى أنها تعمل على أنظمة تشغيل متعددة مثل لينكس وويندوز.

PHP/FI

ظهرت php أولا في سنة 1995 على يد راسموس ليردورف (Rasmus Lerdorf) كانت تسمى وقتها بـ PHP/FI وفي الحقيقة لم تكن لغة برمجه وقتها وانما كانت مجموعه من التطبيقات التي كتبت باستخدام لغة Perl أطلق راسموس اسم Personal Home Page Tools على هذه التطبيقات، لانه احتاج فيما بعد إلى تطبيقات أكثر فائده قام راسموس بكتابة تطبيق أكبر باستخدام لغة C حيث أصبحت قادرة على الاتصال بقواعد البيانات كما أنها كانت تسمح للمستخدمين بتطوير تطبيقات مواقع ديناميكيه بسيطه، اختار راسموس ان تكون الشيفرة المصدرية الخاصة بـ PHP/FI متوفره للجميع لذا كان يمكن لأي شخص ان يستخدمها ويقوم بتحسينها والمشاركة في حل اخطاءها ومشاكلها.
كانت PHP/FI وقتها تحوي على بعض الوظائف المتوفرة بالإصدارات الحالية من اللغة، كما أن المتغيرات كانت تشبه متغيرات Perl، وكانت تركيبها النحوي يشبه Perl بالرغم من بساطتها ومحدودياتها.
في عام 1997 تم إطلاق الإصدار 2.0 من PHP/FI، حيث بلغ عدد مستخدميها آنذاك 50,000 نطاق، وكان هناك مجموعه من الأشخاص الذين يشاركون في التطوير، وتم إطلاق الإصدار الرسمي من 2.0 في شهر نوفمبر من نفس العام بعد العديد من الإصدارات التجريبية بيتا.

PHP 3

في عام 1997 تم إعادة كتابة PHP/FI على يد زيف سوراسكي وأندي جتمانز بعدما وجدوا ان PHP/FI 2.0 ليست قويه بما فيه الكفاية من أجل كتابة تطبيق تجارة إلكترونية والذي كانوا يعملون عليه كمشروع تخرج لجامعتهم، كان هناك تعاون بينهم وبين مؤسس اللغة راسموس ليردورف على ان تكون PHP 3.0 هي النسخة الرسمية بعد PHP/FI.
أحد أهم الميزات التي تميزت بها PHP 3 عن سابقتها انها أصبحت قابلة للتوسع وتوفر مع هذا الإصدار العديد من المكتبات والدوال، وادت قابلية التوسع إلى إقبال العديد من المطورين على تطوير المكتبات الجديدة واضافتها مع اللغة، ويقال ان هذا هو السبب الأساسي للنجاح الذي حققه هذا الإصدار، ومن الميزات الأخرى التي تمت اضافتها في هذا الإصدار هي البرمجة كائنية التوجة.
وفي هذا الإصدار تم تغيير اسم اللغة ووضعها تحت اسم جديد وهو PHP (بدون FI) والذي كان يحمل اختصارا لمعنى جديد مختلف عن الإصدار السابق والمعنى هو "PHP: Hypertext Preprocessor".
تم إطلاق PHP 3.0 في يونيو 1998 بعد 9 أشهر من الاختبارات.

PHP 4

في عام 1998، وبعد الانطلاق الرسمي لـ PHP 3.0 بقليل بدأ زيف سوراسكي وأندي جتمانز بإعادة كتابة أساس لغة PHP وكان الهدف من ذلك هو تحسين الأداء للبرامج المعقدة والضخمة وتحسين قابلية اللغة للتوسع.

المحرك الجديد الذي ظهر بعد إعادة الكتاب تم تسميته بـ محرك زيند (بالإنجليزية: ZEND) واسم "زيند" مأخود من أوائل حروف أسماء مطوريه، ونجحوا في تحقيق اهدافهم عن طريق هذا المحرك، وتم الإعلان عنه في عام 1999.

تعتمد PHP 4.0 على هذا المحرك وتم الإعلان عن هذا الإصدار من PHP في مايو 2000، بالإضافه إلى تحسين الأداء في هذا الإصدار احتوى كذلك على مجموعه جديده من الميزات مثل دعمه لعدد أكبر من خوادم الوب، الجلسات، طرق آمنه جديده لمعالجة دخل المستخدم وغيرها.

وقد تم إعلان إيقاف تطوير ودعم PHP 4 في 13 يوليو، 2007.

المصدر ويكيبيديا

ماذا ينبغي عليك أن تعرف ؟

قبل البدء بهذه اللغة يجب أن تكون على معرفة باللغات التالية :

- HTML/XHTML
- JavaScript

إن أردت دراسة هذه اللغات يمكنك التوجه الى صفحة المعهد الرئيسية .

ماهي لغة PHP ؟

- **ان PHP هي اختصار HypertextPreprocessor :PHP أي إعادة معالجة النصوص التشعبية .**
- **ان لغة PHP هي لغة برمجية تتعامل مع السيرفر مثل لغة ASP .**
- **يتم إنشاء و استدعاء أكواد PHP على السيرفر .**

- تدعم لغة PHP العديد من قواعد البيانات مثل (MySQL و Informix و Oracle و Sybase و Solid PostgreSQL و Generic ODBC) .
- ان لغة PHP هي لغة ببرمجية مفتوحة المصدر .
- ان لغة PHP مجانية التحميل و الاستخدام .

ماهو ملف PHP ؟

- يحتوي ملف PHP على نصوص و وسوم HTML و أيضاً سكربتات و أكواد أخرى .
- تظهر نتيجة كود PHP كنص عادي .
- ان امتداد ملف PHP هو php. و php3. و phtml.

ماهي MySQL ؟

ان لغة MySQL هي لغة قواعد بيانات .

- تعتبر لغة MySQL لغة مثالية لكل من التطبيقات الصغيرة والكبيرة .
- تدعم MySQL لغة SQL .
- تستجيب لغة MySQL لعدد من اللغات البرمجية .
- ان لغة MySQL مجانية التحميل و الاستخدام .

PHP + MySQL

- يتم استخدام PHP مع لغة MySQL لإنشاء منصة و قاعدة يمكن العمل منها سيرفر محلي و أيضاً يمكن أن يعمل على ويندز و أيضاً على يونكس Unix .

لماذا PHP ؟

- تعمل PHP على أرضيات و برامج مختلفة مثل ويندوز , لينوكس و يونكس .

- تعمل PHP على جميع أنواع السيرفرات المستخدمة حالياً مثل Apache و IIS .
- ان لغة PHP مجانية الاستخدام و التحميل من الموقع الرسمي www.php.net
- ان لغة PHP سهلة التعلم و الاستخدام و تعمل على أكواد السيرفر .

من أين تبدأ ؟

للحصول على صلاحية تمكنك من استخدام سيرفر خاص على الانترنت كالتالي :

- اما تنصيب السيرفر Apache أو IIS على سيرفر شخصي وتنصيب PHP و MySQL .
- أو إيجاد خطط هوست حيث تكون جاهزة و داعمة للغة PHP و MySQL .

تنصيب PHP

ماذا تحتاج ؟

اذا كان السيرفر لديك يدعم PHP عندها لا تحتاج لتنصيب أو فعل أي شيء .

فقط يمكنك إنشاء ملفات php في حافظة الملفات لديك في السيرفر و سيتم تحويل و قراءة الأكواد وعرضها على المتصفح .

أما اذا كان متصفحك لا يدعم PHP عندها يجب تنصيب لغة PHP على السيرفر .

من هنا يمكنك الذهاب الى الموقع الرسمي للغة PHP مع شرح كامل حول تنصيب PHP5 :

http://www.php.net/manual/en/install.php

ان تنصيب PHP على السيرفر ليس بالأمر السهل ويحتاج لعدد من المبرمجين لتنصيبه لذلك لن نغوص في كيفية التنصيب إنما في كيفية إنشاء موقع و صفحات بلغة PHP .

تحميل PHP

لتحميل PHP مجاناً من خلال الرابط :http://www.php.net/downloads.php

تحميل قواعد البيانات MySQL

لتحميل MySQL مجاناً من خلال الرابط :http://www.mysql.com/downloads/

تحميل السيرفر أباتشي Apache Server

لتحميل السيرفر أباتشي من خلال الرابط : http://httpd.apache.org/download.cgi

التركيبة

تعمل أكواد PHP على السيرفر و تظهر النتيجة على المتصفح على شكل نصوص HTML .

التركيبة الرئيسية

تبدأ أكواد PHP دائماً من خلال الوسم <?php وتنتهي بالشكل?> كما يمكن ادراج أكواد PHPP في أي مكان تريده في صفحة الويب لديك .

بعض السيرفرات التي تحتوي على اختصارات لأكواد PHP يمكن البدء بوسم الكود <? والانتهاء ?> .

لكن نحن ننصح بشدة استخدام الوسم الرسمي للكود تجنباً لمشاكل السيرفرات ألا وهي <?php

```php
<?php
?>
```

يحتوي ملف PHP على وسوم HTML تماماً مثل صفحات HTML لكن يحتوي أيضاً على أكواد خاصة PHP .

مثال لصفحة وب تحتوي على كود PHP بسيط ومرفقاً بالنص "مرحباً بكم" الذي سيظهر على المتصفح .

```php
<html>
<body>
<?php
echo "مرحباً بكم";
?>
</body>
</html>
```

في نهاية كل كود PHP يجب أن يحتوي على فاصلة منقوطة أي ; كما تعتبر هذه الفاصلة المنقوطة هي العلامة التي تفصل بين الأكواد وذلك لسهولة قراءتها.

يمكن اظهار الكود على المتصفح من خلال التعابير المستخدمة في PHP مثل gecho . gprint في المثال السابق استخدمنا echo لإظهار النص ألا و هو "مرحباً بكم. "

ملاحظة : يجب أن يكون الملف هو PHP وامتداده .php أما اذا كان كود PHP موجوداً في صفحات html. فإن الكود لن يعمل.

التعليقات و الملاحظات فيPHP

نستخدم في // PHP من أجل تعليق ضمن السطر كما يمكن استخدام /* و */ من أجل تعليق متعدد الأسطر تماماً كما في JavaScript.

مثال

```
<html>
<body>
<?php
// تعليق بسطر واحد/*
تعليق بأكثر من
سطر
*/
?>
</body>
</html>
```

المتغيرات Variables

يستخدم المتغير لحفظ المعلومات .

المتغيرات في PHP

تستخدم المتغيرات لحفظ قيم مثل النصوص و الأرقام و المصفوفات .

يمكن استخدام المتغير عند إنشائه في أي مكان و عدة مرات في صفحات الويب .

جميع متغيرات PHP تبدأ بالإشارة $.

أفضل و أصح طريقة لإنشاء متغير هي كالتالي :

```
$var_name = value;
```

ينسى بعض المبرمجون الجدد وضع إشارة $ قبل اسم المتغير و عند ذلك لن يعمل الكود.

الآن سنقوم بإنشاء متغيرين الأول يحتوي على نص و الثاني يحتوي على رقم.

```
<?php
$txt="Hello World!";
$x=16;
?>
```

لغة PHP متراخية

باستخدام لغة PHP لست بحاجة لإعلان المتغير قبل غضافة القيمة اليه .

في المثال السابق لست بحاجة لتعريف المتغير فيما اذا نوع البيانات الموجودة فيه نص أو أرقام أو غير ذلك لأن لغة PHP تقوم تلقائياً بتحديد نوع بيانات المتغير فور استخدام القيمة المرفقة مع المتغير .

في لغات البرمجة المتشددة يجب عليك تحديد نوع المتغير قبل إضافة البيانات اليه بينما لغة PHP تعتبر لغة متسامحة و متراخية أي يتم تحديد نوع المتغير عند اضافة القيمة اليه .

قواعد اختيار اسم المتغير

- يجب أن يبدأ اسم المتغير بحرف انكليزية صغير أو إشارة _
- يجب أن يحتوي اسم المتغير على أحرف صغيرة أو ارقام (0-9, A-Z, a-z, _)
- لايجب أن يحتوي اسم المتغير على مسافة فاصلة بين الأسماء كما يمكن تفريق الكلمات باستخدام _ (my_string$) أو من خلال الأحرف الكبيرة لأول حرف من كل كلمة (myString$) .

String المتغيرات النصية

تستخدم المتغيرات النصية لحفظ و تعديل النصوص .

