الفصـــل السادس

لغة برمجة قواعد البيانات على الويب

نستعرض فى هذا الفصل أساسيات أحد أكثر لغات برمجة تطبيقات الإنترنت استخداما فى مواقع الويب، والتى تنتمى إلى قائمة لغات البرمجة مفتوحة المصدر. وإذا كانت الغالبية العظمى من تطبيقات قواعد البيانات على الإنترنت تعتمد بشكل أساسى على منظومة MySQL، فإنه لا غنى عن استخدام لغة PHP للتفاعل مع البيانات المخزنة، من حيث الإدخال أو الاسترجاع.

المحتويات

- مفاهيم البرمجة الأساسية.
- أنواع البيانات والمتغيرات.
- التعبيرات والمعاملات.
- المنطق البوليينى والمعاملات المنطقية.
- بناءات التحكم فى التدفق: البناء إذا.. وإلا.
- الحلقات التكرارية: حلقة بينما.
- الدوال.
- استخدام كود PHP فى ملفات HTML.
- إدراج ملفات خارجية وإعادة استخدام كود PHP.

الأهداف التعليمية

١. التعرف على مفاهيم البرمجة الأساسية.
٢. التمييز بين أنواع البيانات والمتغيرات.
٣. القدرة على إنشاء التعبيرات والمعاملات.
٤. فهم المنطق البوليينى والتمييز بين المعاملات المنطقية.
٥. القدرة على استخدام بناءات التحكم والحلقات التكرارية والدوال.
٦. التمكن من استخدام أكواد PHP داخل ملفات HTML.
٧. القدرة على إدراج ملفات خارجية وإعادة استخدام أكواد PHP.

١- مفاهيم البرمجة الأساسية Basic Programming Concepts

البرمجة هى إحدى المهارات التى تحتاج إلى وقت لتعلمها. كما أن منحنى التعلم ليس خطيا كما قد يعتقد البعض.

إن تعلم البرمجة أيضا مليء بالكثير من لحظات التعجب والدهشة. فالكثير من الأمور المعقدة تتكشف فجأة، وفى أحيان كثيرة يفيد قليل من المثابرة فى التغلب على الأمور الغامضة. وبعد وقت قليل من هذه الممارسة، سوف تعجب بقدراتك التى اكتسبتها فى البرمجة.

لكل لغة برمجة أسلوبها فى كتابة الكود الخاص بها، وهو ما يعرف بال syntax. على الرغم من أن كثير من لغات البرمجة تستخدم أسلوبا متقاربا من بعضها البعض.

على سبيل المثال: عند كتابة الكود يمكنك إدراج بعض التعليقات comments وهى عبارة عن جمل ليس لها أى تأثير على البرنامج. لكن الغرض الأساسى منها هو مساعدة المبرمج – أو أى شخص آخر – فى قراءة الكود والتعامل معه لاحقا.

فى لغة PHP يتم وضع شرطتين مائلتين / / ، أو علامة الشباك #، قبل التعليقات التى تأتى على سطر مفرد. أما التعليقات التى تحتاج أكثر من سطر، فيتم وضع شرطة مائلة ونجمة قبلها /* ، وتنتهى بنجمة وشرطة مائلة */.

```
سطر تعليقات مفرد / /
وهذا أيضا سطر تعليقات مفرد #

/*
وهذا تعليق طويل يحتاج
إلى أكثر من سطر
*/
```

أما سطور الكود الفعلى (غير التعليقات) فى PHP فيجب أن تنتهى بفاصلة منقوطة (;). وسوف ترى مثالا على ذلك فى الخطوة التالية.

٢ – أنواع البيانات والمتغيرات Data Types and Variables

إن أول الأشياء التى يجب تعلمها فى البرمجة هى أنواع البيانات. فى معظم لغات البرمجة توجد ثلاثة أنواع أساسية للبيانات: بيانات رقمية Numbers، بيانات نصية Strings، وبيانات منطقية Booleans.

البيانات الرقمية Numbers

مجرد أرقام صحيحة integers أو أرقام عشرية decimals.

مثلا: ٠ ١٠ ٣.١٤

البيانات النصية Strings

هى ببساطة عبارة عن مجموعات من الحروف letters والفراغات spaces والرموز symbols. وبعبارة أخرى هى مجرد نص text.

