

جامعة طرابلس  
كلية تقنية المعلومات  
قسم نظم المعلومات

المقرر الدراسي:

تطوير التطبيقات

Application development

ITIS311

المحاضرة ( 3 )

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

# جمل الطباعة في PHP

في PHP يوجد دالتين مهمة لإخراج البيانات على المتصفح وهي

✓ الدالة Echo

✓ الدالة Print

سوف تلاحظ بجميع الدروس أو أغلبها نستخدم هاتين الدالتين لطباعة المخرجات على المتصفح.

يوجد إختلافات بسيطة بين دالة echo و print عند الإستخدام ممكن تلخيصها بالتالي:

- الدالة echo تستطيع إخراج و طباعة نص أو أكثر من نص مفصولين بفاصلة.
- الدالة print تستطيع إخراج و طباعة نص واحد فقط وعند تنفيذ الأمر بنجاح يتم إرجاع القيمة 1 كعلامة لنجاح تنفيذ الأمر.
- الدالة echo أسرع من الدالة print لأنها لا تقوم بإرجاع أي قيمة إنما تنفذ الأمر بطباعة النص فقط.

# الدالة Echo

الدالة Echo من أساس اللغة language construct ويمكن إستخدامها بأقواس كبقية الدوال أو بدون أقواس.

ملاحظة: يمكن أن تحتوي النصوص المراد طباعتها باستخدام الدالة Echo على أكواد HTML.

Ex:

```
<?php
```

```
// إستخدام صحيح
```

```
Echo "PHP coures";
```

```
// إستخدام صحيح
```

```
Echo ('PHP coures');
```

```
// إستخدام صحيح
```

```
Echo "<h2> PHP coures </h2>"; ?>
```

# الدالة Print

الدالة Print من أساس اللغة language construct ويمكن إستخدامها بأقواس كبقية الدوال أو بدون أقواس.

ملاحظة: يمكن أن تحتوي النصوص المراد طباعتها باستخدام الدالة Print على أكواد HTML.

Ex:

```
<?php
```

```
// إستخدام صحيح
```

```
Print "PHP coures";
```

```
// إستخدام صحيح
```

```
Print ('PHP coures');
```

```
// إستخدام صحيح
```

```
Print "<h2> PHP coures </h2>"; ?>
```

# الطباعة المتعددة

كما قلنا سابقا فإن الدالة `print` لا تدعم طباعة النصوص وأو المتغيرات المتعددة المفصولة بفاصلة وعند المحاولة سينتج خطأ

Ex:

```
<?php  
print "طباعة"، "أكثر من نص"، "مفصولين"، "بفاصلة" ?>
```

بينما تدعم الدالة `echo` طباعة النصوص وأو المتغيرات المتعددة المفصولة بفاصلة بشكل سهل ودون حدوث أي اخطاء.

Ex:

```
<?php  
// استخدام صحيح  
echo "طباعة"، "أكثر من نص"، "مفصولين"، "بفاصلة" ?>
```

# ربط النصوص وأو المتغيرات في الطباعة

توفر PHP ميزة ربط النصوص وأو المتغيرات المتعددة بواسطة النقطة يطلق عليها اسم concatenation operator وتدعم كل من دالة echo و دالة print طباعة النصوص المربوطة على المتصفح مباشرةً.

```
<?php
```

```
echo "طباعة " . "أكثر من نص " . "مربوطين " . "بنقطة";
```

```
echo "<br>";
```

```
Print "طباعة " . "أكثر من نص " . "مربوطين " . "بنقطة";
```

الفرق بين النقطة (.) و الفاصلة (،) عند طباعة النصوص جدا بسيط يمكن تلخيصه بالتالي:

- عند استعمال النقطة لربط النصوص فإن PHP تقوم بمعالجة جميع النصوص المربوطة ثم جمعها كنص واحد ثم تمريرها للدالة لطباعتها لهذا تدعمها الدالتين.
- عند استعمال الفاصلة لربط النصوص فإن PHP تقوم بطباعة النص الأول ثم طباعة النص الثاني ثم الثالث بشكل منفصل لهذا تدعمها فقط الدالة echo.

# التعبير الشرطية في PHP

لدينا 4 تعابير شرطية وهي :

تعبير `if` : يستخدم هذا التعبير لتنفيذ امر أو اوامر معينة إذا كان الشرط `true`.

تعبير `if ... else` : يمكن استخدام هذا التعبير لتنفيذ امر أو اوامر معينة إذا كان الشرط `true`، أما `else` يتم تنفيذ الامر أو الاوامر التي تليها إذا كان الشرط المكتوب في `if` يساوي `false` .