المتغيرات النصية في PHP

تستخدم المتغيرات النصية للقيم التي تحتوي على أحرف و كلمات .

سنطلع في هذا الدرس على أكثر functions و operators المستخدمة كنصوص strings في PHP .

يمكننا التعديل على النص الموجود ضمن المتغير بعد إنشائه كما يمكن استدعائه مباشرة أو يمكن حفظه و التعديل عليه لاحقاً .

في المثال التالي تم إنشاء متغير يحتوي على قيمة نصية و قد تم استدعائه ليتم عرض النتيجة على المتصفح .

```php
<?php
$txt="مرحباً بكم";

echo $txt;

?>
```

ستظهر النتيجة على المتصفح:

مرحباً بكم

ربط الأكواد

يوجد رابطة واحدة مستخدمة مع PHP لربط الأكواد معاً و هي (.) و تستخدم لربط قيمتين نصيتين مع بعضهم.

مثال

```php
<?php
$txt1="مرحباً بكم في";

$txt2="موقع معتز للبرامج";

echo $txt1 . " " . $txt2;

?>
```

ستظهر النتيجة:

مرحباً بكم في موقع معتز للبرامج

ان نظرنا للكود في المثال السابق سنشاهد أننا استخدمنا رابطة لربط بين المتغيرين . و أيضاً قمنا بإضافة قيمة ثالثة و هي مسافة بين الكلمتين.

الكود الوظيفي strlen()

يستخدم الكود الوظيفي strlen()لتحديد طول النص أي عدد أحرف النص المستخدمة.

مثال

```
<?php
echo strlen("Hello world!");

?>
```

ستظهر النتيجة على المتصفح:

12

يستخدم عادة هذا الكود الوظيفي مع الحلقات أو بعض الأكواد الوظيفية الأخرى خاصة عندما يكون من المهم معرفة متى ستتوقف الحلقة.

الكود الوظيفي strpos()

يستخدم الكود الوظيفي strpos() للبحث عن نص أو أحرف ضمن النص الواحد .

ان تم إيجاد النتيجة سيتم عرض مكان الكلمة أو الحرف عند أو مطابقة للنتيجة . و لكن إن لم يتم إيجاد نتيجة عندها ستكون النتيجة. FALSE

سنقوم الآن بالبحث عن الكلمة "world" ضمن النص:

```
<?php
echo strpos("Hello world!","world");

?>
```

ستظهر النتيجة على المتصفح:

6

ان موضع الكلمة في النص هو 6 في المثال السابق . و السبب بأنها 6 وليست 7 بأن الحرف الأحرف من النص يبدأ بالرقم 0 و ليس 1.

الرموز الحسابية في PHP

تستخدم الرموز الحسابية لاضافة او تعيين عناصر معينة

الرموز الحسابية في PHP

تستخدم PHP العديد من الرموز الحسابية .

Arithmetic Operators – الرموز الحسابية الجبرية

الرمز	الشرح	المثال	النتيجة
+	جمع	x=2 x+2	4
-	طرح	x=2 5-x	3
*	ضرب	x=4 x*5	20
/	قسمة	15/5 5/2	3 2.5
%	الباقي – باقي القسمة	5%2 10%8 10%2	1 2 0
++	إضافة – إضافة 1	x=5 x++	x=6
—	نقص – ناقص 1	x=5 x-	x=4

Assignment Operators رموز التعيين الحسابية

الرمز	المثال	تماماً مثل
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

رموز المقارنةComparison Operators –

الرمز	الشرح	مثال
==	مساوٍ لـ	خاطئ x==8 صحيح x==5
===	تماماً مساوٍ لـ – القيمة والنوع	صحيح x===5 خاطئ x==="5"
!=	غير مساوٍ لـ	صحيح x!=8
>	أكبر من	خاطئ x>8
<	أصغر من	صحيح x<8
>=	أكبر من أو يساوي	خاطئ x>=8
<=	أصغر من أو يساوي	صحيح x<=8

تستخدم الرموز المنطقية لتحديد علاقة منطقية بين المتغيرات والقيم.

لنفترض أن x=6 وy=3 في الجدول التالي لشرح الرموز المنطقية:

الرموز المنطقيةLogical Operators –

الرمز	الشرح	مثال
&&	and / و	صحيح (x < 10 && y > 1)
||	or / أو	خاطئ (x==5 || y==5)
!	not / ليس	صحيح !(x==y)

If...Else تعابير اذا .. غير ذلك

تستخدم التعابير الشرطية لإنشاء أوامر مختلفة تعتمد على شروط مختلفة .

التعابير الشرطية

عندما تكتب كود معين غالباً ماتريد إنشاء أمر معين حسب شروط مختلفة ولهذا يمكنك استخدام خاصية التعابير الشرطية لفعل ذلك وسنتعلم كيف في الأمثلة .

لدينا 4 تعابير شرطية وهي :

- **تعبير if : يستخدم هذا التعبير لإظهار كود معين في حال كانت الحالة صحيحة تماماً .**
- **تعبير if ... else : تعبير اذا وغير ذلك : يمكن استخدام هذا النوع من الكود الشرطي اذا كانت حالة الشرط الأول صحيحة والثانية خاطئة .**
- **تعبير if ... else if....else : يستخدم هذا التعبير مع واحد من مجموعة أكواد أي يستخدم عند تحقق شرط معين واذا لم يتحقق سيتم إظهار شرط آخر واذا لم يتحقق سيتم تلقائياً تطبيق الكود عند else .**
- **تعبير switch : يستخدم هذا التعبير لإختيار كود من مجموعة أكواد .**

تعبير If

يستخدم هذا التعبير لإظهار كود معين في حال كانت الحالة صحيحة تماماً .

التركيبة

```
if (condition)
```

الكود الذي سيعرض اذا كانت النتيجة صحيحة;

اذا كان اليوم جمعة "اليوم هو جمعة"سيعرض المثال التالي :

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri") echo "اليوم هو جمعة";
?>

</body>
</html>
```

في هذا المثال . ستظهر نتيجة الكود فقط عندما تكون النتيجة elseلاحظ أننا لم نستخدم تعبير صحيحة .

تعبيرif ... else

تعبير اذا أو غير ذلك : يمكن استخدام هذا النوع من الكود الشرطي اذا كانت حالة الشرط الأول صحيحة والثانية خاطئة.

التركيبة

```
if (الشرط)
{
الكود الذي سيظهر في حال تحقق الشرط
}
else
{
الكود الذي سيظهر في حال لم يتحقق الشرط
}
```

مثال

سيعرض المثال التالي "اليوم هو جمعة "اذا كان اليوم جمعة أما اذا لم يكن جمعة فستعرض النتيجة "اليوم ليس بيوم جمعة: "

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "اليوم هو جمعة";
else
echo "اليوم ليس جمعة";
?>

</body>
</html>
```

اذا كان هناك أكثر من نتيجة سيتم عرضها اذا كان الشرط true/falseعندها يجب وضع الشرط والنتائج ضمن أقواس منحنية{}

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
{
echo "مرحباً<br />";
echo "إنه يوم جميل ";
echo "أراك لاحقاً";
}
?>

</body>
</html>
```

تعبيرelse.... else if ... if

يستخدم هذا التعبير مع واحد من مجموعة أكواد أي يستخدم عند تحقق شرط معين واذا لم يتحقق سيتم إظهار شرط آخر واذا لم يتحقق سيتم تلقائياً تطبيق الكود عند. else

التركيبة

```
if (الشرط 1)
{
الكود الذي سيظهر في حال تحقق الشرط 1
}
else if (الشرط 2)
{
الكود الذي سيظهر في حال تحقق الشرط 2
}
else
{
الكود الذي سيظهر في حال لم يتحقق الشرط 1 أو الشرط 2
}
```

مثال

في المثال التالي سيتم عرض الجملة "Have a nice weekend!"اذا كان اليوم جمعة واذا لم يكن سيتم عرض "Have a nice Sundayy!" اذا كان يوم الأحد و سيتم عرض "Have a nice day!" اذا كان غير ذلك.

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "Have a nice weekend!";
elseif ($d=="Sun")
echo "Have a nice Sunday!";
else
echo "Have a nice day!";
?>

</body>
</html>
```

تعبير switch

تستخدم التعابير الشرطية لإنشاء أوامر مختلفة تعتمد على شروط مختلفة .

تعبير switch في PHP

يستخدم تعبير switch لإختيار كود من مجموعة أكواد .

التركيبة

```
switch(n) {
case 1:
1 إدراج مجموعة الأكواد
break;
case 2:
2 إدراج مجموعة الأكواد
break;
default:
2 إظهار الكود اذا كانت النتيجة غير مطابقة للحالتين 1 و
}
```

كيف يعمل الكود

أولاً يجب تحديد تعبير معين بين قوسيم عند استخدام التعبير switch وغالباً مايكون متغير ويتم تقديره مرة واحدة ثم يتم مقارنة قيمة التعبير الذي سيتم مقارنته مع القيم الأخرى في كل حالة.

إن كان هناك تطابق مع الكود سيت عرضه فوراً وسيتوقف الكود عن الفحص كما يمكن استخدام break بعد كل حالة لتجنب عبور الفحص إلى الحالة الأخرى تلقائياً.

مثال

```
<html>
<body>
<?php
$x=1;
switch ($x)
{
case 1:
echo "1 الرقم";
break;

case 2:
echo " 2 الرقم";
break;

case 3:
echo " 3 الرقم";
break;

default:

echo "3 و 1 بين أرقام لايوجد ";

}

?>

</body>
</html>
```

المصفوفات

تستخدم المصفوفة لتخزين بيانات متعددة في متغير واحد .

ماهي المصفوفة Array ؟

المصفوفة هي متغير خاص حيث يمكنه تخزين قيمة أو أكثر في نفس الوقت .

اذا كان لديك قائمة من العناصر (كتب على سبيل المثال) عندها يمكنك تخزينهم في متغير واحد .

مثال

```
$book1="كتاب البرمجة";
$book2="كتاب الحاسوب";
$book3=" قصص المغامرات";
```

ولكن ماذا لو أردت أن تجلب كتاب معين من بين هذه الكتب وماذا لو كان لديك أكثر من 300 كتاب ليس فقط 3 ؟ ماذا ينبغي أن تفعل ؟

أفضل حل هو إنشاء مصفوفة array()

يمكن للمصفوفة أن تحمل جميع القيم التي لديك وأيضاً جميع المتغيرات التي لديك بمتغير واحد فقط ويمكنك استدعاء القيمة التي تريد من خلال ذكر اسمها فقط او ترتيبها.

يحتوي كل عنصر في المصفوفة على ID خاص به وبذلك يمكن استدعائه بسهولة.

يوجد 3 أنواع للمصفوفات في لغة: PHP

- المصفوفة الرقمية : تحتوي على فهرسة رقمية.
- مصفوفة التعيين : تتكون من مجموعة ID وكل ID يحمل قيمته الخاصة.
- المصفوفة المتعددة الخيارات : تحتوي على مصفوفة أو أكثر من مصفوفة.

المصفوفة الرقمية

تخزن المصفوفة الرقمية كل عنصر من المصفوفة برقم مفهرس.

هناك طريقتين لإنشاء مصفوفة

.1في هذا المثال تم تعيين الفهرسة تلقائياً (تبدأ الفهرسة من الرقم 0).

```
$ myBooks = array("كتاب البرمجة","كتاب الحاسوب"," قصص المغامرات");
```

.2في المثال التالي سيتم إنشاء مصفوفة وتعيين الفهرسة يدوياً.

```
$myBooks [0]=" كتاب البرمجة";
$myBooks [1]=" كتاب الحاسوب";
$myBooks [2]=" قصص المغامرات";
```

مثال

في المثال التالي سيتم اختيار قيمة معينة من المصفوفة.

```
<?php
$myBooks [0]=" كتاب البرمجة";
$myBooks [1]=" كتاب الحاسوب";
$myBooks [2]=" قصص المغامرات";
echo "يعتبر " . $myBooks[0] . " و " . $myBooks[1] . " من أفضل الكتب العلمية";
?>
```

ستظهر النتيجة على المتصفح:

يعتبر كتاب البرمجة و كتاب الحاسوب من أفضل الكتب العلمية

مصفوفة التعيين

يمكن تحديد في مصفوفة التعيين كل قيمة مع رقم ID الخاص بها.

ليس من الجيد استخدام المصفوفة الرقمية عند طلب قيمة معينة من المصفوفة لذلك أفضل طريقة هي مصفوفة التعيين.

