

PHP

Personal Home Page



برمجة مواقع الإنترنت باستخدام لغة PHP

مع تحيات

تعلم البرمجة

www.learn-barmaga.com

المحتويات

- الدرس الأول : ما هي لغة PHP ؟
- الدرس الثاني : المزودات
- الدرس الثالث: إنشاء أول صفحة PHP
- الدرس الرابع: التعامل مع الوقت والتاريخ
- الدرس الخامس: الحلقات
- الدرس السادس: الشروط
- الدرس الثامن: المصفوفات
- الدرس السابع: ضع تعليقات في برامجك
- الدرس التاسع: الوظائف
- الدرس العاشر: تمرير المتغيرات من خلال عنوان الموقع (URL)
- الدرس الحادي عشر: تغيير المتغيرات من خلال النماذج
- الدرس الثاني عشر: الجلسات (Sessions)
- الدرس الثالث عشر: الكوكيز (Cookies)
- الدرس الرابع عشر: نظام الملفات
- الدرس الخامس عشر : القراءة من ملف نصي
- الدرس السادس عشر: الكتابة في ملف نصي
- الدرس السابع عشر: قواعد البيانات
- الدرس الثامن عشر : إنشاء قاعدة البيانات والجداول
- الدرس التاسع عشر : إدخال البيانات في قاعدة البيانات
- الدرس العشرون : استرجاع البيانات من قاعدة البيانات
- الدرس الواحد والعشرون : حذف البيانات من قاعدة البيانات
- الدرس الثاني والعشرون : تحديث البيانات في قاعدة البيانات

مقدمة



المصطلح PHP هو اختصار للكلمات

Personal Home Page وهي عبارة عن لغة برمجة تعطيك الحرية لإضافة خصائص متقدمة في موقعك الإلكتروني.

الهدف من هذا الدرس هو توفير مقدمة سهلة متكاملة ودقيقة للغة PHP ، الدرس يبدأ من الصفر لكنه يتطلب منك أن يكون لديك معرفة جيدة بلغة HTML ، إن كنت لا تعرف HTML فعليك أن تبدأ [بدرس HTML](#).

PHP يمكن استخدامها في الكثير من الحالات، لإنشاء المنتديات، الاستفتاءات، المتاجر، بوابات للرسائل النصية القصيرة، قوائم بريدية وغير ذلك، الحد الوحيد لما يمكنك فعله في PHP هو خيالك، PHP ليست صعبة لكن تذكر أنها أكثر تطوراً وتطلباً من HTML لذلك الصبر عند تعلمها ضروري..

هذا الدرس لا يمكنه تعليمك كل شيء، لذلك المطلوب منك أن تشارك وتجرب بنفسك، إن احتجت لمساعدة فننصحك بزيارة [المنتديات](#) للحصول على دعم، في المنتديات ستلتقي بخبراء يمكنهم مساعدتك بالتلميحات والمقترحات والنصائح.

ما المطلوب؟

نفترض أنك تملك محرراً نصياً وتعرف كيف تستخدمه.

بعد ذلك أنت بحاجة لحاسوب أو مزود يمكنه تشغيل PHP ، على النقيض من HTML و CSS لغة البرمجة PHP لا تتأثر بأي متصفح يستخدمه زوار موقعك بل بنوع المزود الذي يستضيف موقعك، لأن PHP هي تقنية من جهة المزود.

في الدروس القليلة القادمة ستتعلم كيف تعمل PHP وكيف تعد حاسوبك لتشغيلها، بعد ذلك ستتعلم وظائفها وطرقها.

عندما تنتهي من هذا الدرس، ستتمكن من البرمجة باستخدام PHP وبالتالي ستحصل على إمكانيات غير محدودة لإضافة التفاعل في صفحات موقعك.

استمتع!

الدرس الأول : ما هي لغة PHP ؟

عندما يبدأ أي شخص في تعلم PHP السؤال الذي يتكرر طرحه هو: ما هي PHP ؟ وكيف تعمل؟

هذا بالضبط ما سنجيب عنه في هذا الدرس، من المفيد أن تفهم الأساسيات المتعلقة بلغة PHP قبل أن تبدأ تطوير صفحات موقعك بها، فهم الأساسيات سيعجل من سرعة تعلمك بشكل كبير.

لنبدأ!

ما هي لغة PHP ؟

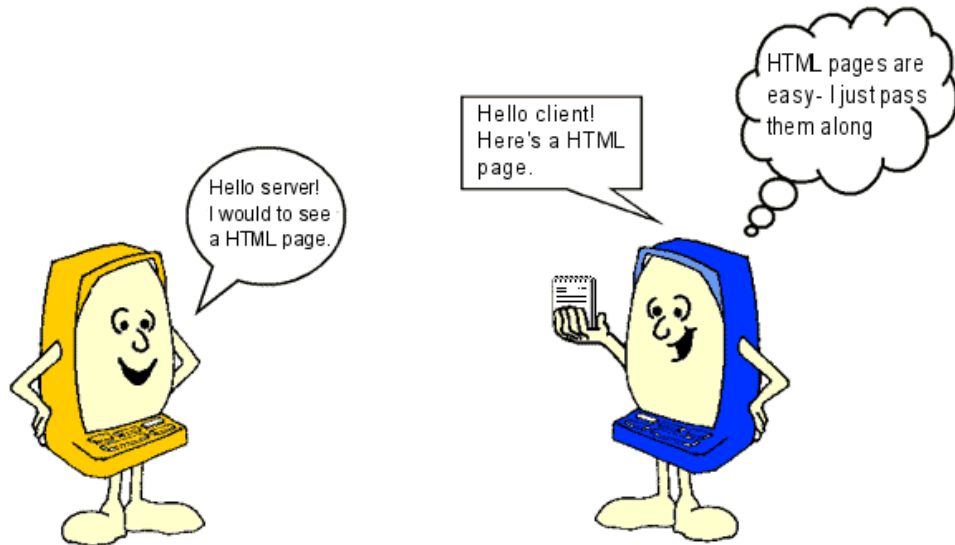
PHP هي اختصار **Personal Home Pages** ، لكنها الآن تعني **PHP: Hypertext Preprocessor**. طورت في البداية من قبل المطور الدينماركي راسموس ليدروف، بعد ذلك تحولت لمشروع برنامج حر، PHP ليست من المعايير القياسية للويب لكنها تقنية مفتوحة المصدر، PHP ليست لغة برمجة حقيقية لكنها تسمح لك بدمجها في صفحات موقعك.

لوصف ما هي صفحة PHP يمكنك أن تقول بأنها ملف بامتداد **.php**. يحوي خليطاً من HTML ولغة البرمجة PHP وهي تعمل على مزود ويب.

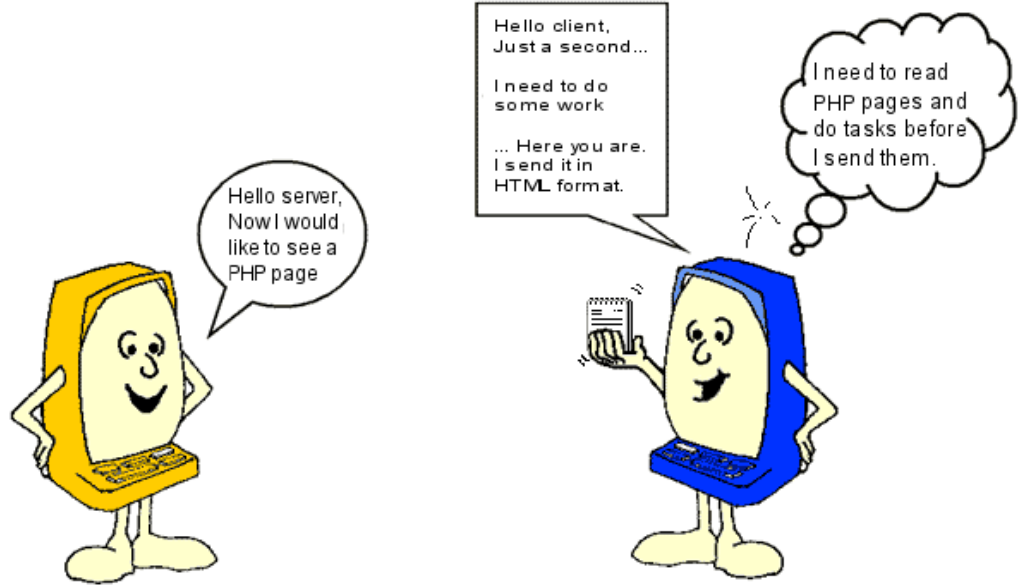
ملاحظة المترجم: المقصود هنا أن PHP ليست لغة برمجة كبقية اللغات الأخرى، بل هي لغة متخصصة لتطوير المواقع فقط وهي لغة مرنة يمكن دمجها في صفحات المواقع.

كيف تعمل PHP ؟

أفضل طريقة لشرح كيف تعمل PHP هي بمقارنتها مع HTML ، تصور أنك كتبت عنوان صفحة HTML (<http://www.mysite.com/page.htm>) في حقل العناوين في المتصفح، بهذه الطريقة أنت **تطلب** صفحة HTML ، يمكن تصوير هذه العملية بهذا الشكل:



كما ترى المزود ببساطة يرسل ملف HTML إلى الزبون أي إلى حاسوب الزائر، لكن إذا كتبت - <http://www.mysite.com/page.php> أي طلبت صفحة - PHP سيحتاج المزود لمزيد من بذل الجهد:



المزود عليه أولاً أن يقرأ ملف PHP بعناية ليرى إن كان هناك أي مهمات عليه تنفيذها، فقط عندما ينتهي المزود مما يجب عليه فعله سيرسل النتيجة إلى الزبون، من الضروري أن تفهم بأن الزبون يرى فقط نتيجة عمل المزود وليس الأوامر الموجودة في ملف PHP.

هذا يعني أنك إن نقرت على "view source" أو "استعراض المصدر" في صفحة PHP فلن ترى أوامر PHP بل فقط وسوم HTML ، لذلك لا يمكنك أن ترى كيف كتبت صفحة PHP عندما تحاول استعراض مصدرها، عليك أن تتعلم اللغة بطريقة مختلفة، بقراءة هذا الدرس مثلاً.

ما تتعلمه في هذا الدرس هو ما تكتبه من أوامر للمزود!

في البداية عليك أن تحصل على مزود، لكن لا تقلق، لا يجب عليك شراء حاسوب جديد، أنت بحاجة فقط إلى تثبيت برنامج على حاسوبك ليحمله يعمل كمزود، الخيار الآخر هو أن تملك موقعاً مستضافاً على مزود يدعم PHP ، عندها عليك أن تكون على اتصال بالإنترنت عندما تبرمج.

الدرس التالي سيكون حول كيفية إعداد حاسوبك ليعمل كمزود.

الدرس الثاني : المزودات

PHP هي تقنية من جهة /المزود، لذلك تحتاج لمزود يشغل PHP ، لكن هذا لا يعني أنك بحاجة لشراء أي شيء، هناك خيارات مختلفة للحصول على مزود.

أنت تحتاج لخيار واحد، هذا الدرس مقسم إلى ثلاثة أقسام، الأول مقدمة للخيارات الثلاثة واختر منها ما يناسبك، عندما يكون لديك مزود يعمل سنكمل الدرس في الجزء الثالث لنكتب أول صفحة PHP.

الخيار ١: موقع مستضاف على مزود

يمكنك أن تختار مزوداً يستضيف موقعك ويدعم PHP.

- [اختبر مزود موقعك لتعرف إن كان يدعم PHP](#)
- إن لم يكن لديك موقع في مزود يمكنك أن تسجل في حساب مجاني على 000webhost.com وهي خدمة استضافة تدعم PHP.

الخيار ٢: تثبيت PHP على حاسوبك

تثبيت PHP على حاسوبك ليس نزهة في الحديقة، هذا الخيار مناسب لخبراء مستخدمي الحاسوب، ويمكن بالطبع إنجاز تثبيت PHP على حاسوبك، هنا ستجد روابط لتنزيل PHP ودليلاً لتثبيتها على أنظمة مختلفة:

- [دليل التثبيت على ويندوز](#)
- [دليل التثبيت على ماك](#)
- [دليل التثبيت على لينكس](#)

خيار ٣: XAMPP :

XAMPP هو برنامج يجعل من السهل والممكن لنا أن نشغل PHP على حواسيبنا دون أن نثبتها بأنفسنا.

- [تعلم كيف تثبت XAMPP](#)


الدرس الثالث: إنشاء أول صفحة PHP

من الدرس الأول والثاني تعلمت القليل حول ماهية لغة PHP ، ثبت مزوداً على حاسوبك أو يمكنك الوصول لمزود يدعم هذه اللغة، الآن أنت جاهز لتبدأ في كتابة أول صفحة PHP ، الدرس سيكون سهلاً وبسيطاً، لكن بعد أن تنجز الدرس ستفهم ما هي لغة PHP وماذا يمكنك إنجازه باستخدامها.

ببساطة ملف PHP هو ملف نصي بامتداد **.php** ويتألف من:

- نص
- وسوم HTML
- برمجة بلغة PHP

أنت تعرف مسبقاً ما هو النص وما هي وسوم HTML ، لذلك لنلقي نظرة على برمجة PHP.

برامج PHP فريق توثيق PHP أصدر  وثائق مفصلة للغة PHP ، خلال الدرس سنضع كثيراً من الروابط لهذه الوثائق، الهدف هو أن تعتاد على البحث وإيجاد الأجوبة لأسئلتك، PHP لغة واسعة ولا يمكنك أن تتعلم كل إمكانياتها في هذا الدرس، لكنها ليست لغة صعبة، على العكس، PHP تبدو في بعض الأحيان كاللغة الإنجليزية.

لنبدأ بإنشاء صفحة PHP الأولى.

مثال: Hello World :

ابدأ في إنشاء صفحة HTML عادية لكن أعطي للملف اسم *page.php* واحفظه في المجلد الرئيسي للموقع:

- إن كنت تستخدم XAMPP اقرأ [الدرس ٢](#)، المسار إلى المجلد الرئيسي هو "c:\xampp\htdocs\page.php" في حاسوبك الذي يعمل كمزود الآن. [اقرأ المزيد حول حفظ ملفات PHP في XAMPP](#).
- أن كنت تستخدم استضافة ببساطة ارفع الملف إلى مزود موقعك.

وسوم HTML يجب أن تكون بهذا الشكل:

```
<html>
<head>
<title>My first PHP page</title>

</head>
<body>

</body>
</html>
```

كما نتذكر من الدرس الأول، كل شيء في PHP متعلق بكتابة أوامر للمزود، لذلك دعنا نكتب بعض الأوامر.

في البداية نحن بحاجة لأن نخبر المزود أين ستبدأ PHP ومتى ستتوقف في PHP. نستخدم وسوم `<?php` و `?>` لتحديد بداية ونهاية أوامر PHP التي يجب على المزود تنفيذها، في معظم المزودات استخدام `?>` كوسم بداية فقط سيكون كافياً لكن `<?php` هي الطريقة الأفضل لكي عندما نستخدم PHP لأول مرة في الملف.

الآن أضف التالي لوسوم HTML:

```
<html>
<head>
<title>My first PHP page</title>
</head>
<body>

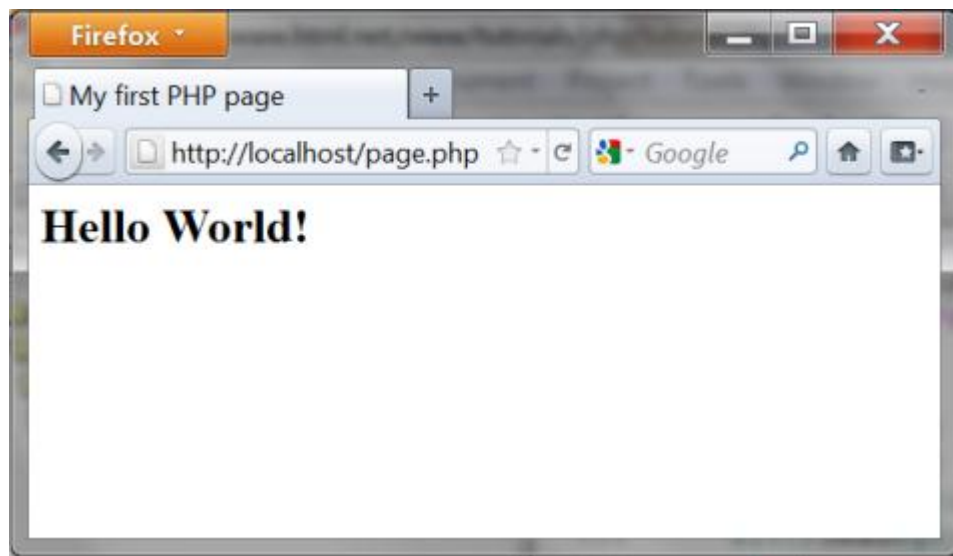
<?php

echo "<h1>Hello World!</h1>";

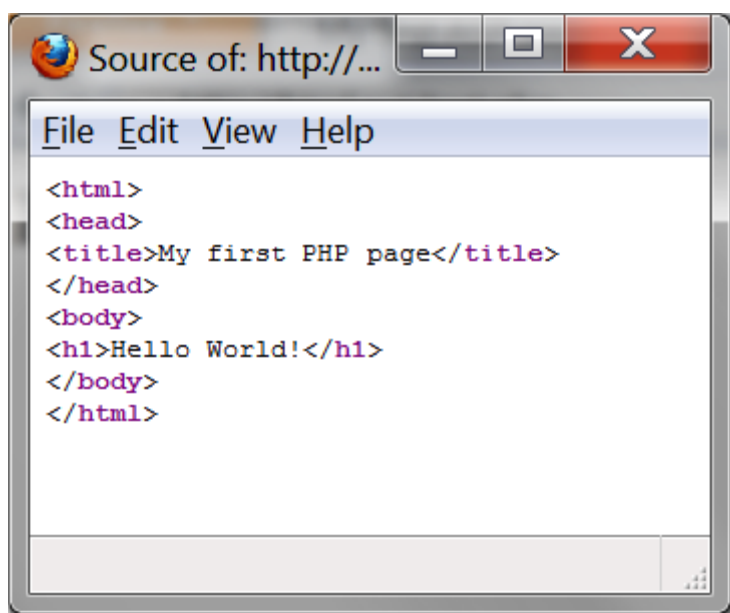
?>

</body>
</html>
```

عندما تستعرض ملف PHP في المتصفح يجب أن يكون الملف بهذا الشكل:



الأمر يصبح مثيراً أكثر عندما تنتظر لمصدر الصفحة في المتصفح باختيار: "view source"



أوامر PHP ذهبت! كما تتكرر في الدرس الأول المزود وحده الذي يستطيع أن يرى أوامر PHP ، - الزبون (أي المتصفح) يرى النتائج فقط!

