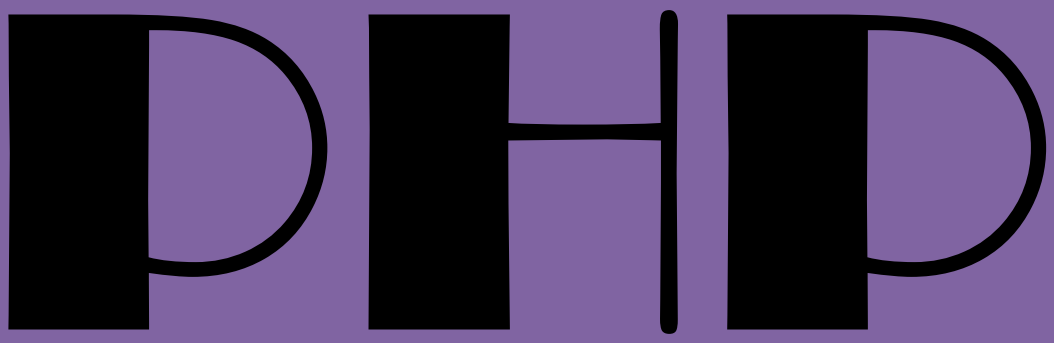
تطوير مواقع انترنت تفاعلية باستخدام



تطبيقات عملية متنوعة تُساعد في تقريب الصورة

أ. طلال الكريمي talalh@tvtc.gov.sa قسم تقنية الحاسب الكلية التقنية ببيشة

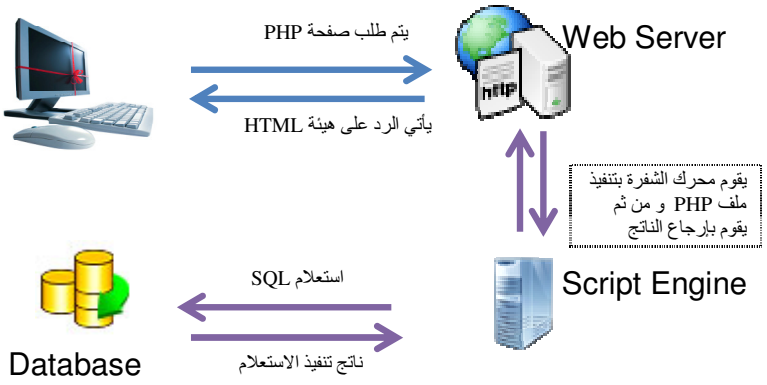
المحتويات

هذا الفصل يغطي:

- **تقديم للغة PHP**
- **متطلبات التركيب**
- **تجهيز بيئة العمل**

ما هي PHP

PHP (ترمز إلى PHP:Hypertext Preprocessor) عبارة عن لغة برمجة مفتوحة المصدر تُستخدم بشكل واسع لأغراض متعددة ، و صُممت خصيصا لتطوير مواقع إنترنت تفاعلية. و تختلف PHP عن JavaScript في أن الشفرة البرمجية يتم تنفيذها على الخادم web server و من ثم يُرسل ناتج التنفيذ على هيئة HTML للمستخدم.



ماذا يمكنها أن تعمل

يمكن أن تُستخدم PHP لعمل ما يلي:

١. إنشاء تطبيقات جهة خادم server-side application ، و هو الهدف الأساسي من PHP .
٢. إنشاء برامج تعمل على موجه الأوامر command line prompt بدون الحاجة لمتصفح إنترنت أو خادم ويب.
٣. إنشاء تطبيقات سطح مكتب desktop applications تحتوي على واجهة مستخدم graphical user interface.

ما هي مميزاتها

تتميز PHP بعدد من المزايا منها:

١. يمكن تنفيذها على العديد من أنظمة التشغيل منها
   أ- Microsoft Windows
   ب- Linux
   ت- Unix
   ث- Mac OS X

٢. تدعم العديد من خوادم الويب كـ Apache و IIS.
٣. تدعم العديد من أنظمة قواعد البيانات كـ
   أ- Oracle
   ب- MySql
   ت- SQL Server
   ث- Microsoft Access
   ج- Postgre SQL
٤. تدعم البرمجة الكينونة object oriented programming والبرمجة الاجرائية procedural-programming و يمكن الخلط بينهما.
٥. لا تدعم PHP عرض البيانات على هيئة HTML فقط بل يمكن عرض البيانات على هيئة صورة ، PDF أو حتى اخراج البيانات على هيئة ملفات فلاش.

متطلبات PHP

قبل أن نستطيع تنفيذ ملفات PHP سنحتاج إلى خادم server و ذلك لأن PHP من لغات برمجة الخادم server side scripting .

لغات البرمجة النصية Scripting Languages : يقصد بها لغات البرمجة عالية المستوى و التي يتم تفسيرها عن طريق برنامج آخر وقت التنفيذ Run Time بدلا من عملية الترجمة و التي تتم بواسطة معالج الكمبيوتر كما هو الحال في لغات برمجة كـ C و C++. من أمثلة هذا النوع Javascript,PHP,ASP,JSP,Perl .

خادم ويب الاباتشي (Apache Web Server)

الاباتشي Apache عبارة عن خادم ويب وظيفته الأساسية استقبال الطلبات Http Request و تسليمها إلى محرك الشفرة Script Engine و من ثم لاحقا يقوم Apache باستلام الرد Http Response من محرك الشفرة و تسليمها الى مرسل الطلب على هيئة HTML. وقد صمم الاباتشي ليعمل مع نظام التشغيل يونكس Unix . لاحقا تم تهيئته للعمل مع نظام التشغيل ويندوز Windows وبعض أنظمة التشغيل الخاصة بالشبكات.
و يمتاز خادم الأباتشي بمجموعة المزايا و الإمكانيات منها:

- إمكانية عمل صفحات خطأ مخصصة Customized error pages.
- إمكانية إخفاء عناوين الانترنت URL.
- سجلات الأخطاء و الاستخدام بأكثر من هيئة.

و يمكن الحصول على نسخة من البرنامج من خلال زيارة موقع فريق عمل الاباتشي التالي:
http://httpd.apache.org/download.cgi .

نظام قواعد البيانات MySql

MySql عبارة عن نظام قواعد بيانات مفتوح المصدر. و تصنف MySql على انها من أنظمة إدارة قواعد البيانات العلائقية RDBMS صُممت للعمل مع الاستعلامات المعقدة و الأحمال الثقيلة. كما تتيح إمكانية ربط عدد كبير من الجداول ببعضها البعض للوصول إلى أعلى قدر من الكفاءة و السرعة.
أخيرا يمكن تحميل نسخة من البرنامج من خلال الرابط التالي:
/http://dev.mysql.com/downloads .

برنامج XAMMP

الكثيرون يعرفون أن عملية تركيب خادم ويب Apache و تهيئته للعمل مع PHP و MySql ليست بالعملية السهلة و دائما ما تستهلك وقت و جهد للتوفيق بينها. لذلك يأتي دور برنامج XAMMP و الذي جعل من عملية التركيب تتم بشكل سهل و سريع جدا . يمكن تحميل نسخة من البرنامج من خلال الرابط التالي: http://www.apachefriends.org/en/xampp.html .

تحميل XAMMP

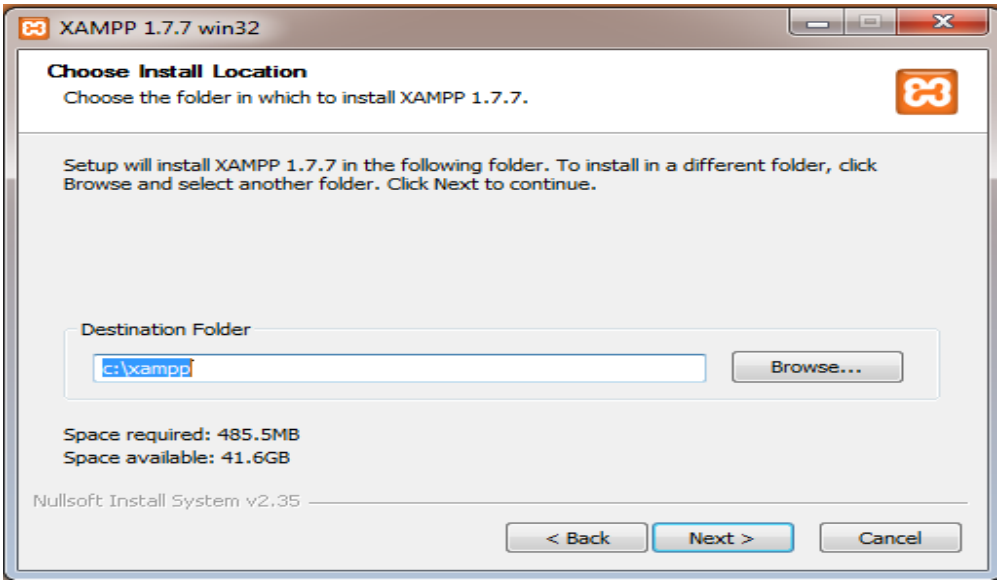
قبل عملية التركيب سنقوم بتحميل البرنامج من خلال الخطوات التالية:

١. قم بالتوجه إلى الصفحة التالية: http://www.apachefriends.org/en/xampp.html .
٢. أسفل الصفحة قم باختيار إصدارة البرنامج المتوافقة مع نظام التشغيل لديك. في حالتنا سنقوم باختيار النسخة المتوافقة مع نظام التشغيل Windows لذلك سنقوم بالنقر على الرابط : XAMMP for Windows .
٣. من الصفحة الجديدة نسحب شريط التمرير إلى منتصف الصفحة وصولا عند القسم تحميل (Download) ، بعد ذلك قم بالنقر على Installer .
٤. أخيرا ستظهر نافذة تطالب بتحديد مكان لحفظ الملف ، قم باختيار سطح المكتب أو المستندات لحفظ الملف.

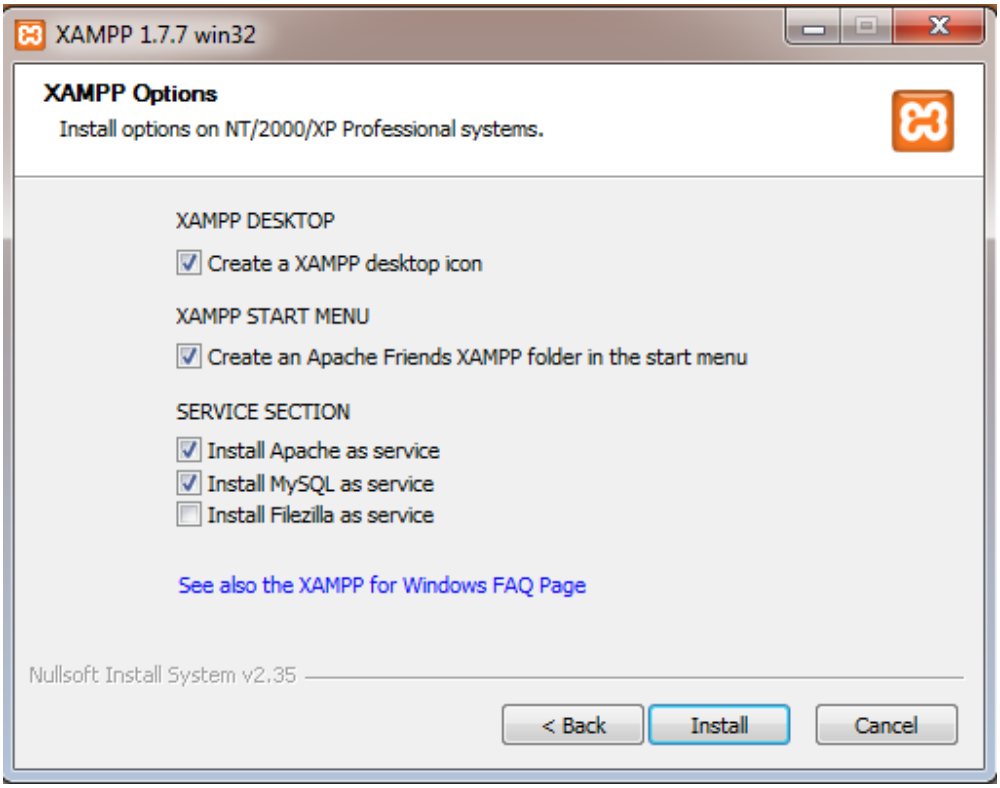
تركيب XAMMP

بعد الانتهاء من تحميل برنامج XAMMP سنبدأ الآن في خطوة تركيب البرنامج و الذي سيقوم بتهيئة Apache ، PHP و MySql للعمل معا. خطوات التركيب:

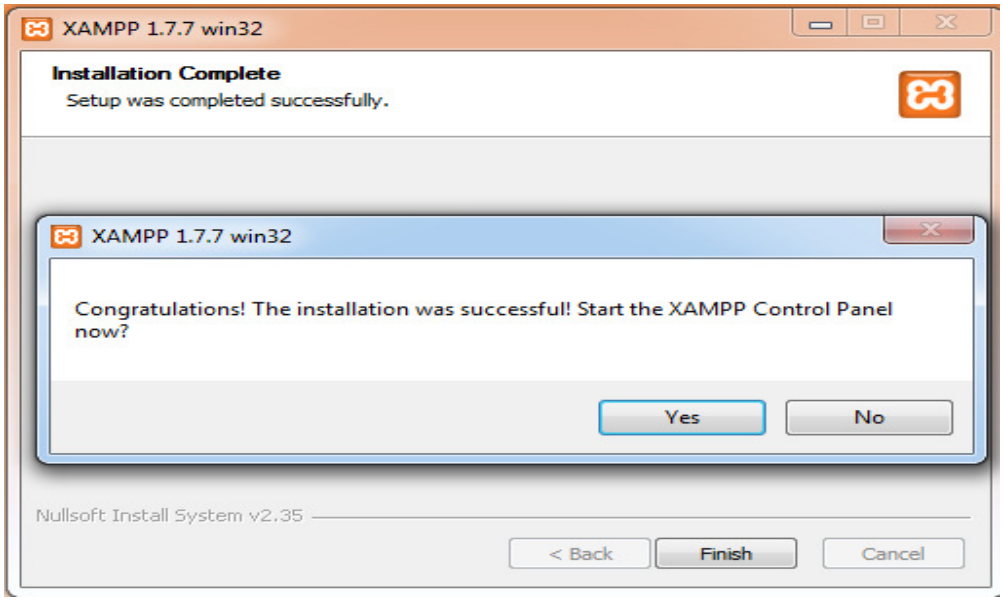
١. انقر بزر الفأرة الايسر مرتين على ايقونة البرنامج.
٢. أول نافذة هي نافذة تحديد اللغة و التي من خلالها سنختار الخيار الافتراضي English و من ثم نضغط Next. و من النافذة التالية نضغط على Next أيضا.
٣. في النافذة التالية نحدد مكان مسار البرنامج (و يفضل C:\xampp).

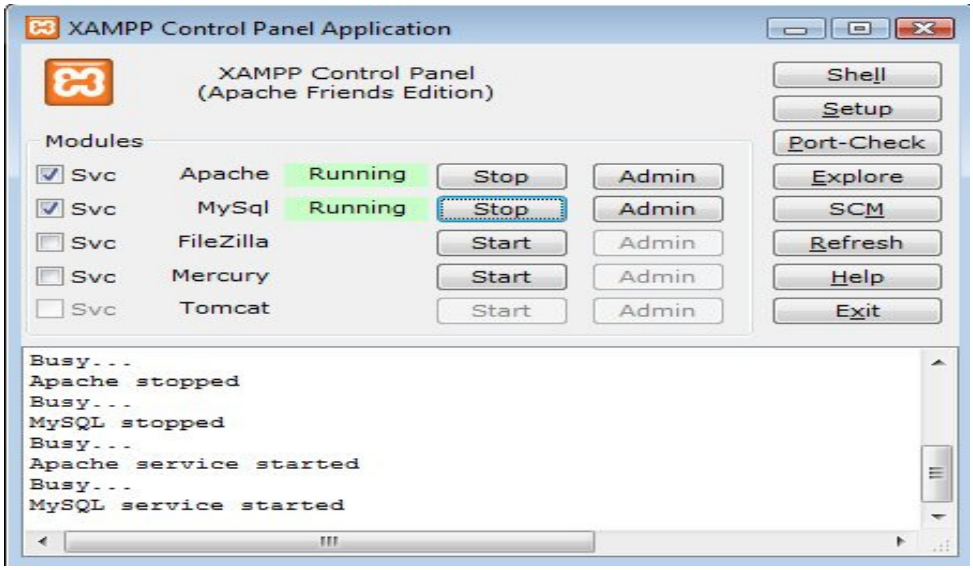


٤. النافذة التالية من خلالها نحدد ما إذا كنا نرغب في تركيب Apache و Mysql كخدمة Service من خدمات ويندوز، بمعنى أنهما سيعملان بمجرد تشغيل نظام التشغيل.

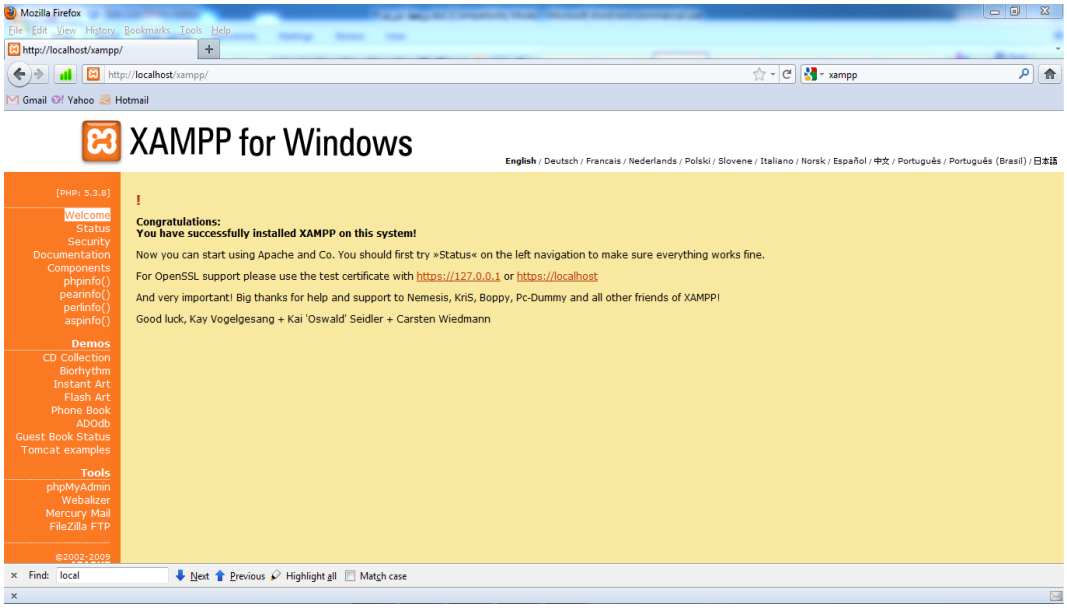


٥. بعد ذلك سيقوم البرنامج باستخراج و تركيب الملفات المطلوبة. و من ثم عند الانتهاء ستظهر النافذة التالية و التي ستخبرنا بانتهاء عملية التركيب بنجاح و نستطيع تشغيل لوحة تحكم البرنامج بالضغط على زر Yes.





٦. بعد الانتهاء من تركيب XAMMP سنقوم بفتح متصفح الانترنت و كتابة العنوان التالي: http://localhost . عندها ستظهر صفحة إختيار اللغة لتطبيق XAMPP و التي من خلالها سنقوم بإختيار اللغة الانجليزية English لكي ننتقل الى الصفحة الرئيسية لـ XAMMP.



أين سيتم حفظ ملفات PHP

على عكس HTML ، يجب أن نقوم بحفظ ملفات PHP في المكان المخصص لها و الذي يسمى بالمجلد الجذر root directory. و المجلد الجذر هو htdocs و الذي يقع في المسار التالي : C:\xammp\htdocs . أي ملف يوضع داخل هذا المجلد سيكون قابل للوصول عن طريق كتابة http://localhost/ أو http://127.0.0.1/ في شريط عنوان متصفح الانترنت .

يمكن تغيير المجلد الجذر من خلال تغيير القيمة الافتراضية المحددة في ملف إعدادات الاباتشي و الموجودة في c:\xammp\apache\conf\httpd.conf و تغيير قيمة لـ DocumentRoot . أخيرا لا تنسى القيام بإعادة تشغيل الاباتشي لكي تصبح التغييرات الجديدة نافذة.