مثال 1

في هذا المثال سيتم تحديد قيم محددة اي سيتم تحديد اسم الشخص مع عمره.

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

مثال 2

يعتبر هذا المثال مماثل للمثال 1 ولكن يظهر طريقة أخرى لإنشاء المصفوفة.

```
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";
```

يمكن استخدام ID في كود المثال التالي :

```php
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";

echo "Peter is " . $ages['Peter'] . " years old.";

?>
```

ستظهر النتيجة على المتصفح

Peter is 32 years old.

المصفوفة المتعددة

يمكن لأي عنصر في المصفوفة المتعددة الأصلية أن يصبح مصفوفة فرعية وأيضاً أي عنصر في المصفوفة الفرعية يمكن أن يصبح مصفوفة مستقلة وهكذا.

في المثال التالي سنقوم بإنشاء مصفوفة متعددة و تلقائياً محدد بـآي دي ID خاص.

```
$families = array (
"Griffin"=>array
(
"Peter",
"Lois",
"Megan"
),
"Quagmire"=>array
(
"Glenn"
),
"Brown"=>array
(
"Cleveland",
"Loretta",
"Junior"
)
);
```

سيصبح شكل المصفوفة السابقة كالتالي في حال تم كتابتها كنتيجة على التصفح.

```
Array
(
[Griffin] => Array
(
[0] => Peter
[1] => Lois
[2] => Megan
)
[Quagmire] => Array
(
[0] => Glenn
)
[Brown] => Array
(
[0] => Cleveland
[1] => Loretta
[2] => Junior
))
```

مثال 2

الآن سنقوم باستدعاء أحد عناصر المصفوفة السابقة في:

```
<?php
echo "Is " . $families['Griffin'][2] .
" a part of the Griffin family?";
?>
```

ستظهر النتيجة على المتصفح

Is Megan a part of the Griffin family?

الحلقات – الحلقة while

تستخدم الحلقات لإنشاء مجموعة من الأكواد تحدد عدد مرات الأرقام أو اذا كان الشرط صحيح .

حلقات PHP

عند كتابتك لكود معين وتريد أن يكون هذا الكود مماثل لعدة مرات بالظهور نفسه مجدداً ضمن سطر معين .

عوضاً عن كتابة الكود أكثر من مرة يمكنك استخدام الكود مرة واحدة وبشكل حلقات متتالية عندها يظهر الكود الذي تريد أكثر من مرة .

هناك 4 أنواع من الحلقات في PHP

- الحلقات باستخدام while : من خلال استخدام مجموعة أكواد عند تحقق شرط معين .
- الحلقات باستخدام do...while : من خلال استخدام مجموعة أكواد لمرة واحدة ثم تكرار الحلقة طالما يتحقق ذلك شرط .
- الحلقات باستخدام for : من خلال استخدام مجموعة أكواد لتحديد عدد مرات معينة .
- الحلقات باستخدام foreach : من خلال استخدام مجموعة أكواد لكل عنصر في المصفوفة .

الحلقة while

تستخدم الحلقات باستخدام while من خلال استخدام مجموعة أكواد عند تحقق شرط معين .

التركيبة

```
while (condition)
{
code to be executed;
}
```

مثال

يشرح المثال التالي حلقة تبدأ بالمتغير i=1 . ستستمر الحلقة بالعمل طالما المتغير i هي أصغر من أو يساوي 5 . سيتم زيادة رقم واحد 1 في كل مرة تعمل بها الحلقة حتى تستوفي الشرط.

```
<html>
<body>
<?php
$i=1;
while($i<=5)
{
echo "الرقم هو " . $i . "<br />";
$i++;
}
?>

</body>
</html>
```

النتيجة:

الرقم هو 1

الرقم هو 2

الرقم هو 3

الرقم هو 4

الرقم هو 5

التعبيرdo...while

تستخدم مجموعة أكواد لمرة واحدة ثم تكرار الحلقة طالما يتحقق ذلك شرط.

التركيبة

```
do
{
code to be executed;
}
while (condition);
```

مثال

يشرح المثال التالي حلقة تبدأ بالمتغير i=1 . ستستمر الحلقة بالعمل طالما المتغير i هي أصغر من أو يساوي 5 . سيتم زيادة رقم واحد 1 في كل مرة تعمل بها الحلقة حتى تستوفي الشرط

```
<html>
<body>
<?php
$i=1;
do
{
$i++;
echo "The number is " . $i . "<br />";
}
while ($i<=5);
?>

</body>
</html>
```

النتيجة:

الرقم هو 2

الرقم هو 3

الرقم هو 4

الرقم هو 5

الرقم هو 6

الحلقات – الحلقةFor

تستخدم الحلقات لإنشاء مجموعة من الأكواد تحدد عدد مرات الأرقام أو اذا كان الشرط صحيح .

الحلقة For

تستخدم الحلقة For عند معرفة عدد المرات التي يجب أن يعمل بها الكود .

التركيبة

```
for (init; condition; increment)
{
code to be executed;
}
```

البارامترزParameters

- **Init : تستخدم عادة لضبط العداد (لكن يمكن أن يكون أي نوع من الأكواد التي ستظهر مرة في بداية كل حلقة.)**
- **Condition : الشرط و هو الشرط الذي سيرفق في الحلقة في حال كان الشرط true فإن الحلقة ستكمل الدوران ولكن في حال كان الشرط false ستتوقف الحلقة.**
- **Increment : تستخدم عادة لزيادة العداد.**

ملاحظة : كل واحد من البارامتر في التركيبة السابقة يمكن أن يكون خالياً و يمكن أن يكون أكثر من تعبير مفصول بفاصلة.

يعرّف المثال التالي حلقة تبدأ من . i=1 ستستمر الحلقة بالدوران طالما أن i هي أصغر أو تساوي 5 . سيتم زيادة i بالرقم 1 في كل مرة تدور فيها الحلقة.

```
<html>
<body><?php
for ($i=1; $i<=5; $i++)
{
echo "الرقم هو " . $i . "<br />";
}
?>
</body>
</html>
```

النتيجة

الرقم هو 1
الرقم هو 2
الرقم هو 3
الرقم هو 4
الرقم هو 5

حلقة foreach

من خلال استخدام مجموعة أكواد لكل عنصر في المصفوفة.

التركيبة

```
foreach ($array as $value)
{
code to be executed;
}
```

في المثال التالي يشرح كيفية إظهار النتيجة من المصفوفة :

```
<html>
<body><?php
$x=array("ثلاثة","اثنان","واحد");
foreach ($x as $value)
{
echo '<ul>';
echo '<li>' . $value . "</li>";
echo '</ul>';
}
?>
</body>
</html>
```

النتيجة

- **واحد**
- **اثنان**
- **ثلاثة**

الوظائف و الدوال

تستمد لغة PHP قوتها من خلال الأكواد الوظيفية functions

هناك أكثر من 700 كود وظيفي معرّف مسبقاً من خلال اللغة نفسها .

الأكواد الوظيفية المعرّفة مسبقاً

لمراجع كاملة عن الكواد الوظيفية المعرّفة مسبقاً يمكنك زيارة مرجعنا الشامل عن PHP

دوال PHP

سندرس في هذا الدرس كيفية إنشاء الكود الوظيفي الخاص بك .

يمكنك وضعه ضمن وظيفة للحفاظ على الكود من ظهوره عند تحميل الصفحة .

يمكن إظهار function من خلال استدعائه .

يمكنك استدعائه في أي مكان في الصفحة .

إنشاء كود وظيفي PHP

يمكن إظهار function من خلال استدعائه .

التركيبة

```
function functionName()
{
code to be executed;
}
```

بعض النصائح عند إنشاء الكود الوظيفي

- ضع اسم مناسب للكود الوظيفي بحيث يشرح ماهية عمل ذلك الكود.

- يمكن أن يبدأ اسم الكود بأحرف أو اشارة _ ولكن ليس رقم.

إضافة parameters

لإضافة فعالية إضافية الى الكود الوظيفي يمكنك اضافة . parameters يعتبر parameters مثل المتغير تماماً.

يتم تحديد parameters بعد اسم function داخل الأقواس.

ارجاع القيمReturn values

لإرجاع قيمة من الكود الوظيفي يمكن استخدام التعبيرreturn

مثال

```
<html>
<body>
<?php
function add($x,$y)
{
$total=$x+$y;
return $total;
}
echo "1 + 16 = " . add(1,16);
?>

</body>
</html>
```

النتيجة

1 + 16 = 17

النماذج و طرق الادخال

تستخدم المتغيرات $_GET و $_POST لإسترجاع معلومات من النماذج مثل اسم المستخدم وغيرها .

التحكم بالنماذج

أكثر الأشياء أهمية عند التعامل مع نماذج HTML و PHP هو أن أي عنصر من النماج في صفحات HTML ستكون متوفرة تلقائياً مع أكواد PHP .

مثال

يحتوي المثال التالي على نموذج HTML مع حقلين من نماذج الادخال و أيضاً أيقونة إرسال :

```
<html>
<body>
<form action="welcome.php" method="post">
الاسم : <input type="text" name="fname" />
العمر : <input type="text" name="age" />
<input type="submit" value="ارسال" />
</form>

</body>
</html>
```

عند تعبئة الحقول في المثال السابق والضغط على الأيقونة ارسال سيتم ارسال البيانات الى ملف PHP و المسمى welcome.php

يجب أن يكون ملف welcome.php يحتوي على الكود التالي:

```
<html>
<body>
مرحباً <?php echo $_POST["fname"]; ?>!<br />
عمرك هو <?php echo $_POST["age"]; ?> عام.

</body>
</html>
```

ستظهر النتيجة على المتصفح حسب وضع البيانات في ذلك الملف

مرحباً محمد !

عمرك هو 25 عام.

فعالية النماذج

يجب التأكد من فعالية نماذج الادخال التي يستخدمها الزائر ان أمكن لأن عمليه التأكد على المتصفح تعتبر سريعة و تقلل من ضغط التحميل على السيرفر.

يجب استخدام خاصية التأكد من نماذج الادخال وخاصة ان كان النموذج سيتم ارساله الى قاعدة البيانات . افضل طريقة لإنشاء تفعيل للنموذج على السيرفر هو ادراجه في النموذج نفسه بدلاً من الذهاب الى صفحة أخرى و عندها سيحصل المستخدم على رسالى الخطأ في نفس الصفحة و بتلك الطريقة يمكن استكشاف الأخطاء بسرعة و سهولة.

المتغير GET

يستخدم المتغير $_GET في PHP لجمع القيم في النماذج مع الطريقةmethod="get"

المتغير $_GET

ترسل المعلومات عبر النماذج من خلال الطريقة GET وهي مرئية للجميع أي تظهر المعلومات على رابط المتصفح كما تحتوي على كمية محدودة من المعلومات المرسلة .

مثال

```
<form action="welcome.php" method="get">
الاسم : <input type="text" name="fname" />
العمر : <input type="text" name="age" />
<input type="submit" value="ارسال" />

</form>
```

عند الضغط على رز الارسال سيظهر على المتصفح المعلومات التي قمت بإدخالها في حقول النموذج.

```
http://www.w3arabiconline.com/welcome.php?fname=Peter&age=37
```

يستخدم الملف welcome.php المتغير $_GET لجمع البيانات أي ستصبح الأسماء الخاصة بالنموذج عبارة وسوم للمصفوفة : $_GET

```
مرحباً بك <?php echo $_GET["fname"]; ?>.<br />
عمرك <?php echo $_GET["age"]; ?> عام
```

متى نستخدم الطريقة "method="get

عند استخدام الطريقة "method="getفي نماذج HTML عندها جميع الأسماء و القيم الموجودة ضمن وسم <inputt>ستظهر على رابط المتصفح.

ملاحظة

- لا يجب استخدام هذه الطريقة عند ارسال بيانات هامة مثل كلمة المرور أو أي معلومات حساسة . و على أي حال لأن المتغيرات يتم عرضها على رابط المتصفح يمكن استخدام bookmarkعلامة للصفحة و التي يمكن أن تكون مفيدة في بعض الحالات.
- ان الطريقة get غير مناسبة عند ارسال كميات كبيرة من البيانات و أيضاً لايجب استخدامها عندما تتجاوز كمية القيم عند 2000 حرف.