فى البرمجة يتم وضع البيانات النصية بين علامتى تنصيص. على سبيل المثال: "هذه هى بيانات نصية". وفى لغة PHP يمكن أيضا استخدام علامات التنصيص المفردة single quotes. `هذه أيضا عبارة عن بيانات نصية`.

البيانات المنطقية Booleans

وهى البيانات التى تكون قيمتها إما صحيحة true أو خاطئة false.

المتغيرات Variable

المتغيرات عبارة عن وسيلة لتخزين المعلومات لاستخدامها لاحقا فى البرنامج. تأخذ المتغيرات اسما name قد يتكون من حروف وأرقام، ويتم تخصيص قيمة value على حسب نوع من أنواع البيانات المذكورة أعلاه. فى لغة PHP تبدأ المتغيرات بعلامة الدولار $ ومن ثم يسهل التعرف عليها داخل الكود.

```
$myName = "شاكر على";
```

فى المثال السابق قمنا بإنشاء متغير باسم $myName وخصصنا له القيمة النصية "على شاكر" باستخدام علامة التساوى =. كذلك يمكن تخزين الأرقام داخل المتغيرات.

```
$a = 3;
$b = 4;
$c = $a + $b;
```

فى المثال السابق قمنا بتخصيص القيمة ٣ للمتغير $a والقيمة ٤ للمتغير $b ثم استخدمنا تعبيرا رياضيا لإعطاء المتغير $c قيمة جمع كل من المتغير $a والمتغير $b أو رقم ٧.

سوف تلحظ أيضا أننا أنهينا كل جملة فى الكود بالفاصلة المنقوطة semicolon. وهذا أمر ضرورى فى لغة PHP للفصل بين تعبير expression وآخر داخل الكود. سوف نتناول التعبيرات فى الخطوة التالية.

٣ – التعبيرات والمعاملات Expressions and Operators

فى عالم البرمجة، يتم تعريف التعبير expression بأنه جزء من الكود ينتج عنه قيمة معينة (رقم – نص – قيمة منطقية).

المعاملات الحسابية Arithmetic operators

فى الخطوة السابقة، استخدمنا تعبيرا بسيطا:

```
$c = $a + $b;
```

هنا التعبير فى اللغة الإنجليزية هو (a زائد b)، وينتج عنه الرقم ٧. علامة الجمع (+) فى هذا التعبير هى المعامل operator. والمتغيرات $a و $b هى المتعاملات operands، أو القيم التى يتم التعامل عليها.

هناك أربعة أنواع أساسية من المعاملات الحسابية:

+ معامل الجمع plus operator: يجمع قيمتين.

– معامل الطرح minus operator: يطرح قيمة من أخرى.

/ معامل القسمة divide operator: يقسم قيمة على أخرى.

* معامل الضرب multiply operator: يضرب قيمة فى أخرى.

معاملات المقارنات Comparison operators

تقوم هذه المعاملات بمقارنة قيمة بأخرى، وينتج عنها قيمة منطقية (صح أو خطأ). فالجملة رقم ٥ أقل من رقم ٦ صحيحة، بينما الجملة رقم ٥ يساوى رقم ٦ خاطئة.

```
$a = 5;
$b = 10;
$c = 12;

// the following expressions are all false:
$a > $b; // read “a is greater than b”
$c < $a; // read “c is less than a”
$a = $b; // read “a is equal to b”

// the following expressions are all true:
$a < 100;
$b > $a;
$b != $a; // read “b does not equal a”
```

٤ – المنطق البوليينى والمعاملات المنطقية

Boolean Logic and Logical Operators

تحدثنا آنفا عن القيم البوليينية Boolean values والتى تكون قيمتها إما صحيحة أو خاطئة. فالمنطق البوليينى أحد أهم أركان البرمجة، ويقصد به المنطق الذى يحدد صحة أو خطأ التعبيرات expressions.

ونحن كثيرا ما نستخدم المنطق البوليينى فى حياتنا اليومية: "أنا اسمى أحمد، وأعمل طبيبا." إذا كان الشخص الذى يقول ذلك يدعى أصلا محمد، ويعمل مهندسا، سوف تكون العبارة السابقة خاطئة.