ما هو المحرر المناسب لكتابة شفرة PHP

تستطيع استخدام أيا من محررات النصوص كبرنامج المفكرة Notepad أو برنامج الدفتر WordPad أو برنامج المفكرة المطور Notepad++ أو البرامج ذات الواجهة الرسومية كبرنامج Adobe Dreamweaver.
و المحرر الذي سنقوم باستخدامه في الشرح هو برنامج Notepad++ و الذي يمكن تحميله بشكل مجاني من خلال زيارة الموقع التالي: http://notepad-plus-plus.org/download/ .

تطبيق عملي

قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<html>
  <head>
     <tilte>تطبيق عملي</tilte>
  </head>
  <body>
    <font color="blue">تجربة كتابة كود PHP</font>
    <p>
    <?php
            echo 'PHP عالم جميل';
    ?>
    </p>
  </body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم test.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/test.php و من ثم لاحظ الناتج.
أخيرا ، اضغط على زر الفأرة الأيمن (على نافذة المتصفح) و من ثم اختر عرض المصدر view source ، أو من القائمة المنسدلة عرض view اختر المصدر source (قد تختلف من متصفح لآخر) ، **ماذا تلاحظ؟**

أسئلة نهاية الفصل

١. ما هو الهدف الرئيسي من استخدام لغة PHP.

٢. عدد ثلاث من مميزات PHP.

٣. أين سيتم حفظ ملفات PHP لكي نتمكن من عرضها لاحقا. و كيف سيتم استدعاء الصفحة؟

٤. ماذا يقصد بلغات البرمجة النصية scripting languages؟

٥. قم بتحميل برنامج XAMMP و تركيبه على جهازك الشخصي متبعا خطوات التركيب المذكورة سابقا.

هذا الفصل يغطي:

- **كيف كتابة و تنفيذ شفرة PHP**
- **تركيب لغة PHP**
- **تنفيذ شفرة PHP من موجه الأوامر**

تنفيذ ملفات PHP

كما ذكرنا سابقا، تعتبر PHP من لغات برمجة جهة الخادم Server-side programming. بمعنى أن خادم الويب Apache Web Server يقوم بإرسال شفرة PHP إلى محرك الشفرة PHP script engine و الذي يقوم بتنفيذ الشفرة و من ثم إرجاع نتيجة التنفيذ إلى خادم الويب و الذي يقوم بإرجاع ناتج التنفيذ الى جهاز العميل و غالبا يكون على هيئة HTML (راجع الرسم التوضيحي في الفصل السابق) . و لكي يتمكن خادم الويب من فهم و معالجة شفرة PHP لابد من إخباره بأن صفحة الانترنت تحتوي على شفرة PHP. و لتحقيق ذلك يجب أن نتبع الخطوات التالية:

١. يجب حفظ صفحة الانترنت بامتداد php. مثال index.php
٢. يجب حصر شفرة PHP بين وسوم PHP.

```
<?php

يتم كتابة الكود هنا

?>
```

وسوم PHP

يمكن كتابة شفرة PHP بين وسوم PHP الموضحة في الجدول التالي:

الوسوم القياسية Standard tags	`<?php`
	`الشفرة هنا`
	`?>`

الوسوم المختصرة Short tags	`<?`
	`الشفرة هنا`
	`?>`

وسوم سكريبت Script tags	`<script language="php">`
	`الشفرة هنا`
	`</script>`

وسوم لغة ASP	`<%`
	`الشفرة هنا`
	`%>`

و ينصح دائما باستخدام الطريقة القياسية <?php ?> لضمان عدم حصول مشاكل تتعلق بالتوافقية compatibility و لأن كل خوادم الويب web servers تدعم العمل بها.

الفاصلة المنقوطة (; Semicolon)

تتطلب PHP أن يتم إنهاء أغلب تعليماتها بواسطة فاصلة منقوطة (;) . وسم إغلاق PHP ?> يشتمل على فاصلة منقوطة لذلك ليس هناك حاجة لإنهاء آخر سطر من شفرة PHP بواسطة فاصلة منقوطة.

```
<?php
    echo 'This is a test';
?>

<?php echo 'This is a test' ?>

<?php echo 'We omitted the last closing tag';

php.net(المصدر)
```

التعليقات (Comments)

لإدراج تعليق في سطر واحد فقط ، نستخدم (//) أو (#). أما في حالة الرغبة في إدراج تعليق لأكثر من سطر واحد فنستخدم (/* */).

```
<?php

// تعليق في سطر واحد

# تعليق في سطر واحد

/*
تعليق في
أكثر من
سطر */
?>
```

يقصد بعبارة (تعليق comment)، كل تعليمة سيتجاهلها مفسر (Parser) لغة PHP أثناء عملية التنفيذ.

المسافة الفارغة whitespace

لا تعتبر PHP حساسة للمسافة الفارغة إلا في عدد من الحالات :

- لا يمكننا ترك مسافة فارغة بين وسم PHP ، كأن نكتب : php ?> (غير صحيح) .
- لا يسمح بترك مسافة فارغة بين الكلمات المحجوزة من قبل PHP ، كأن نكتب : fo r , wh ile (غير صحيح).
- لا يسمح باستخدام المسافات الفارغة مع أسماء المتغيرات و الدوال كأن نكتب : $first name (غير صحيح).

إستخدام الاقواس الكبيرة { }

يتم استخدام الاقواس الكبيرة عند الحاجة لتنفيذ مجموعة من الأوامر في حالة محددة. كما في حالة الدوال functions أو جمل الشرط condition statements أو حلقات التكرار loops.

```
<?php
if( $i > 2 )
{
   //Statement1
   //Statement2
}
?>
```

PHP و HTML

يمكن تضمين شفرة HTML داخل ملف PHP بطريقتين:

١. أن تُكتب HTML خارج الجزء الخاص بشفرة PHP.

```
<html>
<head><title>PHP inside HTML</title>
</head>
<body>
<h1>تعلم بي اتش بي بطريقة سهلة</h1>
<?php
echo "هذه الطريقة الأولى" ;
?>
</body>
</html>
```

٢. أن يتم كتابة HTML داخل عبارة echo أو الدالة print لكن لابد أن تكون محصورة بين علامات الاقتباس المفردة أو المزدوجة.

```
<?php
echo '<html>';
echo '<head><title>الطريقة الثاني</title></head>';
print '<body>';
print '<h1> هذه هي الطريقة الثانية و التي تدل على مرونة
PHP</h1>';
echo'</body></html>';
?>
```

تنسيق شفرة PHP

عند كتابة أي شفرة برمجية باستخدام PHP ينصح الأخذ في الاعتبار تحسين مظهر الشفرة المكتوبة. و يتم ذلك عادة بإضافة مسافة محددة (أربع خانات مثلا) قبل كل تعليمة تكون تابعة للتعليمة التي تسبقها بالإضافة إلى استخدام التعليقات comments .

```
<?php
// هذا البرنامج يقوم باختبار قيمة المتغير x
$x=5;
if(x < 1)
   echo "  x is less than 1 " ;
else
   echo "  x is bigger than 1 " ;
?>
```

علما أن تنسيق شفر PHP ليس له أي تأثير في عملية التنفيذ. لكنها مفيدة في :

- عملية تنظيم الشفرة و جعلها أكثر قابلية للقراءة.
- المساعدة كذلك في عملية الصيانة و التطوير لاحقا.
- تسهيل عملية الصيانة.

print و echo

لطباعة النتائج في PHP نستطيع استخدام إما echo أو print . والفرق الوحيد بينهما أن print تعتبر من دوال PHP بينما echo تُعتبر من تصاريح PHP (راجع المثال في الصفحة السابقة).

علامة الاقتباس المفردة (' Single quote) و المزدوجة (" Double quotes)

عند التعامل مع النصوص يجب أن يتم تضمينها بين علامات الاقتباس المفردة أو المزدوجة. و الفرق بين العلامتين هو أن المفسر PHP interpreter سيقوم بالتعامل مع أي بيانات موضوعة بين علامة الاقتباس المفردة و كأنها نص عادي. بينما في حالة علامة الاقتباس المزدوجة فسيقوم المفسر بالبحث عن أي متغيرات أو حروف خاصة لكي يقوم بمعالجتها.

```
<html>
<head><title>Single and Double quotes</title>
</head>
<body>
<?php
$x = 1;
echo   "   x =\t$x   <br/> ";
echo   '   x =\t$x   <br/> ';
?>
</body></html>
```

PHP حساسة لحالة الحرف case-sensitive

PHP من لغات البرمجة الحساسة لحالة الحرف. بمعنى أن $x تختلف عن $X و test( ) تختلف أيضا عن Test( ) و هكذا.

```
<?php
$number = 60;
echo'$number='.$number;
echo'<br>$number='.$Number;
?>
```

```
$number=60
$number=
```

تنفيذ PHP من موجه الأوامر command line

بدلا من الطريقة القياسية لتنفيذ ملفات PHP و التي نستخدم فيها متصفح الانترنت ، نستطيع تنفيذ شفرة PHP من خلال موجه الأوامر. و الطريقة كالتالي:

١. قم بكتابة شفرة PHP التالية:

```
<?php
echo 'executing PHP in command line…';
?>
```

٢. احفظ الملف السابق باسم commandline.php داخل المجلد الافتراضي (c:\xammp\htdocs\commandline.php).

٣. انتقل إلى كل البرامج --> البرامج الملحقة --> موجه الأوامر.

٤. بعد فتح موجه الأوامر قم بكتابة سلسلة الأوامر التالية بالترتيب:

- cd\
- cd c:\xampp\php
- php c:\xampp\htdocs\commandline.php الآن قم بالضغط على زر Enter و لاحظ الناتج.

```
C:\Users\user>cd\
C:\ cd c:\xampp\php
C:\xampp\php>php c:\xampp\htdocs\commandline.php
executing PHP in command line…

C:\xampp\php>
```

أسئلة نهاية الفصل

١. لماذا لا ينصح باستخدام <? ?> في كتابة شفرة PHP ؟

٢. هل يجب إستخدام الفاصلة المنقوطة (;) مع كل تعليمات PHP؟

٣. كيف يمكن تنسيق شفرة PHP و ما الفائدة منها؟

٤. ما الفرق بين علامة الاقتباس المفردة و المزدوجة؟

٥. هل تعتبر PHP حساسة لحالة الحرف، اضرب مثال لذلك؟

٦. قم بكتابة شفرة PHP ، الغرض منها طباعة اسمك. مع الاخذ في الاعتبار ضرورة أن يظهر الاسم في وسط الصفحة و بحجم كبير جدا؟

٧. صح أم خطأ:

لا يوجد فرق بين كلا من : echo و print .

هذا الفصل يغطي:

- **المتغيرات في PHP**
- **أنواع البيانات data types**
- **كيفية التعامل مع الثوابت**
- **المعاملات الخاصة و كيفية استخدامها**

المتغيرات variables

المتغير هو عبارة عن حاوية container تستخدم لحفظ البيانات بشكل مؤقت . في PHP يمكن أن تحتوي المتغيرات على أي نوع من البيانات كالأرقام أو النصوص أو المصفوفات . و تعتبر PHP من اللغات الغير صارمة من ناحية نوع البيانات loosely typed. بمعنى أن نوع البيانات يختلف باختلاف القيمة المخزنة داخل المتغير.

تسمية المتغيرات naming variables

لتعريف المتغيرات في PHP يجب الأخذ في الاعتبار عدد من القواعد:

- تبدأ المتغيرات في PHP بعلامة الدولار $.
- الذي يلي علامة $ يجب أن يكون إما حرف [a-z A-Z] أو شرطة سفلية _ .
- لا يسمح بالمسافات space و علامات الترقيم punctuation.
- أسماء المتغيرات حساسة لحالة الحرف case sensitive.

كما ينصح عند اختيار اسم لمتغير أن يكون الاسم المختار ذا معنى. أي عند النظر الى اسم المتغير نستطيع أن نعرف ما هي البيانات التي سيتم حفظها في ذلك المتغير.

```
$name = ' صالح ';
$_name = 'صالح ';
$2name = 'غير صالح لأنه بدأ برقم '; // غير صالح
```

اسناد القيم للمتغيرات assign values to variables

عملية إعطاء المتغيرات قيم تتم بطريقة سهلة في PHP . كل ما علينا عمله هو استخدام رمز المساواة (=). عند استخدام هذا الرمز تتم عملية اسناد القيمة الموجودة في اليسار الى المتغير الموجود في الطرف الايمن.

```
$name = " محمد ";
```

في المثال السابق:
$name هو اسم المتغير
= علامة المساواة
"محمد" هي قيمة المتغير
كما يجب أن لا ننسى أن ننهي العبارة بفاصلة منقوطة. المثال التالي يوضح كيفية التعامل مع المتغيرات:

```
<html>
<head><title>Variables</title>
</head>
<body>
<?php
$name = " محمد عبد الله ";
?>
<b> مرحبا يا </b> <h4> <?php echo $name; ?> </h4>
</body></html>
```

كما هو واضح من المثال السابق ستقوم echo بطباعة قيمة المتغير $name.
كما يمكن إسناد قيمة متغير إلى متغير آخر أو أن تكون قيمة المتغير ناتجة من تنفيذ عملية حسابية أو يتم إدخالها عن طريق المستخدم لقيمة المتغير عن طريق نموذج HTML form.

```
<?php
// إعطاء متغير قيمة
$now = 2012;

// اسناد قيمة متغير الى متغير آخر
$currentYear = $now;

// قيمة متغير عن طريق عملية حسابية
$lastYear = $currentYear - 1;
echo "<div dir=rtl>";
echo 'إنتهى العام ';
echo $lastYear;
echo'<br/> مرحبا بكم في العام ';
echo $currentYear;
echo "</div>";
?>
```

إنتهى العام ٢٠١١
مرحبا بكم في العام ٢٠١٢

الاسناد بواسطة المرجع assign by reference

تدعم PHP ميزة أخرى عند التعامل مع المتغيرات و ذلك عن طريق الاسناد بواسطة المرجع. بمعنى أن المتغير الجديد يقوم بالإشارة إلى المتغير الأساسي. وللإسناد بواسطة المرجع نقوم بإدراج العلامة & قبل اسم المتغير الذي ستتم الاشارة إليه. في هذه الحالة ستصبح المتغيرات مرتبطة ببعضها البعض.

```
<?php
$foo = 'Bob';              // Assign the value 'Bob' to $foo
$bar = &$foo;              // Reference $foo via $bar.
$bar = "My name is $bar<br/>";  // Alter $bar...
echo $bar;
echo $foo;                 // $foo is altered too.
?>
```

```
My name is Bob
My name is Bob
```

لحذف قيمة متغير، نستخدم الدالة : unset( ) . مثال:

```
unset( $name );
```

أنواع البيانات في PHP

تدعم PHP العديد من أنواع البيانات data types. و يمكن تصنيف أنواع البيانات في PHP إلى ثلاث فئات:

- أنواع بيانات بقيمة واحدة scalar و تنقسم إلى أربعة أقسام:

int	رقم صحيح بدون فاصلة عشرية كـ ٤٥٦٠
float	رقم صحيح بفاصلة عشرية كـ ٤٥,٣٢٥
string	بيانات ثنائية كالنصوص أو الصور .... الخ
boolean	قيمة تحتمل أن تكون TRUE أو FALSE

- أنواع بيانات تدعم أكثر من قيمة composite

array	سلسلة من البيانات المرتبة في عناصر
object	يمكن أن تحتوي على بيانات أو شفرة برمجية

- أنواع بيانات خاصة

NULL	تدل على أن المتغير لا يحتوي على قيمة (قيمة خالية)
resources	تدل على أن المتغير يحتوي على مصدر بيانات خارجي

```
<?php

$valid_student = TRUE;

$student_id = 181280000;

$student_name = "Ali Mohammad";

$student_gpa = 3.75;

$student_phone=null;

$student_courses = ("cs101","cs102","cs103");

?>
```

فرض نوع بيانات type casting

كما سبق وأن اشرنا بأن PHP من اللغات التي لا تفرض تحديد نوع بيانات المتغير عند تعريفه كما هو الحال في لغات البرمجة كـ C و java. لكن مفسر اللغة PHP interpreter يقوم بتحديد نوع البيانات تلقائيا استنادا على القيمة المخزنة في المتغير. لكن قد نحتاج في بعض الحالات لفرض نوع بيانات محدد.

```
<?php
$value1 = 10; // عدد صحيح
$value2 = 20.38; // عدد صحيح عشري
$value3 = $value1 + $value2;
echo $value3;
?>
```

```
30.38
```

و كما هو ملاحظ فإن ناتج المتغير $value3 عبارة عن رقم صحيح عشري float. و في حال أردنا أن يكون الناتج عدد صحيح فإننا نقوم بفرض نوع البيانات int كما يلي:

```
<?php
$value1 = 10; // عدد صحيح
$value2 = 20.38; // عدد صحيح عشري
$value3 = $value1 + $value2;
echo $value3.'<br/>';// before type casting
echo (int)$value3; // type casting
?>
```

```
30.38
30
```

التعامل مع الثوابت constants

على عكس المتغير و الذي من الممكن أن تتغير قيمته في أثناء سير البرنامج، فإن الثابت يحتوي على قيمة ثابتة لا تتغير. و لتعريف الثابت في PHP يجب الأخذ في الاعتبار مايلي:

- إتباع القواعد الخاصة بتعريف المتغير باستثناء علامة $ و التي ليست مطلوبة مع الثوابت.
- استخدام الدالة define( ).
- ينصح عند تسمية الثوابت بأن تكتب بأحرف كبيرة capital letters .

```
<?php

define("COURSE","PHP Programming"); // تعريف الثابت
define("YEAR",1430);

echo COURSE; // طباعة الثابت
echo '</br>'.YEAR;
?>
```

في الشفرة السابقة :

- COURSE هو اسم الثابت و PHP Programming هي القيمة.
- YEAR هو اسم الثابت و 1430 هي القيمة (ولأنها قيمة رقمية لم توضع بين علامات اقتباس).

ربط النصوص string concatenation

لربط النصوص في PHP نقوم باستخدام اداة الربط (.) .

```
<?php
$name = "Ali Abdullah";
$course = " Web Programming";

echo '<h1>Welcome '.$name.' ,you are enrolled in '.$course
.' course<h1>';
?>
```

Welcome Ali Abdullah ,you are enrolled in Web Programming course

المعاملات الحسابية arithmetic operators

المعاملات الحاسبية تساعدنا على إجراء العمليات الحسابية الأساسية.

$x = 20 + 14 ;	الجمع addition
$x = 30 – 10 ;	الطرح subtraction
$x = 12 * 10 ;	الضرب multiplication
$x = 30 / 10 ;	القسمة division
$x = 10 % 3 ;	باقي القسمة modulus

```
<?php
// كيفية التعامل مع المعاملات الحسابية
$num1 = 10;
$num2 = 3;
echo $num1 * $num2;
echo '</br>'.$num1 / $num2;
echo '</br>'.$num1 % $num2;
?>
```

```
30
3.3333333333333
1
```

أسبقية المعاملات operators precedence

عند التعامل مع المعاملات الحسابية في PHP، فإننا نتبع نفس القواعد المتبعة مع العمليات الحسابية في علم الحساب. لذلك عند تنفيذ أي عملية حسابية ، يجب الأخذ في الاعتبار أسبقية المعاملات من حيث التنفيذ. الجدول التالي يوضح ترتيب المعاملات وفقا لأسبقيتها في التنفيذ.

( )	أعلى ↑ أقل
* / %	
+ -	

معاملات الزيادة و النقصان increment and decrement operators

معاملات الزيادة (++) و معاملات النقصان(--) من الممكن أن تأتي قبل أو بعد المتغير. في حال أتيا قبل المتغير فستتم عملية الزيادة أو النقصان بمقدار (١) قبل إجراء أي عملية على المتغير. و على العكس تماما إذا أتى أياً من المعاملين بعد المتغير سيتم إجراء أي عملية على المتغير و من ثم ستتم عملية الزيادة أو النقصان.

```
<?php
$num1 = 10;
$num2 = 3;

echo $num1++; // $num1 = 10
echo '</br>'.++$num1; // $num1 = 12
echo '</br>'.++$num2; // $num2 = 4
?>
```

معاملات المقارنة comparison operators

تتيح PHP لنا مقارنة المتغيرات أو القيم عن طريق استخدام معاملات المقارنة. الجدول التالي يوضح معاملات المقارنة في PHP:

==	المساواة (Equality)
!=	عدم المساواة (Inequality)
>	أكبر من (Greater than)
<	أقل من (Less than)
>=	أكبر من أو يساوي (Greater than or equal)
<=	أقل من أو يساوي (Less than or equal)
===	التماثل (Identical)
!==	غير متماثل (Not identical)

من الجدول السابق يجب الإشارة إلى أن المقصود بالتماثل Identical، هو أن تكون المتغيرات متساوية في القيمة و من نفس نوع البيانات data type.

