جامعة طرابلس
كلية تقنية المعلومـــــات
قسم نظم المعلومات

المقرر الدراسي:

تطوير التطبيقات

Application development

ITIS311

المحاضرة (2)

اعداد: أ. فاطمة بشير القاضي
F.alqadhi@uot.edu.ly

PHP Syntax - PHP بناء جملة

يمكن وضع كود الـ PHP في أي مكان في المستند. يبدأ كود الـ PHP بـ <?php وينتهي بـ ?>:

```
<?php
  // PHP code goes here
 ?>
```

امتداد الملف الافتراضي لملفات PHP هو ".php".

يحتوي ملف PHP عادةً على علامات HTML وبعض أكواد الـ PHP.

هنا، لدينا مثال لملف PHP بسيط، عبارة عن برنامج PHP يستخدم وظيفة PHP المضمنة "echo" لإخراج النص "Hello World!" على صفحة ويب:

```
<html>
 <body>
    <h1>My first PHP page</h1>
    <?php echo "Hello World!"; ?>
</body>
</html>
```

بناء جملة PHP Syntax - PHP

ملاحظة: جميع جمل الـ PHP تنتهي بفاصلة منقوطة (؛).

حساسية حالة الاحرف case-sensitive في الـ PHP

الكلمات الرئيسية keywords في PHP (على سبيل المثال، if، else، while، echo، وما إلى ذلك)، والفئات classes، والدوال functions، والدوال المعدة من قبل المستخدم user-defined functions ليست حساسة لحالة الأحرف. في المثال أدناه، جميع عبارات العرض الثلاثة أدناه متساوية وصحيحة:

```
<html>
 <body>
   <?php ECHO "Hello World!<br>";
      echo "Hello World!<br>";
      EcHo "Hello World!<br>";
   ?>
</body>
 </html>
```

PHP Syntax - PHP بناء جملة

ملاحظة: في الـ PHP جميع أسماء المتغيرات حساسة لحالة الأحرف case-sensitive.

في المثال ادناه؛ عند التنفيذ فقط العبارة الأولى ستعرض قيمة المتغير $color، وذلك لأنه يتم التعامل مع $color و$COLOR و$coLOR كثلاثة متغيرات مختلفة:

```
<html>
 <body>
  <?php
    $color = "red";
    echo "My car is " . $color . "<br>";
    echo "My house is " . $COLOR . "<br>";
    echo "My boat is " . $coLOR . "<br>";
 ?>
</body>
</html>
```

PHP Syntax - PHP بناء جملة

هناك طريقتين لكتابة أكواد php و HTML معًا إما باستخدام جملة الطباعة أو بغلق وسم كود php والبدأ في كتابة أكواد HTML ومن ثم إعادة فتح وسم php لتكملة كتابة أكواد php

```
<?php  $var1 = 'value1';
   $var2 = 'value2'; ?>
<html dir="rtl">
 <head>
    <title> التمرين  </title>
 </head>
<body>
   <div style="color:#F00;">
   <?php echo $var1; ?>  </div>
   <div style="color:#00F; font-size:28px;">
   <?php echo $var2; ?>  </div>
</body>
</html>
```

```
<?php $var1 = 'value1';
$var2 = 'value2';
echo ' <html dir="rtl">
 <head>
  <title> التمرين </title>
 </head>
<body>
 <div style="color:#F00;">  '.$var1.'  </div>
 <div style="color:#00F; font-size:28px;"> '.  $v
ar2.‘ </div>
</body>
</html> ';
?>
```

Comments in PHP - PHP التعليقات في

التعليق في اكواد الـ PHP هو سطر لا يتم تنفيذه كجزء من البرنامج.

والغرض الوحيد منه هو أن يقرأه اي شخص لفهم الكود بشكل افضل.

نستخدم التعليقات :

- ✓ لجعل الآخرين يفهمون تعليماتك البرمجية.
- ✓ لتذكّير نفسك بما فعلته: يمكن أن تذكرك التعليقات بما كنت تفكر فيه عندما كتبت الكود.
- ✓ لمنع تنفيذ أسطر التعليمات البرمجية.

تدعم لغة PHP عدة طرق للتعليق:

- // This is a single-line comment
- # This is also a single-line comment
- /* This is a
  multi-line comment */

Comments in PHP - PHP تابع التعليقات في

- تعليقات سطر واحد تبدأ التعليقات ذات السطر الواحد بـ // أو #، سيتم تجاهل أي نص بين // أو # ونهاية السطر (لن يتم تنفيذه).

Ex1:

```
// Outputs a welcome message:
 echo "Welcome Home!";
```

Ex2:

```
echo "Welcome Home!"; # Outputs a welcome message
```

- التعليقات متعددة الأسطر تبدأ بـ /* وتنتهي بـ */، سيتم تجاهل أي نص بين /* و*/.