وإذا ما فحصنا العبارة السابقة بدقة، سوف نجد أنها تتكون من تعبيرين أو قيمتين: "أنا اسمى أحمد" و "أنا أعمل طبيبا." كل تعبير / قيمة منهما – بطبيعته – إما أن يكون صحيحا أو خاطئا. وحرف الواو (and فى اللغة الإنجليزية) فى الجملة هو المعامل البوليينى. ووظيفته تكوين تعبير واحد بالجمع بين تعبيرين مستقلين.

كذلك فى استخدامنا للمنطق فى حياتنا اليومية، إذا كان الشخص اسمه أحمد، لكنه لا يعمل طبيبا، يمكننا أن نقول أن هذا التعبير صحيح بنسبة ٥٠٪ أو أن التعبير نصفه صحيحا half true. لكن فى المنطق البوليينى، لا يمكن تجزئة الحقيقة، فالتعبير إما صحيحا أو خاطئا.

المعاملات المنطقية Boolean operators

"&&" And (و): التعبير صحيح إذا كان كلا المتعاملين both operands صحيح.

"||" Or (أو): التعبير صحيح إذا كان أحد المتعاملين – على الأقل – صحيح.

"xor" Exclusive OR (أو الحصرية): التعبير صحيح إذا كان أحد المتعاملين – وفقط أحد المتعاملين – صحيح.

"!" علامة التعجب: التعبير صحيح إذا كان المتعامل المفرد single operand خطأ.

```
$a = true;
$b = false;
$c = true;

// the following expressions are all false:
! $a; // read “not a”. Since “a” is true, “not a” is false
$a && $b; // read “a and b”
$a xor $c; // read “a or c, but not both”

// the following expressions are all true:
! $b;
$a && $c;
$a || $b; //read “a or b”
```

٥ – بناءات التحكم فى التدفق: البناء إذا.. وإلا

Flow Control Structure: if / else Structure

يعتبر التحكم فى تدفق البرنامج من الأركان الهامة جدا فى البرمجة. ففى العادة، يتم تنفيذ كود PHP من أعلى لأسفل. لكننا أحيانا نريد من البرنامج أن يتخذ قرارا معينا، وهذا يتطلب أن يتدفق التنفيذ بشكل مختلف وفقا للمتغيرات. ولكى نقوم بذلك يجب أن نستخدم بناءات التحكم فى تدفق البرنامج.

البناء إذا.. وإلا if / else Construct

خذ على سبيل المثال الجملة التالية: إذا كان لديك كوبون تخفيضات coupon فإن سعر هذا المنتج ٥٠ جنيها. فى هذه الجملة يمكننا تحديد تعبيرين: "إذا كان لديك كوبون تخفيضات"، وهو تعبير منطقى يحتمل أن يكون صحيحا أو خاطئا. والتعبير الآخر "سعر هذا المنتج ٥٠ جنيها"، وهو يحدد قيمة السعر.

تقوم هذه الجملة بعمل قرار معين: السعر فقط ٥٠ جنيها إذا كان هناك كوبون تخفيضات. لعمل هذا القرار فى برنامج PHP، نقوم باستخدام البناء إذا( ) if( ) construct.

```
$haveCoupon = true;
if( $haveCoupon )
{
$itemPrice = 50;
}
```

تتحقق جملة ($haveCoupon) if من المتغير $haveCoupon، فإذا كان صحيحا، تقوم بتنفيذ الكود الواقع بين الأقواس المتعرجة curly braces. أما إذا كان المتغير خاطئا، فإن الكود بين الأقواس المتعرجة يتم تجاهله.

إذا أردت كتابة الكود بحيث يتم تحديد السعر ٥٠ جنيها إذا كان الشخص يحمل كوبون تخفيضات، أو أن يكون السعر ٧٥ جنيها فى حالة عدم وجود الكوبون، يجب أن تستخدم البناء (وإلا) else construct، وهو النصف الثانى الاختيارى optional من البناء إذا( ) if( ) construct.

```
$haveCoupon = true;
if( $haveCoupon )
{
$itemPrice = 50;
}
else
{
$itemPrice = 75;
}
```

فى الجزء التالى سوف نتعلم كيفية استخدام الحلقات التكرارية Loops وهى نوع آخر من بناءات التحكم.