علامة (=) المفردة لا تستخدم في عملية المقارنة ، تستخدم فقط عند إسناد القيم للمتغيرات.

المعاملات المنطقية logical operators

تُستخدم المعاملات المنطقية عادة مع الشروط المركبة و تكون النتيجة إما TRUE أو FALSE .

المعامل	الوصف	مثال
and	(و)هذا المعامل يقوم بإرجاع القيمة TRUE فقط في حال كان ناتج كل الشرطين TRUE. بخلاف ذلك يقوم بإرجاع القيمة FALSE.	x = 6 y = 4 (x>5 and y<3) ➔ FALSE (x>4 and y<6) ➔ TRUE
or	(أو)هذا المعامل يقوم بإرجاع القيمة FALSE فقط في حال كان ناتج كل الشرطين FALSE. بخلاف ذلك يقوم بإرجاع القيمة TRUE.	x = 6 y = 4 (x>5 or y<3) ➔ TRUE (x>7 or y<2) ➔ FALSE
&&	(و)هذا المعامل يقوم بإرجاع القيمة TRUE فقط في حال كان ناتج كل الشرطين TRUE. بخلاف ذلك يقوم بإرجاع القيمة FALSE.	x = 6 y = 4 (x>5 && y<3) ➔ FALSE (x>4 && y<6) ➔ TRUE
||	(أو)هذا المعامل يقوم بإرجاع القيمة FALSE فقط في حال كان ناتج كل الشرطين FALSE. بخلاف ذلك يقوم بإرجاع القيمة TRUE.	x = 6 y = 4 (x>5 || y<3) ➔ TRUE (x>7 || y<2) ➔ FALSE
!	(نفي)هذا المعامل يقوم بعكس قيمة الشرط.	x = 6 y = 4 !(x>y) ➔ FALSE

معامل منع عرض الخطأ error supersession operator

يُستخدم المعامل @ في PHP لإجبار مفسر اللغة interpreter بمنع عرض رسائل الخطأ error messages أو التحذير warning messages للمستخدم.

```
<?php
/*
في هذه الحالة لن يقوم مفسر اللغة بعرض رسالة التحذير الناتجة من
محاولة القسم على الصفر و السبب لأننا سبقنا المتغير x بالمعامل @
*/
@$x = 10/0;
echo $x;
?>
```

تطبيق عملي

في هذا التطبيق سنقوم بتعريف اربع متغيرات a و b و c و d . و سنقوم بإسناد القيم التالية للمتغيرات على التوالي:

- PHP
- 40
- 25.32
- TRUE

سنستعين في هذا التطبيق بالدالة gettype( ) و التي تقوم بإرجاع نوع البيانات و من ثم طباعة الناتج. قم بفتح برنامج Notepad++ ثم قم بكتابة الشفرة التالية:

```
<?php
//هنا قمنا بتعريف المتغيرات المطلوبة
$a = 'PHP';
$b = 40;
$c = 25.32;
$d = TRUE;
/*
قمنا باستدعاء الدالة gettype()
لكي نقوم بعرض نوع البيانات لكل متغير
*/
echo'<table border="1" dir="rtl" align="center">';
echo'<tr><td>المتغير $a نوع بياناته:'.gettype($a).'</td></tr>';
echo'<tr><td>المتغير $b نوع بياناته:'.gettype($b).'</td></tr>';
echo'<tr><td>المتغير $c نوع بياناته:'.gettype($c).'</td></tr>';
echo'<tr><td>المتغير $d نوع بياناته:'.gettype($d).'</td></tr>';
echo'</table>';
?>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم variables.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/variables.php و من ثم لاحظ الناتج.

المتغير a نوع بياناته:string
المتغير b نوع بياناته:integer
المتغير c نوع بياناته:double
المتغير d نوع بياناته:boolean

أسئلة نهاية الفصل

١. قم بكتابة برنامج يقوم بتحويل العملة من دولار امريكي الى ريال سعودي. علما بأن سعر صرف الدولار= ٣,٧٥ ريال .

٢. متى نستخدم المتغيرات و متى نستخدم الثوابت.

٣. ما الفرق بين كلا من: (= = =) ، (= =) و (=).

٤. كيف نستطيع منع عرض رسائل الخطأ للمستخدم ؟

هذا الفصل يغطي:

- **كيف التعامل مع جمل الشرط**
- **أنواع جمل التكرار المختلفة**
- **كيفية التحكم في التكرار**

جملة الشرط if

تعتبر جملة الشرط if من ابسط جمل الشرط في PHP . و تستخدم لإختبار شرط ما و من ثم تنفيذ جواب الشرط إذا كان ناتج الشرط TRUE. وصيغتها العامة كالتالي:

```
if ( condition is TRUE )
{
      // الكود الذي سينفذ في حال تحقق الشرط
}
```

جملة الشرط if else

في هذه الحالة سيتم تنفيذ الجزء التابع لجملة if إذا كان ناتج الشرط TRUE. بخلاف ذلك سيتم تنفيذ الجزء البرمجي التابع لـ else:

```
if ( condition is TRUE )
{
      // الكود الذي سينفذ في حال تحقق الشرط
}
else
{
      // الكود الذي سينفذ في حال لم يتحقق الشرط
}
```

جمل الشرط المتداخلة if elseif else

في هذه الحالة سيتم اختبار أكثر من شرط و لن يتم تنفيذ إلا الجزء البرمجي الذي تكون فيه نتيجة الشرط TRUE بخلاف ذلك سيتم تنفيذ الجزء البرمجي التابع لجملة else :

```
if ( condition is TRUE )
{
      // الكود الذي سينفذ في حال تحقق الشرط
}
elseif( condition 2 is TRUE)
{
      // الكود الذي سينفذ في حال تحقق الشرط الثاني
}
.
.
.
else
{
      // الكود الذي سينفذ في حال لم يتحقق أي شرط
}
```

يمكن كتابة elseif ككلمة واحدة أو كلمتين منفصلتين: else if

```
<?php
$student_mark = 73;
if($student_mark <60)
{
      $result = "هـ";
}
elseif($student_mark<70)
{
      $result = "د";
}
elseif($student_mark<80)
{
      $result = "ج";
}
elseif($student_mark<90)
{
      $result = "ب";
}
else
{
      $result = "أ";
}
echo'<div style="direction:rtl;">';
echo 'تقدير في المادة هو: '.$result;
echo'</div>';
?>
```

تقدير في المادة هو: ج

جملة القرار switch

تستخدم جملة القرار switch كبديل لجملة الشرط if else . في حالة جملة switch نقوم باختبار قيمة متغير ما مقابل مجموعة من القيم . الصيغة العامة لجملة القرار switch كالتالي:

```
switch(variable)
{
      case  value1:
//الكود الذي سينفذ في حال كانت قيمة المتغير تساوي هذه القيمة
                  break;

      case  value2:
//الكود الذي سينفذ في حال كانت قيمة المتغير تساوي هذه القيمة
                  break;

      ……
      ……

      default:
// الكود الذي سينفذ في حال لم تساوي قيمة المتغير ايا من القيم
السابقة
}
```

المتغير الذي سيتم اختبار قيمتة سيوضع بين القوسين التابعين لـ switch .أما العبارة case فتمثل القيمة أو القيم التي سيتم اختبار قيمة المتغير مقابلها. بعد ادخال المتغير المراد اختباره سيقوم مفسر PHP باختبار قيمة ذلك المتغير مقابل كل جملة case . في حال كانت نتيجة اختبار ايا من جمل case تساوي TRUE سيقوم مفسر PHP بتنفيذ الشفرة التابعة لها حتى يصل عند جملة break، بعد ذلك سيقوم بالخروج من جملة switch.

لكن ما الذي سيحدث في حال نسي المبرمج كتابة عبارة break لأي من عبارة case ؟

في هذه الحالة، سيقوم مفسر PHP بتنفيذ الشفرة البرمجية و التوقف عند أقرب عبارة break حتى و ان كانت تتبع لعبارة case اخرى.
اخيرا، اذا لم يجد مفسر PHP ايا من جمل case قيمتها تساوي قيمة المتغير المراد اختباره، سيقوم المفسر بتنفيذ الشفرة التابعة لعبارة default.

```
<?php
/* date() دالة التاريخ
تقوم بتحويل الختم الزمني timestamp
لصيغة تاريخ قابلة للقراءة
*/
$d=date("D");
echo'<div style="direction:rtl;">';
switch ($d)
{
case "Mon":
  echo "اليوم هو الاثنين";
  break;
case "Tue":
  echo "اليوم هو الثلاثاء";
  break;
case "Wed":
  echo "اليوم هو الاربعاء";
  break;
```

```
case "Thu":
  echo "اليوم هو الخميس";
  break;
case "Fri":
  echo "اليوم هو الجمعة";
  break;
case "Sat":
  echo "اليوم هو السبت";
  break;
case "Sun":
  echo "اليوم هو الأحد";
  break;
default:
  echo "اليوم غير معروف؟";
}
echo'</div>';
?>
```

القيم النصية لعبارة case يجب أن تحصر بين علامتي تنصيص، على عكس القيم الرقمية التي لا تتطلب حصرها بين علامات التنصيص.

المعامل الشرطي ternary operator ?:

المعامل الشرطي ?: يستخدم كطريقة مختصرة للتحقق من الشروط البسيطة. الصيغة العامة للمعامل الشرطي كالتالي:

```
condition? value if ture : value if FALSE ;
```

```
<?php
$age=20;
$result = $age>18? 'بالغ' : 'طفل' ;
echo $result;
?>
```

جملة التكرار while

تستخدم جملة التكرار while لتكرار تنفيذ شفرة برمجية طالما كانت نتيجة الشرط التابع لها تساوي TRUE.
و الصيغة العامة لجملة while هي كالتالي:

```
while ( condition )
{
  // الجزء البرمجي الذي سيتم تنفيذه في حال كانت نتيجة الشرط صحيحة
}
```

```
<?php
$i = 0;
$num = 50;
while( $i < 10 )
{
   $num--;
   $i++;
}
echo'<div align="right">';
echo'جملة التكرار توقفت عند'.'<br/>';
echo'$i='.$i.'<br/>';
echo'$num='.$num;
echo'</div>';
?>
```

جملة التكرار توقفت عند
$i=10
$num=40

جملة التكرار do-while

كما هو ملاحظ مع جملة التكرار while، يتم اختبار الشرط في بداية التكرار. على العكس في جملة التكرار do-while ، فإن الشرط يتم اختباره في نهاية التكرار. هذا يعني أن التكرار سيتم تنفيذه مرة واحدة على الأقل. المثال التالي نسخة معدلة من المثال السابق:

```
<?php
$i = 0;
$num = 50;
/*
في جملة التكرار هذه يتم اختبار الشرط في نهاية الجزء الخاص بها
لذلك سيتم تنفيذ الجزء البرمجي التابع لجملة التكرار مرة واحدة
على الأقل
*/
do
{
   $num--;
   $i++;
}while( $i < 10 );
echo'<div align="right">';
echo'جملة التكرار توقفت عند'.'<br/>';
echo'$i='.$i.'<br/>';
echo'$num='.$num;
echo'</div>';
?>
```

جملة التكرار for

جملة التكرار تستخدم لتكرار جزء من الشفرة البرمجية لعدد محدد يتم تحديده في جملة التكرار for (على عكس جملة التكرار while). و صيغتها العامة كالتالي:

```
for( initial value ; condition ; increment or decrement)
{
   // الجزء البرمجي الذي سيتم تنفيذه في حال كان قيمة الشرط صحيحة
}
```

و كما هو ملاحظ من الصيغة العامة لجملة التكرار for فهي تتكون من ثلاث اجزاء:

- Initial value القيمة الابتدائية للعداد.
- Condition الشرط.
- Increment or decrement الزيادة أو النقصان.

المثال التالي يقوم بطباعة مجموع الأعداد الزوجية من ٠ الى ١٠٠:

```
<?php
 $count=0;
for($i=0;$i<=100;$i+=2)
{
      $count = $count + $i;
}
echo $count;
?>
```

```
2550
```

جمل for المتداخلة

كما عملنا مع جملة الشرط if، فإنه من الممكن أن تتداخل أكثر من جملة for. المثال يوضح كيفية التعامل مع جمل for المتداخلة:

```
<?php
// يقوم بإنشاء جدول يتكون من
// 3 صفوف, 4 أعمدة
echo "<table border=\"1\">";
for ($row=1; $row<4; $row++) {
  echo "<tr>";
  for ($col=1; $col<5; $col++) {
    echo "<td>Row $row, Column $col</td>";
  }
  echo "</tr>";
}
echo "</table>";
?>
```

كما هو واضح من المثال السابق، لدينا جملتي for، الأول هي جملة for الخارجية outer loop و التي تقوم بطباعة الصفوف. و الأخرى هي جملة for الداخلية inner loop والمسئولة عن طباعة الخلايا داخل الصفوف.

ايقاف تنفيذ التكرار interrupting loop

في حال أردنا ايقاف تنفيذ التكرار لأي سبب كان فإننا نقوم بإدراج break داخل جملة شرط. عند تحقق ذلك الشرط ستقوم break بإنهاء جملة التكرار كاملة.

تجاهل تنفيذ التكرار skipping loop

على النقيض من عبارة break التي تقوم بالخروج من التكرار، تُستخدم عبارة continue في حال أردنا تجاهل التكرار الحالي و الانتقال الى قيمة التكرار التالية.

جملة التكرار foreach

هذا النوع من جمل التكرار يُستخدم فقط مع المصفوفات. و تأتي على شكلين . و في كلا الحالتين يتم استخدام متغير مؤقت temporary variable للتعامل مع عناصر المصفوفة.

- في حال أردنا التعامل فقط مع القيم لكل عنصر داخل المصفوفة elements values ، فتكون الصيغة العامة لجملة foreach كالتالي:

```
foreach( array_name    as   temporary_variable)
{
      شفرة برمجية تنفذ على المتغير المؤقت//
}
```

هذا النوع جملة foreach يستخدم مع المصفوفات الرقمية numeric arrays .

- الشكل الثاني من جملة التكرار foreach يُستخدم عادة في حال أردنا التعامل مع القيم values و المفاتيح keys لعناصر المصفوفة. هذا النوع يستخدم عادة مع المصفوفات الترابطية associative arrays. الصيغة العامة لهذا النوع من جملة foreach كالتالي:

```
foreach( array_name as key => value)
{
      // شفرة برمجية تنفذ على المفتاح و القيمة للعنصر
}
```

لاحقا، في الفصل الخامس سنتطرق الى العديد من الأمثلة لجملة التكرار foreach عندما نتعامل مع المصفوفات.

تطبيق عملي

في هذا التطبيق سنقوم بكتابة شفرة برمجية تقوم بطباعة جدول الضرب.

```
<html>
<head>
<style>
/*
شفرة css
و التي تهتم بتنسيق مظهر الجدول
*/
table{
	border:1px solid #333;
	text-align:center;
}
td{
	border:1px solid #CCC;
}
.higlight{
	background-color:#0C9;
}
</style>
</head>
<body>
<?php
echo'<table>';
echo'<tr>';
//هنا قمنا بإنشاء خلية فارغة للحفاظ على مظهر الجدول
echo'<td>*</td>';
/*
جملة التكرار التالية مسئولة عن طباعة الصف الأول من الجدول
*/
for($i=1;$i<=10;$i++)
{
	echo "<td class='higlight'>$i</td>";
}
echo'</tr>';

/*
حلقة التكرار الخارجية تقوم بطباعة العمود الأول في الجدول و الذي
يبدأ عن الرقم ١
حلقة التكرار الداخلية تقوم بمهمة عملية الضرب
j هنا نقوم بضرب المتغير
و الذي قيمة العمود
t بالمتغير
والذي سيكون عبارة قيمة متزايدة تبدأ ب ١ وتنهي عند ١٠
*/
for($j=1;$j<=10;$j++)
{
	echo "<tr><td class='higlight'>$j</td>";
	for($t=1;$t<=10;$t++)
```

```
            echo '<td>'.($t*$j).'</td>';
      echo'</tr>';
}
echo'</table>';
?>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم multitable.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/ multitable.php
و من ثم لاحظ الناتج.

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

أسئلة نهاية الفصل

١. أكتب برنامج يقوم بحساب مجموع الاعداد الفردية من ١ الى ١٠٠ .

٢. ما الفرق بي كلا من: while و do-while ؟

٣. ما هي أنواع جملة التكرار foreach و متى يُستخدم كل نوع؟

٤. ما الفرق بين break و continue ؟

٥. أكتب برنامج يقوم بطباعة القيم من ١ الى ١٠ مع تجاهل طباعة القيمة ٥ .

هذا الفصل يغطي:

- **مفهوم المصفوفات**
- **المصفوفة الرقمية و كيفية التعامل معها**
- **المصفوفة الترابطية و كيفية التعامل معها**
- **المصفوفات ذات البعدين و المصفوفات الجاهزة**

ما هي المصفوفة

المصفوفة array عبارة عن متغير لديه الامكانية لتخزين أكثر من قيمة حتى لو اختلفت أنواعها. كل مصفوفة تتكون من مجموعة من العناصر. و كل عنصر يتكون من جزئين قيمة value و مفتاح key . و تنقسم المصفوفات في PHP إلى ثلاثة أنواع و هي :

- المصفوفة الرقمية numeric array.
- المصفوفة الترابطية associative array.
- المصفوفات ذات البعدين multidimensional array.

المصفوفة الرقمية numeric array

في هذا النوع من المصفوفات يتم استخدام الأرقام لفهرسة عناصر المصفوفة لتسهيل عملية التعامل معها لاحقا.

key	value
0	أحمر
1	أخضر
2	أزرق
3	أبيض
4	أسود

كما هو واضح من الشكل السابق ، فإن اسم المصفوفة هو colors و تحتوي على خمس عناصر. يتم بدء فهرسة العناصر من الرقم صفر الى ن-١ (n-1 .... 0) ، حيث تمثل ن مجموع عناصر المصفوفة. لاسترجاع قيمة عنصر ما في المصفوفة، نقوم بكتابة اسم المصفوفة و من ثم قوسين مربعين ، و بينهما رقم المفتاح لذلك العنصر كالتالي:

```
$colors[2]
```

طريقة انشاء المصفوفة الرقمية

يتم انشاء المصفوفة الرقمية بإحدى الطرق التاليه:

الطريقة الأولى

```
$colors[0]= 'أحمر';
$colors[1]= 'أخضر';
$colors[2]= 'أزرق';
$colors[3]= 'أبيض';
$colors[4]= 'أسود';
```

الطريقة الثانية

```
$colors = array ('أحمر','أخضر','أزرق','أبيض','أسود');
```

في هذه الطريقة ، يلاحظ أننا لم نقم بتحديد المفتاح لكل عنصر في المصفوفة. بدلا من ذلك استخدمنا الدالة array( ) لإنشاء المصفوفة . و لذلك فإن عملية الفهرسة تتم بشكل تلقائي بدءا من أول عنصر (المفتاح 0).

طريقة استرجاع قيم المصفوفة

يمكن استرجاع قيم المصفوفة بأكثر من طريقة منها ما يلي:

- باستخدام جمل التكرار و أفضلها جملة التكرار for .

```
<?php

$colors = array ('أحمر'،'أخضر'،'أزرق'،'أبيض'،'أسود');

for($i=0;$i<sizeof($colors);$i++)
{
      echo "<h3>".$colors[$i]."</h3>";
}

?>
```

في هذا المثال استخدمنا جملة التكرار for لطباعة قيم المصفوفة colors . و كما هو واضح فإننا استخدمنا الدالة sizeof( ) لتحديد حجم المصفوفة لكي نتمكن من عرض جميع قيمها. أخيرا بدأنا التكرار عند القيمة صفر لأن المفتاح لأول عنصر في المصفوفة الرقمية هو 0 .

- باستخدام جملة التكرار foreach كما في المثال التالي:

```
<?php

$colors = array ('أحمر','أخضر','أزرق','أبيض','أسود');

foreach($colors as $value)
{
      echo "<h3>".$value."</h3>";
}

?>
```

في هذا الطريقة استخدمنا النوع الأول من جملة التكرار foreach (تم شرحه في الفصل السابق) لاسترجاع قيم المصفوفة الرقمية . في هذا المثال ستقوم جملة foreach بتخزين قيمة أول عنصر داخل المتغير المؤقت $value و من ثم نقوم بطباعة قيمة ذلك المتغير. ستقوم foreach بعد ذلك بالعودة و تكرار نفس الخطوات السابقة مع العنصر الثاني و هكذا حتى الانتهاء من جميع عناصر المصفوفة الرقمية تلقائيا.