المتغير POST

يستخدم المتغير $_POST في PHP لجمع القيم في النماذج مع الطريقة"method="post

---

المتغير $_POST

ترسل المعلومات عبر النماذج من خلال الطريقة POST وهي غير مرئية للجميع أي لا تظهر المعلومات على رابط المتصفح كما تحتوي على كمية غير محدودة من المعلومات المرسلة .

ملاحظة : الحجم الكلي للطريقة POST هي 8 ميجابايت و بشكل افتراضي يمكن تغيير هذه الاعدادات من خلال ملف php.ini من خلال الضبط post_max_sizee .

مثال

```
<form action="welcome.php" method="post">
الاسم : <input type="text" name="fname" />
العمر : <input type="text" name="age" />
<input type="submit" value="ارسال" /></form>
```

عند الضغط على زر الارسال سيصبح الرابط كالتالي:

http://www.example.com/welcome.php

يستخدم الملف welcome.php المتغير $_POST لجمع البيانات أي ستصبح الأسماء الخاصة بالنموذج عبارة وسوم للمصفوفة : $_POST

متى نستخدم الطريقة "method="post

ان المعلومات التي ترسل من النماذج من خلال الطريقة POST تكون مخفية عن الاخرين و غير محدودة بكمية المعلومات المرسلة.

وعلى أي حال وبسبب أن هذه الطريقة مخفية و لاتظهر على الرابط فهذه الطريقة غير صالحة لإنشاء علامات للصفحات. bookmarks

المتغير $_REQUEST

يحتوي المتغير المعرّف $_REQUEST في لغة PHP على محتوى كلاً من $_GET و$_POST و . $_COOKIE

يمكن أن يستخدم المتغير $_REQUEST لجمع بيانات النموذج و إرساله الى كلاً من الطريقتين GET و. POST

```
مرحباً بك <?php echo $_REQUEST["fname"]; ?>.<br />
عمرك <?php echo $_REQUEST["age"]; ?> عام
```

الدروس المتقدمة

التاريخ و الوقت

تستخدم الوظيفة date() لتشكيل الوقت و / أو التاريخ .

الوظيفة Date()

يستخدم الكود الوظيفي date() لتحديد الوقت و التاريخ وجعله واضح على المتصفح .

التركيبة

```
date(format,timestamp)
```

الكود	الشرح
format	مطلوب . لتحديد التاريخ و الوقت الزمني
timestamp	خياري . تحديد الوقت الزمني وبشكل افتراضي تكون النتيجة التاريخ والوقت الحالي

تشكيل الزمن Date()

يحدد الجزء المطلوب format في الكود date()كيفية تشكيل التاريخ أو الوقت.

يمكن استخدام بعض الأحرف كالتالي:

- d : يعبر عن أيام الشهر و هي من 01 الى 31.
- m : يعبر عن الشهر من الشهر 01 الى 12.
- y : تعبر عن السنة من خلال أربع خانات.

من هنا يمكن مشاهدة جميع تشكيلات الوقت من خلال المرجع الخاص بالتاريخ و الوقت.

اشارات أخرى مثل / او . أو – يمكن أن توضع بين الأحرف لتعديل تشكيل التاريخ.

مثال

```php
<?php

echo date("Y/m/d") . "<br />";

echo date("Y.m.d") . "<br />";

echo date("Y-m-d");

?>
```

ستظهر النتيجة على المتصفح

2009/05/11

2009.05.11

2009-05-11

اضافة تفاصيل للزمن

يمكن اضافة تفاصيل معينة للزمن و استخدام تاريخ رما قديم أو حديث ولكن هذه الاضافة اختيارية وان لم تقم بضبطها سيتم اختيار التاريخ الحالي.

يرجع الكود الوظفي mktime() الزمن العالمي للتاريخ.

يحتوي الزمن العالمي على أجزاء صغيرة من الثانية والتي تبدأ من كانون الثاني 00:00:00 1 1970 GMT) .

تركيبة mktime()

```
mktime(hour,minute,second,month,day,year,is_dst)
```

للذهاب الى وقت معين الى المستقبل يمكن اضافة تعبير اليوم من خلال الكود: mktime()

```
<?php

$tomorrow = mktime(0,0,0,date("m"),date("d")+1,date("Y"));

echo "Tomorrow is ".date("Y/m/d", $tomorrow);

?>
```

ستظهر النتيجة على المتصفح

```
غداً هو 12/5/2013
```

المرجع الكامل في PHP

من هنا يمكن مشاهدة جميع تشكيلات الوقت من خلال المرجع الخاص بالتاريخ و الوقت

include الملفات الضمنية

الملفات الضمنية في السيرفر Server Side Includes (SSI)

يمكنك ادراج محتوى ملفات PHP الى ملفات PHP أخرى قبل قيام السيرفر بعرضها ويمكن استخدام include() أو require() .

تعتبر هذه الوظيفتين ثابتتين باستثناء كيفية التحكم بالأخطاء :

- **تقوم الوظيفة include() بتوليد التحذيرات ولكن سيتم استكمال عرض باقي الكود .**
- **تقوم الوظيفة require() بتوليد خطأ فادح عندها سيتم إيقاف الكود ولن يتم عرضه .**

يمكن باستخدام هاتين الخصيصتين إنشاء أكواد و ترويسات و تذيلات و أيضاً عناصر يمكن استخدامها في أكثر من مكان كالاعلانات وغيرها .

بهذه الطريقتين تحفظ الكثير من الوقت والجهد في إنشاء وتعديل الصفحات أي يمكنك إنشاء صفحة واحدة واستخدامها بأكثر من مكان وعندما سيتم تحديث أو تعديل الترويسة في الموقع يمكنك تعديل صفحة الترويسة header عندها سيتم تغيير الهيدر او الترويسة في جميع الصفحات أو عند إضافة صفحة جديدة يمكنك وبسهولة إضافتها في جميع صفحات الموقع عوضاً عن إضافة الصفحة يدوياً .

Function include() PHP

يأخذ ()include جميع محتوى الملف المخصص و يضمّنه في الملف الأصلي .

عند حدوث خطأ مع ()include سيتم توليد تحذير ولكن باقي الكود سيتم عرضه وسيعمل بشكل جيد .

مثال 1

لنفترض أن لديك ملف ترويسة header بملف تم تسميته header.php ولتضمين هذا الملف داخل الصفحة الرئيسية يمكن استخدام ()include كالتالي :

```
<html>
<body>
<?php include("header.php"); ?>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

مثال 2

لنفترض أن لديك قائمة رئيسية تسمى menu.php والتي سيتم استخدامها في جميع الصفحات كالتالي:

```
<a href="/default.php">الرئيسية</a>
<a href="/tutorials.php">الدروس</a>
<a href="/about.php">من نحن</a>
<a href="/contact.php">الاتصال بنا</a>
```

تلك القائمة يجب استخدامها في جميع الصفحات وإليكم طريقة استخدامها في جميع صفحات الموقع

.

```
<html>
<body>
<div class="leftmenu">
<?php include("menu.php"); ?>
</div>
<h1>Welcome to my home page.</h1>
<p>Some text.</p>

</body>
</html>
```

اذا قمت بإظهار مصدر الصفحة لمشاهدة الأكواد من خلال المتصفح سيتم عرضها كالتالي:

```
<html>
<body>
<a href="/default.php">الرئيسية</a>
<a href="/tutorials.php">الدروس</a>
<a href="/about.php">من نحن</a>
<a href="/contact.php">الاتصال بنا</a>

</div>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

PHP require() Function

تعتبر require()مشابهة تماماً للوظيفة include()باستثناء طريقة التعامل مع الأخطاء.

عند حوث خطأ برمجي معين تقوم الوظيفة include()بتوليد التحذيرات ولكن سيتم استكمال عرض باقي الكود أما الوظيفة require()تقوم بتوليد خطأ فادح عندها سيتم إيقاف الكود ولن يتم عرضه .

مثال رسالة الخطأ باستخدام include()

```
<html>
<body>
<?php
include("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

رسالة الخطأ:

```
Warning: include(wrongFile.php) [function.include]:
failed to open stream:
No such file or directory in C:homewebsitetest.php on line 5
Warning: include() [function.include]:
Failed opening 'wrongFile.php' for inclusion
(include_path='.;C:php5pear')
in C:homewebsitetest.php on line 5
Hello World!
```

لاحظ بأن الجملة Hello World! تم عرضها وذلك لأن التحذير لم يقوم بإيقاف الكود بشكل كامل.

مثال رسالة الخطأ باستخدام require()

الآن لنقم بإنشاء الكود التالي الخاص بالوظيفة : require()

```
<html>
<body>
<?php
require("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

رسالة الخطأ:

```
Warning: require(wrongFile.php) [function.require]:
failed to open stream:
No such file or directory in C:homewebsitetest.php on line 5
Fatal error: require() [function.require]:
Failed opening required 'wrongFile.php'
(include_path='.;C:php5pear')
in C:homewebsitetest.php on line 5
```

لم تظهر الجملة الموجودة بعد الكود وذلك لأنها لاتقوم بعرض باقي الكود عند استكشاف خطأ معين

.

من الأفضل استخدام require()مع الصفحات التي تحتوي على أكواد ولأنها بشكل طبيعي لن تعمل الأكواد في حال الخطأ أما include()يمكن استخدامها مع الصفحات التي لاتحتوي على الأكواد مثل HTMLأو. CSSS

التعامل مع الملفات

تعتبر الوظيفة fopen() مسؤولة عن فتح الملفات في لغة PHP .

فتح الملف

تعتبر الوظيفة fopen() مسؤولة عن فتح الملفات في لغة PHP .

يستخدم أول رمز داخل الكود الوظيفي لإدراج اسم الملف المراد فتحه أما في الرمز الثاني سيتم تحديد حالة الملف بعد فتحه :

```
<html>
<body>
<?php
$file=fopen("welcome.txt","r");
?>

</body>
</html>
```

يمكن تحديد حالة الملف من خلال الحالات التالية :

الحالة	الشرح
r	للقراءة فقط وتبدأ مع بداية فتح الملف .
r+	قراءة / كتابة وتبدأ مع بداية فتح الملف .
w	كتابة فقط . فتح الملف ومسح محتويات الملف أو إنشاء ملف جديد إن لم يكن موجوداً .
w+	كتابة و قراءة . فتح الملف ومسح محتويات الملف أو إنشاء ملف جديد إن لم يكن موجوداً .
a	ملحق . يفتح و يكتب الملف حتى نهايته أو إنشاء ملف جديد اذا لم يكن موجوداً .
a+	ملحق . يقدم محتوى الملف من خلال كتابة الملف من بدايته لنهايته .
x	كتابة فقط . إنشاء ملف جديد . يعيد نتيجة خاطئة او خطأ اذا كان الملف موجود بالفعل .
x+	كتابة و قراءة . إنشاء ملف جديد . يعيد نتيجة خاطئة او خطأ اذا كان الملف موجود بالفعل .

ملاحظة : اذا كانت الوظيفة fopen() غير قادرة على فتح ملف معين عندها ستعود القيمة 00.

مثال:

في المثال التالي سيتم اظهار رسالة اذا كانت الوظيفة fopen() غير قادرة على فتح الملف.

```
<html>
<body>
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
?>

</body>
</html>
```

إغلاق الملف

تستخدم الوظيفة fclose() لإغلاق ملف معين.

```
<?php
$file = fopen("test.txt","r");

//some code to be executed
fclose($file);

?>
```

فحص آخر الملف

يفحص الوظيفة feof() نهاية الملف أي في حالة تم قراءة الملف بشكل كامل.

تعتبر الوظيفة feof() مثالية في استخدام الحلقات عبر البيانات الغير معروفة الطول.

ملاحظة : لايمكنك قراءة الملفات المفتوحة من خلال w و a و. x

```
if (feof($file)) echo "End of file";
```

قراءة الملف سطر بسطر

تستخدم الوظيفة fgets() لقراءة سطر معين من الملف.

ملاحظة : بعد استخدام هذه الوظيفة سينتقل المؤشر للسطر الثاني.

مثال

في المثال التالي سيتم قراءة سطر واحد من الملف أي حتى نهاية الملف.

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
//Output a line of the file until the end is reached
while(!feof($file))
{
echo fgets($file). "<br />";
}
fclose($file);

?>
```

قراءة الملف كلمة بكلمة

تستخدم الوظيفة fgetc() لقراءة كلمة واحدة من الملف.

ملاحظة : بعد استخدام هذه الوظيفة سينتقل المؤشر للكلمة الثاني.

مثال

في المثال التالي سيتم قراءة كلمة واحدة من الملف أي حتى نهاية الملف.