لنلقي نظرة على ما حدث، نحن طلبنا من المزود أن يكتب <h1> Hello World!</h1>، بلغة تقنية أكثر سنقول بأننا استخدمنا الوظيفة [echo](#) لنكتب سلسلة محارف محددة للزبون (المتصفح) والفاصلة المنقوطة تنهي الأمر، لكن لا تخف، في هذا الدرس سنحاول أن نبقي اللغة التقنية في حدها الأدنى.

في مثالنا الأول من الواضح أنه ليس مثيراً للحماس، لكن انتظر فقط! من الآن وصاعداً ستزداد الأمور حماساً، لنلقي نظرة على مثال آخر.

مثال: الآن!

لنجعل المزود يكتب شيئاً آخر، يمكننا مثلاً أن نطلب منه أن يكتب تاريخ اليوم والوقت:

```
<html>
<head>
<title>My first PHP page</title>

</head>
<body>

<?php

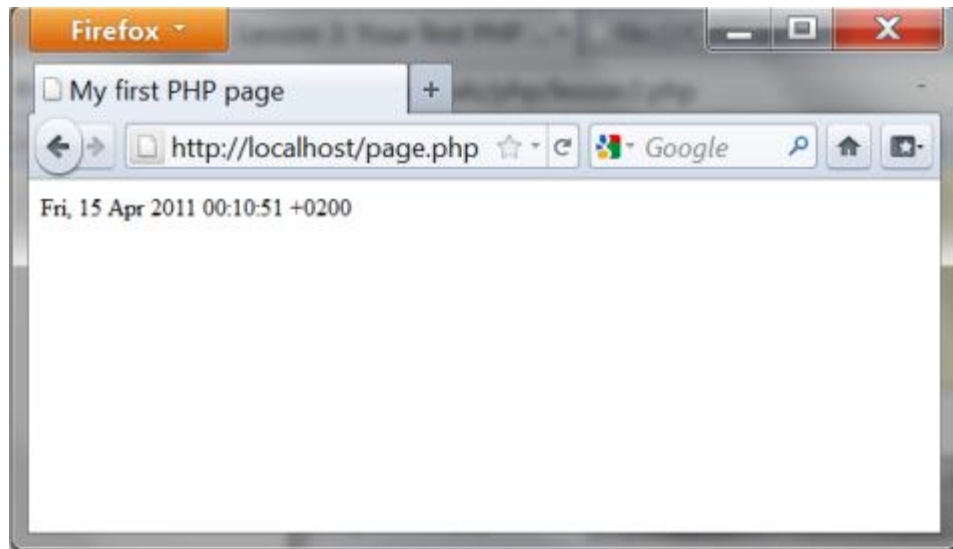
echo date("r");
```

```
?>
```

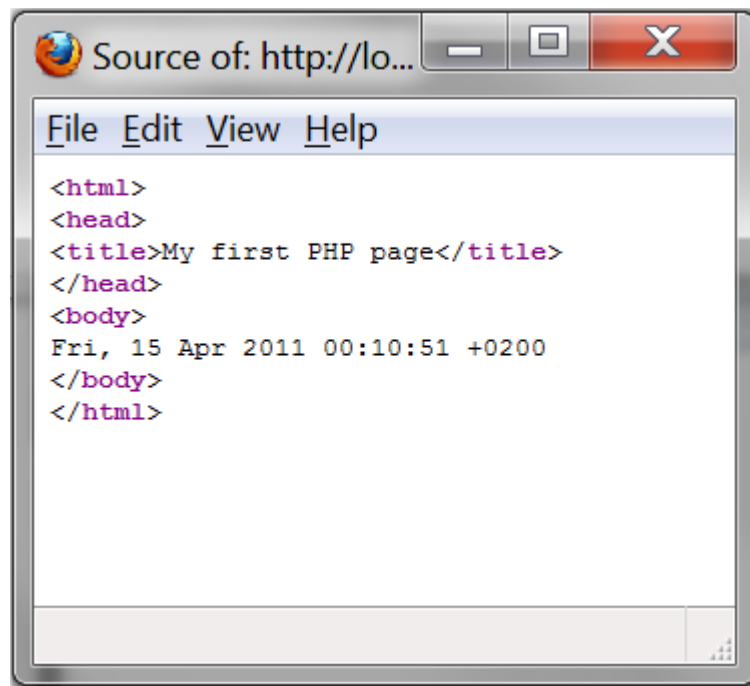
```
</body>
```

```
</html>
```

هذا سيظهر بهذا الشكل في المتصفح:



ووسوم HTML في المقابل ستكون:



الآن الأمور تبدو مسلية أكثر، أليس كذلك؟

جعلنا المزود يكتب التاريخ والوقت عندما تعرض صفحة PHP ، لاحظ أنك إذا جربت تحديث الصفحة سترى وقتاً جديداً يكتب، المزود يكتب التاريخ والوقت الحالي في كل مرة ترسل الصفحة إلى الزبون.

من المهم أيضاً أن تلاحظ بأن مصدر HTML يحتوي فقط التاريخ وليس أوامر PHP ، لذلك المثال لا يتأثر بأي متصفح استخدم، في الحقيقة كل الخصائص التي تعتمد على تقنيات جهة المزود ستعمل في كل المتصفحات!

ومرة أخرى لاحظ الفاصلة المنقوطة بعد سطر الأمر، هذه الفاصلة مهمة جداً ويجب أن تضاف وإلا لن يعمل البرنامج.

في المثال استخدمنا وظيفة `date` ، وهي وظيفة تعطينا الوقت والتاريخ الحاليين في المزود.

لنحرب إضافة المزيد للمثال من خلال كتابة سلسلة محارف ووظيفة برمجية - يفصل بينهما "." (نقطة) - هكذا:

```
<html>
<head>
<title>My first PHP document</title>
</head>
<body>

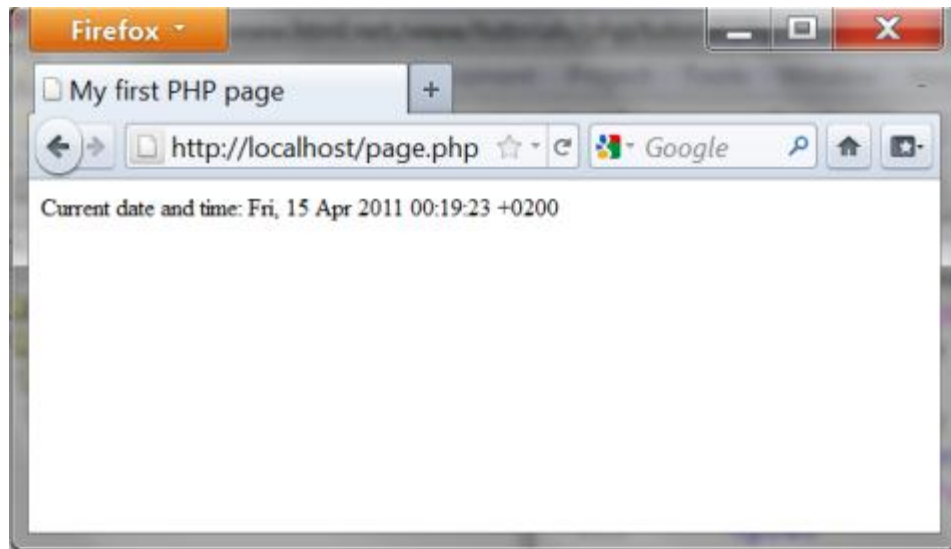
<?php

    echo "<p>Current date and time: " . date("r")
. "</p>";

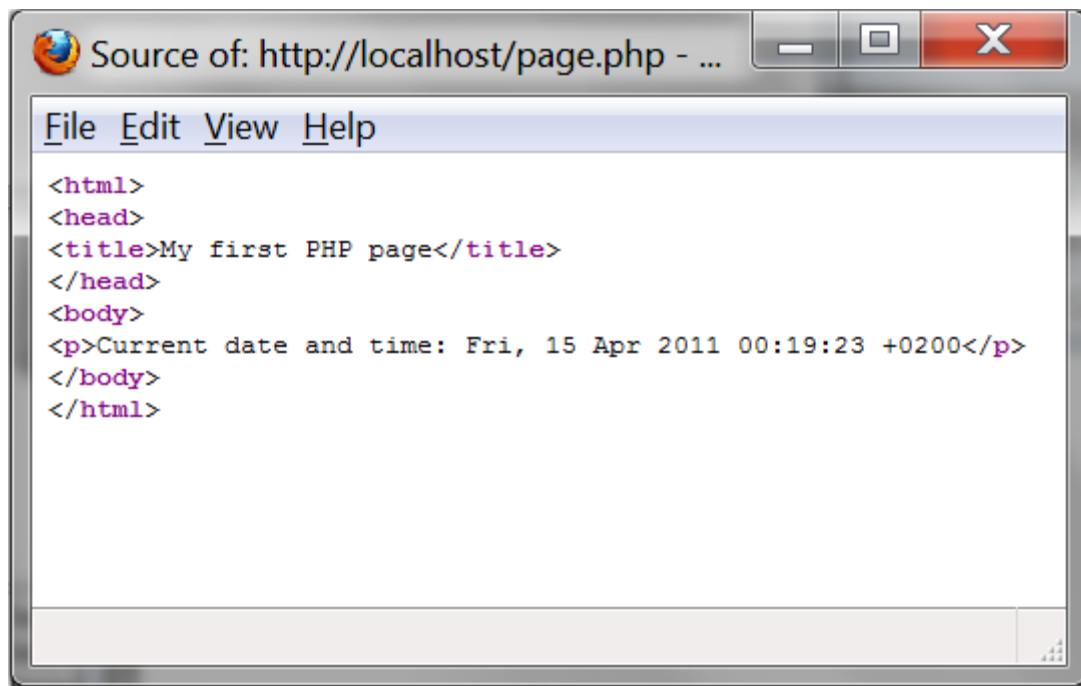
?>

</body>
</html>
```

ستظهر بهذا الشكل في المتصفح:



ووسوم HTML في المقابل ستكون:



في الدرس التالي سنلقي نظرة مفصلة أكثر على وظيفة [date](#) والأشكال المختلفة للتاريخ والوقت.

الدرس الرابع: التعامل مع الوقت والتاريخ

في هذا الدرس سنلقي نظرة على الخيارات المختلفة للتعامل مع الوقت والتاريخ في PHP ، في الدرس السابق استعرضنا أمثلة بسيطة لنشرح ماهية اللغة، في هذا الدرس سنلقي نظرة على وظيفة `date`.

وظائف الوقت والتاريخ

PHP تقدم مجموعة واسعة من الوظائف المتعلقة بالوقت والتاريخ، في هذا الدرس سنلقي نظرة على أهم هذه الوظائف `date`:

مع وجود الكثير من العوامل المختلفة يمكن لوظيفة `date` أن تعطي التاريخ الحالي أو الوقت بأشكال مختلفة، بعض أكثر العوامل المفيدة هي:

`date("y")`

15 :يعطينا السنة الحالية، وبتاريخ اليوم فهو يعطينا

`date("m")`

12 :يعطينا الشهر الحالي، وبتاريخ اليوم فهو يعطينا

`date("F")`

December :يعطينا اسم الشهر الحالي، وبتاريخ اليوم فهو يعطينا

`date("d")`

13 :يعطينا اليوم الحالي في الشهر، وبتاريخ اليوم فهو يعطينا

`date("l")`

Sunday :يعطينا اسم اليوم، وبتاريخ اليوم فهو يعطينا

`date("w")`

0 :يعطينا رقم اليوم في الأسبوع، وبتاريخ اليوم فهو يعطينا

`date("H")`

07 :يعطينا الساعة الحالية، وبالتوقيت الحالي فهو يعطينا

`date("i")`

42 :يعطينا الدقيقة الحالية، وبالتوقيت الحالي فهو يعطينا

`date("s")`

52: يعطينا الثانية الحالية، وبالتوقيت الحالي فهو يعطينا

هذا المثال يوضح استخدام وظيفة: [date](#)

```
<html>
<head>
<title>Time and date</title>

</head>
<body>

<?php

echo "<p>Today it's " . date("l") . "</p>";

?>

</body>
</html>
```

[شاهد المثال](#)

الوقت هو ١٤٤٩٩٨٨٩٧٢

وظيفة [time\(\)](#) يعطينا الوقت الحالي كعدد من الثواني منذ الواحد من يناير ١٩٧٠ وبالتحديد من الساعة ١٢ مساء بتوقيت غرينتش.

```
<html>
<head>
<title>time and date</title>
</head>
<body>

<?php

echo "<p>It's been exactly " . time() . "
seconds since January 1, 1970, 12:00 PM, GMT </ p>
";

?>

</body>
</html>
```

👉 شاهد المثال

الوقت المعبر عنه بعدد الثواني منذ ١ يناير ١٩٧٠ الساعة ١٢ مساء بتوقيت غرينتش يسمى "timestamp" أو الطابع الزمني ويسمى أيضاً الطابع الزمني ليونكس وهو مفيد عند التعامل مع التواريخ والأوقات في المستقبل أو الماضي.

تلقائياً وظيفة `date` تستخدم الطابع الزمني الحالي أي القيمة الحالية لوظيفة `time()` ، لكن من خلال عوامل إضافية يمكنك أن تحدد طابعاً زمنياً مختلفاً وهكذا يمكنك التعامل مع الماضي والمستقبل، في المثال أدناه حددنا الثانية صفر من تاريخ ١ يناير ١٩٧٠ الساعة ١٢ مساء بتوقيت غرينتش، هكذا يمكننا معرفة أي يوم كان هذا:

```
<html>
<head>
<title>time and date</title>
</head>
<body>

<?php

    echo "<p>January 1, 1970 was a " . date("l",0)
    . "</p>";

?>

</body>
</html>
```

👉 شاهد المثال

يزداد تعقيد عد الثواني منذ ١ يناير ١٩٧٠ إلى وقت محدد في المستقبل أو الماضي، إلا إن كنت عبثي رياضي! لكن يمكنك لوظيفة مفيدة أن تساعدك `mktime` ، التي يمكنها أن تحسب عدد الثواني.

طريقة كتابة قيم `mktime` (هي) ساعة، دقيقة، ثانية، شهر، يوم، عام)، المثال أدناه يحول الوقت لأول خطوة على القمر ٢١ يوليو ١٩٦٩ في ٠٢:٥٦:

```
<html>
<head>
<title>time and date</title>
```

```

</head>
<body>

<?php

echo mktime (2,56,0,7,21,1969) ;

?>

</body>
</html>

```

[شاهد المثال](#)

لاحظ أنه يعطينا رقماً سلبياً لأن التاريخ الذي طلبناه كان قبل ١ يناير ١٩٧٠.

يمكننا أن نجمع بين هذه الوظيفة ووظيفة [date](#) لنجد أي يوم وقع هذا الحدث التاريخي.

```

<html>
<head>
<title>time and date</title>
</head>
<body>

<?php

echo date ("l", mktime (2,56,0,7,21,1969)) ;

?>

</body>
</html>

```

[شاهد المثال](#)

كيف يمكنك استخدامها؟

كل ما سبق يبدو نظرياً حتى الآن، كيف يمكنك أن تستخدم وظيفة مثل [time\(\)](#) ، الأهم من ذلك متى سنتعلم شيئاً يمكنك أن فعلاً أن تضيفه في صفحاتك؟

الإجابة هي أن ما تتعلمه هنا هي لبناء البناء الأساسية، الحد الوحيد لما يمكنك فعله بها هو إبداعك وخيالك! سأجرباً لأقول بأنك تعلمت أكثر مما تظن، مثلاً هل تظن أنه بإمكانك إنشاء موقع يعرض خلفية للصفحة تتغير بحسب أيام الأسبوع ويعمل على كل المتصفحات؟

بالتأكيد يمكنك، ألق نظرة على المثال:

```
<html>
<head>
<title>time and date</title>
</head>

<body      background="background_<?php      echo
date("w");  ?>.png">

</body>
</html>
```

[شاهد المثال](#) ➡

المثال أعلاه بصور متغيرة، ببساطة يتطلب منك أن تصنع ٧ صور لخلفية الموقع وتسمي كل واحدة منها background_1.png و background_2.png و ... background_3.png إلخ.

إن زار مستخدم موقعك يوم الثلاثاء الموقع سيعرض صورة background_2.png كخلفية، في اليوم التالي سيعرض background_3.png ، بسيط وسهل!

في الدرس التالي سنتعلم لبنات بناء جديدة يمكن استخدامها لصنع حلقات وتكراراً في برنامجك.

PHP ممتعة، أليس كذلك؟

الدرس الخامس: الحلقات

في PHP من الممكن أن تدير تنفيذ البرامج بهيكل تحكم مختلفة، في هذا الدرس سنلقي نظرة على الحلقات (loop) ، الحلقات يمكن استخدامها لتكرار أجزاء من البرنامج بعدد محدد من المرات أو إلى أن يتحقق شرط محدد.

حلقات while

طريقة كتابة حلقة [while](#) هي:

```
while (condition) {  
    Statement  
}
```

بناء الحلقة يمكن ترجمته إلى العربية *بِفعل كذا بينما الشرط المحدد محقق*.

