الفصل الخامس

الحلقات التكرارية

دار الجزاء

الحلقات التكرارية **LOOPS**:

تعطينا لغة PHP امكانية تنفيذ مجموعة من التعليمات لعدد من المرات ويتم ذلك عن طريق عبارات الحلقات التكرارية ويكون للحلقة شرط معين اذا لم يتحقق تستمر الحلقة فى التكرار حتى يتم الوصول الى تحقيق الشرط فيتم انهاء الحلقة او قد تستخدم عبارات اخرى تؤدى الى الخروج الاجبارى من الحلقة كما سنرى:

العبارة **While**:

و الصيغة القياسية لهذه العبارة هى:

```
while (شرط الايقاف){
//يتم تكرار هذا الجزء
}
```

معنى العبارة السابقة انه يتم تكرار الحلقة طالما الشرط متحقق (true) و يجب ان تقوم فى داخل الحلقة بتغيير هذا الشرط الى false والا يتم تنفيذ الحلقة الى مالانهاية .

مثال (تجاهل ارقام السطور هى للتوضيح فقط):

```
1: <html dir="rtl">
2: <head>
3: <titel>مثال للحلاقات التكرارية</title>
4: </head>
5: <body>
6: <?php
7: $counter = 1;
8: while ( $counter <= 12 )
9: {
10: print "$counter × 2 = ".($counter*2)."<br>";
11: $counter++;
12: }
13: ?>
```

```
14: </body>
15: </html>
```

نلاحظ مما سبق اننا قمنا بتعريف متغير فى السطر رقم 7 واعطيناه القيمة المبدئية 1 ثم بدئنا الحلقة و وضعنا الشرط لهذه الحلقة انها لاتكرر الا اذا كان العداد اقل من او يساوى 12 اى ان الحلقة ستتكرر 12 مرة .

ثم وضعنا عبارة print بسيطة فى السطر رقم لتقوم بطبع ناتج رقم العداد مضروبا فى 2 .

واخيرا تم زيادة قيمة العداد بقدار واحد صحيح فى كل مرة يتم فيها تكرار الحلقة وتم ذلك عن طريق العبارة الشهيرة لزيادة العداد وهى :

```
$var++;
```

وهذه العلامة ++ تساوى العبارة :

```
$var = $var + 1;
```

الحلقة **Do..While**:

ولها الصيغة:

```
} do
سطور الكود التى تتكرر //
}
while ( شرط ايقاف الحلقة );
```

هذه الحلقة تعتبر مثل حلقة while ولكنها مقلوبه رأسا على عقب و الفرق الاساسى بينهم انه يتم فى هذه الحلقة تنفيذ اول تكرار قبل ان يتم اختبار الشرط اذا كان متحقق ام لا

مثال:

```
1: <html dir="rtl">
2: <head>
```

```
3: <title>loop Do..While</title>
4: </head>
5: <body>
6: <?php
7: $num = 1;
8: do
9: {
10: print "رقم الحلقة هو :  $num<br>\n";
11: $num++;
12: }
13: while ( $num > 200 && $num < 400 );
14: ?>
15: </body>
16: </html>
```

تنفيذ الكود السابق هو النتيجة:

و الان هل تعرف لماذا تم تنفيذ الحلقة مرة واحدة فقط ؟

من المثال السابق يتضح ان الحلقة do..while قمانت بالتكرار قبل ان تقوم بتقدير قيمة متغير الحلقة num و لايتحقق التقدير لذلك فأن الحلقة تتنهى عند هذا الحد ؟ ، حاول الان تغيير شرط الحلقة الى

while (num <400);

وراقب ماذا يحدث

الحلقة **For**:

وهى تشبه الحلقة While تماما فيما عدا انها امنه اكثر و يتضح ذلك من الصيغة القياسية لهذه الحلقة:

for (***variable assignment***; ***test expression***; ***variable increment***)
{
// الكود الذى يتم تكراره
}

تذكر من المثال السابق الخاص بالحلقة While كيف قمنا اولا بتعريف متغير و اعطاؤه قيمة مبدئية ثم كتبنا الشرط الخاص بالمتغير بعد العبارة while ثم قمنا داخل الحلقة بزيادة قيمة المتغير حتى تتنتهى الحلقة عند حد معين فكل ذلك يمكن تنفيذه مع الحلقة FOR فى سطر واحد مما يجعل احتمالات ان تنسى زيادة المتغير مثلا منعدمة مما قد يؤدى الى حلقة لانهائية .