```php
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
echo fgetc($file);
}
fclose($file);

?>
```

رفع الملفات

من الممكن رفع الملفات الى السيرفر باستخدام لغة. PHP

إنشاء نموذج لرفع الملفات

يمكن إنشاء نموذج يسمح للمستخدم رفع ملفاته بسهولة.

لاحظ نموذج HTML التالي والذي يسمح برفع الملفات:

```
<html>
<body>
<form action="upload_file.php" method="post"enctype="multipart/form-data">

<label for="file">Filename:</label>
<input type="file" name="file" id="file" />

<br />

<input type="submit" name="submit" value="Submit" />

</form>
</body>
</html>
```

ان المثال السابق هو إنشاء نموذج من خلال لغة: HTML

- **تحدد اللاحقة** enctype **الموجودة في وسم النموذج** <form>**نوع المحتوى المستخدم عند الضغط على زر "ارسال" . أما القيمة** "multipart/form-data" **تستخدم عندما يطلب النموذج بيانات ثنائية مثل محتوى مجلد سيتم رفعه.**
- **تحدد اللاحقة** "type="file**الموجودة في الوسم** <input>**حيث يجب أن تكون العملية مجلد أي على سبيل المثال عند الاستعراض على المتصفح سيتواجد بجانب حقل النص زر لرفع الملف.**

ملاحظة : تعتبر مخاطرة كبيرة في الحماية عند السماح للمستخدم برفع ملفات على السيرفر لذلك مكنك فقط السماح للمستخدمين الموثوقين برفع الملفات على السيرفر.

إنشاء سكربت لرفع الملفات

يحتوي ملف الرفع upload_file.php **على الكود التالي:**

```php
<?php
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
}

?>
```

باستخدام المصفوفة $_FILES يمكنك رفع الملفات من حاسوب المستخدم الى السيرفر.

ان أول قيمة في حقل النموذج هي الاسم أما الثانية فيمكن أن تكون اما الاسم name أو النوع type أو الحجم size أو tmp_name أو الخطأ error كالتالي:

- **$_FILES["file"]["name"] : اسم الملف المرفوع.**
- **$_FILES["file"]["type"] : نوع الملف المرفوع.**
- **$_FILES["file"]["size"] : حجم الملف المرفوع**
- **$_FILES["file"]["tmp_name"] : اسم النسخة المؤقتة للملف الذي تم تخزينه في السيرفر.**
- **$_FILES["file"]["error"] : نتيجة الخطأ اذا كان هناك خطأ في الكود أو الملف المرفوع**

.

تعتبر هذه الخطوات أسهل طرقة لرفع ملف ولكن اذا أردت حماية الملفات على السيرفر يجب وضع قيود على الرفع.

قيود الرفع

لقد قمنا بإضافة بعض الأكواد لوضع قيود على الرفع أي يمكن فقط للمستخدم رفع ملفات بامتداد gif.أو jpg.و يجب أن يكون الحجم أقل من 20: kbb

```
<?php
if (((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
} else {
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
} else {
echo "Invalid file";
}

?>
```

حفظ الملف المرفوع

في المثال السابق تم رفع نسخة مؤقتة للملفات المرفوعة في ملف PHP temp على السيرفر.

تختفي الملفات المؤقتة بعد انتهاء الكود و لحفظ ملفاتك المرفوعة يجب اختيار مكان آخر للملفات على السيرفر:

```php
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"],"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
}
else
{
echo "Invalid file";
}

?>
```

يفحص الكود السابق الملف اذا كان موجوداً من قبل واذا لم يكن موجوداً عندها سيتم نسخ الملف الى المجلد المخصص.

ملاحظة : يتم حفظ الملف في المثال السابق في المجلد. "upload"

Cookies الكوكيز

يستخدم الكوكيز عادة لتحديد هوية المستخدم .

ماهو الكوكيز ؟

يستخدم الكوكيز عادة لتحديد هوية المستخدم . ان الكوكي cookie عبارة عن ملف صغير يتم ارساله من خلال السيرفر الى حاسوب المستخدم . و في كل مرة يتم طلب الصفحة من خلال المتصفح سيتم ارسال ملف الكوكي ايضاً . يمكنك إنشاء و استرجاع ملفات و قيم الكوكي .

كيف يمكن إنشاء cookie ؟

يستخدم الكود الوظيفي setcookie() لإنشاء كوكي .

ملاحظة : يجب وضع كود setcookie() قبل وسم <html> .

التركيبة

```
setcookie(name, value, expire, path, domain);
```

مثال 1

في المثال التالي سيتم إنشاء cookie اسمه user والقيمة هي Ahmad . كما سيتم تحديد انتهاء cookieبعد ساعة:

```php
<?php
setcookie("user", "Ahmad", time()+3600);

?>

<html>

.....
```

ملاحظة : يتم ترميز قيمة cookie تلقائياً عند ارسال cookie كما يتم ترميزه تلقائياً عند استرجاعه ولتجنب ترميز الروابط URL يمكنك استخدام setrawcookie()عوضاً عنها.

مثال 2

يمكنك تحديد وقت انتهاء cookie بطريقة أخرى أيضاً . كما يمكن أن تكون أسهل عند استخدام الثواني:

```php
<?php
$expire=time()+60*60*24*30;
setcookie("user", "Ahmad", $expire);

?>

<html>

.....
```

في المثال السابق تم تحديد انتهاء وقت الملف لشهر (60 ثا , 60 د , 24 سا , 30 يوم).

كيفية استرجاع قيمة Cookie ؟

يمكن استخدام المتغير $_COOKIE لاسترجاع قيمة cookie.

في المثال التالي تم استرجاع قيمة cookie المسماة user وعرضها في الصفحة:

```
<?php
// اظهار الكوكي

echo $_COOKIE["user"];

// طريقة أخرى لعرض جميع الكوكيز

print_r($_COOKIE);

?>
```

تم استخدام isset() في المثال التالي لمعرفة فيما اذا تم ضبط cookie أم لا:

```
<html>
<body>
<?php
if (isset($_COOKIE["user"]))
echo "Welcome " . $_COOKIE["user"] . "!<br />";
else
echo "Welcome guest!<br />";
?>

</body>
</html>
```

كيف تحذف Cookie ؟

عند حذف cookie ينبغي التأكد بأن تاريخ الانتهاء هو في الماضي.

مثال عن حذف كوكيcookie

```php
<?php
// set the expiration date to one hour ago
setcookie("user", "", time()-3600);

?>
```

ماذا لو كان المتصفح لا يدعم cookie ؟

اذا كنت تتعامل مع متصفحات لا تدعم cookie عندها يمكنك استخدام طرق أخرى لتمرير المعلومات من صفحة الى أخرى . إحدى هذه الطرق هي تمرير المعلومات من خلال النماذج) forms تم شرح النماذج و المدخلات في دروس ماضية. (

يمرر النموذج مدخلات المستخدم إلى الصفحة "welcome.php" عند ضغط المستخدم على زر الارسال:

```
<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>

</body>
</html>
```

مثل welcome.php استرجاع القيم من المجلد :

```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old.

</body>
</html>
```

Sessions الجلسات

تستخدم متغيرات الجلسات session لحفظ المعلومات حول أو تغيير الضبط للمستخدم . تحمل متغيرات session معلومات حول مستخدم واحد كما أنها تكون متوفرة في جميع الصفحات وبتطبي واحد .

متغيرات Session

عندما تعمل على برنامج فإنك تفتحه ثم تقوم بالتعديل عليه ثم تقوم بإغلاقه . ذلك تماماً مانسميه جلسة أو session . يعلم الحاسوب من انت ويعلم متى تبدأ العمل على البرنامج ويعلم متى تنتهي من البرنامج و لكن على شبكة الانترنت هناك مشكلة صغيرة وهي أن السيرفر لايعلم من انت و لايعلم ماتفعل لأن عنوان HTTP لايحافظ على نفس الحالة .

ان PHP session تحل هذه المشكلة وذلك بالسماح لك بتخزين معلومات على السيرفر لآخر استخدام قمت به (مثل اسم مستخدم أو عناصر التسوق) . على أي حال تعتبر معلومات session دائمة التخزين أي يمكنك تخزين المعلومات في قاعدة البيانات .

تعمل session بإنشاء id مميز UID و ذلك لكل زائر وتخزن المتغيرات المتمركزة في هذا UID . يتم تخزين UID في cookie أو متوالد على الرابط التشعبي URL .

بدء جلسة PHP Session

قبل إمكانية تخزين معلومات المستخدم على في جلسة PHP session عندها يجب البدء أولاً بالجلسة session .

ملاحظة : يجب أن تظهر session_start() **قبل وسم** <html> :

```
<?php session_start(); ?>
<html>
<body>
.....
</body>
</html>
```

سيتم تسجيل المتستخدم من خلال الكود السابق مع السيرفر كما سمح لك ببدء حفظ معلومات المستخدم وتثبيت الاي دي الفريد UID لتلك جلسة المستخدم.

تخزين متغيرات الجلساتSession

أفضل طريقة لتخزين المعلومات واستردادها هو المتغير المعرّف مسبقاً من PHP وهو: $_SESSION

```
<?php
session_start();
// store session data
$_SESSION['views']=1;
?>
<html>
<body>

<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];

?>

</body>
</html>
```

ستظهر النتيجة على المتصفح

```
Pageviews=1
```

في المثال التالي قمنا بإنشاء عدّاد بسيط لإحصاء عدد مشاهدة الصفحة . يقوم ()issetبفحص فيما اذا كانت المشاهدات تم مشاهدتها بالفعل واذا تمت المشاهدة من قبل سيتم زيادة رقم عدد الزيارات . اذا لم يوجد سيتم إنشاء متغير و سيتم ضبطه الى 1 :

```php
<?php
session_start();
if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

حذف session

إن أردت حذف بعض البيانات في الجلسة session يمكنك استخدام unset() أو استخدام session_destroy() .

تستخدم unset() **لتحرير جزء معين من متغير الجلسة:**

```php
<?php
unset($_SESSION['views']);

?>
```

كما يمكنك أيضاً حذف الجلسة session بشكل كامل من خلال استدعاء: session_destroy()

```php
<?php
session_destroy();

?>
```

سيتم حذف جميع البيانات عند استخدام session_destroy() كما سيتم حذف جميع معلومات المستخدمين المخزنة فيها.

إرسال البريد الالكتروني

تسمح PHP بإرسال بريد الكتروني مباشرة من خلال كود معين .

---

mail() function PHP

تستخدم الوظيفة mail() لارسال البريد الالكتروني من داخل السكربت .

التركيبة

```php
mail(to,subject,message,headers,parameters)
```

القيم	الشرح
to	مطلوب . تحدد المستقبل و المستقبلون للبريد
subject	مطلوب . تحدد موضوع الرسالة . ملاحظة : لايمكن أن تحتوي القيمة على أحرف خاصة
message	مطلوب . تحدد الرسالة التي سيتم ارسالها . ولكل سطر يجب أن يكون مفصول بسطر جديد (n) . يجب أن يحتوي السطر على أكثر من 70 كلمة .
headers	خياري . تخصيص ترويسات اضافية مثل cc أو Bcc يجب على الترويسات أن تكون مفصولة (rn) .
parameters	خياري . تحديد قيمة إضافية لبرنامج sendmail

إنشاء بريد الكتروني بسيط

أبسط طريقة لإنشاء و ارسال بريد باستخدام PHP هو ارسال بريد نصي.

في المثال التالي سيتم أولاً إنشاء متغيرات وهي ($to, $subject, $message, $from, $headers)
ثم نستخدم المتغيرات في mail() functionnلارسال بريد الكتروني:

```
<?php
$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someonelse@example.com";
$headers = "From:" . $from;
mail($to,$subject,$message,$headers);
echo "Mail Sent.";

?>
```

نموذج البريد

باستخدام PHP يمكنك انشاء نموذج لارسال بريد الكتروني في موقعك . سنقوم بارسال رسالة نصية في المثال التالي لبريد الكتروني محدد:

```
<html>
<body>
<?php
if (isset($_REQUEST['email']))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail("someone@example.com", "$subject",
$message, "From:" . $email);
echo "Thank you for using our mail form";
}
else
//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}

?>

</body>
</html>
```

كيف يعمل كود المثال السابق:

- أولاً , فحص فما اذا تم تعبئة حقول مدخلات البريد.

