

الفصل الرابع

أوامر الشرط

أوامر الشرط:

الأوامر الشرطية من أساسيات أى لغة برمجة وتتحكم هذه العبارات فى خط سير البرنامج فمثلا اذا قمت بعرض عدة اختيارات للمستخدم بين ان يقوم باختيار حفظ تغييرات او الغاؤها وقام المستخدم باختيار الالغاء فيمكن التحكم فى طريقة استجابة البرنامج للمستخدم عن طريق اختبار رد فعل المستخدم .

وتستخدم العبارة IF بالصيغة الاتية لاختبار قيمة تعبير معين اذا كان صحيحا (true) ام خطأ (false) :

```
if ([condition = value]) {
// سطور تحقق الشرط
}
else{
// سطور عدم تحقق الشرط
}
```

ومن الصيغة السابقة يتبين لنا انه يتم تنفيذ السطور المحصورة بين الاقواس {} فى حالة معينة و لا يتم تنفيذ الحالتين معا فى نفس الوقت
مثال:

اذا كان قيمة المتغير e تساوى 10 فيتم طبع كلمة "النهاية"

```
<?
$e=10;
if ($e=10){
    echo "النهاية";
}
?>
```

مثال:

اذا كنت تريد السفر ولا تعرف الملابس المناسبة للجو هناك فيمكن اختبار ومعرفة اذا كان الجو دفيى او بارد باختبار درجة الحرارة كالاتى:

```
if ($temp > 24){
    echo "دفيى";
}
```

أوامر الشرط

```
else {  
echo "بارد";  
}
```

نلاحظ من المثال السابق انه تم استخدام العلامات الحسابية المنطقية (<) لاختبار درجة الحرارة و تعنى اكبر من و يوجد ايضا اكثر من معامل يمكن استخدامهم حسب الوظائف الموضحة :

الوظيفة	المعامل
لمقارنة قيمتين بالتساوى	=
القيمة val1 اصغر من القيمة val2	val < val2
القيمة val1 اكبر من القيمة val2	val > val2
القيمة val1 اصغر من او تساوى القيمة val2	val <= val2
القيمة val1 اكبر من او تساوى القيمة val2	val >= val2
القيمة val1 لاتساوى القيمة val2	val <> val2

مثال:

سنقوم الان بتنفيذ لعبة بسيطة وهى تخمين رقم عشوائى من 1 الى 10 فإذا كان الرقم الذى خمنه الزائر اقل من الرقم الصحيح يجب ان تظهر رسالة لتدل على ذلك و بالمثل اذا كان الرقم أكبر أما اذا كان الرقم صحيح فيجب أظهار رسالة ان الزائر قد كسب اللعبة .

قم الان بكتابة الكود الاتى:

```
<html dir="rtl">  
</head>
```

```
<body>  
<form name="form1" method="POST"  
action="result.php">
```

```

<p>قم بادخال الرقم الصحيح<br>
  <label>
    <input type="text" name=" user_no">
</label>
</p>
<p>
  <label>
    <input type="submit" name="Submit" value=" ارسل
الاجابة ">
</label>
</p>
</form>
</body>
</html>

```

و يكون الشكل كما يلي:



قم الان بكتابة الكود الاتي في الصفحة result.php و التي ستحتوى على الكود الفعلى لحساب الرقم العشوائى:

```

<html dir = "rtl">
<?

```

أوامر الشرط

```
$correct_no = rand(1,10);
```

```
if ($_POST["user_no"] > $correct_no){  
echo "الرقم الذى قمت بتخمينه خطأ وهو أكبر من الرقم الصحيح"  
".$correct_no."<br>";
```

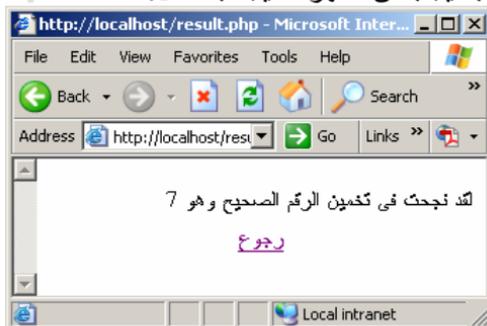
```
}  
elseif ($_POST["user_no"] < $correct_no){  
echo "الرقم الذى قمت بتخمينه خطأ وهو أصغر من الرقم الصحيح"  
".$correct_no."<br>";
```

```
}  
else {  
echo "لقد نجحت فى تخمين الرقم الصحيح وهو "  
".$correct_no."<br>";  
}
```

```
//echo "أسم المستخدم: ".$_POST["user_name"]."<br>";  
//echo "كلمة السر: ".$_POST["user_pw"];
```

```
?>  
<div align="center"><a  
href="javascript:history.back()">رجوع</a></div>  
</html>
```

و عند تجربة اللعبة يجب ان تظهر النتيجة بالشكل:



ولقد نجحت أخيرا في تخمين الرقم الصحيح بعد عناء ولكن لا تقلق إذا لم تصل الرقم الصحيح بعد عدة محاولات حاول مرة أخرى !

