

الفصل الثالث الأوامر الشرطية

لقد أخذنا في الفصول السابقة فكرة عن المتغيرات وكيفية تعامل البيانات مع النماذج... في هذا الفصل سنتعلم كيفية التحكم بالكود بمعنى تنفيذ سطر معين من الكود عند حصول شرط معين وعند عدم حصوله نتجاهل السطر ونتجه إلى السطر الذي يليه.. هذا يمنحنا تحكماً أكبر بالكود ويجعلنا نستخدم قرارات وتنفيذ أشياء ممتازة وبرامج رائعة بالـ PHP.

دعنا نعطيك فكرة من حياتنا اليومية....

تقوم في الصباح وتريد أن تحضر فطورك الذي يتكون من التالي:

عسل
جبنة
خبز
شاي

ستقوم بالذهاب إلى الثلاجة ثم تقوم بالبحث عن الأشياء التي يتكون منها فطورك، فإذا لم تجد ما تريد تستعد للذهاب إلى المركز التجاري لشرائه، تذهب إلى المطبخ وتتأكد مرة أخرى وتبحث عن المؤونة التي يحتاجها البيت بشكل عام.

- تبحث عن جبنة وإذا لم تجدها تنتقل إلى الخطوة ٣.
- إذا وجدت جبنة فإنك تبحث عن العسل فإذا وجدته تنتقل إلى الخطوة ٤، وإذا لم تجده تنتقل إلى الخطوة ٥.
- تقوم بكتابتها في ورقة جانبية وتقوم بالبحث عن العسل.
- تتجهز للذهاب إلى المركز التجاري.
- تكتبه في ورقة جانبية ثم تتجهز للذهاب إلى المركز التجاري.

هل لاحظت أنك كنت تقوم بالبحث عن أشياء معينة فإذا وجدتتها (true) قمت بالبحث عن التي تليها وإذا لم تجدها (false) تقوم بتسجيلها في قائمة المشتريات لديك.

القيم المنطقية والدوال الشرطية

في الواقع لقد تكلمنا عن المتغيرات سابقاً وذكرنا بأن هناك متغيرات منطقية (قيمتها إما صحيح إم خطأ) ولم نقم بشرحها ، وهذا الفصل سيتولى شرحها وإعطاء أمثلة على كيفية التعامل معها.

العبارة IF

IF condition is true (إذا كان الشرط صحيحاً)

excute this code (قم بتنفيذ هذا الكود)

إن الدالة IF معروفة تقريباً في جميع لغات البرمجة... حيث أنها تقوم بعملية التحقق من شي معين وتنفيذ بعض الأشياء إذا كان الشرط صحيحاً (true) والقيام بتنفيذ أشياء أخرى إذا لم يكن صحيحاً....

سيقوم الـ PHP بتنفيذ الكود التي بين { و } فقط إذا كان الشرط صحيحاً.

أما إذا لم يكن صحيحاً فسيقوم بتجاوزه وتنفيذ الكود الذي يليه.

ويمكنك أيضاً أن تقوم بجعلها بسطر واحد ولا تستخدم الأقواس بل تكتب الأمر مباشرة:

```
IF condition is true excute function;
```

لاحظ أنه لا بد من استخدام { و } إذا كان الكود يتكون من عدة أسطر أما إذا كان يتكون من سطر واحد فلا داعي لاستخدامها.

فالمثالين التاليين كلهما صحيحين.

مثال (١)

```
<?
$S=10
IF ($S=10) echo 11;
?>
```

مثال (٢)

```
<?
$S=10
IF ($S=10){
echo 11;
}
?>
```

لنتخيل مثلاً أن الجو ممطر وسنقوم بإعطاء المطر متغيراً ونسميه rain ونقوم بإعطاء المظلة اسم متغير آخر ونسميه umbrella وسنقوم بافتراض أن هناك أمر في الـ php يسمى go out حسناً الآن الكود الذي نريد أن نقوم بكتابته هو:

```
If $rain = true
{
$umberrlla = true
}
go out();
```

فائدة هذا الكود هو أن تأمر الـ PHP بحمل المظلة (\$umberrlla=true) معه إذا كان الجو ممطراً (\$rain=true) وإذا لم يكن ممطراً ولم يتحقق الشرط فإنه سيخرج إلى النزهة بدون أي مظلة.