لنلقي نظرة على مثال بسيط:

```
<html>  
<head>  
<title>Loops</title>  
  
</head>  
<body>  
  
<?php  
  
$x = 1;  
  
while ($x <= 50) {  
    echo "<p>This text is repeated 50  
times</p>";  
    $x = $x + 1;  
}  
>  
  
</body>  
  
</html>
```

➡ شاهد المثال

في المثال استخدمنا/المتغير المسمى "\$x" ، كما ترى المتغيرات في PHP دائماً تبدأ برمز "\$" ، من السهل نسيان هذه النقطة في البداية لكن من الضروري تذكر ذلك وإلا لن يعمل البرنامج.

بعد هذه النقطة المثال يشرح نفسه تقريباً، في البداية المتغير \$x أعطيناها القيمة ١ ، ثم الحلقة تسأل المزود أن يكرر النص ما دام أن \$x أقل أو يساوي ٥٠ ، في كل تكرار المتغير \$x ستزداد قيمته بإضافة ١.

حلقات for

هناك طريقة أخرى لاستخدام الحلقات وهذا من خلال الأمر [for](#) بهذا الشكل:

```
for (Initialization; Condition; Step) {  
    Statement  
}
```

البيان (statement) يكرر ما دام أن التهيئة (Initialization) والخطوة (step) يحققان الشرط (Condition)، إن لم تفهم هذا فانظر إلى المثال:

```
<html>  
<head>  
  
<title>Loops</title>  
</head>  
<body>  
  
<?php  
  
for ($x=0; $x<=50; $x=$x+5) {  
    echo "<p>variable $x is now = " . $x .  
"</p>";  
}  
?>  
  
</body>  
</html>
```

➡ شاهد المثال

في المثال أعلاه المتغير \$x يزداد بقيمة ٥ في كل حلقة، التكرار سيستمر ما دام أن \$x أقل من أو تساوي ٥٠، لاحظ أيضاً أن القيمة \$x تستخدم كجزء من الجملة.

هذا مثال آخر:

```
<html>
<head>

<title>Loops</title>
</head>
<body>

<?php

for ($x=1; $x<=6; $x=$x+1) {
    echo "<h" . $x . ">Heading level " . $x .
"</h" . $x . ">";
}
?>

</body>
</html>
```

➡ شاهد المثال

هل فهمت المثال؟ في البداية حددنا قيمة المتغير \$x بالعدد واحد، ثم في كل تكرار نكتب عنواناً على مستوى \$x أي (h1) ثم h2 ثم ... h3 إلخ) إلى أن تصبح قيم \$x تساوي ستة.

حلقات داخل حلقات

كمبدأ، ليس هناك حد لكيفية استخدام الحلقات، فمثلاً يمكنك بسهولة أن تضع حلقات داخل حلقات وهكذا تنشأ العديد من التكرار.

لكن احذر PHP !تصبح بطيئة ومعقدة ويكبر حجم البرنامج ، ألقى نظرة على المثال التالي حيث باستخدام ٣ حلقات يمكننا كتابة ١٦ مليون لوناً!

لكي لا نجعل الصفحة بطيئة قللنا العدد بأن جعلنا الخطوة محدودة بالرقم ٣٠، وبالتالي قللنا عدد الألوان إلى ٥١٢.

```

<html>

<head>
<title>Loops </title>
</head>
<body>

<?php

    for          ($intRed=0;          $intRed<=255;
$intRed=$intRed+30) {

        for      ($intGreen=0;      $intGreen<=255;
$intGreen=$intGreen+30) {

            for          ($intBlue=0;          $intBlue<=255;
$intBlue=$intBlue+30) {

                $StrColor = "rgb(" . $intRed . "," .
$intGreen . "," . $intBlue . ")";

                echo "<span style='color:" . $StrColor
. "'>" . $StrColor . "</span>";

            }
        }
    }
?>

</body>
</html>

```

[شاهد المثال](#)

في هذا المثال كل لون أولي (أحمر أخضر وأزرق) يمكنه أن يحمل قيمة ما بين ٠ و ٢٥٥، أي مركب من الألوان الثلاثة يكون لوناً بشكل (255,255,255) rgb ، رقم اللون يستخدم في وسم .

الحلقات تصبح أكثر فائدة عندما تتعلم المزيد، عندما تفهم أساس الحلقات يمكنك أن تنتقل إلى الدرس التالي لنلقي نظرة على الشروط.

الدرس السادس: الشروط

الشروط تستخدم لتنفيذ أجزاء من البرنامج فقط عندما تستوفي متطلبات محددة مسبقاً، فمثلاً الشرط يمكن أن يكون تاريخ بعد الأول من يناير ٢٠١٢ أو متغير أكبر من ٧.

If...

أول نوع من الشروط سنلقي نظرة عليه هو [if](#) ، وهو يكتب بهذا الشكل:

```
if (condition) {  
    statement  
}
```

مرة أخرى، طريقة كتابة الأمر قريبة جداً من اللغة الطبيعية، إذا (**If**) تحقق شرط فننفذ كذا وكذا، لنلقي نظرة على مثال بسيط:

```
<html>  
  
<head>  
<title>Loops </title>  
</head>  
<body>  
  
<?php  
  
$x = 2;  
  
if ($x > 1) {  
    echo "<p>variable $x is greater than 1  
</p>";  
}  
  
?>  
  
</body>  
</html>
```

if ... else ...

النوع التالي من الشروط الذي سنلقي نظرة عليه هو [else](#) ، وهو ما يمكن أن يكتب بهذا الشكل:

```

if (condition) {
    statement
}
else {
    statement
}

```

مرة أخرى، أسلوب كتابة الأمر قريب من اللغة الطبيعية **إذا (if) تحقق شرط نفذ كذا وكذا، وإلا else نفذ شيئاً آخر.**

في [الدرس الرابع](#) تعلمت كيف تجد رقم الشهر، في المثال التالي سنستخدم رقم الشهر في حلقة **if** و **else** والشرط وضعت لمعرفة الموسم:

```

<html>
<head>
<title>Conditions</title>
</head>
<body>

<?php

if (date ("m") == 3) {
    echo "<p>Now it's spring!</p> ";
}
else {
    echo "<p>I do not know what season it
is!</p> ";
}

?>

</body>
</html>

```

[شاهد المثال](#) ➡

كما ترى، الشرط ليس شرطاً ذكياً وفقط يعمل إن كان الشهر مارس!

على أي حال هناك كثير من الطرق لتحسين الشرط وزيادة دقته، أدناه قائمة العوامل (operators) التي يمكن استخدامها في الشروط:

			== يساوي
من			< أقل
من			> أكبر
يساوي	أو	من	<= أقل
يساوي	أو	من	>= أكبر
			!= لا يساوي

بالإضافة إلى ذلك هناك عوامل منطق:

&& و
|| أو
! لا

العوامل يمكن استخدامها لوضع شروط أكثر دقة، الآن يمكن توسيع المثال أعلاه لإدراج كل شهور الربيع:

```
<html>
<head>
<title>Conditions</title>

</head>
<body>

<?php

if (date("m") >= 3 && date("m") <= 5) {
    echo "<p> Now it's spring!</p> ";
}
else {
    echo "<p> Now it's either winter, summer or
autumn!</p> ";
}

?>

</body>
</html>
```


لنلقي نظرة مقربة على الشرط المحسن::

```
date("m") >= 3 && date("m") <= 5
```

يمكن ترجمة الشرط إلى:

إن كان الشهر أكبر من أو يساوي ٣، والشهر أقل من أو يساوي ٥

شرط ذكي، أليس كذلك؟ العوامل تلعب دوراً مهماً في كثير من أجزاء PHP.

لكن لا زال البرنامج يعمل لشهور مارس، أبريل ومايو، بقية الشهور لم تذكر في الشرط، لذلك دعنا نطور الشرط أكثر.

if ... elseif ... else...

باستخدام [elseif](#) يمكننا توسيع الشرط ليعمل على كافة الشهور:

```
<html>
<head>
<title>Conditions</title>

</head>
<body>

<?php

if (date("m") >= 3 && date("m") <= 5) {
    echo "<p>Now it's spring!</p>";
}

elseif (date("m") >= 6 && date("m") <= 8) {
    echo "<p>Now it's summer!</p>";
}

elseif (date("m") >= 9 && date("m") <= 11) {
    echo "<p>Now it's autumn!</p>";
}
```

```

else {
    echo "<p>Now is winter!</p>";
}

?>

</body>
</html>

```

[شاهد المثال](#) ➔

لكتابة الشروط عليك أن تفكر بشكل منطقي وتكون منهجياً، المثال أعلاه بسيط لكن الشروط يمكنها أن تصبح معقدة جداً.

switch ... case

هناك طريقة أخرى لكتاب الشروط باستخدام: [switch](#) 📖

```

switch (expression) {

case 1:
    statement
    break;
case 2:
    statement
    break;
default:
    statement
    break;
}

```

هذه الطريقة تعتمد على الشرط أو **expression** ثم تضع قائمة بمختلف الأجوبة والقيم من خلال البيان أو **statements** ، أسهل طريقة لشرح هذا الأمر هو من خلال مثال.

كما نتذكر في الدرس الرابع وظيفة [date\("w"\)](#) 📖 تعطينا رقم اليوم في الأسبوع، هذا يمكن استخدامه في المثال بكتابة اسم اليوم بدلاً من رقمه:

```
<html>
```

```

<head>
<title>Conditions</title>
</head>
<body>

<?php
switch(date("w")) {

case 1:
    echo "Now it's Monday";
    break;
case 2:
    echo "Now it's Tuesday";
    break;
case 3:
    echo "Now it's Wednesday";
    break;
case 4:
    echo "Now it's Thursday";
    break;
case 5:
    echo "Now it's Friday";
    break;
case 6:
    echo "Now it's Saturday";
    break;
default:
    echo "Now it's Sunday";
    break;
}

?>
</body>
</html>

```

[شاهد المثال](#) ➡

غالباً **switch** نستخدم كبديل لأمر **if** و **else**، ما يجب عليك استخدامه في أي ظرف يعتمد على أي طريقة تجدها أسهل وأكثر منطقية، محاولة أن تجعل برنامجك منطقياً وواضحاً سيكون تحدياً كبيراً. في الدرس التالي سنلقي نظرة على كيفية إضافة تعليقات لبرنامجك وكيف تعمل، التعليقات الجيدة يمكنها أن تكون ضرورية لشخص آخر عليه تغيير شيء ما في برنامجك في مرحلة لاحقة.

الدرس السادس: الشروط

الشروط تستخدم لتنفيذ أجزاء من البرنامج فقط عندما تستوفي متطلبات محددة مسبقاً، فمثلاً الشرط يمكن أن يكون تاريخ بعد الأول من يناير ٢٠١٢ أو متغير أكبر من ٧.

If...

أول نوع من الشروط سنلقي نظرة عليه هو **if** ، وهو يكتب بهذا الشكل:

```
if (condition) {  
    statement  
}
```

مرة أخرى، طريقة كتابة الأمر قريبة جداً من اللغة الطبيعية، إذا (**If**) تحقق شرط فننفذ كذا وكذا، لنلقي نظرة على مثال بسيط:

```
<html>  
  
<head>  
<title>Loops </title>  
</head>  
<body>  
  
<?php  
  
$x = 2;  
  
if ($x > 1) {  
    echo "<p>variable $x is greater than 1  
</p>";  
}  
  
>  
  
</body>  
</html>
```

if ... else ...

النوع التالي من الشروط الذي سنلقي نظرة عليه هو **else** ، وهو ما يمكن أن يكتب بهذا الشكل:

```

if (condition) {
    statement
}
else {
    statement
}

```

مرة أخرى، أسلوب كتابة الأمر قريب من اللغة الطبيعية **إذا (if) تحقق شرط نفذ كذا وكذا، وإلا else نفذ شيئاً آخر.**

في [الدرس الرابع](#) تعلمت كيف تجد رقم الشهر، في المثال التالي سنستخدم رقم الشهر في حلقة **if** و **else** والشرط وضعت لمعرفة الموسم:

```

<html>
<head>
<title>Conditions</title>
</head>
<body>

<?php

if (date ("m") == 3) {
    echo "<p>Now it's spring!</p> ";
}
else {
    echo "<p>I do not know what season it
is!</p> ";
}

?>

</body>
</html>

```

[شاهد المثال](#) ➡

كما ترى، الشرط ليس شرطاً ذكياً وفقط يعمل إن كان الشهر مارس!

على أي حال هناك كثير من الطرق لتحسين الشرط وزيادة دقته، أدناه قائمة العوامل (operators) التي يمكن استخدامها في الشروط:

			== يساوي
من			< أقل
من			> أكبر
يساوي	أو	من	<= أقل
يساوي	أو	من	>= أكبر
			!= لا يساوي

بالإضافة إلى ذلك هناك عوامل منطق:

&& و
|| أو
! لا

العوامل يمكن استخدامها لوضع شروط أكثر دقة، الآن يمكن توسيع المثال أعلاه لإدراج كل شهور الربيع:

```
<html>
<head>
<title>Conditions</title>

</head>
<body>

<?php

if (date("m") >= 3 && date("m") <= 5) {
    echo "<p> Now it's spring!</p> ";
}
else {
    echo "<p> Now it's either winter, summer or
autumn!</p> ";
}

?>

</body>
</html>
```

لنلقي نظرة مقربة على الشرط المحسن::

```
date("m") >= 3 && date("m") <= 5
```

يمكن ترجمة الشرط إلى:

إن كان الشهر أكبر من أو يساوي ٣، والشهر أقل من أو يساوي ٥

شرط ذكي، أليس كذلك؟ العوامل تلعب دوراً مهماً في كثير من أجزاء PHP.

لكن لا زال البرنامج يعمل لشهور مارس، أبريل ومايو، بقية الشهور لم تذكر في الشرط، لذلك دعنا نطور الشرط أكثر.

if ... elseif ... else...

باستخدام [elseif](#) يمكننا توسيع الشرط ليعمل على كافة الشهور:

```
<html>
<head>
<title>Conditions</title>

</head>
<body>

<?php

if (date("m") >= 3 && date("m") <= 5) {
    echo "<p>Now it's spring!</p>";
}

elseif (date("m") >= 6 && date("m") <= 8) {
    echo "<p>Now it's summer!</p>";
}

elseif (date("m") >= 9 && date("m") <= 11) {
    echo "<p>Now it's autumn!</p>";
}
```

```

else {
    echo "<p>Now is winter!</p>";
}

?>

</body>
</html>

```

[شاهد المثال](#) ➡

لكتابة الشروط عليك أن تفكر بشكل منطقي وتكون منهجياً، المثال أعلاه بسيط لكن الشروط يمكنها أن تصبح معقدة جداً.

switch ... case

هناك طريقة أخرى لكتاب الشروط باستخدام: [switch](#) 📖

```

switch (expression) {

case 1:
    statement
    break;
case 2:
    statement
    break;
default:
    statement
    break;
}

```

هذه الطريقة تعتمد على الشرط أو **expression** ثم تضع قائمة بمختلف الأجوبة والقيم من خلال البيان أو **statements** ، أسهل طريقة لشرح هذا الأمر هو من خلال مثال.

كما نتذكر في الدرس الرابع وظيفة [date\("w"\)](#) 📖 تعطينا رقم اليوم في الأسبوع، هذا يمكن استخدامه في المثال بكتابة اسم اليوم بدلاً من رقمه:

```
<html>
```



```

<head>
<title>Conditions</title>
</head>
<body>

<?php
switch(date("w")) {

case 1:
    echo "Now it's Monday";
    break;
case 2:
    echo "Now it's Tuesday";
    break;
case 3:
    echo "Now it's Wednesday";
    break;
case 4:
    echo "Now it's Thursday";
    break;
case 5:
    echo "Now it's Friday";
    break;
case 6:
    echo "Now it's Saturday";
    break;
default:
    echo "Now it's Sunday";
    break;

}

?>
</body>
</html>

```

[شاهد المثال](#) ➡

غالباً **switch** نستخدم كبديل لأمر **if** و **else**، ما يجب عليك استخدامه في أي ظرف يعتمد على أي طريقة تجدها أسهل وأكثر منطقية، محاولة أن تجعل برنامجك منطقياً وواضحاً سيكون تحدياً كبيراً. في الدرس التالي سنلقي نظرة على كيفية إضافة تعليقات لبرنامجك وكيف تعمل، التعليقات الجيدة يمكنها أن تكون ضرورية لشخص آخر عليه تغيير شيء ما في برنامجك في مرحلة لاحقة.

الدرس السابع: ضع تعليقات في برامجك

كما قد لاحظت، برامج PHP يمكنها أن تبدو مثيرة للحيرة، في هذا الدرس سنتحدث عن أهمية التعليقات وكيف تضعها في برامجك.

لماذا من المهم أن تضع تعليقات في برامجك؟

عندما تبرمج فأنت تكتب لمزود أو حاسوب وتحتاج لكتابة لغة منهجية قد لا تعبر عن أفكارك عندما تكتب البرنامج.

لذلك قد يكون من الصعب على الآخرين وعليك أن تفهم هيكل البرنامج وبالتالي من الصعب تصحيح الأخطاء في البرنامج.

التعليقات تساعد على كتابة نص قصير يشرح جزء من البرنامج، المزود سيتجاهل هذا التعليق ولن يؤثر التعليق على وظائف البرنامج.

في عالم الأعمال، يكون من متطلبات مواصفات البرامج أن تحوي تعليقات وإلا ستكون مخاطرة كبيرة من الشركة أن تستلم برنامجاً تجد صعوبة في إيجاد وتصحيح أخطاءه.