Ex1:

```
/* The next statement will print a welcome message */
echo "Welcome Home!";
```

Ex2:

```
$x = 5 /* + 15 */ + 5;
echo $x;
```

PHP Variables - PHP المتغيرات في

المتغيرات : هي "حاويات" لتخزين المعلومات.

إنشاء (الإعلان) عن متغيرات PHP

في PHP، يبدأ المتغير بالعلامة $، متبوعًا باسم المتغير:

```
$txt = “Hello world”;
$x = 5;
$y = 10.2;
```

يمكن أن يكون للمتغير اسم قصير (مثل $x و $y)أو اسم أكثر وصفًا ($age, $carname, $total_vol)

قواعد متغيرات PHP:

- ✓ يبدأ المتغير بالعلامة $، متبوعًا باسم المتغير.
- ✓ يجب أن يبدأ اسم المتغير بحرف أو بشرطة سفلية لا يمكن أن يبدأ اسم المتغير برقم.
- ✓ يمكن أن يحتوي اسم المتغير على أحرف أبجدية و ارقام وشرطات سفلية (A-z، 0-9، و _) فقط.

تابع المتغيرات في PHP Variables - PHP

«تعتبر PHP لغة فضفاضة»

في المثال أعلاه، لاحظ أنه لم يكن علينا إخبار PHP بنوع البيانات الذي ينتمي إليه المتغير، حيث تقوم PHP تلقائيًا بربط نوع البيانات بالمتغير اعتمادًا على قيمته.
نظرًا لعدم تعيين أنواع البيانات بالمعنى الدقيق للكلمة، يمكنك القيام بأشياء مثل إضافة سلسلة إلى عدد صحيح دون التسبب في خطأ.

```
$x = 5;    // $x is an integer
$y = "John";    // $y is a string
 echo $x;
 echo $y;
```

تدعم لغة PHP أنواع البيانات التالية:

(String, Integer, Float _also called double_ , Boolean, Array, Object, NULL, Resource)

ملاحظة : القيمة المنطقية false والقيمة الفارغة null لاتظهر في الطباعة والقيمة المنطقية true يطبع عوضًا عنها 1.

تابع المتغيرات في PHP Variables - PHP

الدالة **isset()**

تستخدم هذه الدالة للتعرف على المتغير هل موجود ومسند له قيمه أم لا، و تـــعيد القيمة true في حال وجود المتغير ووجود قيمة مسنده له وتعيد القيمة false في حالة عدم وجود المتغير أو عدم وجود قيمة مسنده له أو أن تكون القيمة المسندة للمتغير هي القيمة الفارغة null

```
<?php
$var1 = 12;
$var2;
If  (isset($var1))
    echo '<br>$var1    is          set';
If  (isset($var2))
    echo '<br>$var2    is          set';
If  (isset($var3))
    echo '<br>$var3    is          set';  ?>
```

String Variables - PHP المتغيرات النصية في

تستخدم المتغيرات النصية للقيم التي تحتوي على أحرف أو كلمات أو رموز.

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

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

```
<?php
$txt= "مرحباً بكم";
echo $txt; ?>
```

ملاحظة: يوجد رابطة واحدة مستخدمة مع PHP لربط مخرجات جمل الطباعة معاً وهي (.) و تستخدم لربط قيمتين نصيتين مع بعضهم أو متغيرين مع بعض أو قيمة نصية ومتغير .

EX1:

```
<?php
 $txt1="مرحباً بكم في";
$txt2="كورسPHP";
echo $txt1 . " " . $txt2; ?>
```

EX2:

```
<?php
echo "Welcome to". " " . "PHP course"; ?>
```

دوال التعامل مع المتغيرات النصية في PHP
Functions of String Variables

- الدالة **strlen()**

تُستخدم الدالة strlen() لتحديد طول السلسة النصية أي عدد أحرف النص المستخدمة. تستخدم هذه الدالة عادة مع الحلقات أو بعض الدوال الاخرى خاصة عندما يكون من المهم معرفة متى ستتوقف الحلقة .

Ex:

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

- الدالة **strpos()**

تُستخدم الدالة **strpos()** للبحث عن نص أو أحرف ضمن السلسة النصية الواحدة. ان تم إيجاد النتيجة سيتم عرض مكان الكلمة أو الحرف المطابقة للنتيجة . ولكن إن لم يتم إيجاد نتيجة عندها ستكون النتيجة FALSE.

Ex:

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

دوال التعامل مع المتغيرات النصية في PHP
Functions of String Variables

- الدالة **الدالة str_replace()**

تستخدم الدالة **str_replace()** لاستبدال نص داخل السلسة النصية بأخر.
تأخذ هذه الدالة ثلاثة وسائط: أولها هو السلسلة التي سنبحث عنها، وثانيها هي السلسلة التي سيتم الاستبدال بها، وثالثها هي السلسلة التي سنبحث فيها. وستعيد هذه الدالة نسخةً من السلسلة الأصلية مع استبدال جميع أماكن ورود السلسلة التي يتم البحث عنها.