٦ – الحلقات التكرارية : حلقة بينما Loops: the While Loop

أحيانا عند كتابة الكود، تريد تنفيذ أفعال معينة باستمرار. وللقيام بذلك فى PHP هناك عدد من البناءات التكرارية المختلفة. وبشكل محدد، تسمح الحلقة التكرارية بتكرار تنفيذ جزء معين من الكود حتى يتحول تعبير معين إلى تعبير **صحيح true**. كل نوع من الحلقات التكرارية يعمل بشكل مختلف؛ ومن ثم يستخدم لتنفيذ أشياء مختلفة.

حلقات بينما التكرارية While Loops

تعتبر جملة بينما while أبسط أشكال الحلقات التكرارية. وتستخدم لتكرار جزء معين من الكود بشكل مستمر حتى يتحول تعبير معين إلى تعبير صحيح.

```
$count = 0;
while( $count < 10 )
{
$count++;
}
```

فى المثال السابق تزيد الحلقة التكرارية من قيمة المتغير $count حتى تتساوى قيمته برقم ١٠. يبدأ المتغير $count وقيمته صفرا؛ ومن ثم يكون التعبير "$count < 10" صحيحا، وهذا سوف يؤدى إلى تكرار الكود داخل الأقواس المتعرجة.

وعندما يتحول التعبير داخل الحلقة التكرارية إلى تعبير خاطئ، فإن الكود داخل الأقواس المتعرجة لن ينفذ أبدا.

التعبير $count++ هو تعبير حسابى يزيد قيمة المتغير $count بمعدل واحد (١). ومن ثم، فى نهاية الحلقة التكرارية الأولى، تصبح قيمة المتغير $count تساوى واحد، والحلقة الثانية ٢، والثالثة ٣، وهكذا. أما فى نهاية الحلقة التكرارية العاشرة، فإن قيمة المتغير $count سوف تساوى رقم ١٠. وبالتالى يتحول التعبير $count < 10 إلى تعبير **خاطئ false**، ثم تتوقف الحلقة التكرارية.

٧ – الدوال PHP Functions

فى البرمجة، قد تستدعى الضرورة أحيانا تنفيذ مهمة معينة فى أماكن مختلفة من البرنامج؛ ومن ثم فإن كتابة نفس الكود أكثر من مرة سوف يسبب مشاكل كثيرة، ويحتاج إلى وقت وجهد من المبرمج. وهو ما عالجته الدوال functions.

يمكن تعريف الدالة function بأنها مجموعة من أسطر الكود، لها اسم معين، ويمكن استدعاؤها فى أى جزء معين فى البرنامج. تقوم الدوال بأداء مهمة محددة، ويمكن أن تسترجع قيمة value. وبالتالى تعتبر الدوال تعبيرات expressions فى حد ذاتها.

الكثير من الدوال مبنية ضمن لغة PHP، وتشكل جزءا كبيرا من اللغة. من بين هذه الدوال، الدالة ( ) sqrt، والتى تستخدم لحساب الجذر التربيعى square root لرقم معين.

```
// this code will give $b a value of 3, the square root of 9
$b = sqrt( 9 ) ;
```

يتم كتابة جميع الدوال بالشكل التالى: الاسم (الخصائص) Name (arguments)، حيث يتبع اسم الدالة بهلاليتين. ويتم وضع جميع خصائص الدالة بين الهلاليتين، مع وضع فاصلة comma بين كل خاصية وأخرى. والخصائص arguments عبارة عن قيم تحتاجها الدالة، لكى تتمكن من القيام بمهمتها، وهى بالتالى تختلف من دالة لأخرى.

على سبيل المثال: تتطلب الدالة ( ) sqrt رقم مفرد. وتقوم الدالة بعد ذلك بحساب الجذر التربيعى لذلك الرقم. ونظرا لأن (9) sqrt تعطى القيمة ٣، فإن الكود السابق يعطى نفس التأثير إذا ما كتبناه بالشكل التالى: 3 = $b.

إلى جانب الدوال المبنية داخل لغة PHP، يمكنك أيضا كتابة دوال أخرى بنفسك، وهو ما نناقشه فى الجزء التالى.

٨- إنشاء الدوال فى Creating PHP Functions .PHP

فى بعض الأحيان قد يحتاج المبرمج إلى كتابة دوال خاصة بنفسه داخل كود البرنامج.

يتم كتابة الدوال الخاصة فى لغة PHP بكتابة إعلان الدالة function declaration. وهو يبدأ بكلمة **function** متبوعا باسم الدالة، ثم قائمة الخصائص، وبعد ذلك يأتى الكود الخاص بالدالة.