كيف تضع التعليقات؟

من السهل أن تضع تعليقات في برنامجك، ببساطة ابدأ التعليق بعلامة "//":

ألقي نظرة على هذا المثال من الدرس الخامس ، أضفنا فيه تعليقات:

```
<html>
<head>
<title>Loops</title>
</head>
<body>

<?php

// Here we write color codes using three loops

// Red can be between 0 and 255
for ($intRed=0; $intRed<=255;
$intRed=$intRed+30) {

    // Green can be between 0 and 255
    for ($intGreen=0; $ intGreen<=255;
$intGreen=$intGreen+30) {
```

```

        // Blue can be between 0 and 255
        for ($ intBlue=0; $intBlue<=255;
$intBlue=$intBlue+30) {

            // The color code is made on the form
            rgb(red,green,blue)
            strColor = "rgb(" . intRed . "," .
            intGreen . "," . intBlue . ")"

            // Now we write the color code to the
            client
            echo "<span style='color:" . $strColor
            . "'> " . $ strColor . " </span>";

            // Closes the loops
        }
    }

?>

```

من أجل هذا المثال أضفنا المزيد من التعليقات لنوضح كيف أنك تستطيع تصحيح البرنامج بسهولة مع وجود التعليقات.

لذلك، تذكر أن تضع تعليقات في برنامجك!

الدرس الثامن: المصفوفات

في هذا الدرس سنلقي نظرة على المصفوفات، كيف تستخدم وماذا يمكنها أن تفعل.

فهم المصفوفات يمكن أن يكون صعباً في البداية، لكن لنجرب على أي حال ... حاولنا أن نجعلها سهلة بقدر الإمكان.

ما هي المصفوفة؟

المصفوفة (array) هي مجموعة من من العناصر المفهرسة حيث كل عنصر له رقم تعريف خاص به.

يبدو هذا مثيراً للحيرة، لكنها في الواقع ليست بهذا القدر من التعقيد.

تصور قائمة من الكلمات فصلنا بينها باستخدام فاصلة، تسمى القائمة مفصولة بفواصل، ويمكنها أن تكون بهذا الشكل:

apples, pears, bananas, oranges, lemons

جرب أن تتخيل تقسيم القائمة عند كل فاصلة، ثم أعطي كل قسم رقماً مميزاً كما في الصورة:

apples	pears	bananas	oranges	lemons
0	1	2	3	4

ما تراه هو مصفوفة، يمكننا مثلاً أن نسمي المصفوفة فواكه أو "fruits"، الفكرة هي أنك تستطيع الوصول إلى المصفوفة برقم وتحصل على قيمة، كما في القائمة أدناه:

fruits(0)	=	apples
fruits(1)	=	pears
fruits(2)	=	bananas
fruits(3)	=	oranges
		fruits(4) = lemons

هذه هي فكرة المصفوفات، الآن لنستخدمها عملياً.

كيف تستخدم المصفوفة؟

سنكمل مع مثال الفواكه، خطوة بعد خطوة سنجعلها تعمل كمصفوفة حقيقية، أولاً علينا أن نضع متغيرات مساوية لقائمة الفواكه:

```
<?php

$fruitlist = "apples, pears, bananas, oranges,
lemons";

?>
```

بعد ذلك نستخدم وظيفة [explode](#) لفصل القائمة عند كل فاصلة:

```
<?php

$fruitlist = "apples, pears, bananas, oranges,
lemons";

$arrFruits = explode(",", $fruitlist);

?>
```

مرحى! المتغير "\$arrFruits" أصبح مصفوفة الآن!

لاحظ أننا استخدمنا الوظيفة [explode](#) بخاصيتين:

١. الأولى هي القائمة التي يجب تقسيمها.
٢. والثانية هي المحدد أو الحرف الذي استخدم لتقسيم القائمة (في حالتنا استخدمنا الفاصلة: ",").

هنا نستخدم الفاصلة كمحدد أو مقسم لكن يمكنك استخدام أي حرف أو كلمة كمقسم للمصفوفة.

لنجرّب إضافة تعليق على البرنامج ووضعه في صفحة: PHP

```
<html>
<head>
<title>Array</title>
</head>
<body>

<?php
```

```

// Comma separated list
$fruitlist = "apples, pears, bananas, oranges,
lemons";

// Create an array by splitting the list (with
comma as delimiter)
$arrFruits = explode(",", $fruitlist);

// Write the values from our array
echo "<p>The list of fruits:</p>";

echo "<ul>";
echo "<li>" . $arrFruits[0] . "</li>";
echo "<li>" . $arrFruits[1] . "</li>";
echo "<li>" . $arrFruits[2] . "</li>";
echo "<li>" . $arrFruits[3] . "</li>";
echo "<li>" . $arrFruits[4] . "</li>";
echo "</ul>";

?>

</body>
</html>

```

[شاهد المثال](#)

هذا مثال بسيط جداً ويمكن أن تجد صعوبة في فهم فائدة المصفوفات في هذه المهمة بالتحديد، لكن انتظر، المصفوفات يمكن أن تكون تستخدم في أشياء كثيرة مفيدة.

حلقة داخل مصفوفة

في الدرس الخامس تعلمت الحلقات، الآن يمكننا استخدام حلقة للتعامل مع المصفوفة.

عندما تعلم كم عدد العناصر التي تحويها مصفوفة فلن تكون هناك مشكلة في تحديد حلقة، ببساطة تبدأ من الصفر وتجعل الحلقة تستمر في عملها بعدد العناصر المتوفرة، في المثال مع الفواكه الحلقة ستكون بهذا الشكل:

```

<html>
<head>
<title>Array</title>

```

```

</head>
<body>

<?php

// Comma separated list
$fruitlist = "apples, pears, bananas, oranges,
lemons";

// Create an array by splitting the list (with
a comma as delimiter)
$arrFruits = explode (",", $fruitlist);

echo "<p>The list of fruits:</p>";
echo "<ul>";

// Loop through the array $arrFruits
for ($x=0; $x<=4; $x++) {
    echo "<li>" . $arrFruits[$x] . "</li>";
}

echo "</ul>";

?>

</body>
</html>

```

[شاهد المثال](#)

كما ترى، المتغير $\$x$ الذي ترتفع قيمته من ٠ إلى ٤ في الحلقة استخدم لجلب المصفوفة

كيف تجد حجم المصفوفة؟

ماذا لو أضفنا فاكهة أخرى إلى القائمة؟ ستحتوي القائمة عنصراً إضافياً وسيحمل رقم التعريف 5 ، هل ترى مشكلة هنا؟ نحن بحاجة لتغيير الحلقة حتى تستطيع أن تكرر نفسها من ٠ إلى ٥ وإلا لن نستطيع التعامل مع كل العناصر في المصفوفة.

ألن يكون من المفيد أن تكون هناك طريقة لإيجاد عدد العناصر في أي مصفوفة؟

هذا بالضبط ما يمكن أن تفعله وظيفة [foreach](#) ، الآن يمكننا أن نصمم حلقة تعمل بغض النظر عن عدد العناصر في المصفوفة:

```
<?php
    foreach ($arrFruits as $x) {
        echo arrFruit[$x];
    }
?>
```

هذه الحلقة ستعمل بغض النظر عن عدد العناصر التي تحويها المصفوفة.

مثال آخر المثال أدناه يوضح كيفية استخدام المصفوفة لكتابة اسم كل شهر:

```
<html>
<head>
<title>Array</title>

</head>
<body>

    // Creates array with each month.
    // Creates array with the months. Note the
    comma before January - because there is no month
    with the number 0
    $arrMonths =
array(,"January","February","March","April","May","J
une","July","August","September","October","November
","December");

    // Call the array with the number of the month
- write to the client
    echo $arrMonths[date("m")];
?>

</body>
</html>
```

[شاهد المثال](#) ➡

لاحظ أننا استخدمنا وظيفة [array](#) بدلاً من وظيفة [explode](#) لإنشاء مصفوفة.

حسناً، اكتفينا بالمصفوفات، في الدرس التالي سنتعلم كيف تطور وظائف خاصة بك.

الدرس التاسع: الوظائف

في الدروس السابقة تعلمنا أن وظائف مثل `date()` و `array()`، في هذا الدرس سنتعلم إنشاء وظائف بأنفسنا باستخدام أمر `function`.

ما هي الوظيفة؟

الوظيفة تعالج مدخلات وتعطينا مخرجات، يمكنها أن تكون مفيدة إذا مثلاً لديك طيف واسع من البيانات التي تحتاج لمعالجة أو لديك حسابات أو إجراءات يجب أن تجريها مرات عديدة.

الوظيفة تكتب بهذا الشكل:

```
Function Name(list of parameters) {  
    Statement  
}
```

بهذه الطريقة يمكننا إنشاء وظيفة بسيطة تضيف القيمة ١ إلى أي رقم، ستكون بهذا الشكل:

```
function AddOne($x) {  
    $x = $x + 1;  
    echo $x;  
}
```

الوظيفة سمينها **AddOne** وهي تطلب مع رقم وفي المثال وضعنا الرقم ٣٤:

```
echo AddOne(34);
```

وستعطينا الوظيفة القيمة ٣٥.

المثال أعلاه يعالج رقماً لكن الوظيفة يمكنها معالجة النصوص والتواريخ وأي شيء آخر، يمكنك أيضاً إنشاء وظائف تتطلب العديد من المدخلات، في هذا الدرس سترى أمثلة مختلفة للوظائف.

المثال ١: وظيفة بمزيد من المدخلات

كما ذكرنا في السابق يمكنك بسهولة إنشاء وظيفة تحتاج للعديد من المدخلات، في المثال التالي سننشأ وظيفة تتطلب ٣ مدخلات وهي ثلاث أرقام وستعطينا قيمة جمع الأرقام الثلاثة:

```

<html>
<head>
<title>Functions</title>

</head>
<body>

<?php

function AddAll($number1,$number2,$number3) {
    $plus = $number1 + $number2 + $number3;
    return $plus;
}

echo "123 + 654 + 9 equals " .
AddAll(123,654,9);

?>

</body>
</html>

```

[شاهد المثال](#)

حسناً، هذا كان بسيطاً جداً، لكن النقطة هنا هي توضيح أن الوظيفة يمكنها أن تقبل المزيد من المدخلات.

مثال ٢: الوقت والتاريخ بالإنجليزية

لنحرب إنشاء وظيفة معقدة أكثر، وظيفة تتطلب إدخال التاريخ فتحوله لنا إلى هذا الشكل
Wednesday, 15 February, 2012, 10:00:00 AM

```

<html>
<head>
<title>Functions</title>
</head>
<body>

<?php

function EnglishDateTime($date) {

```

```

        // Array with the English names of the days
of the week
        $arrDay
        =
array("Monday","Tuesday","Wednesday","Thursday","Fri
day","Saturday","Sunday");

        // Array with the English names of the
months
        $arrMonth
        =
array("", "January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "Novemb
er", "December");

        // The date is constructed
        $EnglishDateTime
        =
$arrDay[(date("w",$date))] . " " . date("d",$date);
        $EnglishDateTime = $EnglishDateTime . " " .
$arrMonth[date("m",$date)] . " " . date("Y",$date);
        $EnglishDateTime = $EnglishDateTime . " " .
. date("H",$date) . ":" . date("i",$date);

        return $EnglishDateTime;

    }

    // Test function
    echo EnglishDateTime(time());

?>

</body>
</html>

```

[شاهد المثال](#)

لاحظ كيف أن '\$arrMonth' و '\$EnglishDateTime' كتبنا على عدة أسطر، فعلنا ذلك لتظهر بشكل جيد لمن يملك شاشة صغيرة، طريقة كتابة البرنامج لا تؤثر على البرنامج.

الوظيفة أعلاه ستعمل في كل المزودات بغض النظر عن اللغة، هذا يعني أنك تستطيع استخدام هذه الوظيفة إن كان موقعك مستضافاً في مزود فرنسي لكنك تريد التواريخ بالإنجليزية.

في هذه المرحلة لن نتعمق أكثر في الوظائف، لكنك الآن تعلمت القليل حول كيفية عمل الوظائف.

الدرس العاشر: تمرير المتغيرات من خلال عنوان الموقع (URL)

عندما تترجم بلغة PHP يمكنك أن تمرر المتغيرات من صفحة لأخرى، في هذا الدرس حول تمرير المتغيرات من خلال عنوان الموقع.

كيف تعمل؟

لعلك تساءلت لم بعض عناوين المواقع تبدو بهذا الشكل:

```
http://html.net/page.php?id=1254
```

لم هناك علامة استفهام بعد اسم الصفحة؟

الإجابة هي أن الأحرف بعد علامة الاستفهام هي سلسلة استعلام HTTP ، سلسلة استعلام HTTP يمكنها أن تحوي متغيرات وقيم، في المثال أعلاه السلسلة تحوي متغيراً اسمه "id" وقيمه "1254".

هذا مثال آخر:

```
http://html.net/page.php?name=Joe
```

مرة أخرى، هناك متغير اسم "name" وقيمه "Joe".

كيف تحصل على المتغير من خلال PHP ؟

لنفترض أن لديك صفحة PHP اسمها **people.php** ، الآن يمكنك أن تطلب هذه الصفحة من خلال العنوان التالي:

```
people.php?name=Joe
```

باستخدام PHP ، يمكنك الحصول على قيمة المتغير 'name' بهذا الأمر:

```
$_GET["name"]
```

أنت تستخدم أمر `$_GET` لتجد قيمة متغير محدد، لنجربه في مثال:

```
<html>
<head>
<title>Query string</title>
</head>
<body>

<?php

// The value of the variable name is found
echo "<h1>Hello " . $_GET["name"] . "</h1>";

?>

</body>
</html>
```

👉 شاهد المثال (keep an eye on the URL)

عندما تطلع على المثال أعلاه جرب أن تغير الاسم "Joe" بإسمك في عنوان الموقع واطلب الصفحة مرة أخرى.

متغيرات عديدة في نفس العنوان

يمكنك تمرير أكثر من متغير في عنوان الموقع، بفصل المتغيرات باستخدام حرف `&` يمكن تمرير متغيرات متعددة:

```
people.php?name=Joe&age=24
```

في عنوان الموقع هناك متغيرين، `age` و `name`، بنفس الطريقة أعلاه يمكنك الحصول على قيمتهما بهذا الشكل:

```
$_GET["name"]
$_GET["age"]
```

لنضيف متغيراً آخرًا للمثال:

```
<html>
<head>
<title>Query string </title>
</head>
<body>

<?php

// The value of the variable name is found
echo "<h1>Hello " . $_GET["name"] . "</h1>";

// The value of the variable age is found
echo "<h1>You are " . $_GET["age"] . " years
old </h1>";

?>

</body>
</html>
```

[شاهد المثال \(keep an eye on the URL\)](#)

الآن تعلمت طريقة واحدة لتمرير القيم بين الصفحات باستخدام عنوان الموقع، في الدرس التالي سنلقي نظرة على طريقة ثانية، النماذج.

الدرس الحادي عشر: تغيير المتغيرات من خلال النماذج

المواقع التفاعلية تتطلب مدخلات من المستخدمين، أحد أكثر الطرق استخداماً للحصول على المدخلات هي النماذج.

في هذا الدرس سنلقي نظرة على كيفية إنشاء نموذج ومعالجة مدخلاته في المزود.

<form>

عندما تبرمج نموذجاً فهناك خاصيتان مهمتان **action** و **method**.

action

تستخدم لإدخال عنوان الصفحة التي ستقبل مدخلات النموذج، ستكون ملف PHP تريده أن يتعامل مع المدخلات.

method

يمكنها أن تقبل إحدى القيمتين إما "post" أو "get" ، وهما طريقتان مختلفتان لتمرير المدخلات، في هذه المرحلة لست بحاجة لمعرفة الكثير، باستخدام "get" البيانات ترسل من خلال عنوان الموقع، وباستخدام "post" البيانات ترسل من خلال كتلة بيانات من خلال تقنية (STDIN) ، في بقية الدرس سنلقي نظرة على كيفية الحصول على البيانات من خلال عنوان الموقع باستخدام [\\$_GET](#) ، في هذا الدرس سنلقي نظرة على كيفية استرجاع البيانات المرسلة بطريقة "post".

صفحة HTML تحوي نموذجاً

الصفحة التي تحوي نموذجاً لا تحتاج لأن تكون ملف PHP ، يمكنها أن تكون إن أردت، ولا تحتاج حتى لأن تكون في نفس الموقع الذي يحوي ملفاً سيتلقى البيانات.

في المثال الأول سنلقي نظرة على نموذج بسيط يحوي حقلاً نصياً واحداً::

```
<html>
<head>
<title>Form</title>
</head>
<body>

<h1>Enter your name</h1>

<form method="post" action="handler.php">
<input type="text" name="username">
<input type="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

النتيجة في المتصفح ستكون:



هنا يأتي الجزء الممتع: تلقي البيانات ومعالجتها باستخدام PHP.

طلب بيانات النموذج باستخدام PHP

عندما تحتاج لطلب بيانات أرسلت من خلال نموذج بطريقة `post` فعليك استخدام أمر `$ _POST`:

```
$ _POST["fieldname"];
```

الذي يأخذ قيمة حقل في النموذج، لنجرب استخدامه في مثال.

في البداية اكتب صفحة تحوي النموذج كما في المثال أعلاه، ثم اكتب صفحة PHP وسمها "handler.php" لاحظ أن هذا هو اسم الصفحة الذي كتبناه في خاصية `action` في النموذج.

ملف "handler.php" سيحوي التالي:

```
<html>
```



```

<head>
<title>Form</title>
</head>

<body>

<?php

    echo    "<h1>Hello    "    .    $_POST["username"]    .
"</h1>";

    ?>

</body>
</html>

```

[شاهد المثال ➡](#)

مدخلات المستخدم والشروط

في المثال التالي سنجرب استخدام مدخلات المستخدم لإنشاء شروط، في البداية نحن بحاجة لنموذج:

```

<html>
<head>
<title>Form</title>
</head>
<body>

<form method="post" action="handler.php">

<p>What is your name:</p>
<input type="text" name="username"></p>

<p>What is your favorite color:
    <input type="radio" name="favoritecolor"
value="r" /> Red
    <input type="radio" name="favoritecolor"
value="g" /> Green
    <input type="radio" name="favoritecolor"
value="b" /> Blue </p>

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

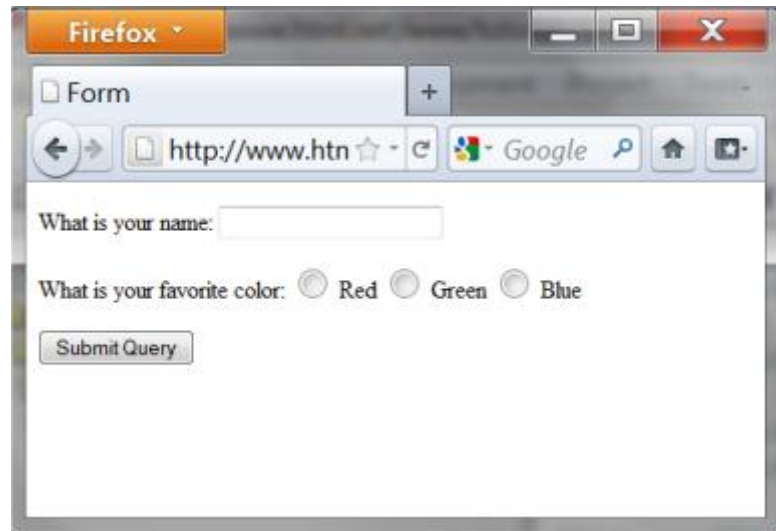
```

```
</form>
```

```
</body>
```

```
</html>
```

سيبدو بهذا الشكل في المتصفح:



الآن سنستخدم هذه المدخلات لإنشاء صفحة تغير لون خلفيتها تلقائياً بحسب اختيار المستخدم للونه المفضل، يمكننا فعل ذلك من خلال إنشاء شرط، (اقرأ [الدروس السادس](#)) يستخدم البيانات التي أدخلها المستخدم في النموذج.