طبعاً ليس هناك دالة تقوم بذلك إنما قمنا بذلك من أجل التوضيح للمستخدم هيكلية عمل الدالة بشكل عام.

مقدمة إلى القيم المنطقية [Boolean Values]

القيم المنطقية ترمز إلى الأشياء التي لا تحتل أكثر من احتمالين وهما إما صح وإما خطأ، وهي نوع جديد من القيم غير التي كنت تعرفها سابقاً (مثل الرقمية والنصية).

مثال

```
<?  
$variable=true;  
echo "$variable";  
>
```

لو قمت برؤيه النتيجة ستجد أنه يطبع الرقم واحد وهو قيمة المتغير إذا كان صحيحاً، أما إذا كان خطأ أو غير صحيح فقيمته ستكون (0).

المعاملات المنطقية

لقد أخذنا المعاملات الرياضية فيما سبق بشيء من التفصيل (+، -، /، *) والآن سنأخذ شيئاً جديداً من المعاملات وهي المعاملات المنطقية التي تساعدنا في صناعة الشروط والتقييدات على شيء معين وتعطينا تحكماً أكبر في الكود.

المعاملات: < و >

من المفترض أن تكون متآلفاً مع علامتي (الأكبر من) و (الأصغر من) في الرياضيات التي تتعلمها في المدرسة مما يجعل فهم هذا الأمثلة بسيطاً.

```
<?
If (6>5)
{
"echo الرقم ستة أكبر من الرقم خمسة ";
}
Echo "end";
?>
```

سيقوم الـ PHP في مثالنا هذا بفحص الشرط (6>5) فإذا كان صحيحاً (true) سيقوم بطباعة السطر (الرقم ستة أكبر من الرقم خمسة) ثم يقوم بطباعة end، وإذا لم يكن صحيحاً فسيقوم بتجاهل الكود وطباعة (end) فقط.

يمكننا أيضاً استعمالها في المقارنة بين متغير ورقم أو بين متغير وثابت (constant) أو العكس أو المقارنة بين متغيرين.

مثال (1)

```
<html dir ="rtl" >
<?
$LuckeyNumber = 5;
If ($LuckeyNumber<6)
{
"echo الرقم الحظ أصغر من الرقم ستة ";
}
?>
```

مثال (٢)

```
<html dir="rtl">
<?
$f=5;
$r=10;
If ($f >$r)
{
r";$ المتغير $f أكبر من المتغير "
}
?>
```

تطبيق عملي:

قم بتشغيل محرر النصوص واكتب الكود التالي واحفظه باسم thegame.php

```
<html dir = "rtl">
<body>
<form method =get action="game.php">
ماهو الرقم الذي أفكر به الآن والذي هو بين ١ و١٠؟
<input type="text" name="number">
<br>
<br>
<input type =submit>
</form>
</body>
</html>
```

قم بفتح محرر النصوص لديك من جديد واكتب الكود التالي واحفظه باسم
game.php

```
<html dir="rtl">
<body>
<?
10); $num = rand (1
if ($number>$num)
{
" echo " لقد اخترت رقم أكبر من الذي أفكر فيه ; "
"Echo الرقم الذي أفكر فيه هو ; "
Echo $num;
"Echo "<br>" . "يؤسفنا فعلاً أنك لم تتجح، نتمنى أن نقول لك في المرات القادمة ; "
}
if ($number<$num)
{
" echo " لقد اخترت رقم أصغر من الذي أفكر فيه ; "
"Echo الرقم الذي كان في مخيلتي هو ; "$num"
"Echo "<br>" . "يؤسفنا فعلاً أنك لم تتجح، نتمنى أن نقول لك في المرات القادمة ; "
}
?>
لقد نجحت
</body>
</html>
```

شرح التطبيق:

الدالة rand

تقوم هذه الدالة باختيار رقم عشوائي من بين رقمين يتم اعطاؤها إياهما، الرقم
الاول (X) هو الأصغر والرقم الثاني هو الأكبر (y)

rand (x
y);