سوف نقوم الآن بإنشاء دالة، ونطلق عليها **( ) square**. وهى تسترجع مربع الرقم (الرقم مضروبا فى نفسه، وهو عكس الجذر التربيعي).

```
function square( $number )
{
$newNumber = $number * $number;
return $newNumber;
}
```

فى المثال السابق: نستخدم المتغير **$number** داخل الدالة للإشارة إلى القيمة التى أخذها كخاصية argument للدالة. ثم نخصص القيمة التربيعية للمتغير $number ($number * $number) لمتغير جديد: **$newNumber**.

بعد ذلك استخدمنا جملة **return** لإرسال قيمة المتغير $newNumber مرة أخرى إلى الجزء من الكود الذى استدعينا منه الدالة.

```
// this code will give $a the value 9, the square of 3
$a = square( 3 ) ;
```

عند كتابة إعلانات الدوال فى PHP، غالبا ما يضع المبرمجون هذه الإعلانات فى بداية كود البرنامج، وذلك لتجنب أى لبس مع بقية الكود.

```
function sampleOne()
{
// code
}
function sampleTwo()
{
// code
}
```

```
/* PHP script starts here */
$a = true;
$b = false;
//... the rest of the script goes here
```

٩- الملخص

قمنا فى هذا الجزء بتغطية أساسيات البرمجة، مع بعض أساسيات لغة PHP. والكثير من الأشياء التى تعلمناها هنا لا تنطبق فقط على لغة PHP، وإنما على الكثير من اللغات الشبيهة، مثل C++ و JavaScript و Java وغيرها. من بين مبادئ البرمجة التى تعلمناها معا:

- أنواع البيانات data types (رقمية، نصية، منطقية)
- المتغيرات variables
- التعبيرات expressions
- المعاملات operators (الحسابية، المقارنات، المنطقية)
- المنطق البوليينى Boolean logic
- تدفق البرنامج program flow (جمل if / else والحلقات التكرارية loops)
- الدوال functions

هذه المفاهيم تقريبا هى ذاتها فى الكثير من لغات البرمجة. لكن الكود يختلف قليلا من لغة لأخرى. وبمجرد اجتيازك لمشقة فهم هذه الأساسيات، فإن تعلم البرمجة فى لغة أخرى سوف يكون أسهل كثيرا.

الآن أنت تعرف ما يكفى لإنشاء أول صفحة ويب بلغة PHP.

١٠- استخدام كود PHP فى ملفات HTML

١- حقول PHP داخل صفحات HTML

PHP Tags Inside HTML Pages

لكى تستخدم PHP فى ملفات HTML (أو صفحات الويب)، يجب أن تعرف أولا كيف تقوم بإنشاء ملف HTML باستخدام محرر نصوص text editor، وبالطبع يجب أن تعرف كيفية كتابة كود HTML.

محرر النصوص هذا عبارة عن برنامج يقوم بحفظ النصوص بدون أية تنسيق formatting. ومن هنا يجب أن نلحظ أن أدوات تجهيز النصوص مثل Microsoft Word لا تحفظ الوثائق كنص بدون تنسيق plain text بشكل افتراضى.

إن صفحة ويب بسيطة جدا، وتحتوى فقط على جملة "Hello World!" سوف تتكون من كود HTML التالى:

```
<html>
<body>
Hello World!
</body>
</html>
```

إذا كنت تعرف كيفية إعداد صفحات HTML، وتفهم المثال السابق، تستطيع البدء فى كتابة كود PHP داخل صفحة الويب. وإلا يجب أن تتعلم أولا المزيد حول كيفية إعداد وثيقة HTML.

تستخدم PHP حقولا خاصة تشبه كثيرا حقول HTML حتى يستطيع الخادم أن يتعرف على الجزء الذى يحتوى على كود PHP. مثل حقول HTML، هناك حقل لبداية ونهاية كود PHP:

```
<?php
?>
```

يتم تجهيز أى شيء بين حقل البداية opening tag (<?php) وحقل النهاية closing tag (?>) بواسطة الخادم باعتباره كود PHP.