لاحظ من الكود المكتوب في صفحة result.php استخدام الدالة rand التي تقوم بارجاع الرقم الصحيح العشوائى المحصور بين 1 و 10 و يتم مقارنتها عن طريق العبارة if و elseif لتدل على عدم تحقق الشرط السابق ثم أخيرا العبارة else التي تدل على عدم تحقق أى من الشروط السابقة .

لقد قمت أيضا بوضع رابط يستطيع الزائر عن طريقه الرجوع مرة أخرى الى الصفحة السابقة و تم هذا باستخدام لغة الجافا سكريبت عن طريق العبارة history.back() وهى من الاوامر المفيدة عمليا .

معاملات خاصة:

اوجزنا فيما سبق المعاملات الحسابية المعروفة مثل التساوى و اكبر من و اصغر من و لكن توجد ايضا علامات خاصة تعطى وظائف وفوائد اكبر عند استخدامها وهى:

علامات المقارنة (==,===):

ذكرنا سابقا ان علامة التساوى المنفردة (=) تستخدم للاحاق قيمة بمتغير مثل العبارة:

```
$var=14;
```

و لكن هذه العلامات تعبر عن المقارنة بين قيمتين و تستخدم مع العبارة IF فمثلا لمقارنة ما اذا كان المتغير s1 يساوى المتغير s2 يجب كتابته بالصورة الاتية:

```
$s1=22;
```

```
$s2="22";
```

```
if ($s1 == $s2) {
    echo "القيم متساوية";
} else {
    echo "القيم غير متساوية";
}
```

لاحظ اننا قمنا بالمقارنة بين نوعين مختلفين من المتغيرات حرفي و عددي ورغم ذلك فان تقدير نتيجة المقارنة يؤدي الى حالة التساوى لان استخدام العلامتين == لا يفرق بين انواع المتغيرات و لكن القيم فقط

أما اذا كتبنا عبارة IF كما يلي :

```
if ($s1 === $s2) {
```

فان تقدير النتيجة ستكون عدم التساوى لان العلامات === تفرق بين القيم و الانواع معا

المعاملين (<>) و (!=) :

العلامتين يعبران عن عدم التساوى و يتم استخدامهم كما بالمثال الاتي:

```
if ($s1 <> $s2) { }
```

```
if ($s1 != $s2) { }
```

و السطرين السابقين يدلان على نفس المعنى وهو "اذا كان المتغير s1 لا يساوى المتغير s2 فنفذ السطور القادمة"

المعاملات المنطقية:

هذه المعاملات هي ثلاثة (AND, OR, NOT) وهي من علامات المقارنة في جميع لغات البرمجة و يتم استخدامهم كما يلي
المعامل AND للتعبير عن ضرورة تحقق شرطين قبل و بعد المعامل حتى يتم تقدير العبارة كلها على انها (true) فمثلا يجب ان نكتب الشرط كما يلي:

```
if ($s1 = 20 AND $s2 = 20) { }
```

اذا تحقق الشرطين معا يمكن تنفيذ السطور القادمة أما اذا كان احد الشرطين لا يتحقق (تكون نتيجته false) فإن العبارة كلها لا تحقق (تكون كلها false) ويتم تنفيذ الجزء else.

بالمثل هناك مثل علامة الإضافة AND و هي العلامتين "&&" وهي تؤدي نفس المعنى و نفس النتيجة

```
if ($s1 = 20 && $s2 = 20) { }
```

و نلاحظ هنا ان لغة PHP جمعت بين عناصر اللغات عالية المستوى مثل الفيچوال بيزيك و عناصر لغة السي بلس بلس

المعامل أو " | | ", "OR":

عكس المعامل AND فهذا المعامل يكفي تحقق احد شروطه حتى يتم تقدير العبارة كلها على انها متحققة (true)

مثال:

```
$t=45;
$w=80;
```

```
if ($t = 40 || $t = 70 || $w=80 || $w=23) {
    echo "النتيجة متحققة";
}
```