Ex:

```
<?php
 $myString = "It was the best of times, it was the worst of times,";
 echo str_replace( "times", "bananas", $myString ); ?>
```

دوال التعامل مع المتغيرات النصية في PHP
Functions of String Variables

- **التعامل مع حالات الأحرف**

ستأخذ الدوال الآتية وسيطًا وحيدًا الذي هو سلسلةٌ نصيّةٌ:

strtolower () جعل كل أحرف السلسلة النصية بأحرفٍ ذات الحالة الصغيرة (lowercase).

strtoupper() جعل كل أحرف السلسلة النصية بأحرفٍ ذات الحالة الكبيرة (uppercase).

ucfirst() جعل أول حرف من السلسلة النصية حرفًا بالحالة الكبيرة.

lcfirst() جعل أول حرف من السلسلة النصية حرفًا بالحالة الصغيرة.

ucword() جعل أول حرف من كل كلمة في السلسلة النصية حرفًا بالحالة الكبيرة.

الرموز الحسابية الجبرية
Arithmetic Operators

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

تستخدم PHP العديد من الرموز الحسابية .

رموز التعيين الحسابية
Assignment Operators

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

اختصار PHP لرموز التعيين الحسابية.

عوامل الزيادة والنقصان
Increment / Decrement Operators

تدعم PHP عوامل الزيادة والنقصان السابقة واللاحقة المماثلة للعوامل المستخدمة في لغة C.
ملاحظة: تؤثر عوامل الزيادة والنقصان على الأعداد والسلاسل النصية فقط، ولا تتأثر المصفوفات والكائنات والموارد بهذه العوامل. لا يؤثر عامل النقصان على قيم NULL ولكن استخدام عامل الزيادة يعود بالنتيجة 1.

المثال	الاسم	التأثير
++$a	الزيادة السابقة	زيادة قيمة المتغير $a بمقدار 1، ثم إعادة القيمة الجديدة للمتغير $a .
$a++	الزيادة اللاحقة	إعادة قيمة $a ، ثم زيادة القيمة بمقدار 1.
--$a	النقصان السابق	إنقاص قيمة المتغير $a بمقدار 1، ثم إعادة القيمة الجديدة للمتغير $a .
$a--	النقصان اللاحق	إعادة قيمة $a ، ثم إنقاص القيمة بمقدار 1.

عوامل الزيادة والنقصان
Increment / Decrement Operators

Ex:

```
<?php
   echo "<h3>Postincrement</h3>";
      $a = 5;
   echo "Should be 5: " . $a++ . "<br />";
   echo "Should be 6: " . $a . "<br />";
   echo "<h3>Preincrement</h3>";
      $a = 5;
   echo "Should be 6: " . ++$a . "<br />";
   echo "Should be 6: " . $a . "<br />";
   echo "<h3>Postdecrement</h3>";
      $a = 5;
   echo "Should be 5: " . $a-- . "<br />";
   echo "Should be 4: " . $a . "<br />";
   echo "<h3>Predecrement</h3>";
      $a = 5;
   echo "Should be 4: " . --$a . "<br />";
   echo "Should be 4: " . $a . "<br />";    ?>
```

رموز المقارنة
Comparison Operators

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

تستخدم الرمـــوز المنطقية لتحديد علاقة منطقية بين المتغيرات والقيم أو بين المتغيرات فيما بينها.
لنفترض أن x=5 في الجدول التالي لشرح رمـــــوز المقارنة.

الرموز المنطقية
Logical Operators

لنفترض أن x=6 و y=3 في الجدول التالي
لشرح الرموز المنطقية

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

الثوابت في PHP
constant

الثابت: هو اسم أو معرف لقيمة ثابتة.
الثابت مثل المتغيرات باختلاف بسيط، المتغير كما هو اسمه قد يعطي قيم متغيرة بحسب ظـــروف معينـــة مثل تغير التاريخ أو اختيار المستخدم.
أما الثابت فيعطي قيمة ثابتة ساكنة لا تتغير.
يتم تعريف الثوابت باستخدام الكلمة المحجوزة const قبل اسم الثابت أو من خلال الدالة define ويتبـــع اسم الثابت قواعد كتابة اسم المتغير ذاتها غير أنه لا يبدأ بعلامة **$** و ُيفضل أن ُتكتب بالحروف الكبيـــرة .
ويجب أن ُيعطى الثابت قيمة عند عملية تعريفه ولا يمكن تغيير هذه القيمة فيما بعد.

دالة **defined()**:
للتعرف على الثابت هل هو موجود أم لا وتعيد القيمة true في حالة وجوده وتعيد القيمة false إن لم يكـن موجود.

الثوابت في PHP
constant

Ex:

```
<?php
   const  أحمد= "user1";
   const  AAA = 'user1';
   define("BBB","user2");
   echo أحمد.AAA.BBB;
?>
```