

## الفصل العاشر اختصارات هامة

- عبارة ال if.
- عبارة ال else.
- عبارة ال elseif.
- عبارة ال switch.
- حلقة التكرار while.
- حلقة التكرار for.
- حلقة التكرار do while.

### - عبارة ال if:

استخدام ال if في كتابة السكريبت شيء أساسي، وكما في لغات البرمجة الأخرى فإن ال PHP تتبع نفس الأسلوب في كتابة ال if، فيمكن تحديد شرط معين مقترن بال if وبالتالي إذا كان الشرط صحيحاً (true) فسيتم تنفيذ الأسطر المحددة، وبتفصيل أكثر يجب وضع الشرط بين قوسين ( )، ووضع الأسطر المطلوب تنفيذها بين العلامات { }، مع ملاحظة أنه يمكن التخلي عن العلامات { } في حال وجود سطر واحد فقط.

فلنفترض وجود نموذج بريدي (Mail Fourm)، يحتوي على الاسم والبريد والرسالة، ونرغب في معرفة ما إذا كان المرسل قد ملأ جميع الحقول وبالتالي إرسال الرسالة، أو أنه لم يفعل ذلك وبالتالي عرض رسالة (فضلاً قم بتعبئة البيانات كاملة)، لعمل ذلك نحتاج لمعرفة أسماء المتغيرات في النموذج، ولذلك فلنفترض أن المتغيرات كالتالي:

(الاسم \$name)، (البريد \$email)، (الرسالة \$later)، ولعمل الشرط الأول (إذا كان الاسم لم يُدخل فلن يتم إرسال الرسالة):

```
<?
if ($name == "")
" echo فضلاً قم بتعبئة البيانات كاملة ";
?>
```

والمعنى أنه إذا كان المتغير \$name لا يحتوي على أي قيمة (أي فراغ) فسيتم تنفيذ السطر التالي وطباعة الجملة، مع ملاحظة أن المطلوب تنفيذه هو سطر واحد فقط ولذلك لم نستخدم { }، بل في حالة وجود أكثر من سطر يجب استخدامها كالتالي:

```
<?
if ($name == "") {
<br> " echo فضلاً قم بتعبئة البيانات كاملة ";
" echo لم تقم بإدخال الاسم ";
}
?>
```

### - عبارة ال else:

هذه العبارة تتيح إمكانية وجود إجراء ثاني لعدم تحقق الشرط، ففي مثالنا السابق كان الإجراء طباعة الجملة إذا تحقق الشرط، ولكن في حالة عدم تحقق الشرط فلن يكون هناك إجراء لتنفيذه، بل إن الإجراء سيتم تنفيذه إذا تحقق الشرط ومن ثم سيتم إكمال بقية الأسطر، وفي حالة مثل هذه الحالة يتم استخدام ال else لوضع إجراء آخر في حالة عدم تحقق الشرط، وبالمثال يتضح المقال:

```
<?
if ($name == "") {
" echo فضلاً قم بتعبئة البيانات كاملة ";
}
else
{
" echo تم إرسال الرسالة، شكراً لك ";
}
```

```
}  
>?
```

في هذا المثال ستتم طباعة الجملة (فضلاً قم بتعبئة البيانات كاملة) إذا تحقق الشرط أن المتغير \$name لا يحتوي على أي قيمة، وستتم طباعة الجملة (تم إرسال الرسالة، شكراً لك) في حالة عدم تحقق الشرط، أي في حالة وجود قيمة في المتغير \$name، مع ملاحظة أن هذا المثال يحتوي على شرطين وليس شرط واحد، فالظاهر هو شرط واحد (\$name == "") ولكن العبارة else تعتبر شرطاً بحد ذاتها ولو لم يكن هذا الشرط مكتوباً، وكما هو واضح فمعنى هذا الشرط هو (إذا كان غير ذلك) فقم بطباعة الجملة.

يمكن أن يكون الشرح غير واضح تماماً، ولكن أهمية فهم الطريقة ستوضح في الأسطر القليلة القادمة.