مثال:

```
1: <html dir="rtl">
2: <head>
3: <title>FOR loop</title>
4: </head>
5: <body>
6: <?php
7: for ( $counter=1; $counter<=12; $counter++
)
8: {
9: print "$counter × 2 = ".($counter*2)."<br>";
10: }
11: ?>
12: </body>
13: </html>
```

لاحظ ان هذا المثال يعطى نفس نتيجة المثال الخاص بالحلقة While فيتم تمهيد قيمة مبدئية للمتغير العداد فى اول معامل من معاملات For ثم يتم تحديد الشرط الذى تتوقف عنده الحلقة ثم يتم زيادة العداد فى المعامل الاخير

الخروج من الحلقات:

يمكن الخروج من الحلقات بطريقة اجبارية عند تحقق شرط معين داخلى باستخدام عبارة IF مثلا بحيث لايتم استكمال عملية التكرار ويتم ذلك هكذا:

```
1: <html dir="rtl">
2: <head>
3: <title>الخروج من الحلقات</title>
4: </head>
5: <body>
6: <?php
7: $counter = – 4;
8: for ( ; $counter <= 10; $counter++ )
9: {
10: if ( $counter == 0 )
11: break;
12: $temp = 4000/$counter;
13: print "4000 divided by $counter is...
$temp<br>";
14: }
15: ?>
16: </body>
17: </html>
```

افترض مثلا انك تستقبل قيمة من الزائر وتقوم باجراء عمليات التكرار بناء على هذه القيمة وماذا اذا كتب الزائر قيمة اقل من الصفر فى هذه الحالة يجب ان نختبر قيمة العداد بحيث لا يتم القسمة على الصفر و يتم ذلك عن طريق عبارة IF فأذا تحققت يتم تنفيذ العبارة break; ويتم الخروج من الحلقة فى السطر 11

لاحظ انه يمكن عدم كتابة اى معامل من معاملات For ولكن يجب ترك الفاصلة المنقوطة مكانها ";"

هناك طريقة اخرى لتفادى تنفيذ بعض الحلقات التكرارية و هى العبارة Continue; وفيها بدلا من ان يتم ايقاف الحلقة نهائيا و الخروج منها يتم تفادى تكرار بعض الحلقات بناء على شرط معين كما سنرى:

مثال:

```
1: <html dir="rtl">
2: <head>
3: <title>العبارة Continue</title>
4: </head>
5: <body>
6: <?php
7: $counter = – 4;
8: for ( ; $counter <= 10; $counter++ )
9: {
10: if ( $counter == 0 )
11: continue;
12: $temp = 4000/$counter;
13: print "4000 divided by $counter is...
$temp<br>";
14: }
15: ?>
16: </body>
17: </html>
```

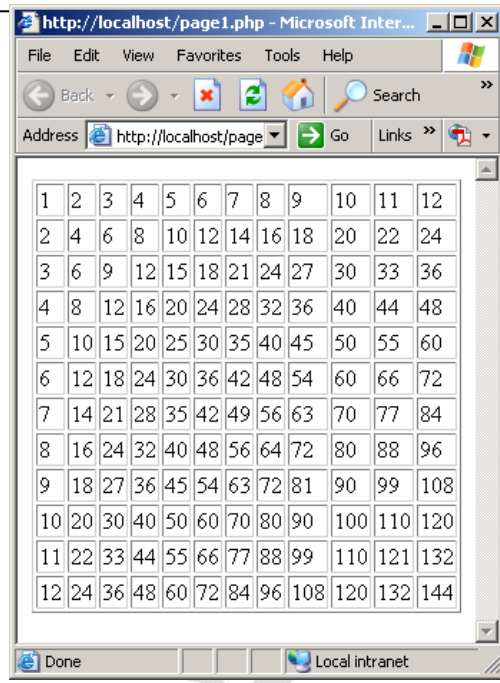
هنا فى حالة اذا كان العداد صفر يتم تفادى تنفيذ الحلقة حتى تكون قيمة العداد غير صفرية و يذهب المعالج الى العبارة For و العبارة while ليعيد تقييم العداد مرة اخرى ولا يتم تنفيذ السطور التى تأتى بعد Continue; .

التداخل بين الحلقات:

يمكن اجراء تداخل لحلقة داخل اخرى بشرط الا يتقاطعا و هذا مفيد اذا كنت تريد انشاء جدول HTML داخل الصفحة و يمكن تنفيذ ذلك كما يلى:

```
1: <html>
2: <head>
3: <title>تداخل الحلقات</title>
4: </head>
5: <body>
6: <?php
7: print "<table border='1'>\n";
8: for ( $y=1; $y<=12; $y++ )
9: {
10: print "<tr>\n";
11: for ( $x=1; $x<=12; $x++ )
12: {
13: print "\t<td>";
14: print ($x*$y);
15: print "</td>\n";
16: }
17: print "</tr>\n";
18: }
19: print "</table>";
20: ?>
21: </body>
86
22: </html>
```

بعد تنفيذ هذا الكود يتم تكوين جدول فى الصفحة يتكون من 12 صف و 12 عمود كما بالشكل:



1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

و يتضح من المثال ان المتغير x يتم تكراره 12 مرة فى كل تكرار واحد من المتغير y وبذلك يتم طبع الخلايا فى كل صف من صفوف الجدول عن طريق الوسم TD .