بالمثل يمكن كتابة العبارة السابقة كما يلي:

```
$t=45;
$w=80;
```

```
if ($t = 40 OR $t = 70 OR $w=80 OR $w=23) {
    echo "النتيجة متحققة";
}
```

و يمكن الحصول على العلامتين "||" بالضغط على المفتاح Shift + المفتاح "\" مرتين

المعامل "!" :

أوامر الشرط

يوجد باللغات الأخرى المعامل NOT الذى يساوى المعامل "!" من حيث الوظيفة ولكن لغة PHP على تحتوى على المعامل NOT و يجب كلما أمكن استخدام المعاملات المنطقية الرمزية ("&&" و "!=" و "||") كلما امكن لانها اسرع فى الترجمة لان هذه اللغة مكتوبة عن طريق لغة ++C .

عندما يجد المعالج هذا المعامل يفهم ان الشرط كله يتحقق اذا كان الشرط بعد علامة "!" لا يتحقق

مثال:

```
$pass="987654";  
if (!$pass == "987654") {  
    echo "كلمة السر خطأ";  
}else {  
    echo "كلمة السر صحيحة";  
}
```

من المثال السابق نرى ان نتيجة الشرط هي التحقق و طباعة كلمة "كلمة السر صحيحة" لان العلامة "!" قد نفت تحقق الشرط الذى بعدها .

و يمكن استخدامها ايضا لاختبار قيمة متغير شرطى من نوع Boolean كما يلى:

```
if !($b) { }
```

وهذا معناه اذا كان المتغير \$b غير متحقق (not true) يتم تنفيذ السطور التالية

يمكن المزج بين الشروط عن طريق الاقواس بحيث يتم التحقق من حدوث شرط مركب فى نفس السطر

```
$x= 8;  
$y=7;  
$z=5;
```

```
if ($x == 10) || ($x == 8) && ($y != 6) && ($z <= 5)
{
    echo "تحقق الشرط";
}
```

نلاحظ من العبارة السابقة كيف تم تحقيق الشرط بالكامل لان اجزائه المختلفه تحققت وتم ذلك كما يلي:

1- (\$x == 10) || (\$x == 8)

تكون نتيجته (true) لان المتغير x يساوى 8

2- (\$y != 6) && (\$z <= 5)

نرى ايضا تحقق هذا الجزء لان المتغير y لا يساوى 6 بل 5 و المتغير z يساوى بالفعل القيمة 5 فأذا كان الجزئين يتم تحقيقهم فان العبارة بالكامل يتم تقديرها على انها (true) كما يتضح من الرسم التوضيحي:

(true) && (true) → (true)

عبارة IF المتعددة:

يمكن ان تتعدد العبارة IF بحيث يتم التحقق من عدة شروط مختلفة وعلى اساس تقييمها يتم تنفيذ سطور معينة من الكود تناسب كل حالة و يتم ذلك بالصيغة الاتية:

```
if (شرط 1) {
    سطور كود
}elseif (شرط 2) {
    سطور كود
}else {
    سطور كود
}
```

مثال:

إذا اردت اختبار درجة الطالب في احد الامتحانات فإذا حصل على 90 فأكثر يتم طبع "امتياز" و إذا حصل على درجة ما بين 60 و 90 يتم طباعة "جيد" و إذا حصل على أقل من 60 يتم طباعة "رسوب" و يمكن تنفيذ ذلك كما يلي:

```
$deg=80;
if ($deg > 90) {
    echo "امتياز";
}elseif ($deg >= 60 && $deg <= 90) {
    echo "جيد";
}else {
    echo "رسوب";
}
```

يجب حتى يتحقق الجزء else الا يتحقق اى من الشروط السابقة :
كما يمكن تنفيذ تداخل بين عبارات IF بحيث لا يتم قطع اى حلقة IF لحلقة اخرى مثيلتها فمثلا المثال التالي صحيح:

```
if ("شرط") {

    if ("شرط") {
        "أجراء معين "
    }else {
        "اجراء اخر"
    }
}
```

جد:

هام

يجب ان تكون الاقواس المفتوحة { بعدد الاقواس المقفلة } كما يتضح من المثال السابق و يجب تنظيم الاقواس اسفل بعضها البعض مع ازاحة العبارات الفرعية الى اليمين مسافتين على الاقل حتى يمكنك قراءة الكود بسهولة و تصحيح اى خطأ قد يوجد نتيجة للسهو فإذا كتبت الكود السابق كما يلي سيكون صعب في الفهم او اكتشاف الاخطاء.

```
if ("اجراء اخر" }else {"شرط" } {"اجراء معين" } {"شرط" }
```