يمكنك حفظ القيمة التي تقوم بإخراجها هذه الدالة في متغير مباشرة

مثال

```
$Num = rand (5.57);
```

وهذا يوضح ما قمنا به في الكود

```
10); $num=rand(1
```

لقد قمنا باختيار قيمة عشوائية ثم قمنا بمقارنتها مع القيمة التي تم إدخالها من قبل المستخدم فإذا كانت القيمة التي أدخلها المستخدم أكبر من قيمة العدد العشوائي أخبرناه بأن الرقم الذي أدخله أكبر من الرقم الصحيح... وهذا ما تجده جلياً في الأسطر التالية:

```
if ($number>$num)
{
"echo لقد اخترت رقم أكبر من الذي أفكر فيه ; "
"Echo الرقم الذي أفكر فيه هو ; "
Echo $num;
"Echo "<br>" ."  
يوسفنا فعلاً أنك لم تتجح، نتمنى أن نقول لك في المرات القادمة ; "
}
```

فإذا لم ينطبق الشرط وكان الرقم الذي اختاره المستخدم أصغر من الرقم العشوائي فإنه يترك الشرط الأول ويتجه إلى الشرط الثاني ويطبق الأوامر التي فيه والتي تقوم بإخباره بأن الرقم الذي قام باختياره أصغر من الرقم المطلوب، وهذا ما تجده جلياً في الأسطر التالية:

```
if ($number<$num)
{
"echo لقد اخترت رقم أصغر من الذي أفكر فيه ; "
$num"; Echo الرقم الذي كان في مخيلتي هو "
"Echo "<br>". "يؤسفنا فعلاً أنك لم تنجح، نتمنى أن نقول لك في المرات القادمة ; "
}
```

فإذا لم يتطبق الشرطين فإنه يتركهما ويكتب الكلمة (لقد نجحت) بدون أي كلمات أخرى مثلما كنا نكتب الكلمة (يؤسفنا فعلاً أنك لم تنجح، نتمنى أن نقول لك في المرات القادمة) قبل كلمة (لقد نجحت)، أتمنى أنك قد فهمت جيداً ما أقول.... وتظهر هذه العبارة جلية في الأسطر التالية:

```
?>
لقد نجحت
</body>
</html>
```

على هذا نكون قد صنعنا لعبة كاملة تقوم بإخبار المستخدم عند نجاحه أو خسارته.

معاملات المساواة: == و ===

لقد قمنا باستخدام علامة المساواة الفردية سابقاً في تخزين قيمة في متغيروها نحن نأخذ نوعاً من علامات المساواة وهي علامة المساواة المزدوجة (==) وعلامة المساواة المضاعفة (===).

لقد كنا نستخدم علامة المساواة الفردية أو العادية في تخزين القيم في المتغيرات.

مثال:

```
<?
$m=12;
?>
```

ولكن العلامات التي نتكلم عنها الآن تستخدم في تحديد إذا ما كانت قيمة معينة تساوي قيمة أخرى.

مثال:

```
<?
$m="11";
$u=11;
If ($m==$u)
{
Echo القيم متساوية ;
}
?>
```

لاحظ أن \$m متغير حريف وأن \$u متغير رقمي.

إذا كنا نريد إرجاع قيمة إلى متغير نستخدم علامة المساواة العادية (=) وإذا أردنا اختبار متغيرين أو قيمة معينة من أنها متساوية نقوم باختبار القيم بواسطة علامة المساواة المزدوجة (==).

في الـ php4.01 تم إصدار علامة مساواة جديدة تقوم باختبار القيم ولا تعطي القيمة (true) إلا إذا كانت أنواع القيم متساوية وأنواع البيانات في المتغيرات أيضاً متساوية.

مثال (1):

```
<?
$m="11";
$u=11;
If ($m==$u)
{
Echo "القيم متساوية ";
}
?>
```

مثال (2):

```
<?
$m="11";
$u=11;
If ($m=== $u)
{
Echo "القيم متساوية ";
}
?>
```

التوضيح

لاحظ أننا في المثال الأول استخدمنا علامة المساواة المزدوجة لاختبار القيم وكانت القيم متساوية في المتغيرين فتمت طباعة أن القيم متساوية (مع أن نوع البيانات مختلف) ولكن في المثال الثاني عندما استخدمنا علامة المساواة المضاعفة لم تتم طباعة أي شيء وذلك لأن القيم متساوية ولكن نوع البيانات مختلف فالمتغير \$m حري في بينما المتغير \$u رقمي.