تغيير قيمة عنصر

عملية تغيير قيمة عنصر ما تتم بسهولة. كل ما علينا عمله هو كتابة اسم المصفوفة متبوع بالمفتاح للعنصر المحدد بين قوسين مربعة و من ثم نقوم بإسناد القيمة الجديدة له.

```
<?php

$colors = array ('أحمر','أخضر','أزرق','أبيض','أسود');
// هنا قمنا بتغيير قيمة العنصر الثاني
$colors[1] = '#FFFFFF';
foreach($colors as $value)
{
      echo "<h3>".$value."</h3>";
}

?>
```

استرجاع حجم المصفوفة

لاسترجاع حجم مصفوفة ما ، بالإمكان استخدام الدالة ( )count أو الدالة ( )sizeof كالتالي:

```
count( array_name );

sizeof( array_name);
```

المصفوفة الترابطية associative array

المصفوفة الترابطية مشابهه للمصفوفة الرقمية من ناحية الوظيفة لكنها تختلف عنها من ناحية عملية الفهرسة. في هذا النوع من المصفوفات، يتم استخدام النصوص string في فهرسة عناصر المصفوفة بدلا من الارقام.

key	value
title	البرمجة باستخدام php
author	محمد عبدالله
publisher	دار النشر
isbn	١٢-٢٥٠-١٦٣

طريقة انشاء المصفوفة الترابطية

يتم انشاء المصفوفة الترابطية بإحدى الطرق التاليه:

الطريقة الأولى

```
$books['title']= ' البرمجة باستخدام php';
$books['author']= 'محمد عبدالله';
$books['publisher']= 'دار النشر';
$books['isbn']= '12-250-163';
```

الطريقة الثانية

```
$books = array('title' => ' البرمجة باستخدام php',
               'author' => 'محمد عبدالله' ,
               'publisher' => 'دار النشر',
               'isbn' => '12-250-163');
```

و كما هو ملاحظ في الطريقة الثانية استخدمنا الدالة array( ). كذلك فإننا استخدمنا الرمز => للفصل بين القيمة value و المفتاح key.

طريقة استرجاع قيم المصفوفة

أسهل طريقة لاسترجاع قيم المصفوفة الترابطية هو عن طريق استخدام جملة التكرار foreach .

```
<?php

$books = array('title' => 'البرمجة باستخدام php',
               'author' => 'محمد عبدالله' ,
               'publisher' => 'دار النشر',
'isbn' => '12-250-163');

foreach($books as $key=>$value)
{
      echo "<h3>$key => $value</h3>";
}

?>
```

كما هو واضح من المثال السابق فإننا قمنا بتعريف المصفوفة books و قمنا بتعبئة قيمها كما هو واضح كذلك. بعد ذلك قمنا باستخدام جملة التكرار foreach لطباعة قيم المصفوفة. في هذه الحالة قمنا باستخدام متغيرين مؤقتين temporary variables لتخزين القيم values و المفاتيح keys لعناصر المصفوفة الترابطية . في كل مرة سنقوم بتخزين قيم و مفاتيح العنصر الأول في المصفوفة في المتغيرين المؤقتين ثم الذي يليه و هكذا حتى الانتهاء من كل عناصر المصفوفة.

تغيير قيمة عنصر

كما سبق أن عملنا مع المصفوفة الرقمية من تغيير قيمة عنصر ، نستطيع عمل نفس الطريقة مع المصفوفة الترابطية عن طريق تحديد المفتاح للعنصر المطلوب و من ثم إسناد القيمة الجديدة له.

```
<?php

$books = array('title' => 'البرمجة باستخدام php',
               'author' => 'محمد عبدالله' ,
               'publisher' => 'دار النشر',
'isbn' => '12-250-163');

//هنا قمنا بتغيير قيمة العنصر الأول
$books['title'] = 'java programming';
echo $books['title'];

?>
```

في حال أردنا مسح عناصر المصفوفة السابقة books فإننا نكتب

```
$books = array( );
```

المصفوفات ذات البعدين multidimensional array

كما ذكرنا سابقا بأن المصفوفة تُستخدم لتخزين أي نوع من البيانات. و كما ذكرنا أيضا بأن المصفوفة تعتبر من أنواع البيانات في PHP . لذلك نستطيع القول بأنه من الممكن تخزين مصفوفة كأحد عناصر مصفوفة أخرى. في هذه الحالة تسمى المصفوفة التي تحتوي على مصفوفة كأحد عناصرها بالمصفوفة ذات البعدين.

```
$books = array(
                array('title' => 'البرمجة باستخدام php',
                'author' => 'محمد عبدالله' ,
                'publisher' => 'دار النشر',
                'isbn' => '12-250-163'),

                array('title' => 'أساسيات قواعد البيانات',
                'author' => 'خالد العلي' ,
                'publisher' => 'دار الفاروق',
                'isbn' => '22-433-987'),

);
```

في هذه الحالة إذا اردنا الاشارة الى عنصر محدد داخل المصفوفة ذات البعدين فنتبع الطريقة التالية:

```
$books[1][ 'isbn']; // value is 12-250-163
```

المصفوفات العامة الجاهزة PHP's built-in super global arrays

تحتوي PHP على عدد من المصفوفات المبنية مسبقا و التي تستخدم بشكل متكرر في PHP و تسهل عملية الوصول للبيانات حتى لو كانت من صفحات مختلفة. جميع هذه المصفوفات تبدأ بعلامة الدولار $ متبوعة بشرطة سفلية _ . من أشهر هذه المصفوفات ، $_GET و $_POST و اللتان تحتويان على بيانات هامة يتم تمريرها عن طريق نماذج HTML (كما سنرى لاحقا). كل المصفوفات العامة تُعتبر مصفوفات ترابطية. المفاتيح لكلا من $_POST و $_GET يتم اشتقاقهما من اسماء العناصر في نموذج HTML.
فلو افترضنا أنه لدينا النموج التالي:

Address:

Send

لنفترض أن أسم مربع النص هذا هو address و الطريقة المستخدمة في ارسال النموذج هي POST. عند الضغط على مربع الارسال submit، يقوم مفسر PHP تلقائيا بإنشاء العنصر $_POST['address'] و الذي ستكون قيمته هي القيمة المدخلة في مربع النص من قبل المستخدم.

من أشهر المصفوفات العامة في PHP ما يلي:

$_POST	تحتوي على بيانات مرسلة عن طريق نموذج HTML باستخدام طريقة post .
$_GET	تحتوي على بيانات مرسلة عن طريق نموذج HTML باستخدام طريقة get .
$_SERVER	تحتوي على معلومات مخزنة بواسطة خادم الويب Web Server كـ: اسم الملف و مساره.
$_FILE	يحتوي معلومات خاصة بالملفات المحملة.
$_SESSION	يحتوي على معلومات يتم حفظها ، لكي تستخدم في أكثر من صفحة.

يجب الأخذ في الاعتبار أن اسماء المصفوفات العامة في PHP تكتب بأحرف كبيرة ، بالإضافة لذلك فهي حساسة لحالة الحرف case sensitive.

استخدام الدالة print_r( ) مع المصفوفات

في حال اردنا معرفة المحتوى لمصفوفة ما، نقوم باستدعاء الدالة print_r( ) و التي تقوم بعرض محتوى المصفوفة المُمررة لها كالتالي:

```
print_r( $book );
```

تطبيق عملي

في هذا التطبيق سنقوم بكتابة شفرة برمجية تقوم بتخزين درجات الطلاب في مصفوفة رقمية ، و من ثم نقوم بطباعة إحصائية عن درجات الطلاب تشمل الطلبة الناجحين و الراسبين و بعض الاحصائيات الأخرى.
قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<html>
<head>
<style>
/*
شفرة css
و التي تهتم بتنسيق مظهر الجدول
*/
body{
	direction:rtl;
}
fieldset{
	width:300px;
}

legend{
	font-weight:bold;
}

p{
	background-color:#CCC;
}
</style>
</head>
<body>
<?php
//إنشاء مصفوفة تحتوي على درجات الطلاب
$grades = array(
25, 64, 23, 87, 56, 38, 78, 57, 98, 95,
81, 67, 75, 76, 74, 82, 36, 39,
54, 43, 49, 65, 69, 69, 78, 17, 91
);
// حساب عدد الدرجات
$count = count($grades);
// إجمالي مجموع الدرجات و القيمة الأولية هي صفر
$total = 0;
//عدد الطلبة الراسبين و القيمة الأولية هي صفر
$fail = 0;
//عدد الطلبة الناجحين و القيمة الأولية هي صفر
$success = 0;

foreach($grades as $grade)
{
	$total += $grade;
```

```
if ($grade < 60)
      {
            $fail++;
      }
      if ($grade >= 60)
      {
            $success++;
      }
}
/* متوسط درجات الطلاب
 تستخدم round للتقريب لأقرب رقم صحيح
 */
$avg = round($total / $count);
// طباعة الإحصائية
echo'<fieldset><legend>إحصائية الدرجات في المادة</legend>';
echo"<p>عدد الطلبة المسجلين في المادة هو: $count</p>";
echo"<p>متوسط الدرجات في المادة هو: $avg</p>";
echo"<p>عدد الطلبة الناجحين في المادة: $success</p>";
echo"<p>عدد الطلبة الراسبين في المادة هو: $fail</p>";
echo'</fieldset>';
?>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم grades.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح (*.*)All types أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/grades.php و من ثم لاحظ الناتج.

إحصائية الدرجات في المادة

عدد الطلبة المسجلين في المادة هو: 27

متوسط الدرجات في المادة هو: 62

عدد الطلبة الناجحين في المادة: 16

عدد الطلبة الراسبين في المادة هو: 11

أسئلة نهاية الفصل

١. ما الفرق بين المصفوفة المفهرسة و المصفوفة الرقمية؟

٢. باستخدام جملة التكرار do-while، اكتب برنامج يقوم بطباعة قيم المصفوفة التالية:

```
$fruit = array('Apple','Orange','Pineapple' ,'Strawberry');
```

٣. كيف نستطيع معرفة حجم المصفوفة السابقة؟

٤. قم بتغيير قيمة العنصر الثالث ليصبح : Banana .

هذا الفصل يغطي:

- **مفهوم الدوال**
- **أنواع الدوال في PHP**
- **مجال رؤية المتغيرات**
- **تضمين الملفات الخارجية**

ما هي الدالة function definition

الدالة function هي عبارة عن شفرة برمجية تقوم بتنفيذ مهمة محددة. و هي مهمة لأنها تساعد على فصل جزء من البرنامج بغرض إعادة استخدامه أكثر من مرة. و تنقسم الدوال في PHP الى قسمين:

- الدوال المبنية مسبقا Built-in functions .
- الدوال المعرفة من قبل المستخدم User defined functions.

الدوال المبنية مسبقا Built-in functions

تحتوي PHP على أكثر من ١٠٠٠ دالة مبنية مسبقا و جاهزة للاستخدام ، كما يتم إضافة العديد من الدوال بشكل مستمر . لذلك تعتبر PHP من أكثر لغات برمجة الويب تكيفا مع التوسع الكبير في التطبيقات الخاصة بشبكة الانترنت. و للمزيد عن دوال PHP الجاهزة يمكن زيارة الموقع: http://php.net/quickref.php و اختيار أي دالة للإطلاع على كيفية التعامل معها .

الدوال المعرفة من قبل المستخدم User defined functions

قبل البدء في إنشاء أي دالة يجب الأخذ في الاعتبار محتويات الدالة. تحتوي كل دالة على ثلاث أقسام:

- معطيات الدالة parameters .
- جسم الدالة body.
- القيمة المرجعة returning value.

عند تعريف أي دالة يجب البدء بالكلمة المحجوزة function متبوعة باسم الدالة و من ثم قوسين صغيرين متبوعة بجسم الدالة كالتالي:

```
function    function_name( )
{
      // الشفرة البرمجية داخل جسم الدالة
}
```

في حال كانت الدالة تحتوي على معطيات فيصبح شكل الدالة كما يلي:

```
function    function_name( arg1, arg2,… )
{
      // الشفرة البرمجية داخل جسم الدالة
}
```

بعد تعريف الدالة يجب أن نقوم باستدعاء الدالة. و عملية الاستدعاء تتم بكتابة اسم الدالة متبوع بقوسين. في حال كانت الدالة مُعرفة بمعطيات فيجب كتابة المعطيات عند استدعاء الدالة. في حال لم يتم استدعاء الدالة فلن يتم تنفيذها.

```
<html>
<head>
<title>كيفية انشاء دالة بدون معطيات</title>
</head>
<body>

<?php
// تعريف الدالة

function writeMessage()
{
  echo'إنشاء دالة بدون معطيات';
}
// إستدعاء الدالة
writeMessage();
?>
</body>
</html>
```

كما هو واضح ، قمنا بتعريف دالة اسمها writeMessage() بدون معطيات . عند استدعاء الدالة ستقوم بتنفيذ جملة الطباعة echo و طباعة النص المضمن بداخلها. المثال التالي يوضح كيفية تعريف و استدعاء دالة بمعطيات.

```
<html>
<head>
<title>كيفية انشاء دالة بمعطيات</title>
</head>
<body>

<?php
// تعريف دالة بمعطيات

function add($number1,$number2)
{
  $result = $number1 + $number2;
  echo'ناتج عملية الجمع هو:'.$result;
}
// إستدعاء الدالة
add(10,20);
?>
</body></html>
```

إرجاع قيمة دالة returning value

يمكن أن تقوم الدالة بإرجاع قيمة عن طريق العبارة return . عند الوصول للعبارة return فإن الدالة ستتوقف. و سيتم ارجاع القيمة للسطر من البرنامج الذي تم فيه استدعاء الدالة.

```
<html>
<head>
<title>كيفية انشاء دالة بمعطيات</title>
</head>
<body>

<?php
// تعريف دالة بمعطيات

function add($number1,$number2)
{
  $result = $number1 + $number2;
  // $result أرجاع قيمة المتغير
  return $result;
}
// استدعاء الدالة و القيمة المرجعة من الدالة سترجع للسطر التالي
$return_value = add(10,20);
echo'ناتج عملية الجمع هو:'. $return_value;
?>
</body>
</html>
```

إسناد قيم افتراضية للمعطيات

يمكن إسناد قيم افتراضية لمعطيات الدالة parameters يتم استخدمها في حال تم استدعاء الدالة بدون تمرير قيم لمعطياتها.

```
<html>
<head>
<title>كيفية انشاء دالة بمعطيات</title>
</head>
<body>

<?php
// تعريف دالة بمعطيات

function add($number1=30,$number2=40)
{
  $result = $number1 + $number2;
  // $result ارجاع قيمة المتغير
  return $result;
}
```

```
// استدعاء الدالة بدون تمرير معطيات
$call_without_values = add();
// استدعاء الدالة مع تمرير معطيات
$call_with_values = add(53,32);

echo'ناتج استدعاء الدالة بدون معطيات:'. $call_without_values;
echo'<br/>ناتج استدعاء الدالة بمعطيات:'. $call_with_values;
?>
</body>
</html>
```

```
ناتج استدعاء الدالة بدون معطيات:٧٠
ناتج استدعاء الدالة بمعطيات:٨٥
```

مجال رؤية المتغير variable scope

من المهم أن نعرف أن أي متغير يتم تعريفه داخل جسم الدالة فسوف يكون متاحا للدالة فقط. بمعنى أنه من غير الممكن الاشارة لذلك المتغير خارج إطار الدالة.

```
<html>
<head>
<title>مجال رؤية المتغير</title>
</head>
<body>

<?php
// تعريف دالة بمعطيات
function add($number1=30,$number2=40)
{
  $result = $number1 + $number2;
}

add(53,32);
echo'قيمة المتغير هي: '.$result;
?>
</body>
</html>
```

```
Notice: Undefined variable: result in C:\xampp\htdocs\test\test2.php on line 16
قيمة المتغير هي:
```

كما هو واضح من المثال السابق ، المتغير $result متغير محلي local خاص بالدالة add . عندما قمنا بمحاولة طباعة قيمة المتغير خارج الدالة ، عرض مفسر PHP رسالة خطأ تفيد بأن المتغير غير معرف.

و في حال أردنا أن يكون المتغير $result متاحا داخل و خارج الدالة أيضا ، فنقوم بكتابة العبارة global قبل اسم المتغير. في هذه الحالة سيصبح المتغير $result متغيرا عاما public. و سيتأثر بأي تغيير يحدث على المتغير خارج إطار الدالة.

```
<html>
<head>
<title>كيفية انشاء دالة بمعطيات</title>
</head>
<body>

<?php

$result = 0;
// تعريف دالة بمعطيات
function add($number1=30,$number2=40)
{
  // المتغير الآن اصبح متغيرا عاما
  global $result;
  $result = $number1 + $number2;
}

add(53,32);

echo'قيمة المتغير هي: '.$result;
?>
</body>
</html>
```

قيمة المتغير هي: ٨٥

تضمين شفرة برمجية من ملف آخر PHP file inclusion

تدعم PHP إمكانية تضمين محتوى ملف PHP في ملف PHP آخر قبل أن يقوم الخادم بإتمام عملية التنفيذ. هذه الطريقة تساعد في إعادة استخدام الشفرة برمجية reusability . و تكون شائعة في تضمين شفرات برمجية تُستخدم بشكل متكرر عند بناء موقع ديناميكي خصوصا ملفات header ، copy right footer .
و لتحقيق ذلك نستطيع استخدام الدالتين التاليتين:

- include( )
- require( )

كلا الدالتين تقوم بتضمين شفرة برمجية من ملف آخر. لكن الفرق بينهما هو في كيفية تعامل الدالتين مع الأخطاء.

في حال استخدمنا include( ) ، و كان الملف المضمن الذي يحتوي على الشفرة البرمجية المدرجة غير موجود ، ففي هذه الحالة سيقوم مفسر PHP بإنشاء رسالة تحذير warning message و من ثم يتابع تنفيذ بقية البرنامج.

على النقيض تماما في حال استخدمنا الدالة require( ) و كان الملف المضمن غير موجود ، فسيقوم مفسر PHP بإنشاء رسالة خطأ مميت fatal error و من ثم ايقاف تنفيذ البرنامج.

menu.php

```
<a href="home.php">الرئيسية</a> -
<a href="services.php">خدمات</a> -
<a href="about.php">حول الموقع</a> -
<a href="contactus.php">اتصل بنا</a> <br />
```

index.php

```
<html>
<body dir="rtl">
<?php include("menu.php"); ?>
<p>هذي طريقة تضمين الملفات في PHP</p>
</body>
</html>
```

الرئيسية - خدمات - حول الموقع - اتصل بنا

هذي طريقة تضمين الملفات في PHP

كما هو واضح من المثال السابق استخدمنا الدالة include( ) لتضمين محتوى الملف menu.php في الملف index.php. و بنفس الطريقة نستطيع استخدام الدالة require( ) كذلك.

تطبيق عملي

في هذا التطبيق سنقوم بتصميم واجهة لموقع تخيلي. و الهدف هو تطبيق ميزة تضمين الملفات في PHP .
عندما نقوم بتصميم الواجهة الرئيسية لموقع فيُنصح دائما بفصل الأجزاء المتكررة من ملفات مستقلة بهدف استخدامها في كامل صفحات الموقع. فمثلا شعار الموقع و القوائم الرئيسية (تسمى رأس الصفحة أو header) تكون ثابتة لذلك نقوم بوضعها في ملف مستقل. ايضا حقوق الموقع و التي تظهر عادة في أسفل الصفحة (تسمى ذيل الصفحة أو footer) تكون ثابتة لذلك نقوم بوضعها في ملف مستقل. و هكذا مع أي محتوى يتم استخدامه عبر كل ملفات الموقع فيُنصح أن يوضع في ملف مستقل. هذه الطريقة تساعد في إعادة استخدام الشفرة البرمجية reusability كما تسهل من عملية التطوير و الصيانة للموقع.
بعد أن نقوم بفصل المحتوى المتكرر نقوم بإنشاء الصفحة الرئيسية للموقع و من ثم تضمين المحتوى المطلوب باستخدام include( ) أو require( ).

قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<html>
<head>
<meta       http-equiv="Content-Type"       content="text/html;
charset=utf-8" />
<style>
body{
      direction:rtl;
      margin:0px 200px 0px 200px;
      font-family:"Times New Roman", Times, serif;
}
a{
      text-decoration:none;
      color:#000;
}
a:hover{
      text-decoration:underline;
      color:#FFF;
}
#logo{
      background-color:#6CC;
}
#titleLogo{
      text-align:right;
      font-size:42px;
      font-weight:bold;
      padding-right:50px;
}
#menu{
      padding-top:5px;
      text-align:center;
}
#menu td{
      background-color:#6CC;
      font-size:16px;
      font-weight:bold;
}
#content{
      margin-top:20px;
      margin-bottom:20px;
      font-size:16px;
      font-weight:bold;
}

#footer{
      margin-top:100px;
      padding:2px;
      background-color:#6CC;
      text-align:center;
      font-size:16px;
      font-weight:bold;
}
```

```
</style>
<title>عالم التصوير الرقمي</title>
</head>

<body>
<div id="header">
    <table width="100%" id="logo">
      <tr>
        <td width="86%" id="titleLogo">عالم التصوير الرقمي</td>
        <td   width="14%"><img   src="cammera.png"   width="128"
height="128" /></td>
      </tr>
    </table>
    <table width="100%" cellpadding="3" id="menu">
      <tr>
        <td><a href="home.php">الرئيسية</a></td>
        <td><a href="tutorials.php">دروس التصوير</a></td>
        <td><a href="camsworld.php">عالم الكاميرات</a></td>
        <td><a href="about.php">حول الموقع</a></td>
        <td><a href="contact.php">اتصل بنا</a></td>
      </tr>
    </table>
</div>
<!-- نهاية رأس الصفحة -->
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم header.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

الآن قم بالضغط على مفتاح Ctrl+N لإنشاء مساحة عمل فارغة ثم قم بكتابة الشفرة التالية.

```
<!-- بداية تذييل الصفحة -->
<div id="footer">
<p>جميع الحقوق محفوظة لموقع البوابة الرقمي&copy;</p>
<p>الرجاء قراءة تعليمات استخدام الموقع</p>
</div>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم footer.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

مرة أخرى ، قم بالضغط على مفتاح Ctrl+N لإنشاء مساحة عمل فارغة ثم قم بكتابة الشفرة التالية.

```
<?php
//قمنا بتضمين محتوى الملف هنا header.php
require('header.php');
?>
<div id="content">
مرحبا بكم في عالم التصوير الرقمي.
هذا الموقع يهدف إلى تقديم نبذة عن مختصرة عن فن التصوير الرقمي
و يسعدنا أن نقدم للزائر الكريم العديد من الدروس و النصائح حول
فن التصوير.
</div>
<?php
//قمنا بتضمين محتوى الملف هنا footer.php
require('footer.php');
?>
```

بعد الانتهاء من كتابة الشفرة السابقه ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم index.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/index.php و من ثم لاحظ الناتج.

أسئلة نهاية الفصل

١. ما هي أنواع الدوال في PHP مع ذكر مثال توضيحي عن كل نوع.

٢. ما هي الطرق المستخدمة في PHP لتضمين شفرة برمجية من ملف آخر، و ما الفرق بينها.

٣. أكتب برنامج يقوم بإنشاء دالة تقوم بتحويل العملة من دولار أسترالي إلى ريال سعودي ، مع الأخذ في الاعتبار أن سعر الصرف غير ثابت.

٤. ما هي الفائدة من جعل المحتوى المتكرر في ملفات مستقلة؟

هذا الفصل يغطي:

- **كيفية التعامل مع البيانات المرسلة من خلال النموذج**
- **الطرق المستخدمة في ارسال البيانات و الفرق بينها.**
- **كيف التعامل مع الروابط التفاعلية**

مع مرور الوقت تصبح تطبيقات PHP أكثر تعقيدا. لذلك تكون الحاجة ملحة للتعامل مع البيانات المدخلة عن طريق المستخدم. و أكثر الشائعة استخداما للتعامل مع مُدخلات المستخدم هو عن طريق النماذج HTML forms.

```
<html>
<head>
<meta        http-equiv="Content-Type"        content="text/html;
charset=utf-8" />
<style>
/*
شفرة css
و التي تهتم بتنسيق مظهر الجدول
*/
body{
      direction:rtl;
}
form{
      font-weight:bold;
}
.button{
      font-weight:bold;
      width:75px;
}

</style>
</head>
<body>
<form name="contact " method="post" action="contact.php">
  <p>الاسم:
    <input type="text" name="name" />
  </p>
  <p>البريد الالكتروني:
    <input type="text" name="email" />
  </p>
  <p>الرسالة:
    <textarea name="message" cols="45" rows="5"></textarea>
   </p>
  <p>
    <input      class="button"      type="submit"      name="send"
value="أرسل" />
    <input class="button" type="reset" name="reset" value="مسح"
/>
  </p>
</form>
</body></html>
```

من خلال الشفرة السابقة ، قمنا بإنشاء نموذج يحتوي على حقل لإدخال الاسم من نوع نص text ، حقل لإدخال البريد الالكتروني من نوع نص كذلك. بالإضافة الى مساحة نص textarea لإدخال الرسالة . أخيرا لدينا زر لإرسال البيانات من نوع submit و زر لمسح قيم الحقول من نوع reset.
كما يجب الإنتباه الى انه عند الرغبة في التعامل مع بيانات حقل معين، يجب تحديد اسم ذلك الحقل عن طريق الخاصية name لنتمكن من استخدامها لاحقا .
بقي أن نشير أن الوسم form يحتوي على خاصيتين مهمتين:

- الخاصية method و تستخدم لتحديد كيفية ارسال البيانات و لها قيمتين:
  - post
  - get
- و الخاصية الأخرى action و تستخدم لتحديد الصفحة التي سوف تستقبل البيانات المرسلة من النموذج.

الفرق بين استخدام post و get

الخاصية	get	post
التاريخ history	البيانات المرسلة يتم حفظها في تاريخ متصفح الانترنت Browser history.	البيانات المرسلة لا يتم حفظها في تاريخ متصفح الانترنت Browser history.
زر العودة أو تحديث الصفحة	يتم تنفيذ البيانات مباشرة.	يتم عرض صندوق حواري للمستخدم لتأكيد عملية التنفيذ
المفضلة Bookmark	يمكن وضعها في المفضلة	لا يمكن وضعها في المفضلة
المتغيرات المرسلة parameters	يمكن ارسالها لكنها محدودة و تختلف من متصفح لآخر.	يمكن ارسالها و لا تعاني من مشاكل طول البيانات.
العبث hacking	سهلة العبث لأن البيانات تُعرض للمستخدم في عنوان URL.	صعبة العبث لأن البيانات يتم تشفيرها و لا تعرض في عنوان URL .
الاستخدام usability	يجب أن لا يتم استخدامها عند ارسال بيانات حساسة مثل كلمات المرور أو أرقام بطاقات الائتمان.	تُستخدم عند التعامل مع بيانات تتطلب سرية عالية.

كيفية قراءة البيانات المُرسلة من النموذج

كما سبق و أن أشرنا في الفصل الخامس ، هناك عدد من المصفوفات العامة في PHP تُستخدم لأغراض محددة. من ضمن هذه المصفوفات المصفوفتين $_POST و $_GET و التي تُستخدمان لجمع البيانات المُرسلة عبر النموذج.
بمعنى أننا نقوم بالنظر للنموذج form و تحديدا للخاصية method . إذا كانت القيمة هي post فإن البيانات المُرسلة من النموذج سيتم تخزينها في المصفوفة العامة $_POST و العكس صحيح إذا كانت القيمة هي get فإن البيانات المُرسلة من النموذج سيتم تخزينها في المصفوفة العامة $_GET.
بعد ذلك نقوم بكتابة قوسين مربعة [ ] و من ثم نقوم بكتابة اسم الحقل و التي تمثل قيمة الخاصية name لعناصر النموذج بين علامات اقتباس مفردة أو مزدوجة.

order.html

```
...
<form action='ordere.php' method='post'>
  الاسم:<input type='text' name='name'/>
  الهاتف: :<input type='text' name='phone'/>
...
```

order.php

```
<?php
...
echo $_POST['name'];
echo $_POST['phone'];
...
?>
```

الصفحة المستقبلة للبيانات

طريقة ارسال البيانات post

سنقوم الآن بالعودة للمثال الأول في هذا الفصل و سنقوم بحفظ الشفرة السابقة باسم contact.html و في نفس المكان الذي تعودنا لحفظ ملفاتنا فيه (داخل المجلد الافتراضي htdocs).

الآن سنقوم بإنشاء الصفحة التي سوف تستقبل البيانات و الموضحة في الخاصية action=' contact.php' في النموذج السابق. و كما لا يجب أن ننسى الخاصية الأخرى المهمة و التي تحدد طريقة ارسال البيانات وهي الخاصية method = 'post'.

```
<?php
/*   في الجزء التالي قمنا بتعريف متغيرات لكي نحفظ فيها القيم
المرسلة من خلال النموذج*/
$name = $_POST['name'];
$email = $_POST['email'];
$message = $_POST['message'];
echo'<div style="direction:rtl;font-size:16px;">';
echo'<h2> شكرا أخي الزائر لتواصلك معنا بيانات رسالتك
كالتالي:</h2>';
echo'<hr align="right" style="width:500px"/>';
echo"<p><b>الاسم:</b>$name</p>";
echo"<p><b>البريد الالكتروني:</b>$email</p>";
echo"<p><b>الرسالة:</b>$message</p>";
echo'</div>';
?>
```

نقوم بحفظ الشفرة السابقة باسم contact.php داخل المجلد الافتراضي و ستكون هذه الصفحة هي المسئولة عن استقبال و التعامل مع البيانات المرسلة من الصفحة contact.html.
نقوم الآن باستدعاء الصفحة التي تحتوي على النموذج كالتالي http://localhost/contact.html

نقوم بتعبئة البيانات كما هو واضح و من ثم نضغط على زر الارسال. عند الضغط على زر الارسال سوف تُرسل البيانات للصفحة contact.php و التي ستقوم بطباعة البيانات كما يلي:

ارسال النموذج لنفس الصفحة

في بعض الحالات قد يكون من الأفضل ارسال النموذج لنفس الصفحة لتتم معالجته بدلا من ارسال الى صفحة أخرى مستقلة. الفائدة من هذه الخطوة أن البيانات المُدخلة في النموذج تبقى ، مما يتيح للمستخدم تعديل البيانات الخاطئة فقط.

لكن ستواجهنا مشكلة بأن مفسر PHP سيقوم بتنفيذ شفرة PHP حتى و إن لم يتم ارسال النموذج؟

لذلك الحل أن يتم حصر الشفرة البرمجية داخل جملة شرط. و يتحقق الشرط فقط عندما يتم ارسال النموذج و بالتالي يتم تنفيذ الشفرة البرمجية التي ستتعامل مع البيانات المرسلة من النموذج.

و لكي نرسل النموذج لنفس الصفحة سنستخدم المصفوفة العامة الجاهزة $_SERVER مع المتغير SCRIPT_NAME و الذي يحتوي على اسم الصفحة التي يتم تنفيذها حاليا و نضعها داخل الخاصية action في النموذج. سنقوم بالتعديل على الملفين السابقين contact.html و contact.php بحيث يصبح النموذج والشفرة البرمجية في صفحة واحدة.

```
<html>
<head>
<meta       http-equiv="Content-Type"       content="text/html;
charset=utf-8" />
<style>
body{
     direction:rtl;
}
form{
     font-weight:bold;
}
.button{
     font-weight:bold;
     width:75px;
}
</style>
</head>
<body>
<?php
if(array_key_exists('send',$_POST))
{
     $name = $_POST['name'];
     $email = $_POST['email'];
     $message = $_POST['message'];
     echo'<div style="direction:rtl;font-size:16px;">';
     echo'<h2> شكرا أخي الزائر لتواصلك معنا بيانات رسالتك
كالتالي:</h2>';
     echo'<hr align="right" style="width:500px"/>';
     echo"<p><b>الاسم:</b>$name</p>";
     echo"<p><b>البريد الالكتروني:</b>$email</p>";
     echo"<p><b>الرسالة:</b>$message</p>";
     echo'</div><br/>';
}
?>
<form   name="contact   "   method="post"   action="<?php   echo
$_SERVER['SCRIPT_NAME'];?>">
  <p>الاسم:
    <input type="text" name="name" />
  </p>
  <p>البريد الالكتروني:
    <input type="text" name="email" />
  </p>
  <p>الرسالة:
    <textarea name="message" cols="45" rows="5"></textarea>
<p>
    <input     class="button"     type="submit"     name="send"
value="أرسل" />
    <input class="button" type="reset" name="reset" value="مسح"
/>
  </p>
</form></body></html>
```

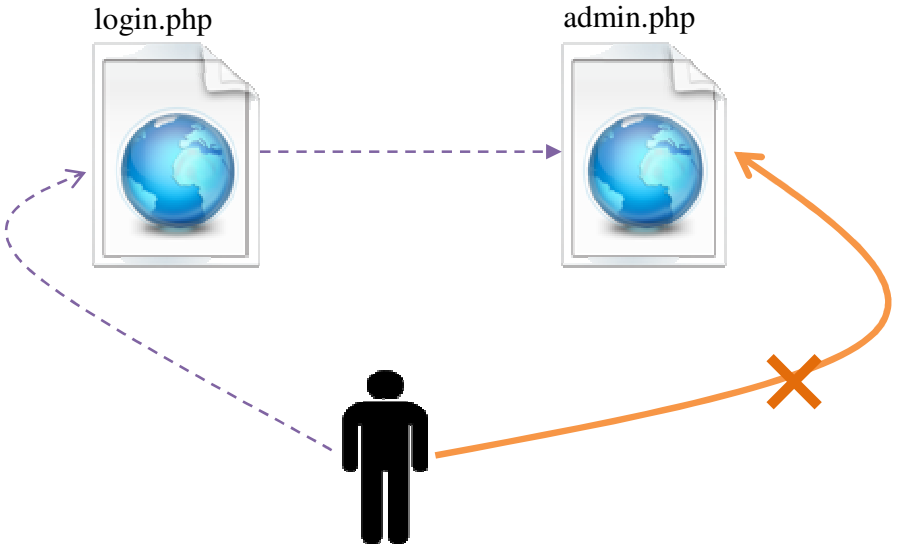
في الشفرة السابقة استخدمنا الدالة array_key_exists( ) و التي تأخذ معطيين. الأول عبارة عن مفتاح key يتم البحث عن وجوده في مصفوفة و التي تمثل المعطى الثاني للدالة. المفتاح في المثال السابق يمثل اسم زر الإرسال send و المصفوفة هي المصفوفة العامة $_POST.
تقوم الدالة بالبحث عن المفتاح send داخل المصفوفة $_POST لأن طريقة ارسال البيانات في النموذج هي post.
عند الضغط على زر الارسال سيتم ارسال المفتاح send للمصفوفة $_POST و ستكون نتيجة اختبار الدالة هي true و يتم تنفيذ الجزء التابع لجملة الشرط if و العكس صحيح اذا لم يتم ارسال النموذج أي سوف تكون نتيجة اختبار الدالة هي false.
في الجزء action الخاص بالنموذج قمنا بطباعة المتغير العام $_SERVER['SCRIPT_NAME'] و الذي سيقوم بإرجاع اسم الصفحة الحالية contact.php.

التعامل مع أزرار الاختيار radio

أزرار الاختيار radio من العناصر البسيطة و الفعالة عند التعامل مع نماذج HTML. في الوضع الافتراضي، نقوم بتحديد أحد أزرار الاختيار radio ليكون هو القيمة الافتراضية وذلك عن طريق أعطاءه الخاصية checked للدلالة على أن هذا الحقل هو القيمة الافتراضية. بمعنى أنه، إذا لم يقم المستخدم بإختيار أيا من أزرار الاختيار، فسيتم ارسال قيمة الحقل المحدد مسبقا كقيمة افتراضية الى ايا من المصفوفتين $_POST أو $_GET عند ارسال النموذج.
في حال كانت قيم أزرار الاختيار radio مرتبطة ببعضها، مثل لو أردنا تحديد ما إذا كان المستخدم ذكر أم أنثى. ففي هذه الحالة يجب أن نعطي جميع أزرار الاختيار نفس الاسم.

حماية صفحة انترنت

في بعض الحالات قد نحتاج الى حماية صفحة انترنت من الوصول اليها مباشرة عن طريق كتابة عنوان الصفحة في شريط عنوان المتصفح.
لنفترض أنه لدينا صفحة انترنت باسم admin.php و للوصول لهذه الصفحة بطريقه صحيحة يجب على المستخدم المرور أولا عبر الصفحة login.php و تعبئة بيانات الدخول. إذا حاول المستخدم الوصول للصفحة admin.php عن طريق كتابة عنوان الصفحة مباشرة في شريط عنوان المتصفح سنقوم بمنعه و طباعة رسالة مناسبة.



```
<html>
<head>
<meta       http-equiv="Content-Type"       content="text/html;
charset=utf-8" />
<style>
body{
     direction:rtl;
}
form{
     font-weight:bold;
}
.button{
     font-weight:bold;
     width:75px;
}
</style>
</head>
<body>
<form method="post" action="admin.php">
  <p>اسم المستخدم
    <label for="user_name"></label>
  <input type="text" name="user_name" id="user_name" />
  </p>
  <p>كلمة المرور
    <label for="password"></label>
    <input type="password" name="password" id="password" />
  </p>
  <p>تذكرني
    <input type="checkbox" name="remeber_me" id="remeber_me" />
    <label for="remeber_me"></label>
  </p>
  <p>
    <input  type="submit"  name="login"  id="login"  value="دخول"
/>
  </p>
</form>
</body>
</html>
```

احفظ الصفحة السابقة باسم login.php

```
<?php
if(array_key_exists('login',$_POST))
{
     echo'<h1>مرحبا بك في صفحة الادارة</h1>';
}
else
{
     echo'ليس لديك صلاحية للوصول لهذه الصفحة';
}
?>
```

احفظ الصفحة السابقة باسم admin.php.
قم الآن بكتابة العنوان التالي http://localhost/login.php و من ثم قم بتعبئة النموذج و اضغط على زر دخول.
بعد ذلك قم بفتح نافذة جديدة من متصفح الانترنت عن طريق الضغط على CRTL + N و من ثم قم بكتابة العنوان التالي http://localhost/admin.php ، الآن لاحظ الناتج.

الروابط التفاعلية dynamic links

في الفصول السابقة كنا نقوم بإرسال البيانات عن طريق النماذج باستخدام POST أو GET إلى صفحات PHP لكي يتم معالجتها. لكن ماذا لو أردنا بدلا من إرسال المعلومات عن طريق النماذج أن نقوم بإرسالها عن طريق الروابط التشعبية ضمن الوسم <a>. في هذه الحالة سيتم إرسال البيانات إلى المصفوفة العامة $_GET (عبارة عن مصفوفة ترابطية associative array) . و كما هو معلوم لدينا أنه في المصفوفات الترابطية المفاتيح عبارة عن نصوص و ليست أرقام. لذلك لو أردنا أن نضيف عنصر داخل المصفوفة $_GET و له القيمة Ahmad فنقوم بما يلي :

```
$_GET[ 'name' ] = "Ahmad" ;
```

لكن كيف سيتم إرسال رابط تشعبي و يتم تخزين قيمه داخل المصفوفة $_GET ؟

لكي نقوم بذلك نقوم أولا بكتابة اسم الصفحة التي ستقوم باستقبال البيانات ، مثلا login.php متبوعة بعلامة استفهام ? و من ثم اسم المفتاح متبوعا بعلامة يساوي = يليها قيمة العنصر كالتالي:

```
<a href= "login.php?name=Ahmad">
```

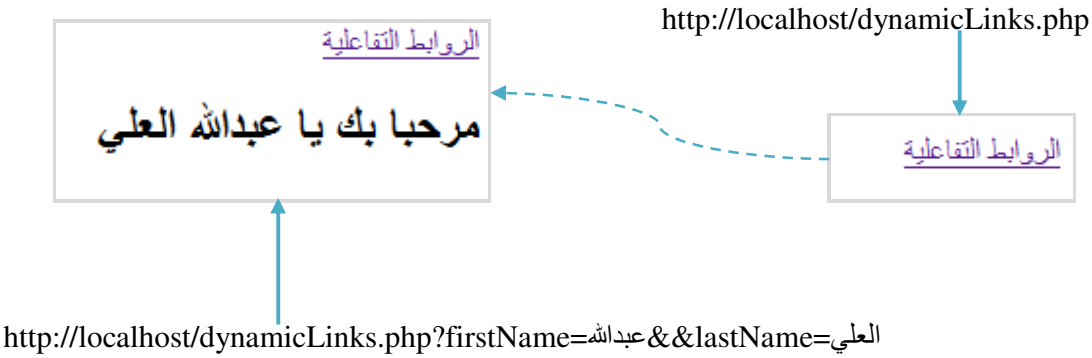
القيمة المفتاح اسم الصفحة متبوع بعلامة ?

في حال أردنا إضافة أكثر من قيمة نقوم بالفصل بين القيم كالتالي:

```
<A href= "login.php?name=Ahmad&password=12345">
```

```
<html>
<head>
<meta       http-equiv="Content-Type"       content="text/html;
charset=utf-8" />
<style>
body{
     direction:rtl;
}
</style>
</head>
<body>
<a
href="dynamicLinks.php?firstName=عبدالله&&lastName=  الروابط العلي"><
التفاعلية </a><br />
```

```
<?php
if(array_key_exists('firstName',$_GET)
&& array_key_exists('lastName',$_GET))
{
     echo'<h2>يا بك مرحبا ' . $_GET['firstName'] . ' ' .
$_GET['lastName'] .'</h2>';
}
?>
</body>
</html>
```



استخدام الدالة trim( )

تعتبر الدالة trim( ) من دوال PHP المبنية مسبقا وتُستخدم لإزالة المسافات الفارغة في بداية و نهاية النص string . و تعتبر من الدوال التي يشيع استخدامها مع نماذج HTML و ذلك لإزالة أي مساحات فارغة في القيم المدخلة من قبل المستخدم قبل إجراء أي عملية على تلك البيانات كتخزينها في قواعد البيانات مثلا.