## - عبارة ال elseif:

في العبارة السابقة ذكرنا أنه يوجد شرطين وإجرائين، أحد هذين الشرطين غير مكتوب بل هو مفهوم من إدراج العبارة else، وفي حالات كثيرة لا يكفي مجرد شرطين وإجرائين لاتمام بعض السكريبات المعقدة، فلذلك يمكن أن نستخدم العبارة elseif مع ال if لعمل مثل هذه السكريبات، فلو افترضنا أن لدينا عداد لزوار الموقع ونريد إظهار العداد بحيث تتم قراءته بشكل جيد، أي بمعنى آخر إذا كان عدد الزوار (١) فستتم طباعة الجملة (عدد الزوار: زائر واحد فقط) وإذا كان (٢) فستتم طباعة الجملة (عدد الزوار: زائرين)... وقس على ذلك، فعندما يكون عدد الزوار (١) سيتم عرض الجملة الأولى فقط وعندما يكون عدد الزوار (٢) فسيتم عرض الجملة الثانية فقط، وهكذا لبقية الشروط.

بافتراض أن المتغير (\$counter) هو عداد الزوار، فالمثال التالي يبين ما تم شرحه سابقاً:

```

<?
if ($counter == 1) {
    "عدد الزوار: زائر واحد فقط"; echo
}
elseif ($counter == 2) {
    "عدد الزوار: زائرين"; echo
}
elseif ($counter >= 3 && $counter <= 10) {
    "عدد الزوار: $counter زوار"; echo
}
else {
    "عدد الزوار: $counter زائر"; echo
}
?>

```

كما هو واضح في المثال السابق سيتم ما يلي:

الشرط: العداد يساوي ١

الإجراء: طباعة (عدد الزوار: زائر واحد فقط)

الشرط: العداد يساوي ٢

الإجراء: طباعة (عدد الزوار: زائرين)

الشرط: العداد أكبر أو يساوي ٣ و اصغر أو يساوي ١٠

الإجراء: طباعة (عدد الزوار: (العداد) زوار)

الشرط: العداد لا يحقق أي من الشروط

الإجراء: طباعة (عدد الزوار: (العداد) زائر)

ملاحظة بسيطة فقط، وهي على العلامة && التي تعني (و)، وهي من علامات

الجمع بين جملتين، فيجب أن تكون الجملتين صحيحتين لتحقيق الشرط.

## - عبارة ال switch:

هذه العبارة قريبة جداً من العبارة if، ولكن يمكن استخدام أكثر من شرطين بأسلوب آخر، غير أنه يجب إسناد قيمة معينة للـ case وهي هنا بمثابة الشرط، لكي يتم تنفيذ الإجراء المحدد لذلك الشرط أو الـ case، وفي النهاية الأمر يعود إلى المصمم وأيهما يفضل، وكما في المثال السابق يمكن كتابة مثال بالـ switch بنفس الطريقة، والمشكلة الوحيدة هي كما قلنا أنه يجب إسناد قيمة معينة لكل case وبالتالي فإن الشرط الثالث من المثال السابق يجب تفريقه لكل قيمة من (٢ إلى ١٠)، وهذه العملية مجهددة لأنه يجب كتابة سطر لكل قيمة كما يلي:

القيمة: ٣

الـ case: 3

الإجراء: طباعة (عدد الزوار: (العداد) زوار)

القيمة: ٤

الـ case: 4

الإجراء: طباعة (عدد الزوار: (العداد) زوار)

القيمة: ٥

الـ case: 5

الإجراء: طباعة (عدد الزوار: (العداد) زوار)

..... الخ...