المعاملات: != و <>

إن عكس علامة المساواة هي علامة عدم المساواة (!=)

مثال:

```
<?  
" If (5!=99) echo القيم غير متساوية ;  
>?
```

لاحظ أن 5 لا تساوي 99 لذلك فإن الشرط صحيح (true) لذلك قام بطباعة أن القيم غير متساوية.

إن الضد من علامة (أكبر من) و(أصغر من) هو علامة ال (<>) وهو يقوم بإرجاع قيمة (true) إذا كانت القيمتين مختلفتين عن بعضهما أي أنه مثل علامة != تقريباً.

مثال:

```
<?  
" If (5<>99) echo القيم غير متساوية ;  
>?
```

تطبيق عملي على علامات المساواة وعدم المساواة
قم بفتح محرر النصوص لديك واكتب الكود التالي:

```
<html>
<head></head>
<body>
<Form method =get ACTION= "quiz.php">
ماهو اسم الرجل الذي يسمى بالفاروق ؟
<br><br>
"><input type ="radio" name = "man" value="
عمر بن الخطاب
<br>
"><input type ="radio" name = "man" value="
أبو بكر الصديق
<br>
"><input type ="radio" name = "man" value="
عثمان بن عفان
<br>
<input type = submit>
</form>
</body>
</html>
```

احفظها باسم ...quiz.html

قم بفتح محرر النصوص لديك واكتب الكود التالي:

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

```
echo "الإجابة صحيحة";  
echo "الإجابة خاطئة";  
?>
```

قم بحفظه باسم quiz.php وضعهما في مجلد السيرفر.

قم بتشغيل الملف quiz.html

العمليات المنطقية [NOT,OR,AND]

إن هذه العمليات المنطقية تتيح لك تنفيذ الكود بعد التحقق من مجموعة شروط وأيضاً تنفيذ الكود إذا تحقق أكثر من شرط: (AND) أو تحقق شيء معين من بين عدة أشياء: (OR) ويمكنك مثلاً التحقق من عدم صحة شيء لكي تقوم بتنفيذ شيء آخر: (NOT)

فيمكنك مثلاً أن تقول: إذا كان الجو ممطراً والعاصفة شديدة فلن أخرج من البيت.

ويمكنك أن تقول: إذا كان الجو هادئاً أو لا يوجد أمطار فسأقوم بالخروج إلى المنتزه.

ويمكنك أيضاً أن تقول: إذا لم يكن الجو ممطراً سأقوم بالخروج إلى نزهة.

ولكن عند استخدامك لهذه الدوال عليك مراعاة أن تقوم بجعل هذه الشروط بين قوسين.

المعامل (AND) ونظيره (&&)

يمكننا استعمال المعامل (AND) والمعامل (&&) للتحقق من صحة عدة شروط لتنفيذ شيء معين

مثال (١)

```
<?
$w=10;
$g=12;
"IF ($w=10 and $g=12) echo ("
?>
```

مثال (٢)

```
<?
$w=10;
$g=12;
"IF ($w=10 && $g=15) echo ("
?>
```

في المثالين السابقين قمنا بعملية التحقق من أكثر من شرط باستخدام المعاملين (&& و and) فعندما تحققت جميع الشروط تم تنفيذ الأمر.... وعندما لم تكن جميع الشروط صحيحة تم تجاهل الأمر.

لاحظ أننا قمنا بجعل الشروط بين قوسين () لكي يعمل الكود بشكل صحيح:

```
($w=10 && $g=15)
($w=10 and $g=12)
```

المعامل (OR) ونظيره (||)

المعامل OR يقوم بالتحقق من عدة شروط وإذا تحقق أي واحد منها فإنه يقوم بتنفيذ الكود ونظيره (||) الذي يقوم بنفس العملية.