تعبير `if ... else if ....else` : يستخدم هذا التعبير مع واحد من مجموعة أوامر أي يستخدم عند تحقق شرط معين واذا لم يتحقق سيتم إظهار شرط آخر واذا لم يتحقق سيتم تلقائياً تطبيق الاوامر التي تلي `else`.

تعبير `switch` : يستخدم هذا التعبير لإختيار تنفيذ اوامر معينة من مجموعة الاوامر.

# تعبير IF

```
if(/* الشرط */)
/*/; الامر المراد تنفيذه في حالة تحقق الشرط
OR
if(/* الشرط */)
{
    1 // أمر
    2 // أمر
    3 // أمر
}
```

ملاحظة: في حالة تحقق شرط جملة if وعدم وجود أقواس يتم تنفيذ الامر البرمجي بعد if وصولاً  
لنهاية الامر البرمجي المنتهي بالفاصلة المنقوطة ;

# أمثلة على تعبير IF

Ex:

```
<?php
  if(true)
    echo "true <br>";
  if(true)
  {
    echo "<h1>>true</h1>";
    echo "<h1>inside if</h1>";
  }
  if(false)
    echo "false <br>";
  if(false)
    echo "<h2>>false</h2>";
  echo "outside if";
?>
```

# تعبير IF Else

```
if (/* الشرط */)
{
//الأمر البرمجية في حالة تحقق الشرط
}
else
{
//الأوامر البرمجية في حالة عدم تحقق الشرط
}
```

**ملاحظة:** في حالة تحقق شرط جملة if أو else وعدم وجود أقواس يتم تنفيذ الأمر البرمجي بعد if أو else وصولاً لنهاية الأمر البرمجي المنتهي بالفاصلة المنقوطة ;

## تعبير IF Else المتعددة

```
if(/* الشرط */)
{
//الاوامر البرمجية هنا
}
else if(/* الشرط */)
{
//الاوامر البرمجية هنا
}
//وهكذا تكرر غير محدود
else if (/* الشرط */)
{
//الاوامر البرمجية هنا
}
else {
//الاوامر البرمجية هنا
}
```

**ملاحظة:** لا يشترط كتابة جملة else المفردة في النهاية، وأيضاً يمكن الاستغناء عن أقواس المجموعة إذا كان لدينا امر واحد داخل المجموعة.

# أمثلة على تعابير IF

أمثلة استخدام if مع أنواع البيانات المختلفة أن أي عدد بخلف الصفر يعبر عن القيمة true، وأي سلسلة نصية بخلف السلسلة النصية الفارغة null فهي أيضاً يعبر عن القيمة true.

Ex:

```
<?php
if(0)
echo "<h3>0 true</h3>";
else
echo "<h3>0 false</h3>";
if(13)
echo "<h3>13 true</h3>";
else
echo "<h3>13 false</h3>";
if(-50)
echo "<h3>-50 true</h3>";
else
echo "<h3>-50 false</h3>";
```

## تابع أمثلة على تعابير IF

```
if(null)
echo "<h3>null true</h3>";
else
echo "<h3>null false</h3>";
if(' ')
echo "<h3>' ' true</h3>";
else
echo "<h3>' ' false</h3>";
if('user1')
echo "<h3>user1 true</h3>";
else
echo "<h3>user1 false</h3>";
?>
```

## حالة IF المختصرة

حيث يتم وضع الشرط المُراد التحقق من صحته و يليه علامة استفهام و بعدها يتم وضع الامر الواجب تنفيذه إذا كان الشرط مُحققًا و الامر الواجب تنفيذه إذا كان الشرط غير مُحقق و يفصل بينهما الرمز ":".

```
condition?true:false;
```

Ex:

```
<?php
```

```
echo true?"yes":"no";
```

```
$a=10;
```

```
echo $a < 2?"yes":"no";
```

```
?>
```

# العمليات المنطقية (And, Or, Not, Xor)

الكود التالي ينفذ جدول الصواب والخطأ للعمليات المنطقية :

Ex:

```
<?php  
echo "AND && <br>-----<br>true and true = ";  
if(true and true)  
echo "true<br>";  
else  
echo "false<br>";  
echo "true and false = ";  
if(true and false)  
echo "true<br>";  
else  
echo "false<br>";
```