- اذا لم يتم ذلك مثل اذا كانت اول زيارة للصفحة عندها سيتم إظهار نموذج. HTML
- و ان تم تعبئة النموذج سيتم ارسال البريد من النموذج.
- عند الضغط على زر ارسال بعد تعبئة النموذج سيتم اعادة تحميل الصفحة وسيتم فحص فيما اذا تم ضبط مدخلات النموذج ثم سيتم ارسال البريد.

ملاحظة : هذه اسهل طريقة لارسال بريد الكتروني و لكنه غير آمن وفي الدرس القادم سيتم شرح كيفية انشاء بريد الكتروني محمي مع الكثير من السكربتات.

مراجع البريد فيPHP

لمزيد من المعلومات حول كيفية إنشاء بريد بسيط او محمي او mail()يمكنك زيارة مراجع بريد PHP

البريد الالكتروني المحمي

هناك نقطة ضعف في انشاء البريد الالكتروني لكن سيتم حل هذه المشكلة في هذا الدرس .

إنشاء البريد الالكتروني المحمي

أولاً لاحظ الكود من خلال الدرس السابق :

```
<html>
<body><?php
if (isset($_REQUEST['email']))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail("someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
else
//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>

</body>
</html>
```

المشكلة في الكود السابق هو أن أي مستخدم غير مرخص له يمكنه ادخال البيانات الى البريد من خلال نماذج الادخال.

ماذا يحصل لو أن المستخدم أدخل النصوص التالية الى حقول مدخلات النماذج ؟

```
someone@example.com%0ACc:person2@example.com
%0ABcc:person3@example.com,person3@example.com,
anotherperson4@example.com,person5@example.com
%0ABTo:person6@example.com
```

يضع الكود mail()النص السابق الى مدخلات النموذج بأنه نص طبيعي و ترويسة النموذج لديها حقول

Cc و Bcc و . Too عند الضغط على زر الارسال عندها سيتم ارسال البريد الى جميع العناوين التي تم وضعها في النص السابق.

إيقاف ادخال البريد الالكتروني

أفضل طريقة لايقاف ادخال البريد الالكتروني هي تفعيل المدخلات.

يعتبر الكود نفسه في الدرس السابق ولكن قمنا بإضافة مفعّل للمدخلات و التي بدورها تفحص مدخلات حقول البريد:

```
<html>
<body><?php
function spamcheck($field)
{
//filter_var() sanitizes the e-mail
//address using FILTER_SANITIZE_EMAIL
$field=filter_var($field, FILTER_SANITIZE_EMAIL);

//filter_var() validates the e-mail
//address using FILTER_VALIDATE_EMAIL

if(filter_var($field, FILTER_VALIDATE_EMAIL))
{
return TRUE;
} else {
return FALSE;
}
}

if (isset($_REQUEST['email']))
{//if "email" is filled out, proceed
//check if the email address is invalid
$mailcheck = spamcheck($_REQUEST['email']);
if ($mailcheck==FALSE)
{
echo "Invalid input";
}
else
{//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail("someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
}
else
{//if "email" is not filled out, display the form
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>

</body>
</html>
```

في المثال السابق قمنا باستخدام فلتر PHP لتفعيل الحقول:

- يمسح الفلتر FILTER_SANITIZE_EMAIL جميع أحرف البريد الغير شرعية من النصوص .
- يغعّل الفلتر FILTER_VALIDATE_EMAIL القيمة مثل عنوان بريد الكتروني.

يمكنك قراءة المزيد عن الفلاتر من خلال درس فلاتر PHP.

معالجة الأخطاء

معالجة الأخطاء الافتراضي في PHP بسيط جداً . يتم ارسال رسالة الخطأ بإسم الملف أو رقم السطر أو شرح الخطأ في رسالة الى المتصفح .

معالجة الأخطاء

تعتبر معالجة الأخطاء مهمة جداً عند إنشاء تطبيقات و سكربتات على الانترنت . سيبدو برنامجط غير احترافي وربما يمكن أن يكون معرض للمخاطر الأمنية اذا كان الكود لديك ينقصه فحص الأخطاء .

يحتوي الدورة لدينا على بعض طرق فواص الأخطاء السائدة في لغة PHP .

سنقوم بعرض طرق مختلفة لمعالجة الأخطاء :

- رسالة بسيطة من خلال die() .
- إحداث رسائل أخطاء تقليدية .
- تقارير الأخطاء .

معالج الأخطاء البسيط die()

يظهر المثال التالي كود بسيط حيث يفتح ملف نصي :

```
<?php
$file=fopen("welcome.txt","r");
?>
```

اذا لم يتم إيجاد الملف سيتم إظهار رسالة خطئ :

```
Warning: fopen(welcome.txt) [function.fopen]: failed to open stream:
No such file or directory in C:webfoldertest.php on line 2
```

لتجنب رسالة الخطأ مثل المثال السابق سيتم فحص فيما اذا كان الملف موجود قبل المحاولة بالدخول اليه:

```
<?php
if(!file_exists("welcome.txt"))
{
die("File not found");
}
else
{
$file=fopen("welcome.txt","r");
}
?>
```

ان لم يتم ايجاد النص ستظهر هذه الرسالة على المتصفح :

```
File not found
```

إنشاء معالج أخطاء

ان إنشاء معالج الأخطاء بسيط جداً . ببساطة يمكن إنشاء كود وظيفي خاص يمكن استدعائه عند حدوث الخطأ في PHP.

يجب أن يكون معالج الأخطاء ويجب أن يحمل قيمتين على الأقل (مستوى الخطأ و رسالة الخطأ) ولكن يمكن قبول خمسة قيم (اختياري : الملف , رقم السطر , محتوى الخطأ).

التركيبة

```
error_function(error_level,error_message,
error_file,error_line,error_context)
```

القيمة	الشرح
error_level	مطلوب . يحدد مستوى تقرير الخطأ نوع الخطأ المعرّف للمستخدم . كما يجب أن يكون رقم للقيمة .
error_message	مطلوب . تحديد رسالة الخطأ
error_file	اختياري . تحدد اسم الملف بحيث وقوع الخطأ .
error_line	اختياري . تحدد رقم السطر حيث وقوع الخطأ .
error_context	اختياري . تحدد المصفوفة التي تحتوي على كل متغير و قيمهم المستخدمة حيث وقوع الخطأ .

مستويات تقارير الخطأ

تحتوي مستويات تقارير الاخطاء على أنواع مختلفة والتي تعالج الأخطاء التي يمكن أن تستخدم لأجل مايلي:

القيمة	الثوابت	الشرح
2	E_WARNING	Non-fatal run-time errors. Execution of the script is not halted
8	E_NOTICE	Run-time notices. The script found something that might be an error, but could also happen when running a script normally
256	E_USER_ERROR	Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()
512	E_USER_WARNING	Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()
1024	E_USER_NOTICE	User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()
4096	E_RECOVERABLE_ERROR	Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
8191	E_ALL	All errors and warnings, except level E_STRICT (E_STRICT will be part of E_ALL as of PHP 6.0)

الآن سنقوم بإنشاء function حيث يعالج الأخطاء :

```
function customError($errno, $errstr)
{
echo "<b>Error:</b> [$errno] $errstr<br />";
echo "Ending Script";
die();
}
```

يعتبر الكود في المثال السابق بسيط جداً . عندما يتم التعيين سيتم معرفة مستوى الخطأ و رسالة الخطأ . ثم سيتم إظهار رسالة الخطأ و مستوى الخطأ ثم حذف السكربت.

و الآن وبعد إنشاء معالج للأخطاء سنحتاج الى معرفة أن سظهر بلضبط.

ضبط معالج الأخطاء

يمكن معالجة الأخطاء من خلال استخدام معالج الأخطاء في PHP . سنقوم بإنشاء function فوق معالج الأخطاء الافتراضي لمدة السكربت.

يمكن تغيير معالج الأخطاء من أجل تطبيق بعض التغيرات الخاصة بالأخطاء البسيطة و بتلك الطريقة يمكن معالجة أخطاء مختلفة بطرق مختلفة . على أي حال في هذا المثال سنقوم باستخدام معالج الأخطاء لجميع أنواع الأخطاء:

```
set_error_handler("customError");
```

يمكن set_error_handler() أن يتحكم بجميع الأخطاء و فقط حتاج قيمة واحدة و ان أردت اضافة قيمة أخرى فهي تحديد مستوى الخطأ.

مثال

اختبار معالجة الأخطاء من خلال المحاولة بإظهار متغير غير موجود أصلاً:

```php
<?php
//error handler function
function customError($errno, $errstr)
{
echo "<b>Error:</b> [$errno] $errstr";
}//set error handler
set_error_handler("customError");//trigger error
echo($test);
?>
```

ستكون النتيجة على المتصفح كالتالي:

```
Error: [8] Undefined variable: test
```

إستكشاف الأخطاء

يمكن استخدام استكشاف الأخطاء عند حدوث ادخال غير شرعي وذلك سيتم حدوثه من خلال trigger_error()**وذلك في السكربت حيث يمكن للمستخدمين وضع بياناتهم.**

مثال

في المثال التالي سيتم حدوث الخطأ اذا كان المتغير test **أكبر من 1:**

```php
<?php
$test=2;
if ($test>1)
{
trigger_error("Value must be 1 or below");
}
?>
```

ستكون النتيجة على المتصفح كالتالي :

```
Notice: Value must be 1 or below
in C:webfoldertest.php on line 6
```

يمكن اظهار الخطأ في أي مكان في السكربت و ذلك بإضافة قيمة أخرى كما يمكنك تحديد مستوى الخطأ الذي سيظهر.

أنواع الأخطاء المحتملة:

- **E_USER_ERROR : خطأ مروع يحدث عند خطأ يحدثه المستخدم و هذا الخطأ لايمكن تغطيته . يكون انجاز السكربت متوقف.**
- **E_USER_WARNING : خطأ غير مروع يتم التحذير فيه و يكون إنجاز السكربت غير متوقف.**
- **E_USER_NOTICE : ملاحظة عند امكانية حدوث خطأ معين و لكن يمكن أن يحدث أيضاً حتى في الحالة العادية للسكربت.**

مثال

في هذا المثال سيتم استدعاء الخطأ E_USER_WARNING اذا كان المتغير test أكبر من الرقم 1 و اذا حدث E_USER_WARNING سنستخدم معالج الأخطاء و سيتم إنهاء السكربت:

```php
<?php
//error handler function
function customError($errno, $errstr)
{
echo "<b>Error:</b> [$errno] $errstr<br />";
echo "Ending Script";
die();
}//set error handler
set_error_handler("customError",E_USER_WARNING);//trigger error
$test=2;
if ($test>1)
{
trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

ستظهر النتيجة على المتصفح كالتالي :

```
Error: [512] Value must be 1 or below
Ending Script
```

و الأن تعلمنا إنشاء أخطاء و كيفية استكشافهم و الآن سنتعلم حول أخطاء تسجيل الدخول.

أخطاء السجلات

بشكل افتراضي ترسل PHP سجلات الأخطاء الى نظام سجل السيرفر أو الملف و ذلك اعتماداً على كيفية ضبط اعدادات error_log في ملف . php.ini باستخدام ()error_logيمكن ارسال سجلات الأخطاء الى ملف أو وجهة مخصصة.

تعتبر طريقة جيدة بارسال سجلات الأخطاء لبريدك الالكتروني و بذلك يتم ارسال التنبيهات لبريدك عند تحديد الأخطاء .

ارسال رسالة خطأ بالبريد الالكتروني

في المثال التالي سنقوم بارسال بريد اللكتروني برسالة الخطأ و إنهاء السكربت اذا حدث ذلك الخطأ:

```php
<?php
//error handler function
function customError($errno, $errstr)
{
echo "<b>Error:</b> [$errno] $errstr<br />";
echo "Webmaster has been notified";
error_log("Error: [$errno] $errstr",1,
"someone@example.com","From: webmaster@example.com");
}//set error handler
set_error_handler("customError",E_USER_WARNING);//trigger error
$test=2;
if ($test>1)
{
trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

ستظهر النتجة على المتصفح كالتالي

```
Error: [512] Value must be 1 or below
Webmaster has been notified
And the mail received from the code above looks like this:
Error: [512] Value must be 1 or below
```

ليس من الضروري استخدام هذه الطريقة لكل الأخطاء لأن الأخطاء النظامية ينبغي أن ترسل السجلات الى السيرفر باستخدام نظام السجلات الافتراضي الخاص بلغة PHP.