مثال (١)

```
<?
$E=100;
$T=8;
"IF ($E=14 OR $E=55 OR $E = 10 OR $T=8) echo ("
?>
```

مثال (٢)

```
<?
$E=100;
$T=458;
"IF ($E=14 || $E=55 || $E = 10 || $T=8) echo ("
?>
```

إذن عندما تحقق واحد من هذه الشروط تمت طباعة السطر (لقد تحقق أحد هذه الشروط).

ملحوظة: قد لا تكون بتلك الأهمية لكن يجب أن تعرف أن الرموز && و || لها الأسبقية والأفضلية على استخدام AND و OR.

المعامل NOT ونظيره (!)

في الواقع لا يمكنك استخدام NOT أبدا لأنها ليست موجودة أصلاً في لغة PHP لكن يمكنك استخدام المعامل (!) كبديل لها فهو يؤدي نفس وظيفتها وهي القيام بالتأكد من أن هناك قيمة غير صحيحة (FALSE) لكي يتم تنفيذ شي معين.

```
<?
"$F="رسالة";
"IF (!$F) echo ("أهلاً بك";
?>
```

في المثال السابق يقوم الـ PHP بالتأكد من أن المتغير \$F لا يحتوي على القيمة الحرفية (نعمان) ويتم ذلك باستخدام المعامل (!) وعندما يتم التأكد من ذلك يقوم بطباعة السطر (أهلاً بك)

ونشير إلى أننا عندما نقوم باختبار متغير بواسطة المعامل (!) فإن الـ PHP إذا وجد المتغير فارغاً أو لم يتم إنشاؤه يعطيه القيمة صفر وهي FALSE.

مثال

```
IF (!($R)) echo (10);
```

استخدام المعاملات < و > و <= و >=

من العلامات المعروفة والمشهورة في الرياضيات هي علامتي (أصغر من أو يساوي) <= أو (أكبر من أو يساوي) >= وهي تستخدم بنفس وظيفتها بالـ php وهي معرفة إذا ما كانت قيمة أصغر أو أكبر من أو تساوي قيمة أخرى، وهذه الأمثلة تعطيك مدخلاً أشمل لفهم هذه الدوال:

```
<?  
$t = 15;  
". "<br>"; If ($t >= 10) echo ("  
$t = 5;  
") If ($t <= 9) echo ("  
?>
```

جميع المعاملات

يمكننا في الشرط أن نتحقق من مجموعة من القيم باستخدام مجموعة من المعاملات، ونقوم بتجميع هذه المجموعات داخل أقواس () مثلما كنا نستخدم سابقاً أكثر من معامل (+، -، /، *) باستخدام الأقواس.

وسيبدو ذلك واضحاً وجلياً في مثالنا التالي:

```
<?
$a=10;
$y=5;
$t =29;
If (($a == 10) or ($a==54) and ($y !=25) and ($t >= 11)) echo "تحققت جميع الشروط";
?>
```

ستتم طباعة ١٨ لأن قيمة تجميع التعبير السابق تكون صحيحة ولو قمنا بشرح المثال فسنقوم برؤية القسم الأول وهو:
(\$a == 10) or (\$a==54)
وطبعاً المتغير يحمل القيمة ١٠ فسيكون هذا الجزء صحيحاً.

ثم نقوم برؤية الجزء:

(\$y !=25) and (\$t >= 11)

وطبعاً تم التحقق من جميع الشروط وتمت طباعة الكلمة (تحققت جميع الشروط).

نعد الشروط [else و else if]

يمكننا استخدام أكثر من هيكلية للعبارة if فهناك مثلاً الهيكلية التالية:

```
If condtion is true
{
Excute code
}
Else
{
Excute other code
}
```

وهي تقوم بالتحقق من الشرط فإذا وجدته صحيحاً قامت بتنفيذ الكود الأول وإذا لم تجده صحيحاً ستقوم بتنفيذ الكود الآخر.