يعتبر الكود التالى صحيحا داخل وثيقة HTML:

```
<html>
<body>
<?php
echo( "Hello World!" ) ;
?>
</body>
</html>
```

هنا بدلا من الاكتفاء بكتابة "Hello World!" داخل وثيقة HTML، نقوم بإخراج النص "Hello World!" باستخدام الوظيفة ( ) echo وهى إحدى وظائف functions لغة PHP. وأيا كان النص الذى نضعه داخل الأقواس الهلالية التابعة لأمر ( ) echo فإنه يتم استبداله وإرساله على أنه كود HTML.

٢ – تقديم ملفات PHP للزائرين Serving PHP Files to Visitors

يتم تقديم ملفات HTML العادية بالامتداد.html أو الامتداد.htm وعلى الجانب الآخر يتم تقديم ملفات PHP مصحوبة بالامتداد.php وذلك بهدف تمييزها عن ملفات HTML العادية.

يمكنك حفظ وثيقة HTML على الكومبيوتر، وعرضها فى متصفح الويب لرؤية نتائج أية تغييرات قمت بها. على الرغم من ذلك يتطلب كود PHP أن يكون لديك خادم ويب على الكومبيوتر، حتى تستطيع تجهيز الوثيقة، قبل إرسالها إلى برنامج متصفح الويب.

لعرض مخرجات ملف PHP، سوف تحتاج خادم مزود بدعم PHP وتحميل الملف على هذا الخادم.

قم بكتابة النص التالى واحفظه كنص عادى plain text فى ملف باسم hello.php ثم ضعه على الخادم.

```
<html>
<body>
<?php
echo( "Hello World!" ) ;
?>
</b>
</html>
```

وبعد ذلك افتح الملف بالطريقة التى تتبعها لفتح أية صفحة ويب مخزنة على الكومبيوتر. سوف تحصل على النتيجة التالية:

Hello World!

وإذا قمت بعرض مصدر صفحة الويب (من خلال القائمة View > Source أو View > Page Source) فى برنامج المتصفح، فإن الكود الذى تراه ليس كود PHP لكن HTML. سوف تكون النتيجة هكذا:

```
<html>
<body>
Hello World!
</body>
</html>
```

يرجع السبب فى عدم رؤيتك لكود PHP أن الخادم الذى يحتوى على صفحة الويب قد قام بتجهيز كود PHP وإرساله إلى المتصفح ككود HTML، وليس ملف PHP الأصلى. وهذا يعنى أن الزائرين لصفحة الويب لن يتمكنوا أبدا من رؤية كود PHP.

٣ – ما هى الخطوة القادمة؟

الآن بعد أن تعرفت على كيفية إدراج كود PHP داخل وثيقة HTML وتقديمها لزائرى الموقع على الويب، يمكنك أن تبدأ فى إنشاء مواقع ويب ديناميكية.

ولكى تبدأ هذه الخطوة، يجب أن تتعلم أولا أساسيات البرمجة فى PHP، وأيضا كيفية استخدام قواعد بيانات MySQL لإضافة المزيد من التفاعلية والديناميكية على موقع الويب.

١١- إدراج ملفات خارجية وإعادة استخدام كود PHP

١ – إدراج ملفات خارجية باستخدام PHP

Including Files with PHP

عند بناء موقع ويب متعدد الصفحات، قد تريد أحيانا وضع كود معين فى صفحات متعددة. على سبيل المثال: قد تريد وضع مجموعة الروابط links الخاصة بالإبحار فى الموقع site navigation فى الجزء العلوى من كل صفحة، أو قد تريد وضع بيان حق المؤلف copyright فى الجزء السفلى من كل صفحة.

فى لغة PHP يمكنك استخدام الدالة ( ) include لإدراج ملفات نصية text أو ملفات PHP فى الموقع. على سبيل المثال: إذا كنت تريد إدراج بيان حق المؤلف copyright notice فى أسفل كل صفحة، يمكنك كتابة الكود التالى، وحفظه فى ملف نصى plain text file باسم copyright.txt.

```
<p>This page <strong>Copyright © 2000</strong> Ali Shaker </p>
```

وبعد ذلك، وفى كل صفحة تريد إدراج هذا البيان بها، يمكنك كتابة كود PHP التالى:

```
<?php
include( “copyright.txt” ) ;
?>
```

مهمة الكود السابق هى وضع بيان حق المؤلف فى كل صفحة نيابة عنك. والميزة الحقيقية من استخدام الوظيفة ( ) include بهذه الطريقة هى أنه عندما نريد تحديث بيان حق المؤلف هذا، نقوم فقط بتغيير ملف واحد، وهو copyright.txt، وسوف ينعكس التغيير على جميع الصفحات التى يوجد بها هذا البيان. وهذا يساعد على الاحتفاظ بصفة الثبات consistency فى صفحات الموقع.