PHP فلاتر

تستخدم فلاتر (تصفيات) PHP لفلترة و جعل البيانات شرعية التي تأتي من جهات غير محمية مثل مدخلات المستخدم .

ماهو فلتر / تصفية PHP ؟

تستخدم فلاتر (تصفيات) PHP لفلترة و جعل البيانات شرعية التي تأتي من جهات غير محمية مثل مدخلات المستخدم .

يعتبر اختبار أو تصفية مدخل مستخدم أو أي بيانات تقليدية خطوة مهمة لأي تطبيق على الانترنت .

تم تصميم زيادات فلاتر PHP لجعل فلترة البيانات أسهل وأسرع .

لماذا نستخدم الفلاتر ؟

غالباً تعتمد جميع تطبيقات الانترنت على مدخلات خارجية و ذلك يأتي من مستخدم أو من خلال تطبيق آخر (مثل خدمات الويب) و باستخدام الفلترة يمكنك أن تكون واثقاً بأن تطبيقك حصل على نوع المدخل الصحيح .

يجب دائماً استخدام الفلترة للبيانات الخارجية

تعتبر فلترة المدخلات واحدة من أهم القضايا الأمنية المهمة للتطبيقات .

ماهي البيانات الخارجية ؟

- البيانات المدخلة من خلال النماذج .
- الكوكيز cookies .

- بيانات الويب الخدمية .
- نتائج أوامر قواعد البيانات .

الوظائف و الفلاتر

لفلترة أو تصفية متغير يمكن استخدام احدى هذه الأكواد الوظيفية :

- filter_var() : فلترة متغير واحد مع فلتر محدد .
- filter_var_array() : فلترة متغيرات متعددة مع فلترات متشابهة أو مختلفة .
- filter_input : الحصول على متغير لمدخل واحد و فلترته .
- filter_input_array : الحصول على متغيرات لمدخلات متعددة و فلترتهم مع فلترات مختلفة او متشابهة .

في المثال التالي سنقوم بتفعيل رقم باستخدام الوظيفة filter_var() :

```php
<?php
$int = 123;
if(!filter_var($int, FILTER_VALIDATE_INT))
{
echo("Integer is not valid");
}
else
{
echo("Integer is valid");
}
?>
```

تم استخدام الفلتر FILTER_VALIDATE_INT لفلترة متغير . بما أن الرقم فعال سيتم إظهار نتيجة الكود. "Integer is valid"

لو كان المتغير ليس رقماً (مثل "123 ("abcسيتم إظهار نتيجة الكود. "Integer is not valid"

للحصول على قائمة كاملة للوظائف و الفلاتر يمكنك زيارة مرجع فلاتر. PHP

التفعيل Validating و التعقيمSanitizing

هناك نوعين من الفلترة:

تفعيل الفلاتر:

- تستخدم لتفعيل مدخل مستخدم.
- تثبيت القواعد) مثل تفعيل URL أو. (Email
- إعادة الأنواع المتوقعة من النجاح أو الفشل.

تعقيم الفلاتر:

- تستخدم للمساح أو عدم السماح بحرف معين في النص.
- لايوجد قواعد معينة للبيانات.
- دائماً تعيد نص.

الخيارات و الرايات

تستخدم الخيارات والرايات لإضافة فلترة اضافية الى فلتر معين.

تختلف الخيارات والرايات مع اختلاف الفلاتر.

في المثال التالي قمنا بتفعيل رقم معين integer باستخدام filter_var()بالاضافة الى خيارات min_rangeو: max_rangee

```php
<?php
$var=300;
$int_options = array(
"options"=>array
(
"min_range"=>0,
"max_range"=>256
)
);
if(!filter_var($var, FILTER_VALIDATE_INT, $int_options))
{
echo("Integer is not valid");
}
else
{
echo("Integer is valid");
}
?>
```

للحصول على قائمة كاملة للوظائف و الفلاتر يمكنك زيارة مرجع فلاتر . PHP يمكنك فحص كل فلتر لمشاهدة الخيارات و الرايات المتوفرة.

المدخلات المفعلة

دعنا نقوم بتفعيل مدخل للنموذج.

أول شيء يجب أن نقوم فيه هو تأكيد بأن بيانات الادخال التي نبحث عنها موجودة . ثم نقوم بفلترة البيانات المدخلة باستخدام الوظيفة . filter_input()

في المثال التالي سيتم ارسال مدخل المتغير "email" **الى صفحة:** PHP

```php
<?php
if(!filter_has_var(INPUT_GET, "email"))
{
echo("Input type does not exist");
}
else
{
if (!filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL))
{
echo "E-Mail is not valid";
}
else
{
echo "E-Mail is valid";
}
}
?>
```

شرح المثال

يحتوي المثال السابق على مدخل تم ارساله الى باستخدام طريقة: GET

1. **سيفحص فيما اذا مدخل البريد email فعال و موجود باستخدام النوع. GET**
2. **اذا كان المتغير موجود سيتم فحص فعالية البريد الالكتروني.**

تعقيم المدخلات

دعونا نقوم بتنظيف الرابط URL المرسل من النموذج.

أولاً يجب التأكيد على أن البيانات المدخلة التي نبحث عنها موجودة . ثم نقوم بتعقيم البيانات المدخلة من خلال استخدام الوظيفة filter_input()

في المثال التالي سيتم ارسال مدخل المتغير url الى صفحة: PHP

```php
<?php
if(!filter_has_var(INPUT_POST, "url"))
{
echo("Input type does not exist");
}
else
{
$url = filter_input(INPUT_POST,
"url", FILTER_SANITIZE_URL);
}
?>
```

شرح المثال

يحتوي المثال السابق على مدخل url تم ارساله الى باستخدام طريقة: POST

1. **سيفحص فيما اذا مدخل url فعال و موجود باستخدام النوع. POST**
2. **اذا كان المتغير موجود سيتم تعقيم (حذف الأحرف الغير فعالة) و حفظها في المتغير . $url**

اذا كان مدخلات المتغير نص عادي مثل http://www.exååmple.com/ سيتم اختيار الرابط الصحيح و تعقيمه من خلال المتغير $url كالتالي:

```
http://www.Example.com/
```

فلترة مدخلات متعددة

دائماً مايحتوي النموذج على واحد أو اكثر من مدخلات النصوص . ولتجنب استدعاء filter_var أو filter_input مرات و مرات يمكن استخدام filter_var_array أو. filter_input_array

في المثال التالي قيمنا باستخدام filter_input_array() لفلترة ثلاث متغيرات من نوع GET . المتغير من نوع GETT المستلم هو الاسم والعمر و عنوان البريد الالكتروني:

```
<?php
$filters = array
(
"name" => array
(
"filter"=>FILTER_SANITIZE_STRING
),
"age" => array
(
"filter"=>FILTER_VALIDATE_INT,
"options"=>array
(
"min_range"=>1,
"max_range"=>120
)
),
"email"=> FILTER_VALIDATE_EMAIL,
);
$result = filter_input_array(INPUT_GET, $filters);
if (!$result["age"])
{
echo("Age must be a number between 1 and 120.<br />");
}
elseif(!$result["email"])
{
echo("E-Mail is not valid.<br />");
}
else
{
echo("User input is valid");
}
?>
```

شرح المثال

يحتوي المثال السابق على ثلاث متغيرات (الاسم و العمر و البريد) تم ارسالهم الى البريد نفسه باستخدام الطريقة: GET

1. تم ضبط مصفوفة تحتوي على متغيرات وفلاتر اسم المدخل المستخدمة مع متغير لمدخل محدد.

2. استدعاء الوظيفة filter_input_array()مع متغيرات GET المدخلة و المصفوفة التي أنشأناها مسبقاً.
3. فحص متغيرات العمر age و البريد الالكتروني email في متغير $result للمدخلات الغير فعالة.

يمكن أن يكون القيمة الثانية في الوظيفة filter_input_array()مصفوفة أو فلتر IDD واحد.

اذا كان المتغير فلتر id واحد عندها سيتم فلترة جميع المتغيرات في مصفوفة مدخلات من خلال فلتر محدد.

اذا كان المتغير مصفوفة عندها يجب اتباع التعليمات التالية:

- يجب أن يكون مرفق بمصفوفة تحتوي على متغير مدخل مثل وسم المصفوفة.
- قيمة المصفوفة يجب أن تكون مفلترة id او مصفوفة تحدد فلتر او راية او خيار.

استدعاء الفلتر

يمكنك استدعاء وظيفة معرفة من خلال المستخدم و استخدامه كفلتر باستخدام ميزة FILTER_CALLBACK . وبهذه الطريقة لديك تحكم كامل بفلترة البيانات.

يمكنك إنشاء وظفة خاصة للمستخدم او استخدام وظائف PHP الحالية.

الوظيفة التي ترغب باستخدامها لفلترتها محددة بنفس الطريقة كخيار مخصص.

في المثال التالي سيتم إنشاء وظيفة تقوم بتحويل جميع "_" الى فراغات:

```php
<?php
function convertSpace($string)
{
return str_replace("_", " ", $string);
}
$string = "Peter_is_a_great_guy!";
echo filter_var($string, FILTER_CALLBACK,
array("options"=>"convertSpace"));
?>
```

ستظهر النتيجة على المتصفح كالتالي :

```
Peter is a great guy!
```

شرح المثال

في المثال التالي سيتم إنشاء وظيفة تقوم بتحويل جميع "_" الى فراغات:

- **إنشاء** function **لاستبدال** "_" **الى فراغات.**
- **استدعاء** filter_var()**مع الفلتر** FILTER_CALLBACK **و مصفوفة تحتوي على** functionn .

استخدام PHP معMySQL

مقدمةMySQL

تعتبر لغة MySQL أشهر نظام قواعد بيانات مفتوح المصدر .

ماهي لغة MySQL ؟

ان MySQL لغة خاصة بقواعد البيانات .

تخزن البيانات في MySQL في كائنات قواعد بيانات تسمى جداول .

ان الجدول هو عبارة عن مجموعة من البيانات المدخلة و المتصلة كما يتألف الجدول من أعمدة و صفوف .

تعتبر قواعد البيانات مفيدة جداً عند تخزين المعلومات بشكل تصنيفي أي عند استخدام شركة لقواعد البيانات يمكن أن تستخدم الجداول التالية :

"Employees" و "Products" و "Customers" و "Orders" .

جداول قواعد البيانات

تحتوي غالباً قواعد البيانات على جدول او أكثر يتم تعريف كل جدول باسم مثل (Customers أو Orders) . كما يحتوي الجدول على صفوف تمسى rows أو records بالاضافة الى البيانات .

في المثال التالي جدول يسمى Persons .

FIRSTNAME	LASTNAME	ADDRESS	COUNTRY
Mutaz	Hakmi	Timoteivn 10	SAR
Abdo	Mando	Borgvn 23	SAR
Abdo	Fajr	Storgt 20	SAR

يحتوي الجدول السابق على 4 أعمدة و 3 صفوف لكل شخص صف يتألف من الاسم و الكنية و العنوان و الدولة.

الأوامرQueries

الأوامر هي عبارة عن طلبات أو أسئلة.

باستخدام MySQL يمكن تقديم أوامر لقواعد البيانات معلومات معينة والتي تحتوي على امكانية تقديم النتائج من خلال الصفوف أو الأعمدة.

لاحظ الأمر التالي:

```
SELECT LastName FROM Persons
```

يختار الأمر السابق جميع البيانات الموجودة في عمود الكنية LastName من الجدول المسمى Personsثم سيقوم بإعادة النتيجة كالتالي :

LASTNAME
Hakmi
Mando
Fajr

تحميل قواعد البياناتMySQL

اذا لم يكن لديك تطبيق يحتوي على PHP مع قواعد البيانات MySQL يمكنك تحميلها من موقعها الرسمي من خلال الرابط التالي: http://www.mysql.com/downloads/

حقائق حول قواعد البياناتMySQL

أحد الحقائق المذهلة عن MySQL أنها تقوم بتخزين البيانات الضمنية الموجودة في التطبيقات و هذه الحقيقة مذهلة بسبب أم معظم الناس تعتقد أن MySQL تستخدم فقط مع التطبيقات التي تخزن بيانات صغيرة أو متوسطة.