مثال

```
<?
$age=10;
If ($age>18)
{
echo"مرحباً بك في أكبر موقع تجاري إلكتروني";
}
else
{
echo "ممنوع دخول الأطفال الموقع لأنهم لا يملكون المال";
}
?>
```

ويمكننا أيضاً استخدام الهيكلية التالية:

```
If condtion is true
{
Excute code
}
Elseif
{
```

```
Excute other code
}
Else
{
Excute other code
}
```

وهي تقوم بتطبيق أكثر من شرط فإذا لم يكن أي شرط من الشروط صحيحاً سيتم تنفيذ الكود الذي يقع بعد كلمه else.

مثال:

```
<?
$age=10;
If ($age<=18)
{
echo"مرحباً بك في أكبر موقع تجاري إلكتروني";
}
}elseif($y >= 44
{
echo " لا يوجد مشكلة أيضاً إذا كنت كبيراً ";
}
else
{
echo"ممنوع البقية";
}
?>
```

نعشيش العبارات الشرطية

يمكنك تعشيش العبارات الشرطية، ونعني بتعشيش العبارات الشرطية هي أن تقوم بعملية تعشيش الشروط فمثلاً إذا كان شرط ما صحيحاً فإنه يجب أن يكون شرط آخر صحيح لكي يتم حصول شيء معين وغير ذلك.

مثال:

```
<?
$h="ahmed";
$f=45;
If ($h == "ahmed")
If ($f = 45)          {
{
"echo الاسم والرقم صحيحان";
}
else
{
"echo الرقم غير صحيح";
}
}
{ else
"echo اسم تسجيل الدخول غير صحيح ;
}
?>
```

هذا مجرد مثال بسيط جداً لتعشيش الدوال الشرطية حيث يقوم بإجراء اختبار على قيمة معينة ثم يقوم عند تجاوز ذلك الاختبار بنجاح بإجراء اختبار ثاني فإذا تم تجاوز الاختبار الثاني تتم طباعة الاسم والرقم صحيحان وإذا لم يتم الاجتياز تتم طباعة عبارة الفشل في الاجتياز.

تطبيق عملي

سنقوم في هذا التطبيق بصناعة مسابقة بسيطة نستخدم فيها ما تكلمنا عنه سابقاً.

- قم بإنشاء ملف Msabqa.html.

- قم بكتابه الكود التالي فيه:

```
<html>
<body>

<form method="POST" action="msabqa.php" dir="rtl">
<br> من هو أول الخلفاء الراشدين
```

```

<p><br><input type="radio" value="abubaker"
<br><input type="radio" value="3mar" أبو بكر الصديق name="s">
عمر name="s">
<br><input type="radio" value="3thman" checked
<br><br><br>عثمان name="s">
</p>
"> <input إرسال <p><input type="submit" value="
"></p>حذف type="reset" value="
</form>
</body><html>

```

قم بفتح ملف وقم بتسميته msabqa.php

```

<?
<html dir = "rtl">
If $s == "3mar"{
الإجابة صحيحة
}
else
{
"الإجابة خاطئة"; echo
}
?>

```

العبارة Switch

```
Switch (VARIABLE) {  
:CASE THING1  
Excute code ;  
    break;  
:CASE THING2  
Excute code ;  
break;  
    Default;  
Excute code ;  
  
}
```

تقوم العبارة بنفس عملية العبارة if ولكن بهيكلية أسهل ومحبة أكثر وتتيح لك اختبار قيمة متغيرة وإجراء أكثر من اختبار عليه.

break;

تقوم بالخروج من عبارة معينة مثل switch و if والذهاب إلى الأوامر والعبارات التي بعدها.

EXIT;

تقوم بعملية الخروج من الكود نهائياً ولا تطبق أي أوامر بعدها، وفي الأمثلة التوضيحية التالية ستجد أن break; تخرج من العبارة فقط (Statement) بينما exit; تقوم بالخروج من كامل الكود (code).

مثال:

```
<?  
$s=10;  
if ($s=10) {  
echo "number=10";
```

```

exit;
}
elseif ($s<11) {
    echo "number is less than 11"
    {
    echo "hello";
    }
?>

```

مثال:

```

<?
$s=10;
if ($s=10) {
    echo "number=10";
    break;
}
elseif ($s<11) {
    echo "number is less than 11"
    {
    echo "Hello";
    }
?>

```

Default;

إذا لم تصلح جميع الحالات (Cases) في العبارة (Switch) فسيتم تنفيذ الأوامر التي تقع بعد هذه الكلمة وهي تؤدي نفس عمل else تقريباً في العبارة .if.