```
<?php
// remove leading and trailing whitespace
// output 'a b   c'
$str = '  a b   c   ';
echo trim($str);
?>
```

تطبيق عملي

في هذا التطبيق سنقوم بتصميم نموذج تسجيل زائر جديد. وقبل أن يتمكن الزائر من البدء في تعبئة النموذج يجب أن يقوم الزائر بالموافقة على الشروط التسجيل.
قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>نموذج التسجيل</title>
<style>
body{
      direction:rtl;
}
.agreement{
      border:1px solid #CCC;
      width:350px;
      padding:5px;
}
form input{
      border:1px solid #CCC;
}
form .button{
      width:80px;
}
form fieldset{
      width:350px;
      border:1px solid #CCC;
}
form fieldset legend{
      font-weight:bold;
      border:1px solid #CCC;
}

</style>

</head>

<body>
<?php
/*
الشرط نتحقق من خلاله من ما اذا تمت الموافق على اتفاقية التسجيل
ام لا.
اذا لم تتم الموافقة لن يتم الانتقال لنموذج التسجيل
*/
if(!array_key_exists('agreement',$_POST) &&
!array_key_exists('register',$_POST))
{
```

```
      echo"
          <p class='agreement'>
            تتطلب عملية التسجيل في الموقع الموافقة على شروط
التسجيل بالموقع و التي تتضمن استخدام الموقع بصورة
           شرعية. وعدم العبث بالموقع أو التهجم على الاشخاص....
           </p>
           <form method='post'>
           <p>أوافق على شروط التسجيل<input type='checkbox'
name='agreement'/></p>
           <p><input type='submit' name='accept_agreement'
value='الخطوة التالية' class='button'/></p>
           </form>
      ";
}
/*
هنا نتحقق من ان المستخدم وافق على اتفاقية التسجيل و لم يتمم بعد
عملية التسجيل
*/
elseif(array_key_exists('agreement',$_POST) &&
!array_key_exists('register',$_POST))
{
?>
    <form method='post'>
        <fieldset >
        <legend>نموذج تسجيل عضو جديد</legend>
        <table>
         <tr>
           <td>الاسم الكامل*: </td>
           <td><input type='text' name='name' id='name'
maxlength="50" /></td>
         </tr>
         <tr>
           <td>البريد الالكتروني*:</td>
           <td><input type='text' name='email' id='email'
maxlength="50" /></td>
         </tr>
         <tr>
           <td>اسم المستخدم*:</td>
           <td><input type='text' name='username' id='username'
maxlength="50" /></td>
         </tr>
         <tr>
           <td>كلمة المرور*:</td>
           <td><input type='password' name='password'
id='password' maxlength="50" /></td>
         </tr>
         <tr>
           <td><br />* الحقل مطلوب</td><td><br /><input
type='submit' name='register' value='اتمام التسجيل'
class="button"/></td>
         </tr>
```

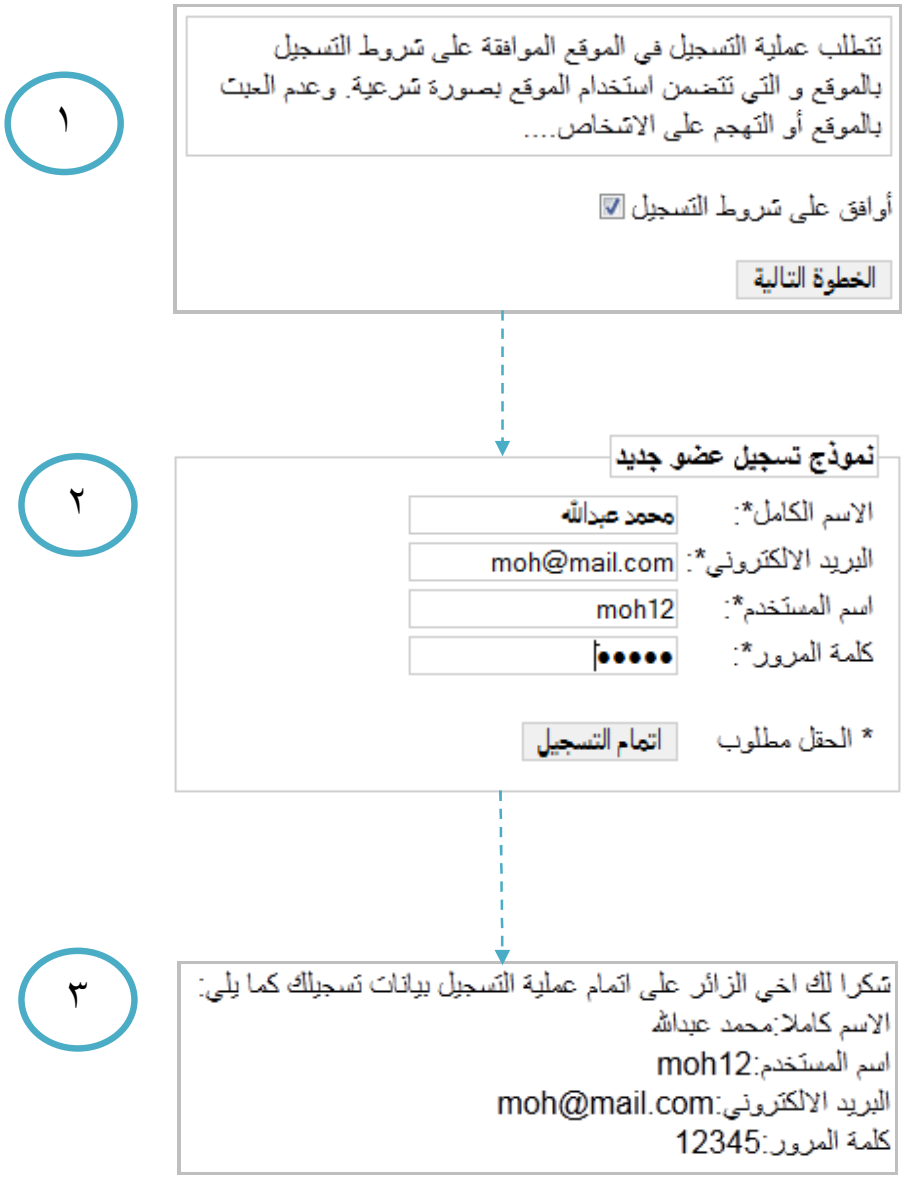
```
</table>
        </fieldset>
    </form>
<?php
}
else
{
      $name = $_POST['name'];
      $username = $_POST['username'];
      $email = $_POST['email'];
      $pass = $_POST['password'];

      echo'شكرا لك اخي الزائر على اتمام عملية التسجيل ';
      echo' بيانات تسجيلك كما يلي:';
      echo'<br/>الاسم كاملا:' . $name;
      echo'<br/>اسم المستخدم:' . $username;
      echo'<br/>البريد الالكتروني:' . $email;
      echo'<br/>كلمة المرور:' . $pass;
}
?>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقة ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم register.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/register.php و من ثم لاحظ الناتج.


١
تتطلب عملية التسجيل في الموقع الموافقة على شروط التسجيل بالموقع و التي تتضمن استخدام الموقع بصورة شرعية. وعدم العبث بالموقع أو التهجم على الاشخاص....
أوافق على شروط التسجيل
الخطوة التالية
٢
نموذج تسجيل عضو جديد
الاسم الكامل*: محمد عبدالله
البريد الالكتروني*: moh@mail.com
اسم المستخدم*: moh12
كلمة المرور*: •••••
* الحقل مطلوب
اتمام التسجيل
٣
شكرا لك اخي الزائر على اتمام عملية التسجيل بيانات تسجيلك كما يلي:
الاسم كاملا:محمد عبدالله
اسم المستخدم:moh12
البريد الالكتروني:moh@mail.com
كلمة المرور:12345

أسئلة نهاية الفصل

١. عند التعامل مع النماذج ما هو الفرق بين طريقة إرسال البيانات POST و GET .

٢. ما وظيفة كلا من الدوال التالية:

- array_key_exists( )
- trim( )

٣. كيف يمكن التحقق من أزرار الاختيار radio في حال كانت تتبع لمجموعة واحدة.

٤. أين سيتم تخزين القيم المرسلة من خلال الرابط التفاعلي .

هذا الفصل يغطي:

- **أهمية عملية التحقق من البيانات المُدخلة.**
- **مفهوم التعبيرات العادية Regular Expressions.**
- **كيف مطابقة الانماط في PHP.**

عند التعامل مع نماذج HTML بغرض السماح للمستخدمين بإدخال بيانات محددة فيجب علينا التحقق من تلك البيانات قبل إجراء أي عملية عليها. و تكمن أهمية إجراء عملية التحقق لأننا لا نضمن أن تكون البيانات المُدخلة صحيحة، وبدون إجراء عملية التحقق قد تتسبب تلك البيانات بحدوث نتائج غير متوقعة قد يصعب معالجتها لاحقا. لذلك يجب أن لا نثق في المستخدم لأن الأخطاء قد تحدث عمدا أو عن غير قصد.
عملية التحقق من البيانات يمكن أن تتم بواسطة طريقتين:

- عن طريق جهة العميل client-side باستخدام JavaScript مثلا.
- عن طريق جهة الخادم server-side باستخدام php.

ما هي الأشياء التي يجب أن نتحقق منها

- التحقق من الحقول الفارغة.
- التحقق من البريد الالكتروني.
- طول البيانات المُدخلة.
- حقول الأرقام فقط.
- حقول الحروف فقط.
- حقول الحروف و الأرقام.
- إزالة وسوم HTML.

و لتحقيق ذلك يوجد طرق كثيرة في PHP و من ضمنها التعبيرات العادية Regular Expressions . في هذه الطريقة يتم البحث عن نمط pattern ضمن نص محدد.

أساسيات التعبيرات العادية Regular Expressions Basics

في التعبيرات العادية مُعظم الحروف تطابق نفسها. فمثلا لو بحثنا عن التعبير "foo" داخل النص "John plays football" فستتم المطابقة لظهور التعبير "foo" داخل النص السابق.
بعض الحروف الخاصة special characters لها معنى خاص في التعبيرات العادية.
فمثلا علامة $ تُستخدم للبحث عن نمط في نهاية النص. و العلامة ^ تُستخدم للبحث عن نمط في بداية النص. و النقطة . تُستخدم للبحث عن أي حرف ماعدا السطر الجديد.
و في حال أردنا البحث عن نمط داخل نص بعدد مرات محدد فيمكن استخدام الرموز التالية:

- الرمز * للبحث عن نمط داخل نص بعدد صفر أو أكثر من المرات.
- الرمز + للبحث عن نمط داخل نص بعدد واحد أو أكثر من المرات .
- الرمز ؟ للبحث عن نمط داخل نص بعدد صفر أو واحد من المرات.

- الرمز { } للبحث عن نمط داخل نص بعدد مرات مختلف. فمثلا {n} على فرض أن n عبارة عن رقم صحيح تعني عدد مرات ظهور النمط داخل النص. و التعبير {n,m} تعني مرات ظهور النمط داخل النص على الأقل n و على الأكثر m .

الجدول التالي يحتوي على عدد من الأمثلة عن التعبيرات العادية.

التعبير العادي Regular Expression	سيتم مطابقة..
foo	سيتم مطابقة الكلمة foo
^foo	سيتم مطابقة الكلمة foo في بداية النص
foo$	سيتم مطابقة الكلمة foo في نهاية النص
^foo$	يجب أن يحتوي النص على الكلمة foo فقط.
[abc]	سيتم مطابقة a ، b أو c
[a-z]	أي حرف صغير
(gif|jpeg)	سيتم مطابقة إما gif أو jpeg
[a-z]+	سيتم مطابقة أي حرف صغير بعدد مرات ظهور مرة أو أكثر.
[0-9\.\-]	سيتم مطابقة أي رقم أو نقطة أو اشارة سالب
^[a-zA-Z0-9_]{1,}$	سيتم مطابقة كلمة واحدة فقط تحتوي على الأقل على حرف أو رقم أو شرطة سفلية.
[^A-Za-z0-9]	أي رمز (يجب أن لا يكون حرفا أو رقما)
([A-Z]{3}|[0-9]{4})	سيتم مطابقة أي كلمة تحتوي على ثلاثة أحرف أو أربعة أرقام.

كل الأنماط عند استخدامها بشكل عملي يجب أن توضع بين شرطتين مائلة / /.

للمزيد حول التعبيرات العادية Regular Expressions يمكن زيارة الموقع التالي:
http://www.regular-expressions.info/

مطابقة الأنماط Matching Patterns

أخيرا لمطابقة الأنماط في PHP سنقوم بإستخدام الدالة preg_match( ) . يتم تمرير عدد من المعطيات parameters لهذه الدالة. من ضمن هذه المعطيات ، النمط pattern و النص string الذي سيتم البحث داخله. في حال و جدت الدالة النمط داخل النص ستقوم بإرجاع القيمة 1 (التي تماثل true) بخلاف ذلك ستقوم بإرجاع القيمة 0 (التي تماثل false).

```
<?php
$string = "abcdef";
$pattern = '/^def/';
echo preg_match($pattern,$string);
//٠ سيكون الناتج
?>
```

```php
<?php
$string = "defabc";
$pattern = '/^def/';
echo preg_match($pattern,$string);
//١ سيكون الناتج
?>
```

تطبيق عملي

في هذا التطبيق سنقوم بتصميم نموذج تسجيل زائر جديد. في هذا التطبيق لن يتمكن الزائر من إرسال النموذج إلا بعد أن يقوم بتعبئة كامل النموذج. كما يجب أن تكون المعلومات المُدخلة صحيحة و لتحقيق ذلك سنقوم باستخدام التعبيرات العادية Regular Expressions مع الدالة preg_match() للقيام بعملية التحقق validation.
قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<html>
<head>
<meta        http-equiv="Content-Type"        content="text/html;
charset=utf-8" />
<title>نموذج تسجيل</title>
<style type="text/css">
body{
      direction:rtl;
}

form input{
      border:1px solid #CCC;
}
#result{
      font-weight:bold;
      color:#930;
      text-align:center;
}

table{
      border:2px solid #0C3;
      width:500px;
      padding:10px;
}
table caption{
      background-color:#0C3;
      font-weight:bold;
}
.submit{
      width:75px;
}

.errorMessage{
      text-align:right;
```

```
font-weight:normal;
}
</style>
</head>
<body>
<form action="" method="post">
<table align="center">
  <caption>نموذج تسجيل مستخدم جديد</caption>
<tr><td colspan="2" id="result">
  <?php
  if(array_key_exists('register',$_POST))
  {
        $errors=array();

        if(empty($_POST['name']))
        {
              array_push($errors,'الرجاء إدخال قيمة للاسم.');
        }
        elseif(preg_match('/[\'\/~`\!@#\$%\^&\*\(\)_\-
\+=\{\}\[\]\|;:"\<\>,\.\?\\\]/', $_POST['name']))
        {
              array_push($errors,' الاسم المدخل يحتوي على قيم غير
صالحة.');
        }

        if(empty($_POST['username']))
        {
              array_push($errors,' الرجاء ادخال قيمة لإسم
المستخدم');
        }
        elseif(!preg_match("/^[A-Za-z][A-Za-z0-
9_]+$/",$_POST['username']))
        {
              array_push($errors,' القيمة المدخلة لإسم المستخدم
غير صالحة.');
        }

        if(empty($_POST['password']))
        {
              array_push($errors,' الرجاء ادخال قيمة لكلمة
المرور');
        }
        elseif(!preg_match("/^[A-Za-z0-9_]*[A-Z][A-Za-z0-
9_]*$/",$_POST['password']))
        {
              array_push($errors,' القيمة المدخلة لكلمة المرور
غير صالحة.');
        }
```

```
elseif(strlen($_POST['password'])<6)
        {
             array_push($errors,' كلمة المرور يجب أن لا تقل عن ٦
خانات.');
        }
        elseif($_POST['password2'] !== $_POST['password'])
        {
             array_push($errors,' تأكيد كلمة المرور لا يطابق كلمة
المرور المُدخلة');
        }
if(empty($_POST['email']))
        {
             array_push($errors,' الرجاء ادخال قيمة للبريد
الالكتروني');
        }
        elseif(!preg_match("/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-
9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})]*$/",$_POST['email']))
        {
             array_push($errors,' القيمة المدخلة للبريد
الإلكتروني غير صالحة.');
        }

        if(empty($_POST['phone']))
        {
             array_push($errors,'الرجاء ادخال قيمة للهاتف');
        }
        elseif(!preg_match("/^[0-9]{10,12}$/",$_POST['phone']))
        {
             array_push($errors,' القيمة المدخلة للهاتف غير
صالحة.');
        }

        if(empty($_POST['gender']))
        {
             array_push($errors,'الرجاء تحديد جنسك');
        }

        if(empty($_POST['from']))
        {
             array_push($errors,'الرجاء تحديد بلدك.');
        }

        if(empty($_POST['agreement']))
        {
             array_push($errors,' الرجاء الموافقة على شروط
التسجيل.');
        }
```