```
<?php

    $strHeading      =      "<h1>Hello      "      .
$_POST["username"] . "</h1>";

    switch ($_POST["favoritecolor"]) {
    case "r":
        $strBackgroundColor = "rgb(255,0,0)";
        break;
    case "g";
        $strBackgroundColor = "rgb(0,255,0)";
        break;
    case "b":
        $strBackgroundColor = "rgb(0,0,255)";
        break;
    default:
```

```

        $strBackgroundColor = "rgb(255,255,255)";
        break;
    }

?>

<html>
<head>
<title>Form</title>

</head>
<body style="background: <?php echo
$strBackgroundColor; ?>;">

<? echo $strHeading; ?>

</body>
</html>

```

لون خلفية الصفحة سيكون أبيضاً إن لم يختار المستخدم لوناً مفضلاً في النموذج، هذا يمكن إنجازه من خلال أمر **default** لتحديد ما يحدث إن لم تتحقق أي من الشروط السابقة.

لكن ماذا لو لم يدخل المستخدم اسمه؟ البرنامج سيقول فقط "Hello" سنضيف شرطاً إضافياً لتغيير هذا.

```

<?php

$strUsername = $_POST["username"];

if ($strUsername != "") {
    $strHeading = "<h1>Hello " .
$_POST["username"] . "</h1>";
}
else {
    $strHeading = "<h1>Hello stranger!</h1> ";
}

switch ($_POST["favorite color"]) {
case "r":
    $strBackgroundColor = "rgb(255,0,0)";
    break;

```

```

        case "g";
            $strBackgroundColor = "rgb(0,255,0)";
            break;
        case "b":
            $strBackgroundColor = "rgb(0,0,255)";
            break;
        default:
            $strBackgroundColor = "rgb(255,255,255)";
            break;
    }

    ?>

<html>

<head>

<title>Form</title>
</head>
<body      style="background:      <?php      echo
$strBackgroundColor; ?>;">

<? echo $strHeading; ?>

</body>
</html>

```

[شاهد المثال](#) ➡

في المثال أعلاه استخدمنا شرطاً **للتحقق** من المعلومات التي أرسلها المستخدم، في هذه الحالة قد لا يكون من المهم أن يكتب المستخدم اسمه، لكن عندما تبدأ في برمجة برامج متقدمة من الضروري أن تضع في الحسبان أن المستخدم قد لا يدخل المعلومات كما ترغب.

مثال: نموذج مراسلة

بمعرفتكم الجديدة حول لغة PHP والنماذج يمكنك إنشاء نموذج مراسلة من خلال استخدام الوظيفة [mail](#) ، التي تكتب بهذه الطريقة:

```
mail(to, subject, message);
```

أولاً نحن بحاجة لنموذج بسيط:

```
<html>
<head>
<title>Contact form</title>
</head>
<body>

<h1>Contact form</h1>

<form method="post" action="handler.php">
  <p>Subject:<br          /><input          type="text"
name="subject" /></p>
  <p>Message:<br          /><textarea
name="message"></textarea></p>
  <input type="submit">
</form>

</body>
</html>
```

ثم نحن بحاجة إلى برنامج PHP ليرسل مدخلات المستخدم:

```
<html>
<head>
<title>Functions</title>
</head>
<body>

<?php

// Recipient (change to your e-mail address)
$strEmail = "name@mydomain.com";

// Get user inputs
$strSubject = $_POST["subject"];
$strMessage = $_POST["message"];

mail($strEmail,$strSubject,$strMessage);
echo "Mail Sent.";
```

```
?>
```

```
</body>
```

```
</html>
```

لاحظ أن المثال سيعمل فقط إن كنت تملك حق الوصول إلى مزود بريد إلكتروني، هذا غير متوفر في XAMPP ومعظم خدمات الاستضافة المجانية، كذلك وبعض شركات الاستضافة تطلب إضافة معلومات المرسل والذي يمكن إضافته بهذا الشكل:

```
mail("you@yourdomain.com", "Test", "This is a  
test mail", "From: me@mydomain.com");
```

الدرس الثاني عشر: الجلسات (Sessions)

عندما تزور موقعاً فأنت تفعل أشياء مختلفة، تنتقل من صفحة لأخرى وربما تضع معلومات في نموذج أو تشتري منتجاً.

كمطور مواقع مثل هذه المعلومات لها أهمية كبيرة لتوفير حلول ناجحة للويب.

افترض مثلاً أنك تطور موقعاً يحوي بضع صفحات محمية باسم وكلمة سر، لكي تكون الحماية فعالة يجب أن تتمكن الصفحات المحمية بكلمة السر من الوصول إلى معلومات حول إمكانية دخول المستخدم في وقت سابق، بمعنى آخر يجب على الصفحة أن تتذكر ما فعله المستخدم سابقاً.

هذا بالضبط ما يدور حوله هذا الدرس، كيف يمكنك استخدام الجلسات في PHP لحفظ واسترجاع المعلومات أثناء زيارة المستخدم لموقعك.

الجلسات

أمر [session](#) في PHP يسمح لك بإدارة المعلومات حول جلسة المستخدم، يمكنك كتابة تطبيقات ذكية يمكنها معرفة المستخدم وجمع معلومات حوله.

الجلسة يمكن أن تبدأ بطرق مختلفة، لن نشرح التفاصيل التقنية هنا لكن سنركز على حالة حيث الجلسة تبدأ بحفظ قيمة، الجلسة تبدأ وتنتهي إذا لم يطلب المستخدم أي صفحة خلال وقت محدد، (٢٠ دقيقة هو المعيار)، بالطبع يمكنك أن تنهي الجلسة في برنامجك.

لنفترض أن ٥٠ شخصاً يتصفحون نفس الموقع - متجر مثلاً - في نفس الوقت، أفضل وسيلة لحفظ معلومات ما قد تحويه سلة المشتريات لكل واحد منهم هي الجلسات، لكي نحدد كل زائر المزود يستخدم رقم تعريف محدد يحفظ في كوكيز (cookie)، الكوكيز هو ملف نصي صغير يحفظ في حاسوب المستخدم، (للمزيد حول الكوكيز اقرأ [الدرس الثالث عشر](#))، لذلك الجلسات تتطلب دعم الكوكيز في متصفح الزائر.

مثال لاستخدام الجلسات

عندما تطلب هذه الصفحة ستحفظ الوقت الحالي في جلسة، فعلت ذلك لكي أوضح لك بالمثل كيف تعمل الجلسات.

سميت المتغير "StartTime" وحفظته بإضافة هذه الأسطر لبرنامج: PHP

```
<?php  
  
session_start();  
$_SESSION["StartTime"] = date("r");  
  
?>
```

هكذا بدأنا جلسة، كما هو مذكور أعلاه كل جلسة لها رقم تعريف خاص بها في المزود.

الجلسة الخاصة بك رقم تعريفها **4cokl3agv0t18mo5ajf9ebtdk7** :

في أي وقت، يمكنني أن أطلب "StartTime" من الجلسة بكتابة التالي:

```
<?php  
  
session_start();  
echo $_SESSION["StartTime"];  
  
?>
```

والذي سيظهر لنا أن هذه الصفحة طلبت في هذا الوقت Sun, 13 Dec 2015 07:50:33
(+0100 بحسب ساعة المزود).

لكن ما هو مثير أن المعلومة ستبقى في الجلسة حتى لو تركت الصفحة، المعلومة ستلاحقك حتى تنتهي الجلسة.

تلقائياً الجلسة تدوم حتى يغلق الزائر المتصفح وهكذا تنتهي الجلسة تلقائياً، لكن إن أردت أن تنتهي الجلسة يمكنك أن تفعل ذلك بهذه الطريقة:

```
<?php  
  
session_destroy();  
  
?>
```

لننظر في مثال آخر حيث الجلسات تستخدم: كلمات السر.

نظام دخول مع الجلسات

في المثال التالي سنطور مثلاً بسيطاً لنظام دخول، سنستخدم كثيراً مما تعلمناه في الدروس السابقة.

أول ما نحتاجه هو نموذج يدخل فيه الناس الاسم وكلمة السر، سيكون بهذا الشكل:


```

<html>
<head>
<title>Login</title>

</head>
<body>
<form method="post" action="login.php">

    <p>Username:          <input          type="text"
name="username" /></p>
    <p>Password:         <input          type="text"
name="password" /></p>

    <p><input type="submit" value="Let me in"
/></p>

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

```

بعد ذلك ننشأ ملف login.php

في هذا الملف سنتأكد ما إذا أدخل المستخدم اسماً وكلمة سر صحيحين، في هذه الحالة نبدأ جلسة تقول بأن المستخدم سجل دخوله باسم وكلمة سر صحيحين.

```

<html>

<head>
<title>Login</title>

</head>
<body>

<?php

    // Check if username and password are correct
    if      ($_POST["username"]      ==      "php"      &&
$_POST["password"] == "php") {

        // If correct, we set the session to YES
        session_start();

```

```

        $_SESSION["Login"] = "YES";
        echo "<h1>You are now logged correctly
in</h1>";
        echo "<p><a href='document.php'>Link to
protected file</a><p/>";

    }
    else {

        // If not correct, we set the session to NO
        session_start();
        $_SESSION["Login"] = "NO";
        echo "<h1>You are NOT logged correctly in
</h1>";
        echo "<p><a href='document.php'>Link to
protected file</a></p>";

    }

?>

</body>
</html>

```

في الملفات المحمية نريد أن نتأكد أن المستخدم سجل دخوله بشكل صحيح، في حال لم ينجز ذلك المستخدم سيعود إلى نموذج الدخول، هكذا تحمي الصفحات:

```

<?php

// Start up your PHP Session
session_start();

// If the user is not logged in send him/her
to the login form
if ($_SESSION["Login"] != "YES") {
    header("Location: form.php");
}

?>

```

```
<html>
<head>
<title>Login</title>
</head>

<body>
<h1>This document is protected</h1>

  <p>You can only see it if you are logged
in.</p>
</body>
</html>
```

[اختبر نظام الدخول](#) ➡

الآن نحن نعرف أمر الجلسات، في الدرس التالي سنعمل في نفس المساحة لكن سنلقي نظرة مقربة أكثر على الكوكيز.

الدرس الثالث عشر: الكوكيز (Cookies)

كيف وما هي المعلومات التي تجمعها المواقع من مستخدميها وكيف تستخدم هذه المعلومات هو موضوع ساخن، الكوكيز تذكر كمثال لكيفية جمع المعلومات وكيف تشكل تهديداً لخصوصياتك، لكن هل عليك فعلاً أن تقلق؟ احكم بنفسك، عندما ننهي من هذا الدرس ستري ماذا يمكن فعله بالكوكيز.

ما هي الكوكيز؟

الكوكيز هو ملف نصي صغير تستخدمه المواقع لحفظ معلومات مختلفة، الكوكيز تحفظ في حاسوب المستخدم وليس في المزود.

معظم الكوكيز تنتهي صلاحيتها (تحذف نفسها) بعد مدة محددة من الزمن تتراوح ما بين دقيقة وحتى سنوات، لكن المستخدم يمكنه تحديد وحذف الكوكيز من حاسوبه.

معظم المتصفحات مثل مايكروسوفت إنترنت إكسبلورر وموزيلا فايرفوكس وغوغل كروم يمكنها أن تعدل إعداداتها لتترك الخيار للمستخدم لاختيار ما إذا كان يريد أو لا يريد قبول الكوكيز، لكن لم لا نقول "لا" لكل الكوكيز؟، يمكن فعل هذا، لكن كثير من المواقع لن تعمل كما يجب بدون الكوكيز لأن الكوكيز تستخدم لتحسين قابلية الاستخدام ولوظائف مختلفة في الموقع.

كيف تحفظ المعلومات في الكوكيز؟

من السهل إنشاء وتعديل الكوكيز في PHP باستخدام [setcookie](#) ، في المثال الأول سنقوم بإنشاء كوكيز وسنضع فيه قيمة.

أولاً أنت بحاجة لتسمية الكوكيز، في المثال سميناه "HTMLTest" ، ثم سنضع فيه قيمة كما في المثال التالي:

```
<?php

// Setting the cookie
setcookie("HTMLTest", "This is a test
cookie");

?>
```

تلقائياً الكوكيز تبقى حتى يغلق المستخدم المتصفح، لكن بإمكانك وبسهولة تعديلها بإضافة معيار آخر لتحديد تاريخ الانتهاء:

```
<?php

// Setting the cookie
setcookie("Name", "C. Wing, time()+3600);
setcookie("Interests", "plane spotting",
time()+3600);

?>
```

"Time()+3600" تحدد أن الكوكيز سينتهي بعد ٣٦٠٠ ثانية - أي ٦٠ دقيقة - من الآن.

في المثال أعلاه حفظنا معلومة حول اسم المستخدم واهتماماته هذه المعلومات يمكنها مثلاً أن تكون مفيدة لتوجيه الموقع ليكون مخصصاً لهذا الزائر.

كيف تسترجع المعلومات من الكوكيز؟

للحصول على القيم في الكوكيز استخدام أمر [\\$_COOKIE](#) ، مثلاً إذا أردنا المعلومات في المثال أعلاه سنحتاج لفعل التالي:

```
<?php

// Retrieve values from the cookie
$strName = $_COOKIE["Name"];
$strInterest = $_COOKIE["Interest"];

// Write to the client
echo "<p>" . strName . "</p>"
echo "<p>Your interest is . " strInterest .
"</p>"

?>
```

من يمكنه قراءة الكوكيز؟

تلقائياً الكوكيز يمكن أن تقرأ في نفس العنوان من المستوى الثاني -أي في نفس الموقع - الذي أنشأ الكوكيز، لكن باستخدام عوامل مثل *domain* و *path*، بإمكانك أن تضع مزيداً من القيود على الكوكيز كما في المثال التالي:

```
setcookie(name, value, expiration time, path, domain);
```

لنلقي نظرة على مثال:

```
<?php

// Setting the cookie: name, value, expiration
time, path, domain
setcookie("Name", "C. Wing",
time()+60*60*24*365, "/tutorials/php/",
"www.html.net");
?>
```

في المثال أعلاه أنشأنا كوكيز اسمها "Name" وتحتوي قيمة "C. Wing." وهذه الكوكيز ستنتهي بعد عام، ٦٠ ثانية x ٦٠ دقيقة x ٢٤ ساعة x ٣٥٦ يوم، ويمكن أن تقرأ فقط في المواقع الموجودة في مجلد "/tutorials/php/" في العنوان "www.html.net".

مثال لكوكيز

يمكننا أن نجرب حفظ مثال لكوكيز في حاسوبك وننظر كيف ستبدو.

المثال التالي ينشأ ويعد الكوكيز:

```
<?php

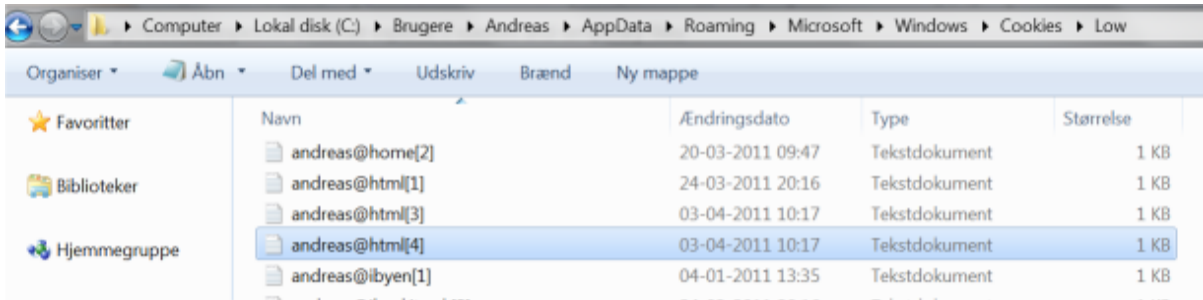
// Setting the cookie
setcookie("HTMLTest", "This text is in a
cookie!", time()+60*60*24, "/tutorials/php/",
"www.html.net");

// Write the information to the client
echo $_COOKIE ["HTMLTest"];

?>
```

[شاهد المثال](#) ➡

الكوكيز وضعت في القرص الصلب في حاسوبك، بحسب نظام التشغيل الذي تستخدمه ستحفظ الكوكيز في مكان مختلف، عندما تجدها ستكون بهذا الشكل:



Organiser	Åbn	Del med	Udskriv	Brænd	Ny mappe
Favoritter	Navn	/Endringsdato	Type	Størrelse	
Biblioteker	andreas@home[2]	20-03-2011 09:47	Tekstdokument	1 KB	
Hjemmegruppe	andreas@html[1]	24-03-2011 20:16	Tekstdokument	1 KB	
	andreas@html[3]	03-04-2011 10:17	Tekstdokument	1 KB	
	andreas@html[4]	03-04-2011 10:17	Tekstdokument	1 KB	
	andreas@ibyen[1]	04-01-2011 13:35	Tekstdokument	1 KB	

كما ترى الكوكيز هي ملفات نصية عادية يمكن استعراضها في محرر نصي مثل المفكرة (Notepad) مثلاً، محتويات الكوكيز التي قمنا بإنشائها ستكون في الغالب بهذا الشكل:

```
HTMLTest      TEXT=This+text+is+in+a+cookie%      21
www.html.net/tutorials/php      0      80973619229399148
4216577264 29399141 *
```

لن ندخل في تفاصيل مختلفة حول الرموز المختلفة، ببساطة لاحظ أن المستخدم يتحكم بشكل كامل بالكوكيز في حاسوبه.

في هذا الدرس ألقينا نظرة على الكوكيز وما يمكنه فعله لكن ليس على كيفية استخدامها، هناك هم مشترك بين الناس حول الكوكيز وكيف أن بعض المواقع تستخدمها في أمور غير لائقة، لكن في معظم الحالات الكوكيز تستخدم لتكون المواقع سهلة أكثر أو متعلقة بكل مستخدم.

إن اخترت أن تستخدم الكوكيز في موقعك فقد تكون فكرة جيدة أن تخبر زواره أنك تستخدمها، هذا يمكن مثلاً أن تذكره في صفحة سياسة الاستخدام أو في نموذج الإلتحاق بالموقع.

الدرس الرابع عشر: نظام الملفات

مع PHP يمكنك أن تصل إلى نظام الملفات في المزود، هذا يسمح لك بتعديل المجلدات والملفات النصية في برامج PHP.

فمثلاً يمكنك استخدام PHP لقراءة أو كتابة ملف نصي، أو يمكنك عرض قائمة الملفات في مجلد محدد، هناك إمكانيات كثيرة و PHP يمكنها أن توفر عليك الكثير من الجهد.

هنا سنلقي نظرة على كيف يمكن استخدام PHP للعمل مع المجلدات والملفات، الهدف هو أن نعطيك نظرة سريعة، في الدرس التالي سنلقي نظرة مقربة على إمكانيات مختلفة، لن نخوض في كل الإمكانيات المتوفرة، مرة أخرى، راجع وثائق لغة PHP لترى كامل قائمة ما يمكن فعله بالملفات والمجلدات.

[filetime](#)

تعطينا الوقت الذي حررت فيه محتويات الملف آخر مرة، (كطابع زمني - اقرأ الدرس الرابع).

[filetime](#)

تعطينا الوقت الذي فتح الملف فيه آخر مرة (كطابع زمني - اقرأ الدرس الرابع).

[filesize](#)

تعطينا حجم الملف بالبايت.

لنجرب إيجاد هذه الخصائص للملف الذي تنتظر له `"/tutorials/php/lesson14.php"` :

```
<html>

<head>
<title>Filesystem</title>
</head>
<body>

<?php

// Find and write properties
echo "<h1>file: lesson14.php</h1>";
echo "<p>Was last edited: " . date("r",
filemtime("lesson14.php"));
echo "<p>Was last opened: " . date("r",
fileatime("lesson14.php"));
```



```

        echo "<p>Size: " . filesize("lesson14.php") .
" bytes";

    ?>

</body>
</html>

```

[شاهد المثال](#) ➔

المجلدات

PHP يمكنها أيضاً أن تسمح لك بالتعامل مع المجلدات في المزود، لن نخوض في كل الإمكانيات المتوفرة في هذا المجال، فقط سنعرض مثلاً، مرة أخرى، راجع التوثيق للمزيد من المعلومات.

 [opendir](#)

يفتح مجلداً محدداً.

 [readdir](#)

يعطينا اسم الملف التالي في مجلد مفتوح ([cf. opendir](#)).

 [closedir](#)

يغلق مجلداً محدداً.

المثال أدناه يكتب قائمة بمحتويات مجلد "tutorials/php/".

```

<html>
<head>
<title>FileSystemObject</title>
</head>
<body>

<?php

// Opens the folder
$folder = opendir("../..tutorials/php/");

// Loop through all files in the folder
while (($entry = readdir($folder)) != "") {
    echo $entry . "<br />";
}

```

```
}

// Close folder
$folder = closedir($folder);

?>

</body>

</html>
```

[شاهد المثال](#) ➡

في المثال مجلد "../tutorials/php/" فتح أولاً، ثم استخدمت حلقة لكتابة اسم الملف التالي في المجلد ما دام أن هناك مزيد من الملفات وفي النهاية أغلق المجلد.

في الدروس التالية سنلقي نظرة على كيفية القراءة والكتابة في ملف نصي.

الدرس ١٥ : القراءة من ملف نصي

في الدرس السابق تعلمنا كيف نستخدم PHP للوصول إلى نظام الملفات في المزود، في هذا الدرس سنستخدم ما تعلمناه لقراءة ملف نصي عادي.

الملفات النصية يمكنها أن تكون مفيدة جداً لحفظ أنواع مختلفة من البيانات، وهي ليست بمرونة قواعد البيانات لكنها لا تتطلب ذاكرة كبيرة، فوق ذلك الملفات النصية بسيطة وتعمل في معظم الأنظمة.

فتح ملف نصي

يمكننا استخدام أمر `fopen` لفتح ملف نصي ويكتب الأمر بهذا الشكل:

```
fopen(filename, mode)
```

filename

اسم الملف الذي تريد فتحه.

mode

الطريقة يمكن أن تحدد بحرف "r" للقراءة (reading) أو "w" للكتابة (writing) أو "a" للإلحاق (appending)، في هذا الدرس سنشرح فقط كيف تقرأ الملف لذلك سنستخدم "r"، في الدرس التالي سنتعلم الكتابة والإلحاق في الملف النصي.

الأمثلة في هذا الدرس تستخدم ملفاً نصياً هو [unitednations.txt](#)، الملف يحوي قائمة بسيط لبرامج ومعونات الأمم المتحدة وعناوين مواقع هذه البرامج، بإمكانك تنزيل الملف أو إنشاء ملف خاص بك لتختبره.

أولاً لنجرب فتح ملف: `unitednations.txt`

```
<?php

// Open the text file
$f = fopen("unitednations.txt", "r");

// Close the text file
fclose($f);

?>
```

مثال ١: قراءة سطر من ملف نصي

مع وظيفة [fgets](#) يمكن قراءة سطر من ملف نصي، هذه الطريقة تقرأ السطر حتى تصل إلى نهايته.

```
<html>

<head>
<title>Reading from text files</title>
</head>
<body>

<?php

    $f = fopen("unitednations.txt", "r");

    // Read line from the text file and write the
    contents to the client
    echo fgets($f);

    fclose($f);

?>

</body>
</html>
```

[شاهد المثال](#) ➡

مثال ٢: قراءة كل الأسطر في ملف نصي

```
<html>

<head>
<title>Reading from text files</title>
</head>
<body>

<?php
```

```

$f = fopen("unitednations.txt", "r");

// Read line by line until end of file
while(!feof($f)) {
    echo fgets($f) . "<br />";
}

fclose($f);

?>

</body>
</html>

```

[شاهد المثال](#) ➔

في المثال الحلقة تقرأ كل سطر وتستخدم الوظيفة [feof](#) (for end-of-file) لتتأكد إن وصلت لنهاية الملف أم لا، فإن لم تصل سيكتب السطر.

بدلاً من استخدام حلقة لكل السطور يمكن تحقيق نفس النتيجة من خلال الوظيفة [fread](#) ، إن كنت تعمل على ملف نصي كبير بآلاف الأسطر فاعلم أن [fread](#) تستخدم المزيد من الموارد أكثر من [fgets](#) ، للملفات الصغيرة لن يشكل هذا فرقاً كبيراً.

المثال ٣: دليل روابط بسيط

كما ذكر في بداية هذا الدرس، الملف النصي هو وسيلة رائعة لحفظ البيانات، ويتضح هذا في المثال التالي حيث سننشأ دليل روابط بسيط من محتويات الملف النصي [unitednations.txt](#)

الملف كتب بطريقة نظامية حيث اسم البرنامج متبوعاً بفاصلة ثم موقع البرنامج، كما تتخيل يمكن إضافة المزيد من المعلومات بسهولة في هذا الملف الذي يحوي بيانات مفصولة بفاصلة.

للحصول على المعلومات من كل سطر علينا استخدام المصفوفة، اقرأ [الدرس الثامن](#) للمزيد من المعلومات حول المصفوفات.

```

<html>
<head>
<title>Reading from text files</title>

</head>
<body>

```

```

<?php
$f = fopen("unitednations.txt", "r");

// Read line by line until end of file
while (!feof($f)) {

    // Make an array using comma as delimiter
    $arrM = explode(",", fgets($f));

    // Write links (get the data in the array)
    echo "<li><a href='http://' . $arrM[1] .
    "'>" . $arrM[0] . "</a></li>";

}

fclose($f);
?>

</body>
</html>

```

[➡ شاهد المثال](#)

مفيد جداً، أليس كذلك؟ مبدئاً يمكنك الآن إضافة المئات من الروابط إلى الملف النصي أو بإمكانك إضافة مزيد من المعلومات.

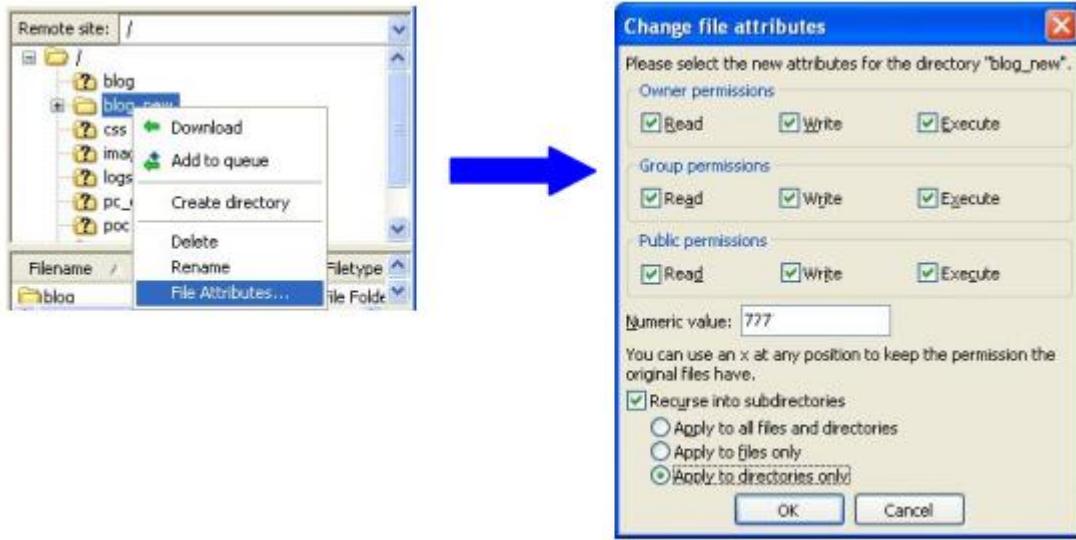
في الدرس التالي سنلقي نظرة على كيفية الكتابة في ملف نصي.

الدرس السادس عشر: الكتابة في ملف نصي

في الدرس الماضي تعلمنا القراءة من ملف نصي، في هذا الدرس سنتعلم كيف نكتب في ملف نصي.

هناك طريقتان متشابهتان لكن بفرق واحد: يجب أن يكون لديك صلاحية الكتابة في ملف، هذا يعني أن الملف يجب أن يوضع في مجلد حيث تملك الصلاحية التي تخول لك فعل ذلك.

إن كنت تعمل في حاسوبك يمكنك أن تحدد هذه الصلاحية بنفسك: اضغط بالزر الأيمن على مجلد ثم اختر "Properties"، في شركات استضافة المواقع سيكون هناك في الغالب مجلد واحد يسمح لك بالكتابة فيه، المجلد يسمى غالباً "cgi-bin" أو "log" أو "databases" أو شيء مماثل، وإذا سمح لك مستضيف موقعك يمكنك أن تغير الصلاحيات بنفسك، غالباً من خلال الضغط بالزر الأيمن على المجلد من خلال برنامج FTP لنقل الملفات، اختر "properties" أو "permissions" أو شيء مماثل، صورة الشاشة أدناه تعرض كيف ينجز ذلك من خلال برنامج FileZilla.



اقرأ المزيد حول الموضوع من خلال صفحات الدعم لمستضيف موقعك.

لاحظ أن الملف النصي - وليس ملف - PHP يحتاج إلى أن يكون في مجلد له صلاحية الكتابة.

فتح الملف النصي للكتابة فيه

بنفس الطريقة عندما عند قراءة ملف نصي نستخدم الوظيفة [fopen](#) أيضاً للكتابة، لكن هذه المرة نغير الطريقة إلى "w" للكتابة (writing) أو "a" للإلحاق (appending).

الاختلاف بين الكتابة والإلحاق هو في مكان مؤشر الكتابة، إما أن يكون في بداية أو نهاية الملف النصي.

في أمثلة هذا الدرس نستخدم ملفاً نصياً فارغاً باسم `textfile.txt`، لكن يمكنك إنشاء ملفك النصي الخاص إذا أردت.

أولاً لنجرب فتح الملف النصي للكتابة:


```
<?php

// Open the text file
$f = fopen("textfile.txt", "w");

// Close the text file
fclose($f);

?>
```

مثال ١: كتابة سطر في الملف النصي

لكتابة سطر يجب أن تستخدم الوظيفة [fwrite](#)  لفعل هذا، بهذا الشكل:

```
<html>

<head>
<title>Writing to a text file</title>
</head>
<body>

<?php

// Open the text file
$f = fopen("textfile.txt", "w");

// Write text line
fwrite($f, "PHP is fun!");

// Close the text file
fclose($f);

// Open file for reading, and read the line
$f = fopen("textfile.txt", "r");
echo fgets($f);

fclose($f);

?>
```



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

[شاهد المثال](#)

بما أننا فتحنا الملف *للكتاب*، السطر سيضاف في أعلى الملف النصي وهذا يعني أنه سيكتب فوق أي سطر آخر موجود، أما إن فتحنا الملف *للاحاق* سيضاف السطر في آخر الملف وهذا يعني زيادة سطر واحد للملف في كل مرة نكتب في الملف.

مثال ٢: إضافة كتلة من النص إلى الملف

بالطبع يمكنك أن تضيف كتلة من النص بدلاً من سطر واحد، بهذا الشكل:

```
<html>
<head>
<title>Write to a text file</title>
</head>
<body>

<?php

// Open the text file
$f = fopen("textfile.txt", "w");

// Write text
fwrite($f, $_POST["textblock"]);

// Close the text file
fclose($f);

// Open file for reading, and read the line
$f = fopen("textfile.txt", "r");

// Read text
echo fgets($f);
fclose($f);

?>

</body>

</html>
```

[شاهد المثال](#) 

في الدرس التالي سنلقي نظرة على وسيلة أخرى لحفظ البيانات: قواعد البيانات.

الدرس السابع عشر: قواعد البيانات

قاعدة البيانات هي مجموعة من المعلومات أو البيانات منظمة ليسهل استرجاعها وإدارتها وتحديثها، لذلك قواعد البيانات تتيح لنا إنشاء مواقع تفاعلية تحوي الكثير من المعلومات، مثلاً كل بيانات أعضاء هذا الموقع ومواضيعهم في المنتديات مخزنة في قاعدة بيانات.

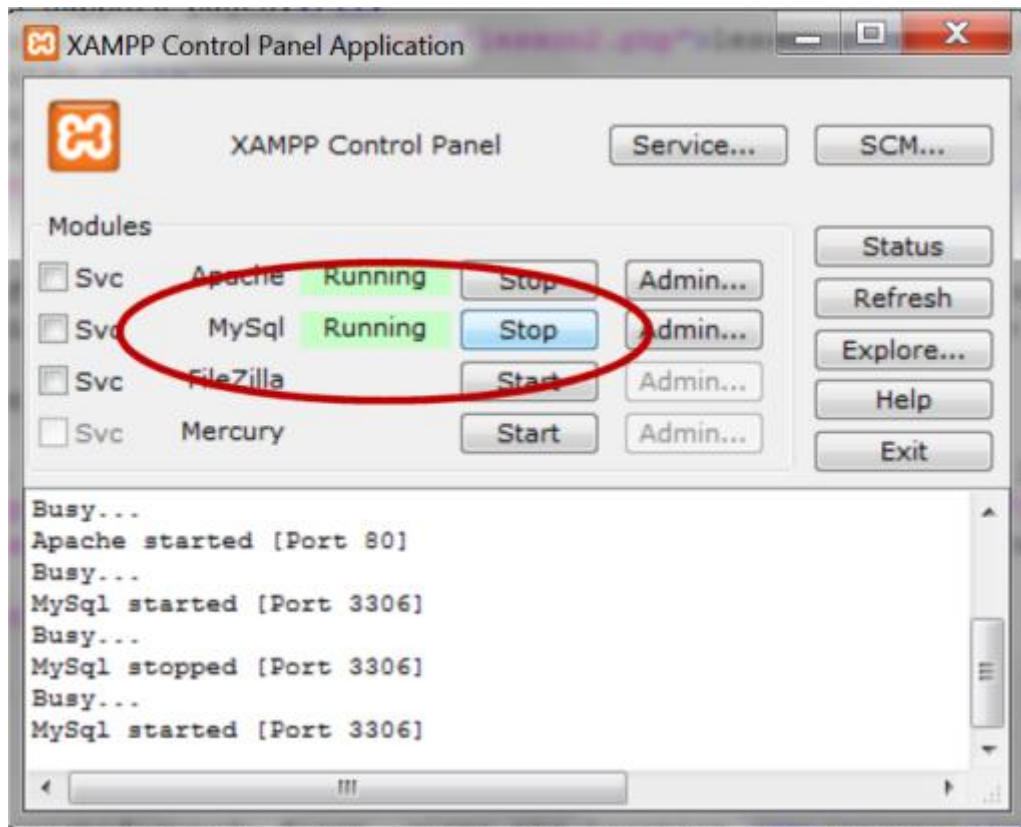
قاعدة البيانات تتكون من جدول أو أكثر، إذا كنت معتاداً على العمل في الجداول الممتدة أو استخدمت قواعد البيانات من قبل ستكون الجداول مألوفة لك:

←T→	id	FirstName	LastName	Phone	BirthDate
<input type="checkbox"/>	22	Donald	Duck	33221100	1954-01-20
<input type="checkbox"/>	23	Gladstone	Gander	44332211	1952-11-14
<input type="checkbox"/>	24	Scrooge	McDuck	55443322	1935-03-06
<input type="checkbox"/>	25	Grandma	Duck	66554433	1932-10-09
<input type="checkbox"/>	26	Mickey	Mouse	77665544	1956-03-14
<input type="checkbox"/>	27	Daisy	Duck	88776655	1959-04-28

هناك أنواع مختلفة من قواعد البيانات، MySQL و MS Access و MS SQL Server و Oracle SQL Server وغيرها الكثير، في هذا الدرس سنستخدم MySQL ، قاعدة بيانات MySQL هي المكان الطبيعي لتبدأ عندما تريد استخدام قواعد البيانات في PHP.

أنت بحاجة للوصول إلى MySQL لتستطيع إنجاز هذا الدرس والدروس التالية:

- أن كنت تملك موقعاً مستضافاً ويدعم لغة PHP وقاعدة بيانات MySQL سيكونان في الغالب مدعومان على المزود، اقرأ صفحات الدعم الخاصة باستضافة موقعك..
- إذا ثبت PHP في حاسوبك بنفسك ولديك الشجاعة لتنصيب MySQL كذلك فيمكنك تنزيل نسخة مجانية في [موقع MySQL](#).
- إن كنت تستخدم XAMPP اقرأ [الدرس الثاني](#) (قاعدة MySQL مثبتة لديك وجهاز للاستخدام، فقط تأكد أن MySQL تعمل من خلال لوحة التحكم:



في بقية الدرس سنلقي نظرة مقربة على كيفية ربط مزود قاعدة البيانات، قبل أن نتعلم كيفية إنشاء قاعدة بيانات واسترجاع وتحديث البيانات.

الربط بمزود قاعدة البيانات

في البداية يجب أن تكون لديك إمكانية الوصول لمزود قاعدة البيانات، هذا ينجز من خلال وظيفة `mysql_connect` ويمكن كتابتها بهذه الطريقة:

```
mysql_connect(server, username, password)
```

تبدو الوظيفة واضحة، أولاً عليك كتابة موقع قاعدة البيانات (`server`)، ثم اسم المستخدم `username` وكلمة السر `password`.

إن كنت تملك موقعك الخاص عليك أن تقرأ عن موقع مزود قاعدة البيانات في صفحات الدعم لمستضيف موقعك، الاسم وكلمة السر ستكونان في الغالب متطابقتان للاسم وكلمة السر الذي تستخدمهما لخدمة نقل الملفات FTP، إن لم يكن الأمر كذلك فتواصل مع مستضيف موقعك.

مثال للاتصال بمزود قاعدة البيانات في موقع مستضاف:

```
mysql_connect("mysql.myhost.com", "user001",
"sesame") or die(mysql_error());
```

مثال للاتصال بقاعدة البيانات في مزود:

```
mysql_connect("localhost", "root", "") or die  
(mysql_error());
```

في الأمثلة أعلاه أضفنا `or die(mysql_error())` والتي باختصار تقاطع البرنامج لتكتب رسالة خطأ في حال لم ينجح الاتصال بقاعدة البيانات.

الآن اتصلنا بمزود قاعدة البيانات، يمكننا إنشاء قواعد البيانات ويمكننا إدخال واسترجاع المعلومات منها، هذا ما سنتحدث عنه في الدروس التالية.

بالمناسبة، تذكر أنه من الأفضل إغلاق الاتصال بقاعدة البيانات عندما تنتهي من طلب المعلومات أو تحديثها، هذا ينجز من خلال وظيفة [mysql_close](#) .

الدرس ١٨ : إنشاء قاعدة البيانات والجداول

في الدرس السابقة ألقينا نظرة على كيفية إنشاء اتصال بمزود قاعدة البيانات، الخطوة التالية ستكون إنشاء قواعد البيانات والجداول.

سنلقي نظرة على طريقتين لإنشاء قاعدة البيانات والجداول، الأولى من خلال PHP والثانية من خلال أداة سهلة الاستخدام وهي PhpMyAdmin وهي أداة متوفرة في معظم خدمات الاستضافة وفي XAMPP.

إن كنت تملك موقعاً مستضافاً يستخدم PHP و MySQL ففي الغالب هناك قاعدة بيانات جاهزة لك ويمكنك تجاوز هذا القسم من الدرس لتبدأ مباشرة في إنشاء الجداول، مرة أخرى، أسأل مستضيف موقعك لمزيد من المعلومات.

إنشاء قواعد البيانات والجداول باستخدام PHP

وظيفة [mysql_query](#) تستخدم لإرسال استعلام لمزود قواعد البيانات MySQL ، الاستعلامات تكتب بلغة تسمى **Structured Query Language** أو **SQL** ، هذه اللغة هي الأكثر استخداماً في استعلامات قواعد البيانات على اختلافها وليس فقط لقواعد بيانات MySQL وهي لغة منطقية وسهلة، في هذا الدرس وفي الدرس التالي سنتعلم أهم استعلامات SQL.

عند إنشاء قاعدة بيانات يستخدم استعلام [CREATE DATABASE](#) بهذا الشكل:

```
CREATE DATABASE database name
```

منطقي وسهل، أليس كذلك؟ لنجرب وضع الاستعلام في برنامج: PHP

```
mysql_connect("mysql.myhost.com", "user",  
"sesame") or die(mysql_error());  
  
mysql_query("CREATE DATABASE mydatabase") or  
die(mysql_error());  
  
mysql_close();
```

في البداية اتصلنا بمزود MySQL ثم أنشأنا قاعدة بيانات باسم "mydatabase" وأخيراً أغلقنا الاتصال بمزود قاعدة البيانات. connection to the MySQL server again.

كل هذا سهل حتى الآن، لكن الأمور تصبح معقدة بعض الشيء عندما نرغب في إنشاء جداول باستخدام لغة PHP ، عند إنشاء جداول نستخدم الاستعلام [CREATE TABLE](#) بهذا الشكل:

```
CREATE TABLE table name
(
  column1_name DATA_TYPE,
  column2_name DATA_TYPE,
  column3_name DATA_TYPE,
  ...
)
```

column_name و *table_name* هما بالطبع اسم الجدول وأعمدته، *DATA_TYPE* تستخدم لتحديد نوع البيانات التي ستوضع في كل عمود من الجدول، الأنواع الأكثر استخداماً للبيانات هي:

❏ [INT](#)

للأرقام بدون كسور عشرية

❏ [DECIMAL](#)

للأرقام بكسور عشرية

❏ [CHAR](#)

لنص قصير بعدد حروف لا يزيد عن ٢٥٥ حرفاً

❏ [TEXT](#)

للنص بعدد حروف يصل إلى ٦٥٥٣٥ حرفاً

❏ [LONGTEXT](#)

للنصوص الطويلة بعدد حروف يصل إلى ٤,٢٩٤,٩٦٧,٢٩٥ حرفاً

❏ [Date](#)

YYYY-MM-DD للتواريخ بتنسيق

❏ [Time](#)

HH:MM:SS للوقت بتنسيق

❏ [DATETIME](#)

YYYY-MM-DD HH:MM:SS للوقت والتاريخ بتنسيق

بشكل عام تبدو أنواع البيانات منطقية وسهلة، لنجربها في مثال:

```
mysql_connect("mysql.myhost.com", "user",
"sesame") or die(mysql_error());
mysql_select_db("people") or
die(mysql_error());

mysql_query("CREATE TABLE MyTable (
    id INT AUTO_INCREMENT,
    FirstName CHAR,
    LastName CHAR,
    Phone INT,
    BirthDate DATE
    PRIMARY KEY(id)
)") Or die(mysql_error());
mysql_close ();
```

في المثال بدأنا بالاتصال بمزود قاعدة البيانات، ثم استخدمنا وظيفة `mysql_select_db` لنختار قاعدة البيانات "people" ثم أنشأنا جدولاً سميناه "persons" يحوي خمس أعمدة.

لاحظ في عمود "id" استخدمنا `INT` لنحدد أن هذا العمود سيحوي أرقاماً ثم أضفنا `AUTO INCREMENT` لنزيد الأرقام بشكل تلقائي لتأكد من أن كل سطر في الجدول يحوي رقم تعريف مميز.

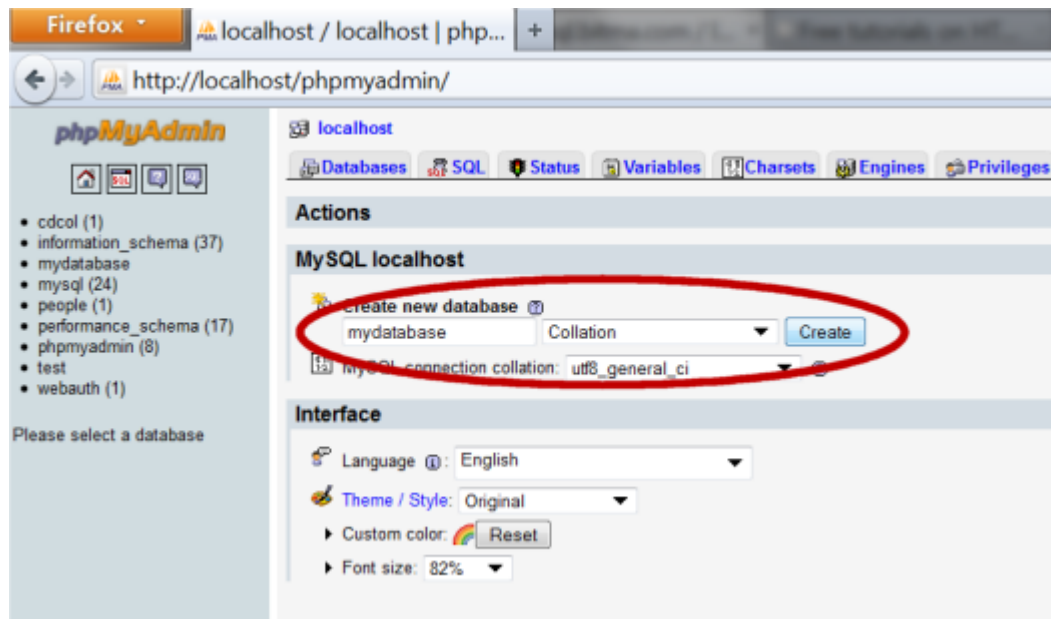
في نهاية الاستعلام استخدمنا `PRIMARY KEY` لتحديد عمود "id" ليكون المفتاح الأساسي، المفتاح الأساسي يحدد كل سجل أو سطر في الجدول برقم خاص وهذا سيكون مفيداً لاحقاً عند تحديث قاعدة البيانات.

إنشاء قاعدة البيانات والجداول في phpMyAdmin

من المفيد أن تتمكن من إنشاء قواعد البيانات والجداول مباشرة في PHP ، لكن غالباً من الأسهل استخدام - phpMyAdmin أو أي أداة مماثلة - وهي الأداة المتوفرة في معظم خدمات الـ XAMPP ، لقطات الشاشة أدناه توضح كيفية إنشاء قواعد البيانات والجداول في phpMyAdmin.

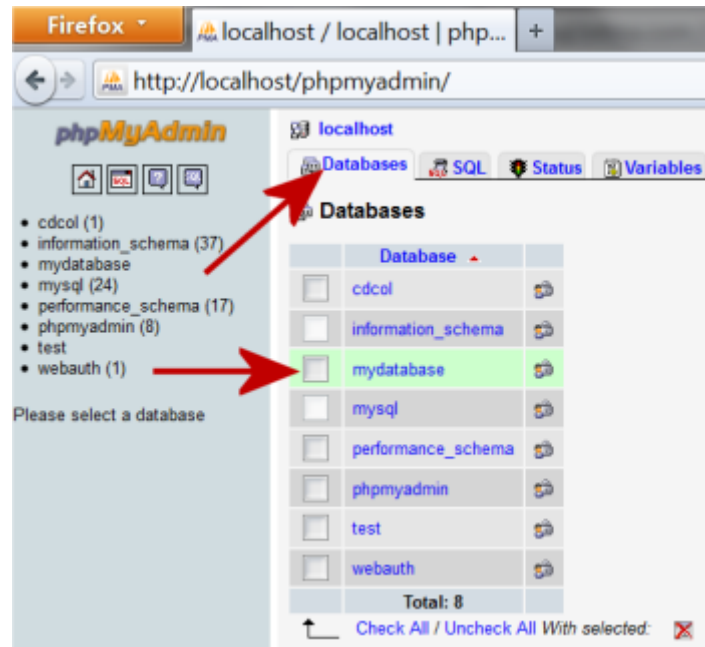
ابدأ بتسجيل دخولك إلى phpMyAdmin ، في الغالب العنوان سيكون نفس عنوان مزود قواعد البيانات (MySQL مثال "http://mysql.myhost.com") وب نفس الاسم وكلمة السر، في XAMPP العنوان هو. http://localhost/phpmyadmin/

بعد تسجيل دخولك اكتب اسماً لقاعدة البيانات ثم اضغط على الزر: "Create"



في بعض شركات الاستضافة قد تجد قاعدة بيانات جاهزة ولا تستطيع أن تنشأ واحدة بنفسك، في هذه الحالة استخدم قاعدة البيانات المتوفرة.

لإنشاء جداول اضغط على "Databases" واختر قاعدة البيانات:



سيكون هناك صندوق بعنوان "Create new table in database" يمكنك أن تكتب فيه اسم الجدول وعدد الأعمدة التي سيحويها ثم اضغط على زر: "Go"

Create new table on database mydatabase

Name: people Number of fields: 5

Go

بعد ذلك يمكنك تسمية الأعمدة وتحديد نوع البيانات كما في مثال SQL أعلاه.

Field	Type	Length/Values ¹	Default ²	Collation	Attributes	Null	Index	A_I
id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
FirstName	CHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
LastName	CHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
Phone	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>
BirthDate	DATE		None			<input type="checkbox"/>	---	<input type="checkbox"/>

لاحظ أننا حددنا "id" كمفتاح رئيسي أي **PRIMARY KEY** وهو يستخدم **AUTO INCREMENT** أو (A_I).

الآن أنشأت قاعدة البيانات الخاصة بك وجدولاً، في الدرس التالي سنلقي نظرة على إدخال واسترجاع وحذف البيانات في قاعدة البيانات.

الدرس ١٩ : إدخال البيانات في قاعدة البيانات

في هذا الدرس سنلقي نظرة على إدخال البيانات في قاعدة البيانات مباشرة من خلال PHP.

إدخال البيانات من خلال SQL

تستخدم SQL لإدخال البيانات في قاعدة البيانات بنفس الطريقة التي تستخدم SQL لإنشاء قاعدة البيانات والجداول، استعلام SQL يكتب بهذا الشكل:

```
INSERT INTO TableName(column1, column2, ...)
VALUES (value1, value2, ...)
```

كما ترى يمكنك تحديث عدة أعمدة في استعلام SQL بتحديد ما في قائمة تفصل بين كلماتها بفاصلة، لكن بالطبع يمكنك تحديد عمود واحد وقيمة واحدة، الأعمدة غير المذكورة في الاستعلام ستبقى فارغة.

مثال: إدخال بيانات شخص في الجدول

في هذا المثال سنستخدم قاعدة البيانات من [الدرس ١٨](#)، لنقل أننا نرغب في إدخال بيانات شخص في قاعدة البيانات، يمكن أن يكون اسمه *Gus Goose* ورقم هاتفه 99887766 وتاريخ ميلاده هو 1964-04-20.

سيكون بهذا الشكل SQL استعلام

```
$strSQL = "INSERT INTO
people (FirstName, LastName, Phone, BirthDate)
VALUES ('Gus', 'Goose', '99887766', '1964-04-20')";

mysql_query($strSQL) or die(mysql_error());
```

كما ترى استعلام SQL يمكن أن يكون طويلاً، ويمكن أن تشعر بأن هذا معقد، لذلك من الإيجابي أن تكتب استعلام SQL بطريقة مختلفة:

```
strSQL = "INSERT INTO people(";

strSQL = strSQL . "FirstName, ";

strSQL = strSQL . "LastName, "
```

```

strSQL = strSQL . "Phone, ";
strSQL = strSQL . "birth) ";

strSQL = strSQL . "VALUES (";

strSQL = strSQL . "'Gus', ";
strSQL = strSQL . "'Goose', ";
strSQL = strSQL . "'99887766', ";

strSQL = strSQL . "'1964-04-20')";

mysql_query($strSQL) or die(mysql_error());

```

بهذه الطريقة استعمال SQL كتب بتقسيم جملة إلى أقسام صغيرة ثم وضع في متغير اسمه **\$strSQL**.

عملياً اختيار أي طريقة منهما لا يشكل فرقاً، لكن ما إن تعمل على جداول كبيرة من الضروري أن تعرف ما الذي تفعله لذلك اختر الطريقة الأنسب لك.

شغل البرنامج التالي لإدخال Gus Goose في قاعدة البيانات:

```

<html>
<head>
<title>Insert data into database</title>
</head>
<body>
<?php

// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// The SQL statement is built

$strSQL = "INSERT INTO people("

$strSQL = $strSQL . "FirstName, ";

```

```

$strSQL = $strSQL . "LastName, ";

$strSQL = $strSQL . "Phone, ";
$strSQL = $strSQL . "BirthDate) ";

$strSQL = $strSQL . "VALUES(";

$strSQL = $strSQL . "'Gus', ";

$strSQL = $strSQL . "'Goose', ";
$strSQL = $strSQL . "'99887766', ";

