

الفصل السابع

المصفوفات

المصفوفات Arrays:

عرفنا سابقا ان المتغيرات تستطيع ان تحمل قيمة مفردة من البيانات سواء العددية او الحرفية و اذا احتجت الى اكثر من متغير فانك تقوم بتعريفه ولكن على عكس المتغيرات فانك اذا قمت بتعريف المصفوفة فتستطيع ان تقوم بتخزين قيم كثيرة فى نفس الوقت و تستطيع الوصول الى كل قيمة عن طريق فهرس للمصفوفة .

كما تستطيع ترتيب هذه البيانات المخزنة فى المصفوفة ترتيبا ابجديا او عدديا او ترتيب خاص يلائمك ايضا يمكنك عن طريق الحلقات المرور على كل القيم داخل المصفوفة وقرائنها وتعديلها اذا لزم الامر .

يبدأ فهرس المصفوفة دائما برقم صفر لذلك فدائما اخر رقم فى المصفوفة يقل بواحد عن عدد العناصر فى المصفوفة .

تعريف المصفوفات:

- استخدام الدالة array():

يمكن استخدام هذه الدالة لتعريف مصفوفة بالطريقة الاتية:

```
$myBooks = array("تعلم اكسيس", "تعلم دلفى 8", "الهاكرز");
```

يمكنك الان اذا اردت طباعة *الثانى* فى المصفوفة ان تستخدم العبارة:

```
print $myBooks[1];
```

- استخدام الاقواس المربعة:

يمكنك ايضا تعريف مصفوفة بطريقة سهلة عن طريق الاقواس كما يلى:

```
$myBooks[] = "تعلم اكسيس";
```

```
$myBooks[] = "تعلم دلفى 8";
```

```
$myBooks[] = "الهاكرز";
```

وبنفس الطريقة يمكنك الوصول الى العنصر الثالث مثلا عن طريق العبارة

```
print $myBooks[2];
```

ويمكن ايضا استخدام هذه الطريقة مع الطريقة السابقة لاضافة عناصر جديدة الى المصفوفة بعد تعريفها

المصفوفات المترابطة:

يمكن ايضا تعريف فهرس المصفوفات عن طريق قيم حرفية وليست عددية فقط و يتم ذلك بطريقتين:

- استخدام الدالة `array()`:

عند تعريف المصفوفة المترابطة يجب تعريف المفتاح و القيمة للمصفوفة كما يلي:

```
) employee = array$  
name=>"أيهاب",  
occupation=>"محاسب",  
age=>20,  
"Department no"=>"5C"  
);
```

يتضح من المثال السابق كيفية انشاء مصفوفة لتخزين بيانات الموظفين و ليس من الضروري ادراج علامتي تنصيب لعناصر هذه المصفوفة الا اذا كانت تتكون من اكثر من كلمة وفي بعض اصدارات PHP الحديثة قد يطلب وضع علامة ' او " حول كل فهرس حرفي .

ويمكن الوصول الى بيان السن مثلا هكذا:

```
$employee[age];
```

- استخدام الاقواس المربعة:

بنفس الطريقة يمكن تعريف المصفوفة بهذه الطريقة:

```
$employee[name] = "أيهاب";  
$employee[occupation] = "محاسب";  
$employee[age] = 20;  
$employee["Department no"] = "5C";
```

المصفوفات المتعددة:

يمكن تعريف المصفوفات المتعددة بتعريف بسيط على انها مصفوفة داخل مصفوفة لذلك حتى تستطيع الوصول الى العنصر الثالث فى المصفوفة الثانية يجب ان تذكر فهرس كلا المصفوفتين هكذا:

```
myarray[1][2];
```