مثال (١)

```

<?
$g= "ahmed";
Switch ($g) {
Case "ahmed":
    "Echo مسموح";
Break ;
Case "khaled ":

```

```

“Echo ممنوع;”
Break ;
Case “salem“:
“Echo ممنوع;”
Break ;
Case “Mohmed “:
“Echo مسموح;”
Break ;
Default ;
“Echo لقد أدخلت اسماً غير صالح;”
}
?>

```

مثال (٢)

```

Switch ($g) {
Case $g>50:
“Echo كبير;”
Break ;
: Case 40
“Echo لا بأس;”
Break ;
Case ($g<15):
“Echo أطفال ممنوع;”
Break ;
Case 30:
“Echo مسموح;”
Break ;
}

```

لاحظ أننا عند اختبارنا لنصوص نحتاج إلى علامتي تنصيص مزدوجة وعند الأرقام فإننا لا نحتاج إلى ذلك.

تطبيق عملي

قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم age.html

```
<html>
<form method=post action="age.php">
كم عمرك ؟
<br>
<input type="text" name = "g">
">إرسال<input type=submit value="
</form>
</html>
```

قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم age.php

```
<?
Switch ($g) {
Case $g>50:
    "؛ كبيرEcho "
Break ;
: Case 40
    "؛ لا بأسEcho "
Break ;
Case ($g<15):
    "؛ أطفال ممنوعEcho "
Break ;
Case 30:
    "؛ مسموحEcho "
Break ;
}
?>
```

الشرح

تقوم العبارة Switch باختبار قيمة متغير ما ويمكنك إجراء أكثر من افتراض عليه ويجب عليك كتابة الكلمة break لكي تقوم بإيقاف تنفيذ العبارة switch فمثلاً لو قمت بكتابة الكود التالي:

```
<?
$g=40
Switch ($g) {
Case $g<50:
Echo "1 ";
: Case 40
Echo "2" ;
}
?>
```

فإذا أدخل المستخدم الرقم ٤٠ فسيتم طباعة الرقمين واحد واثنين كلاهما وذلك لأنك لم تقم بإيقاف العبارة فأكملت التحقق وطبقت جميع العمليات المطلوبة.

الخلاص من وسوم ال html

إذا قمت بوضع مربع نص وأردت من المستخدم كتابة شي فيه فإنه يستطيع إدخال أي شيء ولنفترض أنه كتب في مربع النص كالتالي:

I am ahmed ...

فسيقوم المتصفح بعرضها بعد معالجتها كالتالي:

I am ahmed

ولنقم بتطبيق عملي على ذلك

قم بفتح محرر النصوص واكتب الكود التالي واحفظه باسم htmlch.html

```
<html dir="rtl">
<form method=post action="html.php">
```

أدخل اسمك الكريم

```
<br>
<input type="text" name = "fname">
">إرسال<input type=submit value="
</form>
</html>
```

قم بفتح محرر النصوص واكتب الكود التالي واحفظه باسم `html.php`

```
<?
Echo "هذا هو الشكل الطبيعي للعبارة عند طباعتها";
Echo "<br>". $fname;
?>
```

قم بوضع الملفات في مجلد السيرفر ثم قم بتشغيل الملف `htmlch.html` واكتب في مربع النص أي شيء وضعه بين وسوم `html`

مثال:

```
I am <b><i>alfareees</b></i>
```

ستجد أنه قد تم التعامل مع الوسوم كـ `html` وليس كنص عادي ولكي تعرضها كنص عادي فإنك تقوم باستخدام الدالة

```
htmlspecialchars();
```

حيث أنها ستقوم بمعاملة كود الـ `html` كنص عادي وطبيعي تماماً. إذا نقوم بتعديل ملف الـ `html.php` ليصبح كالتالي:

```
<?
$name = htmlspecialchars($fname);
Echo "هذا هو الشكل بعد استخدام الدالة";
Echo "<br>". $fname;
?>
```