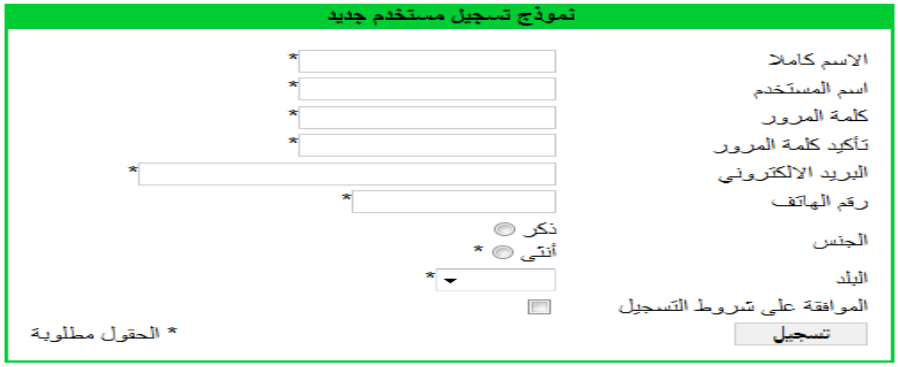
```
      if($errors)
        {
              echo'<ol class="errorMessage">';
              foreach($errors as $value)
              {
                    echo '<li>'.$value.'</li>';
              }
              echo'</ol>';
        }
      else
      {
           echo'.شكرا لك لقد أتممت عملية التسجيل بنجاح';
      }
}
  ?>
</td></tr>
  <tr>
    <td>الاسم كاملا</td>
    <td><input  name="name"  id="name"  size="20"  maxlength="20"
type="text" value="<?php echo @$_POST['name']?>"/>*</td>
  </tr>
  <tr>
    <td>اسم المستخدم</td>
    <td><input    name="username"    id="username"    size="20"
maxlength="20"       type="text"       value="<?php       echo
@$_POST['username']?>"/>*</td>
  </tr>
  <tr>
    <td>كلمة المرور</td>
    <td><input    name="password"    id="password"    size="20"
maxlength="15" type="password">*</td>
  </tr>
  <tr>
    <td>تأكيد كلمة المرور</td>
    <td><input   name="password2"   id="password2"   size="20"
maxlength="15" type="password">*</td>
  </tr>
  <tr>
    <td>البريد الالكتروني</td>
    <td><input     name="email"     size="35"     maxlength="30"
type="text" value="<?php echo @$_POST['email']?>">*</td>
  </tr>
  <tr>
    <td>رقم الهاتف</label></td>
    <td><input     name="phone"     size="15"     maxlength="15"
type="text" value="<?php echo @$_POST['phone']?>">*</td>
  </tr>
  <tr>
    <td>الجنس</td>
    <td>ذكر<input  name="gender"  type="radio"  value="male"><br
/>
```

```
أنثى<input name="gender" type="radio" value="female">*
    </td>
  </tr>
  <tr>
    <td>البلد</td>
    <td>
    <select name="from">
     <option value="" selected></option>
     <option value="السعودية">السعودية</option>
<option value="قطر">قطر</option>
     <option value="الكويت">الكويت</option>
    </select>*
    </td>
  </tr>
  <tr>
<td>الموافقة على شروط التسجيل</td>
    <td><input name="agreement" type="checkbox"/></td>
  </tr>
  <tr>
    <td><input   name="register"   value="تسجيل"   type="submit"
class="submit"/></td>
    <td align="left">* الحقول مطلوبة</td>
  </tr>
</table>
</form>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقة ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم form.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.

بعد الانتهاء من عملية الحفظ ،افتح متصفح الانترنت و اكتب العنوان التالي: http://localhost/form.php و من ثم لاحظ الناتج.



في حال قام الزائر بإرسال النموذج بدون إدخال بيانات فسيتم عرض رسائل الخطأ التالية.

شرح التطبيق:

١. قمنا بتعريف المصفوفة $errors لتخزين رسائل الأخطاء التي ستحدث أثناء عملية التحقق.
٢. في حال حدوث أي خطأ أثناء عملية التحقق سنقوم بإدراج رسالة الخطأ المناسبة داخل المصفوفة السابقة عن طريق استدعاء الدالة ( )array_push و تمرير معطيين لها. الأول يمثل إسم المصفوفة و الثاني رسالة الخطأ.
٣. نقوم بالتحقق من الحقول الفارغة عن طريق الدالة ( )empty عن طريق تمرير المتغير الذي يحتوي على قيمة الحقل المحدد. في حال كان الحقل فارغا ستقوم الدالة بإرجاع القيمة true.
٤. في حقل الاسم استخدمنا النمط التالي

```
/[\'\/~`\!@#\$%\^&\*\(\)_\-\+=\{\}\[\]\|;:"\<\>,\.\?\\\]/
```

و الهدف من هذا النمط أن يتم رفض أي قيم غير الحروف و الأرقام و المسافة.
٥. في حقل اسم المستخدم استخدمنا النمط التالي

```
/^[A-Za-z][A-Za-z0-9_]*+$/
```

و ذلك لكي يتم قبول فقط الحروف والأرقام و الشرط السفلية.

٦. في حقل كلمة المرور استخدمنا النمط

```
/^[A-Za-z0-9_]*[A-Z][A-Za-z0-9_]*$/
```

و هنا فرضنا ان تحتوي كلمة المرور على حروف وأرقام و شرطة سفلية بالإضافة الى وجوب أن تحتوي كلمة المرور على حرف واحد كبير على الأقل.

٧. يجب أن لا تقل كلمة المرور عن ٦ خانات و لتحقيق ذلك استخدمنا الدالة ( )strlen و التي تستقبل معطى واحد يمثل المتغير الذي ستقوم بإرجاع طوله.

٨. يجب أن تتطابق كلمة المرور المُدخلة مع حقل تأكيد كلمة المرور.

٩. في حقل البريد الالكتروني استخدمنا النمط التالي

```
/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-
]+)*(\.[a-z]{2,3})]*$/
```

و هذا النمط يهدف إلى أن يكون البريد المُدخل بريدا صالحا.

١٠. في حقل الهاتف استخدمنا النمط `/^[0-9]{10,12}$/` و باستخدام هذا النمط فرضنا قبول فقط قيمة رقمية لا تقل عن عشرة أرقام و لا تزيد عن اثني عشر رقما.

١١. بعد الانتهاء من عملية التحقق قمنا بالتأكد من أن المصفوفة $errors فارغة، و في هذه الحالة و باستخدام حلقة التكرار foreach قمنا بطباعة محتوى المصفوفة.

١٢. بخلاف ذلك، إذا كانت المصفوفة فارغة أي أن المستخدم أدخل كل البيانات المطلوبة بشكل صحيح ، عندها قمنا بطباعة رسالة مناسبة.

١٣. أخيراً، لضمان أن الزائر لا يقوم بتعبئة الحقول مرة أخرى قمنا بطباعة قيمة الحقل المُرسلة من النموذج كما يلي:

```
<input name="name" id="name" size="20" maxlength="20"
type="text" value="<?php echo @$_POST['name']?>"/>
```

و كما هو ملاحظ أسبقنا المتغير بالعلامة @ لمنع ظهور الخطأ قبل أن يتم إرسال النموذج.

أسئلة نهاية الفصل

١. ما أهمية التحقق من البيانات المُدخلة في النموذج؟

٢. أيهما أفضل التحقق من البيانات بواسطة جهة العميل أو جهة الخادم مع ذكر السبب؟

٣. هل هناك طريقة أخرى غير استخدام الدالة empty( ) للتحقق من الحقول الفارغة ؟

٤. هل يمكن استخدام وسوم HTML بهدف الإضرار بتطبيق الانترنت و كيف يمكن منع المستخدم من إدخالها في النموذج؟

٥. مستخدما شبكة الإنترنت كيف يمكن التغلب على مشكلة SPAM مع النماذج؟

هذا الفصل يغطي:

- **كيفية التعامل مع cookie.**
- **كيفية التعامل مع الجلسة Session.**
- **الطرق المُستخدمة لإعادة توجيه المستخدم.**

كما هو معلوم أن البروتوكول المستخدم في نقل البيانات بين متصفح الإنترنت و مواقع الويب المختلفة هو بروتوكول نقل النص التشعبي HTTP(Hypertext Transfer Protocol) . لكن قد لا يعرف الكثير أن هذا البرتوكول من النوع stateless بمعنى أنه يتعامل مع كل طلب request بشكل مستقل بدون أن يكون هناك أي علاقة بين الطلب الحالي current request و الطلب السابق previous request . لشرح هذه العملية ، لنتخيل معا كيف يتم العمل في المواقع التي تتيح خدمة التسوق الإلكتروني أو ما يسمى بـ shopping carts . الذي يحدث أن الزائر لموقع التسوق يقوم باستعراض المنتجات التي يقدمها الموقع و من ثم يقوم باختيار المنتج و إضافته إلى سلة الشراء. في الوضع العادي ، إذا انتقل الزائر إلى صفحة أخرى داخل الموقع لعرض منتج آخر فإن القيمة الأولى المضافة إلى سلة الشراء و التي تمثل المنتج الأول الذي قام الزائر بإضافته إلى سلة الشراء سيتم فقدها !! و السبب كما ذكرنا أن البروتوكول HTTP من النوع الذي لا يتذكر ما تم مسبقا مع الطلبات السابقة. للتغلب على هذه المشكلة فإن كثير من مواقع الويب تقوم باستخدام ما يسمى بـ sessions و cookies . PHP تقدم دعم كامل لـ sessions و cookies عن طريق عدد من الدوال المبنية مسبقا و الجاهزة للاستخدام من قبل مطوري الويب.

التعامل مع الـ cookies

cookies عبارة عن ملفات نصية صغيرة يتم تخزينها في جهاز المستخدم بواسطة موقع إنترنت. هذه الملفات تحتوي على معلومات تسترجع من قبل موقع الإنترنت الذي قام المستخدم بزيارته. و كمثال عليها المنتديات الحوارية. عندما يقوم المستخدم بتسجيل الدخول إلى أي منتدى للحوار يقوم بإدخال اسم المستخدم و كلمة المرور ، و في حال قام بتحديد صح على مربع (تذكرني) في نموذج الدخول سيقوم الموقع بتخزين ملف cookie داخل جهاز المستخدم يتم استخدامها لاحقا اذا قام المستخدم بزيارة الموقع لاحقا و يتم تسجيل دخوله تلقائيا عن طريق المعلومات المخزنه داخل جهازه. تعاني cookies من كثير من النقد لأنها تستخدم لنقل البيانات بين المستخدم و موقع الويب لذلك توصف دائما بأنها سيئة و غير آمنة. لكن الحقيقة أنها تعتبر كغيرها من التقنيات يمكن استخدامها بشكل سلبي أو إيجابي . لكن أكثر ما يعيبها أنه يمكن قراءتها من قبل أي شخص يستخدم جهاز الكمبيوتر لأنها ببساطة عبارة عن ملفات نصية. لذلك يجب تجنب حفظ كلمات المرور باستخدام cookie إلا في حال أن يتم تشفير كلمة المرور قبل أن يتم حفظها و من ثم يتم مطابقتها بكلمة المرور المشفرة و المخزنة على الخادم web server.

أخيرا تتميز cookie بعض الخواص الآمنة:

- ملفات cookies يمكن قراءتها فقط بواسطة موقع الإنترنت الذي قام بتخزينها.
- كل موقع إنترنت لا يمكنه تخزين أكثر من ٢٠ ملف cookies.
- أقصى حجم لملف الـ cookie الواحد هو ٤ كيلوبايت.
- أقصى عدد لملفات cookies التي يمكن تخزينها داخل جهاز المستخدم هو ٣٠٠ ملف.

لأن ملفات cookies سيتم تخزينها داخل جهاز المستخدم ، لذلك ستكون إمكانية التحكم فيها محدودة. فلو قام المستخدم بإبطال دعم cookies (من خيارات متصفح الإنترنت) فلن يتمكن أي موقع من تخزين ملفات cookies داخل جهاز المستخدم.

إنشاء ملف cookie

لإنشاء ملف cookie سنقوم باستدعاء الدالة ( )setcookie و التي تأخذ القيم التالية:

Parameters المعطى	Description الوصف
name	إسم ملف cookie (إجباري)
value	قيمة cookie
expires	الوقت بالثواني لصلاحية ملف cookie
domain	إسم موقع الإنترنت الذي خزن ملف cookie
path	تقوم بتحديد ما إذا كان ملف cookie سيكون متاحا لكامل الموقع أو سيكون متاح لصفحات محددة داخل مجلد يتم تحديده. الخيار (/) يعني إتاحة الملف لكامل الموقع.
secure	هذا الخيار يحدد أن ملف cookie يتطلب اتصال آمن (https)
httponly	هذا الخيار يتيح إمكانية الوصول لملف cookie فقط عن طريق بروتوكول http

بسبب أن ملفات cookies سيتم إعدادها بواسطة HTTP headers فيجب وضعها قبل أي عملية طباعة بواسطة شفرة PHP . في حال كان ملف PHP مضمنا مع وسوم HTML فيجب وضع الدالة أيضا قبل وسوم HTML.

```
<?php
   setcookie("username",   "Ahmad   Ali",   time()+3600,   "/","",
0,1);
   setcookie("language", "ar", time()+3600, "/", "",  0, 1);
?>
<html>
<head>
</head>
<body>
<?php echo "تم انشاء ملف الكوكي"?>
</body>
</html>
```

قراءة ملفات الـ cookie

نستطيع قراءة ملف cookie عن طريق المصفوفة العامة $_COOKIE بنفس الطريقة التي كنا نتعامل بها مع النماذج.

```
<?php
if(array_key_exists('username',$_COOKIE))
{
      echo $_COOKIE['username'].'<br/>';
      echo $_COOKIE['language'].'<br/>';
}
else
{
      echo'No cookie';
}
?>
```

حذف ملف الـ cookie

لحذف ملف cookie كل ما علينا عمله هو استدعاء الدالة ( )setcookie و نقوم بكتابة نفس قيم ملف cookie الذي قمنا بإنشائه مسبقا لكن مع تحديد وقت الصلاحية ملف cookie الى وقت سابق بالإضافة ترك القيمة فارغة كالتالي:

```
<?php
   setcookie("username", "", time()-3600, "/","", 0,1);
   setcookie("language", "", time()-3600, "/", "",  0, 1);
?>
<html>
<head>
</head>
<body>
<?php echo "تم حذف ملف الكوكي"?>
</body>
</html>
```

العمل مع sessions

الجلسة أو session تعمل نفس عمل cookie بتوفير طريقة للاستمرارية (تذكر الحالة) مع اختلاف بسيط في التعامل. فبدلا من تخزين الملف في جهاز العميل يتم تخزين الملف على جهاز الخادم web server في مجلد مؤقت يتم تخزين متغيرات session وقيمها داخله . و لهذا السبب تعتبر sessions أكثر أمانا من cookies.
أخيرا تنتهي session بمجرد إقفال المستخدم لمتصفح الإنترنت.

إنشاء الجلسة creating session

لإنشاء session كل ما علينا عمله هو إستدعاء الدالة ( )session_start في بداية كل صفحة نرغب بأن تُستخدم في الجلسة مع ضرورة الانتباه لما يلي:

- هذه الدالة يجب أن تستدعى مرة واحدة فقط في كل صفحة.
- يجب أن نستدعي هذه الدالة قبل طباعة أي بيانات بواسطة الشفرة البرمجية بالإضافة إلى أنها يجب أن تسبق وسوم HTML.

إنشاء متغيرات الجلسة creating session variables

لإنشاء متغير الجلسة ، نستطيع إضافته إلى المصفوفة العامة $_SESSION (ترابطية Associative) بنفس الطريقة التي نتعامل بها مع المتغيرات. فلو أردنا مثلا تخزين إسم الزائر و طباعة رسالة ترحيب له. فسوف نقوم بتخزين إسمه في الجلسة كالتالي:

```
$_SESSION['name'] = $_POST['name'];
```

كما هو واضح قمنا بتخزين إسم الزائر و المدخلة عن طريق نموذج التسجيل في المصفوفة العامة $_SESSION.
نستطيع الآن استخدام متغير الجلسة السابق في كل صفحة تبدأ بـ ( )session_start .

حذف متغير الجلسة destroying session variable

في حال أصبح متغير الجلسة غير مطلوب بواسطة الشفرة البرمجية فتستطيع تدميره/حذفه عن طريق استدعاء الدالة unset( ) كالتالي:

```
unset($_SESSION['name']);
```

في هذه لن يصبح المتغير السابق متاحا لأي صفحة.

حذف الجلسة destroying session

في حال لم يعد هناك أي حاجة للجلسة الحالية نستطيع تدمير الجلسة عن طريق استدعاء الدالة session_destroy( ).

إنهاء الجلسة بعد مرور وقت محدد session time-outs

في بعض الحالات عند التعامل مع نظام تسجيل الدخول user login system في موقع الانترنت تبرز الحاجة لاستخدام طريقة تضمن انهاء الجلسة في حال لم يقم المستخدم بأي نشاط بعد مرور وقت محدد. هذي الطريقة تساعد على منع استخدام النظام من قبل شخص غير مخول له بذلك. فلك أن تتخيل في حال نسي المستخدم اقفال صفحة حسابه البنكي أو بريده الالكتروني في مقهى انترنت.

```
<?php
session_start();
/* تحديد الوقت بالثواني للحد الأقصى المسموح
للمستخدم للجلوس بدون نشاط داخل الصفحة*/
$inactive = 5;
// check to see if $_SESSION["timeout"] is set
if (isset($_SESSION["timeout"])) {
    // calculate the session's "time to live"
    $sessionTTL = time() - $_SESSION["timeout"];
    if ($sessionTTL > $inactive) {
        session_destroy();
        echo' الجلسة الحالية انتهت الرجاء إلدخول مرة أخرى
للنظام';
    }
}
$_SESSION["timeout"] = time();  ?>
```

إعادة توجيه المستخدم redirecting user

في أحيان كثيرة تكون هناك حاجة لإعادة توجيه المستخدم إلى صفحة أخرى . فلو أردنا مثلا منع وصول زوار الموقع لصفحة من صفحات الموقع (بسبب سرية محتوى الصفحة و لتقييد الوصول) فنستطيع تحقيق ذلك بإعادة توجيه متصفح الزائر إلى صفحة أخرى غير الصفحة الحالية التي وصل إليها (بكتابة عنوانها مباشرة في URL) . لإعادة توجيه المستخدم إلى صفحة أخرى نستطيع استخدام إحدى الطرق التالية:

- عن طريق استدعاء الدالة ( )header و التي تقوم بإرسال متصفح الزائر إلى صفحة محددة. لكن في هذه الحالة لابد الانتباه إلى الدالة هذه يجب أن يتم استدعاءها قبل أي وسوم HTML و قبل أي مخرجات تتم عن طريق البرنامج.
- الطريقة الثانية عن طريق إضافة شفرة جافا سكرت ضمن شفر PHP. و في هذه الحالة لدينا الحرية في وضعها في أي مكان داخل شفرة PHP.

المثال التالي يوضح كيفية التعامل مع الدالة ( )header:

```
<?php

if(!isset($_POST['login']))
{
      header("Location:login.php");
}

?>
```

في حال أردنا استخدام شفرة جافا سكربت بدلا من ذلك فنستطيع عمل ذلك كما في المثال التالي:

```
<?php

if(!isset($_POST['login']))
{
      echo "سيتم تحويلك الآن إلى صفحة تسجيل الدخول.";
      echo"
            <script>
            window.location = 'login.php';
            </script>
      ";
}

?>
```

أسئلة نهاية الفصل

١. ما الفرق بين : cookie و session.

٢. كيف نستطيع حفظ ملف cookie و جعله متاحا للمجلد /foo/.

٣. ما الفرق بين إستخدام ( )unset و ( )session_destroy مع الجلسات.

٤. ما الطرق المستخدمة في إعادة توجيه المستخدم و ما الفرق بينها.

هذا الفصل يغطي:

- **كيفية الاتصال بقاعدة البيانات .**
- **كيفية انشاء وتنفيذ الاستعلامات.**
- **كيفية استرجاع السجلات من قاعدة البيانات والعمل معها.**

واحدة من أهم مزايا PHP أنها تدعم العمل مع أغلب أنظمة قواعد البيانات المعروفة و الشهيرة. من هذه النظم الشهيرة يبرز نظام قواعد البيانات MySql. و كما سبق أن اشرنا في أول فصل بأن PHP و MySql من الانظمة مفتوحة المصدر و باستخدامهما معا سيوفر مطور الويب الكثير من الأموال لشراء حقوق الاستخدام من الشركات المنتجة.

كيفية الاتصال بقاعدة البيانات

هناك أكثر من طريقة في PHP للاتصال بقاعدة البيانات MySql. و من هذه الطرق استخدام الإضافة mysqli و التي تسمح لنا بالاتصال بنظام قاعدة البيانات MySql 4.1 فما أعلى. و يجب الإشارة أن هذه الاضافة لن تعمل إلا مع النسخة الخامسة من PHP فما أعلى.

إنشاء اتصال بنظام قواعد البيانات MySql

لإنشاء اتصال بـ MySql نقوم باستدعاء الدالة ( )mysqli_connect و التي تأخذ أربع معطيات كالتالي:

```
mysqli_connect(" hostname ",
               " username ",
               " password ",
               " database " );
```

hostname : اسم الجهاز أو عنوانه الذي يحتوي على MySql (غالبا ما يكون localhost).
username : اسم مستخدم النظام.
password : كلمة مرور مستخدم النظام.
database: اسم قاعدة البيانات.

```
<?php

$conn = mysqli_connect('localhost', 'user30','db456','test');
if($conn === false)
{
      echo "Connection failed! Reason:".mysqli_connect_error();
}

?>
```

الدالة ( )mysqli_connect_error المُستخدمة في المثال السابق تقوم بعرض وصف نصي لآخر خطأ حدث أثناء الاتصال بقاعدة البيانات.

إغلاق الاتصال بقاعدة البيانات closing the connection

بالرغم من أن الاتصال يتم قطعه بمجرد الانتهاء من تنفيذ الشفرة البرمجية إلا انه ينصح إغلاقه برمجيا عند الانتهاء من العمل مع MySql و ذلك عن طريق استدعاء الدالة ( )mysqli_close . هذه الدالة ستقوم بإرجاع true في حال تم اغلاق الاتصال بخلاف ذلك ستقوم بإرجاع false.