مثال:

```
1: <html>
2: <head>
3: <title>مصفوفة متعددة</title>
4: </head>
5: <body>
6: <?php
7: $Family = array (
8: ( name=>"ايهاب",
9: occupation=>"محاسب",
10: age=>40,
11: depart=>"5C" ),
12: array ( name=>"جيهان",
13: =>"طالبة",
14: age=>22,
15: Sex=>"انثى" ),
16: array ( name=>"اسامة",
17: occupation=>"مهندس",
18: age=>25,
19: Sex=>"ذكر" )
20: );
21:
22: print $Family[0][occupation];
23: // النتيجة طالبة
24: ?>
25: </body>
```

مما سبق يتضح كيف يمكن تعريف مصفوفة ثنائية البعد و الوصول الى العنصر الثاني من اول عائلة .

الطرق المختلفة للوصول الى المصفوفة:

هناك عدة طرق هامة يجب ان نعرفها لكي نستطيع الوصول الى المصفوفة وقراءة وتعديل قيمها المختلفة .

- الحصول على حجم المصفوفة:

يمكن الحصول على عدد العناصر فى المصفوفة عن طريق استخدام الدالة count كما يلي :

```
$item_color = array("احمر", "اصفر", "اخضر", "ازرق");
print $item_color[count($item_color)- 1];
```

مما سبق يتضح كيفية الوصول الى اخر عنصر فى المصفوفة لان الدالة count تحتوى على عدد العناصر بدء من الرقم واحد وبالتالي فان تنفيذ العبارة السابقة يؤدى الى طباعة اللون الازرق .

- التكرار داخل مصفوفة:

يمكن استخدام حلقة التكرار القوية foreach مع المصفوفة للوصول الى كل عناصرها هكذا :

```
foreach( $array as $temp )
{
الكود الخاص بك هنا//
}
```

حيث المعامل \$array هو المصفوفة التى يتم التكرار داخلها و المعامل \$temp هو المخزن الوقت لكل عنصر داخل المصفوفة

مثال:

```
$item_color = array("احمر", "اصفر", "اخضر", "ازرق");
foreach ($item_color as $c){
    print "$c<br>" ;
}
```

والان ماذا عن المصفوفات المترابطة ؟
 يمكن التكرار داخل المصفوفة المترابطة عن طريق تعريف مفتاح العنصر
 وقيمة العنصر كما يلي:

```
foreach( $array as $key=>$value )
{
    //الكود الخاص بك هنا
}
```

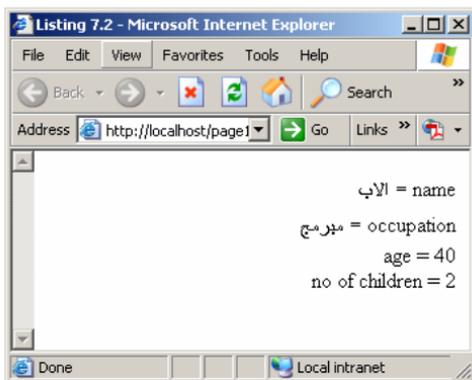
و يمكن اداء ذلك كما بالمثال الاتي:

```
1: <html dir="rtl">
2: <head>
3: <title>التكرار داخل المصفوفات</title>
4: </head>
5: <body>
6: <?php
7: $Family = array (
8: name=>"الاب",
9: occupation=>"ميرمج",
10: age=>40,
11: "no of children"=>"2"
12: );
13: foreach ( $Family as $key=>$val )
14: {
15: print "$key = $val<br>";
16: }
17:
18: ?>
```

19: </body>

20: </html>

مما سبق يتضح كيفية الوصول الى كل عنصر داخل المصفوفة المترابطة ومن المثال السابق يتم طباعة فهرس العنصر و قيمته كما بالشكل:



معالجة المصفوفات:

- دمج مصفوفتين:

يمكن دمج مصفوفتين و الوصول الى كل عناصرها عن طريق الدالة `array_merge` كمايلي

```
$first = array("a", "b", "c");
$second = array(1,2,3);
$third = array_merge( $first, $second );
foreach ( $third as $val )
{
print "$val<BR>";
}
```