٢ – إدراج ملفات PHP. .Including PHP Files

يمكن للدالة ( ) include أن تستخدم أيضا لإدراج ملفات PHP فى ملف ما، بشكل يسمح لك بإعادة استخدام كود PHP.

دعنا نفترض مثلا أننا نريد وضع نص معين فى كل صفحة من صفحات الموقع، يبين تاريخ اليوم.

Welcome to my site! Today is Wednesday, July 17!

للقيام بذلك، يمكنك إنشاء ملف PHP باسم message.php بحيث يحتوى على الكود التالى:

```
<?php
$date = date( "l, F j" ) ;
echo( "Welcome to my site! Today is $date!" ) ;
?>
```

هنا تمر الدالة ( ) date بجملة خاصة تحد لها شكل التنسيق format الذى تريده للتاريخ. فبالنسبة لحرف "l" فهو يمثل يوم من أيام الأسبوع (Wednesday) فى هذه الحالة، و "f" يمثل الشهر، أما "j" فيمثل يوم من أيام الشهر. يمكنك قراءة المزيد حول الدالة ( ) date والجمل الخاصة بتنسيقها formatting فى موقع PHP.

تقوم الدالة ( ) date بعرض التاريخ – بالتنسيق المحدد – فى شكل جملة نصية، وبعد ذلك تقوم بتخزينه فى المتغير $date.

وبعد ذلك نقوم باستخدام الدالة ( ) echo لطباعة الرسالة التى أردناها فى البداية. سوف يتم استبدال المتغير $date بقيمته النصية؛ وهى فى هذه الحالة: "Wednesday, July 17!".

والآن.. فى كل صفحة نريد إدراج رسالة الترحيب هذه، نقوم بوضع النص التالى:

```
<?php
include( "message.php") ;
?>
```

هذا ويمكنك من عرض الرسالة فى أى عدد من الصفحات كما تريد. كما يمكنك تحديث الرسالة فى جميع الصفحات، عن طريق تحديث الملف الأصلى فقط، وهو هنا message.php.

تدريبات عملية

١. مستخدما الإنترنت، ابحث عن الاسم الكامل لـ PHP؟

٢. ما هى المتغيرات؟

٣. اشرح بإيجاز أنواع البيانات التى تستخدمها لغة PHP؟

٤. تبدأ أسطر التعليقات برمز..... فى حين تنتهى أسطر الكود بعلامة.....

٥. ما هى وظيفة الكود التالى:

```
$BorrowerStudent = true;
if( $BorrowerStudent )
{
$CircPeriod = $currentday+14;
}
```

٦. يمكن تعريف..... بأنها مجموعة من أسطر الكود، لها اسم معين، ويمكن استدعاؤها فى أى جزء معين فى البرنامج، حيث تقوم بأداء مهمة محددة، ويمكن أن تسترجع قيمة value.

٧. ما الفرق بين الدالة ( )echo والدالة ( )print؟

٨. حدد صحة أو خطأ العبارات التالية:

أ- تبدأ أسماء المتغيرات بعلامة $.

ب- يتم كتابة الدوال الخاصة فى لغة PHP بكتابة إعلان الدالة، وهو يبدأ بكلمة Create Function متبوعا باسم الدالة.

ج- يتم تقديم ملفات PHP مصحوبة بالامتداد.php

٩. فى لغة PHP يمكنك استخدام الدالة..... لإدراج ملفات نصية text أو ملفات PHP فى الموقع.

١٠. حدد مهمة الكود التالى:

```
<?php
include( "message.php") ;
?>
```

المصادر

1. Julie Meloni. ***PHP essentials [electronic resource].*** 2nd ed. Boston, MA : Premier Press, c2003.
2. Sterling Hughes. ***PHP developer's cookbook.*** Indianapolis, Ind. : Sams, 2001.
3. Julie C. Meloni. ***PHP fast & easy Web development [electronic resource].*** 2nd ed. [Cincinnati, Ohio?] : Premier Press, c2002.