وفي المثال التالي سأتغاضى عن الشرط الثالث بكامله، وأذكر بقية الشروط والحالات لمجرد فهم طريقة عمل هذه العبارة:

```
<?
switch ($counter)
{
case "1";
echo عدد الزوار: زائر واحد فقط ;
```

```

break;
case "2";
    "echo عدد الزوار: زائرين ";
break;
default;
    "echo عدد الزوار: $counter زائر ";
break;
}
?>

```

استخدمنا في هذه المثال بعض الجمل وتعني ما يلي:

**Switch** وتكتب في البداية مع إدراج اسم المتغير الذي سيتم عمل الشروط عليه.

**Case** أي في حالة (... ) ويكتب بجانبها الشرط.

**Break** وتعني إيقاف العملية والخروج من الشرط بعد تنفيذ أحد الإجراءات.

**Defaukt** وهي تقابل العبارة **else** أي بمعنى أنها لأي حالة لم يتم ذكرها في الشروط.

### - حلقة التكرار **while**:

وهي أبسط حلقات التكرار على الإطلاق، بحيث تأخذ شرط واحد فقط وتبني على تنفيذ ما بين علامات الشروط { } ، والفرق الوحيد بينها وبين ال **if** هو أنها ستقوم بتنفيذ الإجراءات طالما كان الشرط صحيحاً، وهذا يعني احتمال تنفيذ الإجراء أكثر من مرة، وهذه الدالة مفيدة في إدراج الحقول من الجداول وغيرها من الاستخدامات، بحيث لو افترضنا وجود جدول معين في قاعدة بيانات ونريد إدراجه في صفحة **PHP**، فسيكون من أهم خيارات استخدام هذه الدالة، وبإذن الله سيتم التطرق لقواعد البيانات في الفصول القادمة، وفي الوقت الحالي سأذكر مثال بسيط على هذه الدالة لفهم طريقة استخدامها:

```

<?
$total = 10;
while ($total <= 50)
{

```

```
<br>"echo العدد أقل من ٥٠";
$total += 10;
}
?>
```

كبير بسيط يمكن معرفة أن الجملة (العدد أقل من ٥٠) ستتم طباعتها ٥ مرات، لأن حلقة التكرار **while** قامت بتنفيذ الاجراء طالما أن الشرط صحيح، وفي المرة الأولى كان المتغير (**\$total**) يساوي (١٠) والشرط صحيح لأن الـ (**\$total**) فعلاً أصغر أو يساوي الـ (٥٠)، فتم تنفيذ ما بين علامات الشرط، ومن ذلك زيادة متغير المجموع (**\$total**) بقيمة (١٠) ومن ثم الرجوع والمقارنة من جديد، وفي هذه الحالة صار المتغير (**\$total**) يساوي (٢٠) وأيضاً الشرط صحيح وبالتالي الدخول مرة أخرى وتنفيذ الأجراء.... وهكذا حتى يتم الوصول إلى أن قيمة الـ (**\$total**) يساوي (٥٠) وبالتالي الشرط صحيح، ومن ثم تصبح قيمة الـ (**\$total**) تساوي (٦٠) وفي هذه الحالة يتم إيقاف تنفيذ الإجراءات لأن الشرط غير صحيح.

### - حلقة التكرار **for**:

يوجد طريقة أسهل للتعامل مع المثال السابق، فاستخدام حلقة التكرار **while** كانت القيمة الابتدائية للمتغير (**\$total**) في سطر، والشرط في سطر والزيادة على المتغير في سطر آخر، وبالتالي زيارة في عدد الأسطر عن ما يمكن استخدامه مع حلقة التكرار **for**، فالمثال التالي يبين طريقة أخرى لاستخدام مثال الـ **while** بطريقة أسهل:

```
<?
for ($total = 10; $total <=50; $total +=10)
{
<br>"echo العدد أقل من ٥٠";
}
?>
```

وللتوضيح فإن تركيب الـ for هو على الشكل التالي:

```
for(القيمة الافتراضية; الشرط; مقدار الزيادة )  
{  
  الإجراء المطلوب تنفيذه  
}
```

### - حلقة التكرار do while :

وهي نسخة أخرى من الـ while والفرق الوحيد بينهما أن التأكد من الشرط وصحته من عدمها يتم بعد تنفيذ الإجراء وليس قبله كما في الـ while وكمثال عليها:

```
<?  
$total = 10;  
do  
{  
  echo "العدد أقل من ٥٠";  
  $total += 10;  
}  
while ($total <= 50);  
>
```