لكن الحقيقة بأن MySQL تستخدم مع قواعد بيانات مع مواقع الانترنت التي تستخدم لتخزين هائل للبيانات و المستخدمين مثل Friendsterو Yahooو Google.

لمشاهدة الشركات التي تستخدم MySQL لقواعد البيانات من خلال الرابط التالي:
http://www.mysql.com/customers/

MySQL الاتصال مع قاعدة البيانات

تستخدم عادة MySQL مع لغة PHP .

إنشاء اتصال لقاعدة البيانات MySQL

قبل الوصول الى البيانات في قاعدة البيانات يجب إنشاء اتصال مع قاعدة البيانات .

باستخدام لغة PHP يمكن انشاء اتصال مع قاعدة البيانات من خلال الوظيفة()mysql_connect .

التركيبة

```
mysql_connect(servername,username,password);
```

المتغيرات	الشرح
servername	اختياري . يحدد وجهة الاتصال مع السيرفر و غالباً مايكون localhost
username	اختياري . يحدد اسم المستخدم مع السيرفر . يمكن استخدام افتراضياً root .
password	اختياري . يحدد كلمة المرور مع السيرفر و غالباً ماتكون فارغة اذا لم تعيين كلمة مرور خاصة مسبقاً .

ملاحظة : هناك الكثير من القيم المستخدمة مع قواعد البيانات ولكن في الجدول أهمها فقط لكن لمزيد من المعلومات يمكنك زيارة مراجع. PHP MySQL

مثال

في المثال التالي سنقوم بتخزين الاتصال مع قاعدة البيانات من خلال المتغير $con من أجل استخدامات لاحقة في السكربت بالاضافى الى استخدام die() من أجل معرفة نوع الخطأ مع قواعد البيانات في حال فشل الاتصال:

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}

// some code

?>
```

إنهاء الاتصال

سيتم اغلاق الاتصال بشكل تلقائي عند انتهاء السكربت لكن اذا أردت انهاء الاتصال يمكن استخدام الوظيفة : mysql_close()

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
// some code
mysql_close($con);

?>
```

إنشاء قاعدة بيانات و الجداول

تحتوي قاعدة البيانات على جدول أو اكثر .

إنشاء قاعدة بيانات

يمكن إنشاء قاعدة بيانات من خلال التعبير CREATE DATABASE .

التركيبة

```
CREATE DATABASE database_name
```

لاستخدام التعبير يمكن استخدام الوظيفة ()mysql_query و بهذه الوظيفة يمكن استخدام لارسال الأوامر الى الاتصال مع MySQL.

مثال

في المثال التالي سنقوم بإنشاء قاعدة بيانات تسمى my_db

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con) {
die('Could not connect: ' . mysql_error());
}
if (mysql_query("CREATE DATABASE my_db",$con))
{
echo "Database created";
} else {
echo "Error creating database: " . mysql_error();
}
mysql_close($con);

?>
```

إنشاء جدول

يمكن إنشاء قاعدة بيانات من خلال التعبير . CREATE TABLE

التركيبة

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
)
```

لإنشاء الأمر يجب اضافة التعبير CREATE TABLE الى الوظيفة. mysql_query()

مثال

في المثال التالي سيتم إنشاء جدول مسمى Persons مع ثلاثة أعمدة . ستكون أسماء الأعمدة كالتالي FirstNameو LastName و: Age

```
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
// Create database
if (mysql_query("CREATE DATABASE my_db",$con))
{
echo "Database created";
}
else
{
echo "Error creating database: " . mysql_error();
}
// Create table
mysql_select_db("my_db", $con);
$sql = "CREATE TABLE Persons
(
FirstName varchar(15),
LastName varchar(15),
Age int
)";

// Execute query
mysql_query($sql,$con);
mysql_close($con);

?>
```

ملاحظة هامة:

- **يجب اختيار قاعدة البيانات قبل إنشاء الجدول حيث يمكن اختيار قاعدة البيانات من خلال الوظيفة.** mysql_select_db()
- **عند إنشاء حقل قاعدة بيانات بنوع** varchar , **يجب تحديد الحد الأقصى للطول في الحقل مثل .** varchar(15)
- **تحدد نوع البيانات نوع العمود الذي سيقوم بحمل البيانات فيها.**

الحقول Primary Keys وAuto Increment

يجب أن يحتوي كل حقل على وسم أساسي. Primary Keys

يستخدم primary key **لتحديد فريد ومميز للصفوف في الجدول . يجب أن يكون** primary key **فريد ضمن الجدول بالاضافة بأن** primary key **لايجب أن يكون فارغ** null **لأن محرك قاعدة البيانات يحتاج الى قيمة لتحديد مكان الصف في الجدول.**

في المثال التالي سيتم تحديد حقل لعنوان id **للجدول** personID **مثل حقل .** primary key **غالباً مايكون** primary key **يحتوي على رقم** ID **و غالباً مايكون مع .** AUTO_INCREMENT **تزيد** AUTO_INCREMENT**تلقائياً قيمة الحقل برقم واحد في كل مرة يزيد عدد الصفوف . وذلك للتأكد بأن** primary key **ليس فارغاً** null **يجب تحديد ضبط بأن القيمة غير فارغة.** NOT NULL

مثال

```
$sql = "CREATE TABLE Persons
(
personID int NOT NULL AUTO_INCREMENT,
PRIMARY KEY(personID),
FirstName varchar(15),
LastName varchar(15),
Age int
)";
mysql_query($sql,$con);
```

إدراج قيمة الى قاعدة البيانات

تستخدم INSERT INTO لإدراج صف جديد في الجدول .

ادراج بيانات الى جدول قاعدة البيانات

تستخدم INSERT INTO لإدراج صف جديد في الجدول .

التركيبة

من الممكن كتابة التعبير INSERT INTO من خلال طريقتين .

لايحدد الشكل الأول اسماء الأعمدة حيث سيتم ادراج البيانات فقط قيمهم :

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
```

يحدد الشكل الثاني كلاً من اسماء الأعمدة و قيمهم التي سيتم ادراجها :

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

لاستخدام التعابير السابقة بشكل جيد يمكن استخدام الوظيفة mysql_query()حيث سيتم ارسال الأوامر و الطلبات الى الاتصال الخاص مع MySQLL.

مثال

في الدرس السابق قمنا بإنشاء جدول باسم Persons مع ثلاثة أعمدة Firstname و Lastname و Ageكما سنقوم باستخدام نفس الجدول في هذا المثال . يضيف المثال التالي صف جديد للجدول Persons :

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Root', 'Griffin', '35')");
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Glenn', 'Quagmire', '33')");
mysql_close($con);

?>
```

ادراج بيانات من نموذج الى قاعدة البيانات

الآن سنقوم بإنشاء نموذج HTML والذي بدوره سيقوم بإضافة صف جديد للجدول Persons.

هنا كود نموذج: HTML

```
<html>
<body>
<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body>
</html>
```

عند ضغط المستخدم على زر الارسال في نموذج HTML سيتم ارسال المعلومات الى الملف insert.php

.

يقوم الملف insert.php بالاتصال بقاعدة البيانات ثم يحوّل المعلومات المسجلة في النموذج الى متغيرات $_POST .

ثم يرسل mysql_query()أمراً الى INSERT INTO ليتم ادراج صف جديد الى جدول. Personss

هناالكود الموجود في صفحة : insert.php

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES
('$_POST[firstname]','$_POST[lastname]','$_POST[age]')";
if (!mysql_query($sql,$con))
{
die('Error: ' . mysql_error());
}
echo "1 record added";
mysql_close($con)

?>
```

SELECT الاختيار

يستخدم التعبير SELECT لاختيار البيانات من قاعدة البيانات

اختيار البيانات من جدول قاعدة البيانات

يستخدم التعبير SELECT لاختيار البيانات من قاعدة البيانات .

التركيبة

```
SELECT column_name(s)
FROM table_name
```

لاستخدام التعبير يمكن استخدام الوظيفة mysql_query() و بهذه الوظيفة يمكن استخدام لارسال الأوامر الى الاتصال مع MySQLL.

مثال

يختار المثال التالي جميع البيانات المخزنة في جدول Persons تستخدم الاشارة * لاختيار جميع البيانات الموجودة في الجدول:

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
while($row = mysql_fetch_array($result))
{
echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />";
}
mysql_close($con);

?>
```

يخزن المثال السابق البيانات العائدة من خلال الوظيفة ()mysql_queryفي المتغير .$resultثم سنقوم باستخدام ()mysql_fetch_arrayلاسترجاع الصف الثاني من صفوف الجدول . أما الحلقة whileeفإنها ستدور حول جميع الصفوف في جدول البيانات وتقدم النتيجة.

لعرض النتيجة لجميع الصفوف يمكن استخدام متغير $rowكالتالي:

```php
($row['FirstName'] and $row['LastName'])
```

ستظهر النتيجة كالتالي:

Mutaz Hakmi

Abdo Mando

Abdo Fajr

عرض النتيجة في جدول HTML

يختار المثال التالي نفس البيانات في المثال السابق لكن سيتم عرض البيانات في جدول HTML كالتالي :

```
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>";
while($row = mysql_fetch_array($result))
{
echo "<tr>";
echo "<td>" . $row['FirstName'] . "</td>";
echo "<td>" . $row['LastName'] . "</td>";
echo "</tr>";
}
echo "</table>";
mysql_close($con);

?>
```

استخدام Where

تستخدم العبارة WHERE لفلترة الصفوف في جداول قاعدة البيانات .

عبارة WHERE

تستخدم عبارة WHERE لتحديد صف معين في جدول قاعدة البيانات .

التركيبة

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value
```

و هي تستخدم لارسال mysql_query()لاستخدام التعبير السابق يمكن استخدام PHP لإنشاء كود MySQLL الأوامر أو الطلبات الى اتصال .

مثال:

يختار المثال التالي جميع الصفوف من الجدول Persons حيث WHERE الاسمMutaz

```
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons
WHERE FirstName=mutaz');
while($row = mysql_fetch_array($result))
{
echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />";
}

?>
```

ستظهر النتيجة على المتصفح كالتالي:

```
Mutaz Hakmi
```

الترتيبOrder By

تستخدم الترتيب ORDER BY لترتيب البيانات المعروضة من قاعدة البيانات .

الترتيب Order By

تستخدم الترتيب ORDER BY لترتيب البيانات المعروضة من قاعدة البيانات .

عند استخدام ORDER BY يكون الترتيب تصاعدي بشكل تلقائي ولكن ان أردت الترتيب بشكل آخر مكن استخدام الكلمة DESC والتي تشكل ترتيب تنازلي .

التركيبة

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s) ASC|DESC
```

الترتيب من خلال عمودين

أيضاً من الممكن ترتيب أكثر من عمود . عند استخدام الترتيب من خلال أكثر من عمود فإن العمود الثاني سيتم استخدامه اذا كانت القيمة الأولى متساوية:

```
SELECT column_name(s)
FROM table_name
ORDER BY column1, column2
```

Update تحديث القاعدة

تستخدم خاصية التحديث UPDATE للتعديل على البيانات في الجدول .

تحديث البيانات في قاعدة البيانات

تستخدم خاصية التحديث UPDATE للتعديل على البيانات الحالية في الجدول .

التركيبة

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

ملاحظة : لاحظ بأن التعبير WHERE يحدد أي الصف أو الصفوف الذي ينبغي تحديثه لكن بينما اذا لم تستخدم WHEREE عندها سيتم تحديث جميع الصفوف.

و هي تستخدم لارسال mysql_query()لاستخدام التعبير السابق يمكن استخدام PHP لإنشاء كود الأوامر أو الطلبات الى اتصال MySQLL .

الحذف Delete

يستخدم الحذف DELETE لحذف بيانات في صفوف الجدول .

حذف بيانات من قاعدة البيانات

يستخدم التعبير DELETE FROM لحذف بيانات من جدول قاعدة البيانات .

التركيبة

```
DELETE FROM table_name
WHERE some_column = some_value
```

ملاحظة : لاحظ بأن التعبير WHERE يستخدم مع تركيبة الحذف DELETE لتحديد معين أي البيانات التي سيتم حذفها من الجدول ولكن ان لم تستخدم WHERE سيتك حذف جميع صفوف الجدول.

لإنشاء كود PHP لاستخدام التعبير السابق يمكن استخدام ()mysql_queryو هي تستخدم لارسال الأوامر أو الطلبات الى اتصال. MySQLL