## تابع العمليات المنطقية

```
echo "false and false = ";
  if(false and false)
echo "true<br>";
else
echo "false<br>";
echo "<br>OR | | <br>-----<br>true or true = ";
  if(true or true)
echo "true<br>";
else
echo "false<br>";
echo "true or false = ";
if(true or false)
echo "true<br>";
else
echo "false<br>";
```

## تابع العمليات المنطقية

```
echo "false and false = ";
  if(false and false)
echo "true<br>";
else
echo "false<br>";
echo "<br>OR | | <br>-----<br>true or true = ";
  if(true or true)
echo "true<br>";
else
echo "false<br>";
echo "true or false = ";
if(true or false)
echo "true<br>";
else
echo "false<br>";
```

## تابع العمليات المنطقية

```
echo "false or false = ";
  if(false or false)
echo "true<br>";
else
echo "false<br>";
echo "<br>XOR<br>-----<br>true xor true = ";
  if(true xor true)
echo "true<br>";
else
echo "false<br>";
echo "true xor false = ";
  if(true xor false)
echo "true<br>";
else
echo "false<br>";
```

## تابع العمليات المنطقية

```
echo "false xor false = ";
if(false xor false)
echo "true<br>";
else
echo "false<br>";
  echo "<br>!<br>-----<br>!true = ";
if(!true)
echo "true<br>";
  else
echo "false<br>";
  echo "!false = ";
if(!false)
echo "true<br>";
else
echo "false<br>";
?>
```

**ملاحظة:** يمكنك استخدام أي صيغة لعمليتي and و or حيث الطريقتين متكافئتين أي استخدام "and" أو "&&" لا يحدث فرق أبداً.

# عمليات المقارنة

(==, !=, >, <, >=, <=, ===, !==)

سنركز هنا على عمل المساواة من نفس النوع وعدم المساواة من نفس النوع لان باقي العمليات واضحة، وكما عرفنا سابقا أن الصفر مساوي للقيمة false وأي عدد خلف الصفر مساوي للقيمة true ، وقيمة السلسلة النصية بخلف السلسلة النصية الفارغة مساوية للقيمة true فلهذا لا يصلح أن نستخدم قيم المساواة العادية، وكمثال إذا أردنا أن نختبر القيمة على أنها false و false فقط إذاً علينا استخدام عملية المساواة من نفس النوع والمثال التالي يوضح العملية :

```
if(" == 0)
echo "" == 0 yes<br>";
else
echo "" == 0 no<br>";
if(0 == false)
echo "0 == false yes<br>";
else
echo "0 == false no<br>";
```

## تابع عمليات المقارنة

```
if(" == false)
echo "" == false yes<br>";
else
echo "" == false no<br>";
if(-10 == true)
echo "-10 == true yes<br>";
else
echo "-10 == true no<br>";
?>
```

ولكن عند استخدام عمليات المساواة من نفس النوع سيتم التعرف على القيم ومساواتها من نفس نوعها  
فالمثال السابق يكون على الشكل التالي :

```
<?php
if(" === 0)
echo "" == 0 yes<br>";
```

## تابع عمليات المقارنة

```
else  
echo "" == 0 no<br>";  
if(0 === false)  
echo "0 == false yes<br>";  
else  
echo "0 == false no<br>";  
if("" === false)  
echo "" == false yes<br>";  
else  
echo "" == false no<br>";  
if(-10 === true)  
echo "-10 == true yes<br>";  
else  
echo "-10 == true no<br>";  
?>
```

# تعبير switch case

```
switch($var)
```

```
{
```

```
case 1:
```

```
    إدراج مجموعة الأكواد 1
```

```
break;
```

```
case 2:
```

```
    إدراج مجموعة الأكواد 2
```

```
break;
```

```
default:
```

```
    إظهار الكود اذا كانت النتيجة غير مطابقة للحالتين 1 و 2
```

```
}
```

**ملاحظة:** إن كان هناك تطابق مع الحالة سيتم تنفيذ الامر الذي يليها فوراً وسيتوقف البرنامج عن الفحص كما يفضل استخدام `break` بعد كل حالة لتجنب عبور الفحص إلى الحالة الأخرى تلقائياً .

# تعبير switch case

```
<?php
$id = 250;
switch($id)
{
case 100:
echo "<h3><a href='page1.php'> go page1 </a></h3>"; break;
case 200:
echo "<h3><a href='page3.php'> go page3 </a></h3>"; break;
case 300:
echo "<h3><a href='page4.php'> go page4 </a></h3>"; break;
default:
echo "<h3><a href='index.php'> go home </a></h3>";
}
?>
```