$strSQL = $strSQL . "'1964-04-20')";

// The SQL statement is executed
mysql_query($strSQL) or die (mysql_error());

// Close the database connection
mysql_close();
?>

<h1>The database is updated!</h1>
</body>
</html>

```

حفظ مدخلات المستخدم في قاعدة بيانات

غالباً سترغب في أن تحفظ مدخلات المستخدم في قاعدة البيانات.

يمكن إنجاز ذلك من خلال نموذج كما شرحنا في [الدرس ١١](#)، حيث القيم من النموذج يمكن إدخالها في استعلام SQL، افترض أن لديك نموذجاً بسيطاً بهذا الشكل:

```

<form action="insert.php" method="post">
<input type="text" name="FirstName" />
<input type="submit" value="Save" />

</form>

```

النموذج يرسل بياناته إلى ملف **insert.php** الذي يتلقى بيانات المستخدم بطلبها من محتويات النموذج، في هذا المثال استعمال SQL سيكون بهذا الشكل:

```
strSQL = "INSERT INTO people (FirstName)
values ('" . $_POST["FirstName"] . "')"
```

بنفس الطريقة من الممكن طلب البيانات من الكوكيز، الجلسات، وغيرها.

الأخطاء الشائعة عند المبتدئين

في البداية قد تواجه كثيراً من رسائل الأخطاء عندما تحاول تحديث قاعدة البيانات، لا يوجد مساحة لأي خطأ أو عدم دقة عندما تعمل مع قواعد البيانات، وجود فاصلة في المكان الخطأ قد يعني عدم تحديث قاعدة البيانات وستتلقى رسالة خطأ، أدناه نشرح الأخطاء الشائعة عند المبتدئين.

أنواع بيانات غير صحيحة

من الضروري أن يكون هناك توافق بين نوع البيانات والعمود في الجدول، كل عمود حدد له نوع بيانات، اللفظة أدناه توضح نوع البيانات للجدول في مثالنا.

Field	Type	Length/Values ¹	Default ²	Collation	Attributes	Null	Index	A.I
id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
FirstName	CHAR	255	None			<input type="checkbox"/>	--	<input type="checkbox"/>
LastName	CHAR	255	None			<input type="checkbox"/>	--	<input type="checkbox"/>
Phone	INT		None			<input type="checkbox"/>	--	<input type="checkbox"/>
BirthDate	DATE		None			<input type="checkbox"/>	--	<input type="checkbox"/>

يحدث الخطأ عندما مثلاً تحاول إدخال نص أو رقم في حقل التاريخ لذلك حاول أن تحدد بدقة نوع البيانات.

أدناه قائمة بأنواع البيانات الأكثر شيوعاً:

الحجم	نوع البيانات	الإعداد
255 حرفاً أو يمكن تحديد عدد الحروف الأقصى من خلال	نص أو خليط من النصوص والأرقام، يمكن استخدامه للأرقام التي لا تستعمل في العمليات الحسابية مثل أرقام	CHAR

	الهواتف	"Length"
TEXT	نصوص طويلة أو خليط من النصوص والأرقام.	يصل الى ٦٥٥٣٥ حرفاً.
INT	أرقام تستخدم في العمليات الحسابية.	4بايت
DATE	تواريخ بتنسيق YYYY-MM-DD	3بايت.
TIME	الوقت بتنسيق hh:mm:ss	3بايت.
DATETIME	الوقت والتاريخ بتنسيق YYYY-MM-DD hh:mm:ss	8بايت.

استعلام SQL بعلامة الاقتباس أو الشرطة المائلة للخلف (backslash)

إذا جربت إدخال نص يحوي علامة اقتباس أحادية (') أو ثنائية (") أو شرطة مائلة للخلف (\) قد لا تستطيع إدخال البيانات في قاعدة البيانات، الحل هو إضافة شرطة مائلة قبل هذه العلامات.

هذا يمكن أن ينجز من خلال وظيفة [addslashes](#) بهذه الطريقة:

```
<?php

$strText = "Is your name O'Reilly?";
$strText = addslashes($strText);

?>
```

كل علامات الاقتباس الأحادية (') والثنائية (") والشرطة المائلة (\) ستحصل على شرطة مائلة قبلها، هذا سيكون فقط لإضافة البيانات إلى قاعدة البيانات، الشرطة المائلة لن تضاف في قاعدة البيانات، لاحظ أن PHP تشغل [addslashes](#) في كل بيانات \$_GET و \$_POST و \$_COOKIE تلقائياً، لذلك لا تستخدم [addslashes](#) في التعامل مع هذه البيانات.

في الدرس التالي سنتعلم كيف تستخرج المعلومات من قاعدة البيانات، لكن أولاً حاول إدخال مزيد من المعلومات في قاعدة البيانات.

الدرس ٢٠ : استرجاع البيانات من قاعدة البيانات

الآن علينا استرجاع البيانات من قاعدة البيانات لنضعها في صفحات PHP.

هذا هو أحد أهم الدروس، عندما تقرأ وتفهم هذا الدرس ستدرك لم حلول مواقع الويب المعتمدة على قواعد البيانات قوية ونظرتك لتطوير المواقع ستتسع بشكل كبير.

استعلامات SQL

لاسترجاع البيانات من قاعدة البيانات سنستخدم الاستعلامات، مثال الاستعلام قد يكون بهذا الشكل: "استرجع كل البيانات من جدول 'people' مرتبة أبجدياً" أو "استرجع كل الأسماء من جدول 'people'".

مرة أخرى، لغة **Structured Query Language** أو (SQL) تستخدم للتواصل مع قواعد البيانات، انظر إلى المثال البسيط:

```
Get all data from the table 'people'
```

Will be written like this in SQL:

```
SELECT * FROM people
```

المثال يشرح نفسه، أكمل القراءة لترى كيف يستخدم استعلام SQL الأمثلة أدناه.

مثال ١: استرجاع البيانات من قاعدة البيانات

هذا المثال يستخدم قاعدة البيانات والجدول من [الدرس ١٩](#) و [الدرس ١٨](#)، لذلك من المهم أن تقرأ هذه الدروس أولاً.

المثال يوضح كيف أن البيانات في جدول "people" من خلال استعلام SQL.

استعلام SQL يعود بسلسلة نتائج على شكل سجلات، هذه السجلات تخزن في ما يسمى **recordset** أو مجموعة السجلات وهو ما يمكن وصفه بجدول في ذاكرة المزود يحوي صفوفاً من البيانات (سجلات) وكل سجل مقسم إلى حقول أو اعمدة.

مجموعة السجلات يمكن مقارنتها بجدول حيث كل سجل يمكن مقارنته بصف في الجدول، في PHP يمكننا أن نمرر مجموعة السجلات في حلقة ووظيفة [mysql_fetch_array](#) التي تعطينا كل صف كمصفوفة.

المثال أدناه يوضح كيف تستخدم وظيفة [mysql_fetch_array](#) للتعامل مع مجموعة السجلات:

```
<html>
<head>
<title>Retrieve data from database </title>
</head>
<body>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// SQL query
$strSQL = "SELECT * FROM people";

// Execute the query (the recordset $rs
contains the result)
$rs = mysql_query($strSQL);

// Loop the recordset $rs
// Each row will be made into an array ($row)
using mysql_fetch_array
while($row = mysql_fetch_array($rs)) {

    // Write the value of the column FirstName
    (which is now in the array $row)
    echo $row['FirstName'] . "<br />";

}

// Close the database connection
mysql_close();
?>
</body>
</html>
```

[شاهد المثال](#) ➡

لاحظ كيف نستخرج البيانات من عمود "FirstName" بكتابة \$row['FirstName'] ، بنفس الطريقة يمكننا استخراج بيانات عمود "Phone" بكتابة \$row['Phone'].

ترتيب مجموعة السجلات هو نفس ترتيبها في الجدول داخل قاعدة البيانات، لكن في المثال التالي سنوضح كيف يمكن إعادة ترتيب مجموعة السجلات.

المثال ٢: ترتيب البيانات أبجدياً، زمنياً أو حسب الأعداد

غالباً من المفيد عرض قائمة أو جدول بيانات بترتيب أبجدي أو زمني أو بحسب الأرقام، هذا الترتيب يمكن إنجازه بسهولة في SQL ، حيث طريقة كتابته **Order By ColumnName** تستخدم لترتيب المحتويات بحسب محتويات أحد أعمدة الجدول.

أنظر إلى استعمال SQL التالي:

```
strSQL = "SELECT * FROM people"
```

السجلات يمكن ترتيبها مثلاً أبجدياً بحسب الاسم الأول للأشخاص بهذه الطريقة:

```
strSQL = "SELECT * FROM people ORDER BY  
FirstName"
```

أو زمنياً بحسب تاريخ الميلاد بهذا الشكل:

```
strSQL = "SELECT * FROM people ORDER BY  
BirthDate"
```

يمكن تغيير الترتيب من تصاعدي إلى تنازلي بإضافة: **DESC**

```
strSQL = "SELECT * FROM people ORDER BY  
BirthDate DESC"
```

في المثال التالي الأشخاص يعرضون في قائمة مرتبة بحسب العمر:

```

<html>
<head>

<title>Retrieve data from database </title>

</head>
<body>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// SQL query
$strSQL = "SELECT * FROM people ORDER BY
BirthDate DESC";

// Execute the query (the recordset $rs
contains the result)
$rs = mysql_query($strSQL);

// Loop the recordset $rs
while($row = mysql_fetch_array($rs)) {

// Write the value of the column FirstName
and BirthDate
echo $row['FirstName'] . " " .
$row['BirthDate'] . "<br />";

}

// Close the database connection
mysql_close();
?>

</body>
</html>

```

➔ شاهد المثال

جرب تغيير استعلام SQL بنفسك ورتب السجلات بحسب الاسم الأول أو الاسم الأخير أو رقم الهاتف.

استرجاع معلومات مختارة

حتى الآن استعلامات SQL تطلب كل الصفوف من الجدول، لكن غالباً ستحتاج لوضع شروط أو معايير في استعلام SQL للبيانات التي تريدها، فمثلاً يمكنك أن تطلب صفوف البيانات للأشخاص الذين يملكون أرقام هواتف محددة أو أسم آخر محدد.

لنقل بأننا نريد استرجاع بيانات للناس الذين يملكون رقم الهاتف "٦٦٥٥٤٤٣٣"، يمكننا فعل ذلك بهذا الشكل:

```
strSQL = "SELECT * FROM people WHERE Phone = '66554433' "
```

هناك ٦ عوامل (operators) اتصال في: SQL

				= يساوي
من				< أصغر
من				> أكبر
يساوي	أو	من		= < أصغر
يساوي	أو	من		= > أكبر
				= ! لا يساوي

بالإضافة لذلك هناك عوامل منطق:

AND
OR
NOT

راجع [الدرس ٦](#) للمزيد من المعلومات حول كيفية استخدام الشروط والعوامل.

في المثال التالي سنستخدم الشروط لإنشاء دفتر عناوين.

مثال ٣: دفتر عناوين

في هذا المثال سنحاول الجمع بين عدة أشياء تعلمناها، سننشأ قائمة أسماء من قاعدة البيانات حيث كل اسم يكون رابطاً لمزيد من التفاصيل حول الشخص.

لهذا نحن بحاجة لملفين **list.php** - و **person.php** يحويان هذا الكود:

```

<html>
<head>
<title>Retrieve data from the database</title>
</head>
<body>

<ul>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// SQL query
$strSQL = "SELECT * FROM people ORDER BY
FirstName DESC";

// Execute the query (the recordset $rs
contains the result)
$rs = mysql_query($strSQL);

// Loop the recordset $rs
while($row = mysql_fetch_array($rs)) {

    // Name of the person
    $strName = $row['FirstName'] . " " .
$row['LastName'];

    // Create a link to person.php with the id-
value in the URL
    $strLink = "<a href = 'person.php?id = " .
$row['id'] . "'>" . $strNavn . "</a>";

    // List link
    echo "<li>" . $strLink . "</li>";

}

```

```

// Close the database connection
mysql_close();
?>

</ul>
</body>
</html>

```

الكود في person.php

```

<html>
<head>
<title>Retrieve data from database</title>
</head>
<body>

<dl>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// Get data from the database depending on the
value of the id in the URL
$strSQL = "SELECT * FROM people WHERE id=" .
$_GET["id"];
$rs = mysql_query($strSQL);

// Loop the recordset $rs
while($row = mysql_fetch_array($rs)) {

    // Write the data of the person
    echo "<dt>Name:</dt><dd>" .
$row["FirstName"] . " " . $row["LastName"] .
"</dd>";

```

```

        echo          "<dt>Phone:</dt><dd>"
$row["Phone"] . "</dd>";
        echo          "<dt>Birthdate:</dt><dd>"
$row["BirthDate"] . "</dd>";

    }

    // Close the database connection
    mysql_close();
    ?>

</dl>
<p><a href="list.php">Return to the
list</a></p>

</body>

</html>

```

[شاهد المثال](#)

مثال دفتر العناوين بسيط إلى حد ما، لكن يظهر لنا الإمكانيات المتوفرة عند العمل مع PHP وقواعد البيانات.

تخيل أن قاعدة البيانات تحوي ١٠ آلاف منتج بتفاصيلها ووصفها، بتعديلات قليلة في الملفين السابقين يمكنك بسهولة إنشاء دليل منتجات يحوي أكثر من ١٠ آلاف صفحة، يمكن إنجاز ذلك بقاعدة بيانات واحدة وملفي PHP فقط.

أهلاً بك في عالم المواقع الواسعة التي يسهل تطويرها وصيانتها، عندما تتعلم العمل مع قواعد البيانات فلن تكون حلول الويب الخاصة بك مثلما كانت من قبل.

الدرس ٢١ : حذف البيانات من قاعدة البيانات

في الدرسين السابقين تعلمت كيف تدخل المعلومات وتسترجعها من قاعدة البيانات، في هذا الدرس سنلقي نظرة على كيفية حذف السجلات وهو أمر أسهل من إضافة البيانات.

الحذف باستخدام SQL

طريقة كتابة استعلام SQL لحذف السجلات يكون بالشكل التالي:

```
DELETE FROM TableName WHERE condition
```

مثال: حذف سجل

عند حذف سجل يمكنك استخدام حقل الرقم المميز **AutoNumber** في قاعدة البيانات، في قاعدة البيانات الخاصة بنا هناك عمود اسمه **id** ، باستخدام معرف مميز تتأكد بأنك ستحذف سجلاً واحداً فقط، في المثال التالي سنحذف سجل رقمه المميز هو ٢٤:

```
<html>
<head>
<title>Delete data in the database</title>
</head>

<body>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());

// Select database
mysql_select_db("mydatabase") or
die(mysql_error ());

// The SQL statement that deletes the record
$strSQL = "DELETE FROM people WHERE id = 24";
mysql_query($strSQL);

// Close the database connection
mysql_close();
?>
```



```
<h1>Record is deleted!</h1>
```

```
</body>
```

```
</html>
```

تذكر أنه لا توجد "سلة المحذوفات" عند العمل مع قواعد البيانات و PHP، عندما تحذف سجلاً فهذا يعني أنك لن تستطيع استرجاعه.

الدرس ٢٢ : تحديث البيانات في قاعدة البيانات

في الدروس السابقة تعلمت إدخال واسترجاع وحذف البيانات من قاعدة البيانات، في هذا الدرس سنلقي نظرة على كيفية تحديث قاعدة البيانات، أي تحرير القيم الموجودة في الجدول.

تحديث البيانات باستخدام SQL

طريقة كتابة استعلام SQL لتحديث البيانات هو:

```
UPDATE TableName SET TableColumn='value' WHERE condition
```

يمكن أيضاً تحديث أكثر من حقل في نفس استعلام SQL:

```
UPDATE TableName SET TableColumn1='value1',
TableColumn2='value2' WHERE condition
```

بما تعلمته الآن من الدروس ١٩، ٢٠ و ٢١ يجب أن يكون من السهل عليك أن تعرف كيف تستخدم المثال أعلاه عملياً، لكن بالطبع سنلقي نظرة على مثال.

مثال: تحديث حقول في جدول "people"

البرنامج أدناه يحدد بيانات اسم Donald Duck إلى D. ويغير رقم الهاتف إلى 44444444 ، المعلومات الأخرى لن تتغير، بإمكانك تغيير معلومات أخرى بكتابة استعلام SQL الخاص بك.

```
<html>
<head>
<title>Update data in database</title>

</head>
<body>

<?php
// Connect to database server
mysql_connect("mysql.myhost.com", "user",
"sesame") or die (mysql_error ());
```

```

// Select database
mysql_select_db("mydatabase") or
die(mysql_error());

// The SQL statement is built
$strSQL = "Update people set ";
$strSQL = $strSQL . "FirstName= 'D.', ";
$strSQL = $strSQL . "Phone= '44444444' ";

$strSQL = $strSQL . "Where id = 22";

// The SQL statement is executed
mysql_query($strSQL);

// Close the database connection
mysql_close();
?>

<h1>The database is updated!</h1>
</body>
</html>

```

بهذا المثال نكمل الدروس حول قواعد البيانات، تعلمت إدخال وحذف وتحديث البيانات في قاعدة البيانات باستخدام PHP ، وهكذا أنت الآن تستطيع أن تنشأ مواقع تفاعلية متقدمة حيث يمكن للمستخدمين إدارة وتحديث قاعدة البيانات باستخدام النماذج.

إذا كنت تريد رؤية مثال متقدم بما يمكن فعله بلغة البرمجة PHP وقواعد البيانات فجرب [الانضمام إلى منتدانا](#)، الانضمام مجاني ولن يأخذ منك أكثر من دقيقة لفعل ذلك، يمكنك أن تدير ملفك الخاص باستخدام النماذج، وقد تحصل على أفكار لموقعك.

هذا أيضاً نهاية الدرس، PHP تعطيك إمكانيات كثيرة لإضافة التفاعل لموقعك، الحد الوحيد لما يمكنك فعله هو خيالك، استمتع!

المرجع

<http://ar.html.net/tutorials/php>

تم بحمد الله وشكره