العبارة Switch:

تقوم العبارة switch بوظيفة مماثلة للعبارة IF و لكن بعكس العبارة IF التي يمكن عن طريقها تقدير نتيجة عدة متغيرات تقوم العبارة switch بتقدير نتيجة متغير واحد و يتم بناء على القيم المختلفة للمتغير تنفيذ اجراءات مختلفة من الكود

ولا يمكن اختبار صحة شرط باستخدام العبارة switch و لكن يتم استخدامها للتعامل مع عدة قيم متوقعة من متغير معين .

و تأخذ العبارة الصيغة القياسية الآتية:

```
switch ( expression )
```

```
{
```

```
case result1:
```

```
// execute this if expression results in result1
```

```
break;
```

```
case result2:
```

```
// execute this if expression results in result2
```

```
break;
```

```
default:
```

```
// execute this if no break statement
```

```
// has been encountered hitherto
```

```
}
```

شرح العبارة:

يتم اختبار قيمة متغير بسيط (حرفي او عددي) بدلا من القيمة *expression* ثم يتم تنفيذ اول حالة من حالات *case* فإذا تحقق تم تنفيذ الكود الذي يوجد بعدها ثم يصل المعالج الى السطر الذي يوجد به العبارة *break*; فيتم الخروج من الجزء *switch* و يتم تنفيذ السطور التي توجد بعده .

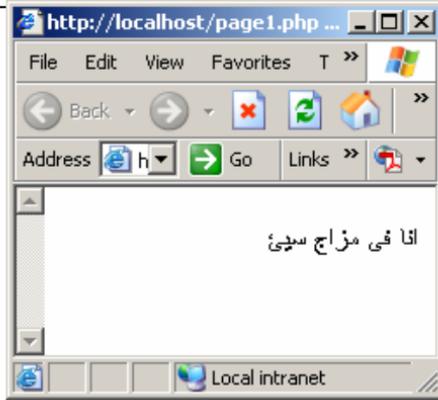
أما إذا لم تتحقق الحالة الاولى يتم تنفيذ الحالة الثانية و هكذا فإذا لم تتحقق اى من الحالات السابقة فيتم تنفيذ الكود الذى يوجد بعد الحالة *default* .

إذا لم تكتب العبارة break; بعد كل نهاية حالة فإن المعالج سيستمر في تنفيذ بقية الحالات حتى إذا تحقق احداها وفي معظم الأحيان أنت لا تريد تحقيق ذلك.

مثال:

```
<html>
<body>
<?php
$mood = "حزين";
switch ( $mood ){
  case "سعيد":
    print "انا فى مزاج جيد";
    break;
  case "حزين":
    print "انا فى مزاج سيئ";
    break;
  default:
    print "$mood: انا لست حزينا او سعيدا بل";
}
?>
</body>
</html>
```

و تكون نتيجة تنفيذ الكود السابق هي :



لاحظ هنا اننا استخدمنا العبارة `print` وهى تعطى نفس وظيفة `echo` تقريبا

الشرط عن طريق الاستفهام `"?:"`.
و يتم ذلك بالصيغة الآتية:

```
( expression
)?returned_if_expression_is_true:returned_if_expression_
is
_false;
```

مثال:

```
$mood = "sad";
$text = ( $mood=="سعيد" )?"انا فى مزاج جيد":"لست فى مزاج
جيد بل $mood";
print "$text";
```

لاحظ مما سبق انه يتم تنفيذ الاجراء بعد علامة الاستفهام مباشرة اذا كان الشرط صحيحا و الا يتم تنفيذ الاجراء الذى يلى علامة التنقيط `:"`. كما بالمثال.

ملحوظة:

اذا كنت تعطى المستخدم امكانية ادخال قيمة ضرورية بحيث تخاف ان يقوم المستخدم بادخال وسوم HTML فى عملية الادخال مما يؤدى الى ان يتعامل معها المعالج على انها كود HTML فيمكن استخدام الدالة

أوامر الشرط

HtmlSpecialChars التي تقوم بحذف العلامات الخاصة بكود HTML بحيث يتم طباعتها كما هي .

مثال:

```
$last_name = htmlspecialchars($last_name);
```

بهذه الطريقة اذا كان المتغير last_name يحتوى على علامات مثل ">" او "<" فيتم طباعتها كما هي .