```
<?php

$conn = mysqli_connect('localhost', 'user30','db456','test');
if($conn === false)
{
      echo "Connection failed! Reason:".mysqli_connect_error();
}
mysqli_close($conn)
?>
```

تنفيذ الاستعلام executing query

عند التعامل مع MySql كنظام قواعد بيانات فيجب أن يكون لدينا القدرة على كتابة الاستعلامات باستخدام لغة SQL (Structured Query Language). و لتنفيذ أي استعلام بواسطة PHP نقوم باستدعاء الدالة ( )mysqli_query و التي تأخذ معطيين الأول المتغير الذي يحتوي على معلومات الاتصال بقاعدة البيانات و الثاني يحتوي على الاستعلام.
تقوم هذه الدالة بإرجاع false في حال فشلت في تنفيذ الاستعلام. و في حال نجحت في تنفيذ الاستعلام فلها حالتين:

- في حال أن الاستعلام سيقوم بإرجاع سجلات من قاعدة البيانات مثل select , describe ,show and explain ، ففي هذه الحالة ستقوم الدالة بإرجاع مجموعة نتائج MySql Result Set نتعامل معها لاحقا بواسطة الدالة ( )mysqli_fetch_array.
- أما في حال أن الاستعلام لن يقوم بإرجاع سجلات من قاعدة البيانات مثل update , delete , insert ,…. فستقوم الدالة بإرجاع true.

إنشاء جدول crating database table

نستطيع إنشاء جدول باستخدام PHP في خطوات بسيطة كما يلي:

١. إنشاء اتصال بقاعدة البيانات باستخدام الدالة ( )mysqli_connect .
٢. كتابة جملة SQL التي ستقوم بإنشاء الجدول.
٣. تنفيذ الاستعلام باستخدام الدالة ( )mysqli_query .

على افتراض أنه لدينا قاعد بيانات باسم test ، سنقوم الآن بإنشاء جدول لتخزين معلومات عن الكتب.

و قبل البدء في إنشاء الجدول يجب نحدد البيانات التي يجب أن نعرفها عن الكتاب و التي بمعرفتها سنقوم بإنشاء الجدول:

- رقم الكتاب و يجب أن يكون رقما مميزا غير متكرر .
- اسم الكتاب سيكون نصا بطول ١٠٠ حرف.
- اسم المؤلف و سيكون نصا بطول ٦٠ حرفا.
- الناشر و سيكون نصا بطول ٦٠ حرفا.
- تاريخ النشر و نوعه تاريخ.
- سعر الكتاب و نوعه رقم .

الآن لم يتبق علينا إلا معرفة انواع البيانات data types في MySql و الموضحة في الجداول التالية:

النوع	الوصف
char	سلسلة نصية ثابتة دائما يتم ازاحتها ناحية اليمين بواسطة مسافة فارغة للوصول للطول المحدد و بحد أقصى ٢٥٥ حرف.
varchar	سلسلة نصية متغيرة بحد أقصى ٢٥٥ حرف.
text	سلسلة نصية تصل إلى طول ٦٥٥٣٥ حرف.
mediumtext	سلسلة نصية تصل إلى طول ١٦٧٧٧٢١٥ حرف.
longtext	سلسلة نصية تصل إلى طول ٤٢٩٤٩٦٧٢٩٥ حرف.
tinyint	رقم صحيح صغير جدا يتراوح بين -١٢٨ إلى ١٢٧ و من ٠ إلى ٢٥٥ للرقم الصحيح الموجب.
int	رقم صحيح يتراوح بين -٢١٤٧٤٨٣٦٤٨ إلى ٢١٤٧٤٨٣٦٤٧ و من ٠ إلى ٤٢٩٤٩٦٧٢٩٥ للرقم الصحيح الموجب.
float	رقم صغير مع علامة عشرية.
double	رقم كبير مع علامة عشرية.
date	تاريخ بالصيغة YYYY-MM-DD
datetime	تركيبة من التاريخ و الوقت بالصيغة YYYY-MM-DD HH:MM:SS
timestamp	الطابع الزمني YYYYMMDDHHMMSS
time	الوقت بالصيغة HH:MM:SS

ويمكن الاطلاع على تفصيل أكثر عن أنواع البيانات من خلال زيارة الموقع الرسمي لـ MySql التالي
http://dev.mysql.com/doc/refman/5.0/en/data-type-overview.html

الآن سنقوم بإنشاء بكتابة شفرة PHP التي ستقوم بإنشاء الجدول books .

```
<?php
//الخطوة الاولى هي الاتصال بقاعدة البيانات المحددة
$conn=mysqli_connect('localhost','root','','test');
if($conn === false)
{
      echo " حدث خطأ أثناء الاتصال بقاعدة البيانات و السبب".mysqli_connect_error();
}
```

```
else
{
      //جملة sql التي تقوم بإنشاء الجدول
      $query = " CREATE TABLE books(
                        isbn                    varchar(16)      not
null,
                        title                           varchar(100)
null,
                        author              varchar(60)      null ,
                        publisher           varchar(60)      null,
                        publish_date    date      null,
                        price float null,
                        primary key(isbn)
                  )";
      //الآن نقوم بتنفيذ الاستعلام
      $result = mysqli_query($conn,$query);
      if($result)
      {
            echo "تم انشاء الجدول المحدد بنجاح";
      }
      else
      {
            echo    "    و    الجدول    انشاء    أثناء    مشكلة    حدث
السبب:".mysqli_error($conn);
      }
}
mysqli_close( $conn );
?>
```

عند تنفيذ الشفرة السابقة سيكون الناتج.

```
تم إنشاء الجدول المحدد بنجاح
```

و في حال أعدنا تنفيذ الشفرة السابقة فستظهر رسالة الخطأ التالية:

```
حدث مشكلة أثناء انشاء الجدول و السبب 'Table 'books:
already exists
```

و معنى رسالة الخطأ أن الجدول books موجود مسبقا و لا يسمح بإنشائه مرة أخرى.

عرض رسائل الأخطاء printing error messages

في أحيان كثيرة قد نواجه كثير من الأخطاء عند كتابة الاستعلامات. لذلك من الجيد أن نقوم بعرض رسائل الأخطاء التي تنتج عند تنفيذ الاستعلامات لكي يتم التعامل معها. لعرض رسائل الأخطاء نقوم باستدعاء الدالة mysqli_error( ) و التي تأخذ معطى واحد كالتالي:

```
echo mysqli_error($conn);
```

حيث $conn يمثل المتغير الذي يحتوي على معلومات الاتصال بقاعدة البيانات.

إضافة البيانات inserting data

لإضافة بيانات داخل جدول في قاعدة البيانات نقوم باستخدام الأمر insert أحد أوامر SQL و الذي يستخدم لإدراج البيانات.
لكن يجب الحذر عند اضافة البيانات داخل قاعدة البيانات لكي لا تتضرر من قيام المستخدم بإدخال قيم غير صحيحة بهدف الاضرار بقاعدة البيانات.
و لتفادي ذلك يجب علينا القيام بالخطوات التالية:

- إجراء عملية التحقق باستخدام الدالة ( )preg_match (تم شرحها سابقا في الفصل الثامن) أو أي دالة أخرى تقوم بعمل مشابه للتحقق من البيانات المرسلة من خلال النموذج (على افتراض ان البيانات ستُرسل لقاعدة البيانات باستخدام نموذج HTML) .
- بعد ذلك يجب أن نقوم باستخدام الدالة ( )mysqli_real_escape_string لتجاهل أي رموز خاصة يكون لها دلالة في SQL كعلامات الاقتباس.

الآن سنقوم بإضافة سجل جديد داخل الجدول السابق books كما يلي

```
<?php
$conn=mysqli_connect('localhost','root','','test');
if($conn === false)
{
      echo "حدث خطأ أثناء الاتصال بقاعدة البيانات و السبب".mysqli_connect_error();
}
else
{
      $isbn = '978-603-500-045-1';
      $title = 'انشاء تطبيقات انترنت باستخدام PHP';
      $author = 'أحمد العبدالله';
      $publisher = 'دار النشر';
      $publish_date ='2006-05-20';
      $price ='123.25';

      $isbn = mysqli_real_escape_string($conn,$isbn);
      $title = mysqli_real_escape_string($conn,$title);
      $author = mysqli_real_escape_string($conn,$author);
      $publisher = mysqli_real_escape_string($conn,$publisher);
      $publish_date = mysqli_real_escape_string($conn,$publish_date);
      $price = mysqli_real_escape_string($conn,$price);

      $query = "insert into books(isbn,title,author,publisher,publish_date,price)

values('$isbn','$title','$author','$publisher','$publish_date',$price)";

      $result = mysqli_query($conn,$query);
```

```
if($result)
      {
            echo "تمت اضافة البيانات بشكل ناجح";
      }
      else
      {
            echo "حدث مشكلة أثناء اضافة البيانات و السبب:".mysqli_error($conn);
      }
}
mysqli_close( $conn );
?>
```

عند تنفيذ الشفرة السابقة سيكون الناتج.

```
تمت اضافة البيانات بشكل ناجح
```

و في حال أعدنا تنفيذ الشفرة السابقة فستظهر رسالة الخطأ التالية:

```
حدث مشكلة أثناء اضافة البيانات و السبب Duplicate entry:
'978-603-500-045-' for key 'PRIMARY'
```

و السبب لأن الحقل isbn يحتوي على القيد primary key و الذي لا يسمح بتكرار القيم.

استرجاع البيانات retrieving data

بعد أن أصبح الآن لدينا مجموعة من البيانات داخل قاعدة البيانات ، فمن المهم أن نستطيع استرجاع تلك البيانات و من ثم التعامل معها. و لاسترجاع سجلات من قاعدة البيانات نتبع الخطوات التالية:

١. إنشاء اتصال بقاعدة البيانات باستخدام الدالة ( )mysqli_connect .
٢. إنشاء الاستعلام عن طريق كتابة أمر select .
٣. تنفيذ الاستعلام باستخدام الدالة ( )mysqli_query .
٤. نقوم باستخلاص نتائج الاستعلام عن طريق استدعاء الدالة ( )mysqli_fetch_array.
٥. نستخدم حلقة التكرار while لكي نتمكن من قراءة كل السجلات التي قام بإرجاعها الاستعلام.

```
<?php
$conn=mysqli_connect('localhost','root','','test');
if($conn === false)
{
      echo "حدث خطأ أثناء الاتصال بقاعدة البيانات و السبب".mysqli_connect_error();
}
else
{
      $query = "select * from books";
```

```
$result = mysqli_query($conn,$query);
      $numberOfRows = mysqli_num_rows($result);
      if($result)
      {
            if($numberOfRows>0)
            {
                  echo'<table border="1" align="center">';
                  echo'<tr><th>                                    اسم
الكتاب</th><th>المؤلف</th><th>الناشر</th><th>                    تاريخ
النشر</th><th>سعر الكتاب</th></tr>';
                  while($row = mysqli_fetch_array($result))
                  {

      echo'<tr><td>'.$row['title'].'</td><td>'.$row['author'].'<
/td><td>'.$row['publisher'].'</td><td>'.$row['publish_date'].'<
/td><td>'.$row['price'].'</td></tr>';
                  }
            }
            else
            {
                  echo'عفوا لايوجد بيانات للكتب حاليا';
            }

      }
      else
      {
            echo    "    و    البيانات    اضافة    أثناء    مشكلة    حدث
السبب:".mysqli_error($conn);
      }
}
mysqli_close( $conn );

?>
```

عند تنفيذ الشفرة السابقة فسيكون الناتج كما يلي

اسم الكتاب	المؤلف	الناشر	تاريخ النشر	سعر الكتاب
انشاء تطبيقات انترنت باستخدام PHP	أحمد الحبدالله	دار النشر	2006-05-20	123.25
البرمجة المتقدمة باستخدام جافا	محمد العلي	دار الفاروق	2008-11-15	203.75

معرفة عدد نتائج الاستعلام

لمعرفة عدد النتائج (عدد السجلات) لاستعلام ما نستطيع استدعاء الدالة mysqli_num_rows() كما فعلنا في المثال السابق. هذه الدالة تأخذ معطى واحد يمثل المتغير الذي يحتوي على ناتج الاستعلام.

```
mysqli_num_rows( $result )
```

إفراغ الذاكرة من ناتج الاستعلام الأخير

في بعض الحالات قد نحتاج لتنفيذ أكثر من استعلام داخل الشفرة البرمجية الواحدة. لذلك يُنصح في هذا الحالة أن يتم إفراغ الذاكرة من ناتج الاستعلام الأخير لكي يتم استخدامه من قبل استعلام آخر. و نستطيع تحقيق ذلك عن طريق استدعاء الدالة mysqli_free_result( ) و التي تأخذ معطى واحد يمثل المتغير الذي يحتوي على ناتج الاستعلام كما يلي

```
mysqli_free_result( $result )
```

تطبيق عملي

في هذا التطبيق سنقوم بتصميم سجل للزوار .هذا التطبيق سيمسح لزائر بطرح تعليقاته حول الموقع. و قبل أن نبدأ في كتابة شفرة سجل الزوار سنقوم بكتابة شفرة PHP تقوم بإنشاء الجدول guestbook و الذي سيُستخدم لتخزين تعليقات الزوار.
قم بفتح برنامج Notepad++. بعد ذلك سيقوم البرنامج بإنشاء مساحة عمل فارغة. قم بكتابة الشفرة التالية:

```
<?php
$conn=mysqli_connect('localhost','root','','test');
if($conn === false)
{
      echo " و البيانات بقاعدة الاتصال أثناء خطأ حدث
السبب".mysqli_connect_error();
}
else
{
      $query = " CREATE TABLE guestbook(
                        id                    int   primary   key
auto_increment,
                        name        varchar(50) not null,
                        email       varchar(50) not null ,
                        comment     text not null,
                        date        datetime    not null
                  )";
      $result = mysqli_query($conn,$query);
      if($result)
      {
            echo "تم انشاء الجدول المحدد بنجاح";
      }
      else
      {
            echo " و الجدول انشاء أثناء مشكلة حدث
السبب:".mysqli_error($conn);
      }
}
mysqli_close( $conn );

?>
```

بعد الانتهاء من كتابة الشفرة السابقة ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم createTable.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.
٥. قم بتنفيذ الملف عن طريق كتابة العنوان http://localhost/createTable.php.

عند تنفيذ الشفرة السابقة سيكون الناتج.

تم انشاء الجدول المحدد بنجاح

بعد ذلك ، قم بفتح برنامج Notepad++ و من ثم قم بكتابة الشفرة التالية:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta       http-equiv="Content-Type"       content="text/html;
charset=utf-8" />
<title>Untitled Document</title>
<style>
body{
      direction:rtl;
}
table{
      border:1px solid #666;
      background-color:#CCC;
}

#messages{
      border:1px solid #666;
      background-color:#fff;
      padding:2px;
}

#messages th{
      background-color:#0FC;
}

#messages td{
      background-color:#0FF;
}

#addMessage{
      margin-right:400px;
      color:#960;
     font-weight:bold;
}
```

```
hr{
      border:1px solid #666;
      width:800px;
}
</style>
</head>

<body>
<form id="form1" name="form1" method="post" action="">
  <table width="40%" align="center" cellpadding="3">
    <tr>
      <td width="27%">الاسم:</td>
      <td width="73%"><label for="name"></label>
      <input       type="text"       name="name"       id="name"
size="50px"/></td>
    </tr>
    <tr>
      <td>البريد الالكتروني:</td>
      <td><label for="email"></label>
      <input       type="text"       name="email"       id="email"
size="50px"/></td>
    </tr>
    <tr>
      <td>التعليق:</td>
      <td><label for="comment"></label>
      <textarea     name="comment"     id="comment"     cols="45"
rows="5"></textarea></td>
    </tr>
    <tr>
      <td colspan="2"><input type="submit" name="add" id="add"
value="إضافة المشاركة" /></td>
    </tr>
  </table>
</form>
<hr />
<?php
$conn = mysqli_connect('localhost','root','','test');
if(!$conn)
{
      exit(mysqli_connect_error());
}
else
{
      if(isset($_POST['add']))
      {
            echo'<div id="addMessage">';
            $errors = array();
if(empty($_POST['name'])      ||      empty($_POST['email'])   ||
empty($_POST['comment']))
            {
```

```
array_push($errors,'الرجاء تعبئة جميع الحقول');
           }
           if(preg_match('/[\'\/~`\!@#\$%\^&\*\(\)_\-
\+=\{\}\[\]\|;:"\<\>,\.\?\\\]/', $_POST['name']))
          {
             array_push($errors,' الاسم المدخل يحتوي على قيم غير
صالحة.');
          }
           if(!preg_match("/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-
9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})]*$/",$_POST['email']))
          {
             array_push($errors,' القيمة المدخلة للبريد الإلكتروني
غير صالحة.');
          }

           if($errors)
           {
                echo'<ol  class="errorMessage">    الرجاء تصحيح
الأخطاء التالية';
               foreach($errors as $value)
               {
                      echo '<li>'.$value.'</li>';
               }
               echo'</ol>';
           }
           else
           {
                    $name                                          =
mysqli_real_escape_string($conn,$_POST['name']);
                    $email                                         =
mysqli_real_escape_string($conn,$_POST['email']);
                    $comment                                       =
mysqli_real_escape_string($conn,$_POST['comment']);

                    $insertQuery       =       "insert       into
guestbook(name,email,comment,date)

values('$name','$email','$comment',NOW())";

                    $result = mysqli_query($conn,$insertQuery);

                    if($result)
                    {
                         echo' شكرا لك ، لقد تم اضافة تعليقك
بنجاح';
                    }

                   else
                    {
                         echo mysqli_error($conn);
```

```
          }
             }
          echo'</div>';
      }
      $query = "select * from guestbook order by date";
      $result = mysqli_query($conn,$query);
      $numberOfRows = mysqli_num_rows($result);
      echo'<table  align="center"  id="messages"  cellspacing="1"
cellpadding="3">';
      if($numberOfRows>0)
      {
            echo'<tr><th>المرسل</th><th>                    البريد
الالكتروني</th><th>الرسالة</th><th>تاريخ الرسالة</th></tr>';
            while($row = mysqli_fetch_array($result))
            {

      echo'<tr><td>'.$row['name'].'</td><td>'.$row['email'].'</td
><td>'.$row['comment'].'</td><td>'.$row['date'].'</td></tr>';
            }
      }
      else
      {
            echo'<tr><td>  لا يوجد مشاركات في سجل زوار الموقع
حاليا.</td></tr>';
      }
      echo'<table>';
}
?>
</body>
</html>
```

بعد الانتهاء من كتابة الشفرة السابقة ، قم بالنقر على زر حفظ . و من النافذة التالية قم بما يلي:

١. قم بتغيير مكان الحفظ عن طريق الانتقال للمجلد C:\xammp\htdocs.
٢. قم بتسمية الملف باسم guestbook.php .
٣. غير نوع الملف Save as type من القائمة المنسدلة ليصبح All types(*.*) أو PHP Hypertext Preprocessor files.
٤. انقر على الزر Save.
٥. قم بتنفيذ الملف السابق عن طريق كتابة العنوان http://localhost/guestbook.php.

لا يوجد مشاركات في سجل زوار الموقع حاليا.

و عند إضافة سجل جديد سيكون الناتج كما يلي:

شكرا لك , لقد تم اضافة تعليقك بنجاح

المرسل	البريد الالكتروني	الرسالة	تاريخ الرسالة
طلال حامد	talal_43@hotmail.com	نتمنى العمل على تطوير الموقع باستمرار و شكرا لكم.	21:32:46 2012-11-24

و كما هو واضح في الشفرة السابقة فقد قمنا بإجراء عملية التحقق validation على البيانات المُدخلة قبل إضافتها لقاعدة البيانات.

الدالة NOW() المُستخدمة في المثال السابق في جملة insert من دوال SQL و تُستخدم لطباعة التاريخ و الوقت الحالي و لا توضع بين علامات الاقتباس.

أسئلة نهاية الفصل

١. ما هي أنظمة قواعد البيانات التي تدعمها PHP (استخدم شبكة الانترنت للبحث).

٢. كيف نستطيع إنشاء اتصال بقاعدة البيانات (user) .

٣. ما الفرق بين كلا من : ( )mysqli_connect_error و ( )mysqli_error.

٤. كيف نستطيع معرفة عدد نتائج استعلام ما.

٥. ما الفائدة من استخدام الدالة ( )mysqli_free_result .