نتيجة المثال السابق سيتم طبع قيم المصفوفتين اسفل بعضهم البعض

- إضافة عناصر الى المصفوفة:

يمكن اضافة المزيد من العناصر عن طريق الدالة `array_push()` التي تقوم بدمج عناصر جديدة في المصفوفة الاصلية و تقوم ايضا بارجاع اجمالي عدد العناصر في المصفوفة و يتضح ذلك من المثال:

```
$first = array("a", "b", "c");
$total = array_push( $first, 1, 2, 3 );
print "اجمالي عدد العناصر $total in \"$first<P>";
foreach ( $first as $val )
{
print "$val<BR>";
}
```

في المثال يتم طبع اجمالي عدد العناصر وهو 6 ثم يتم طبع عناصر المصفوفة الاصلية (a,b,c) ثم العناصر الجديدة (1,2,3)

لاحظ اننا قمنا بوضع علامة "\ " قبل المتغير `$first` حتى يتم طبع حرف الدولار كما هو و لا يعتبره المعالج كمتغير و يقوم بطبع قيمته وهذا ما ليس نريده.

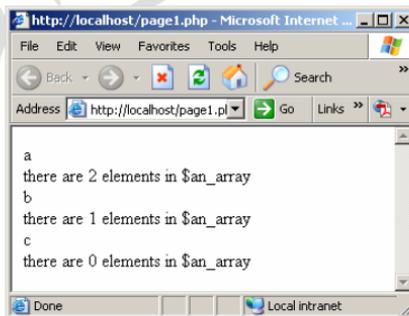
- حذف عناصر من مصفوفة:

تستطيع الدالة `array_shift()` ان تقوم بحذف اول عنصر من المصفوفة مع ارجاعه و نستطيع عن طريق حلقة تكرر `while` ان نحذف جميع عناصر المصفوفة بحيث نختبر انتهاء العناصر من المصفوفة عن طريق الدالة `count` كمايلي:

```
<?php
$an_array = array("a", "b", "c");
while ( count( $an_array ) )
{
    $val = array_shift( $an_array);
    print "$val<BR>";
    print "there are ".count($an_array)." elements
in \$an_array <br>";
}
?>
```

لاحظ ان الدالة `count` تصبح 0 عند حذف جميع العناصر من المصفوفة وهذا يعنى (`false`) بالنسبة لشرط الحلقة

وتكون نتيجة المثال السابق كما بالشكل:



تقطيع المصفوفات:

يمكنك عن طريق الدالة `array_slice()` ان تقوم باستخراج عناصر معينة من مصفوفة الى مصفوفة جديدة وهذه الدالة تقبل ثلاث معاملات الاول هو المصفوفة التي تريد تقطيعها و الثانى وهو فهرس اول عنصر تريد البدء فى استخراجها (`offset`) و المعامل الاخير اختياري وهو عدد العناصر التي تريد استخراجها واذا قمت بحذف هذا المعامل يتم استخراج جميع العناصر بدء من البداية المحددة وحتى النهاية و يتضح ذلك من المثال الاتي:

```
$first = array("a", "b", "c", "d", "e", "f");
$second = array_slice($first, 2, 3);
foreach ( $second as $var )
{
print "$var<br>";
}
```

هل تستطيع الان ان تستنتج نتيجة المثال السابق؟
يطبع المتصفح القيم `c` و `d` و `e` فقط لان المصفوفة تبدء من الصفر واذا قمت بتحديد رقم سالب فى معامل عدد العناصر فان جميع العناصر يتم استخراجها بدء من البداية المحددة .

- ترتيب المصفوفة:

من الاشياء الهامة جدا فى البرمجة عموما هو ترتيب البيانات حتى تكون نتيجة هذه البيانات عند طباعتها او عرضها على متخذى القرار مفهومة و منطقية و بدء من الاصدار الرابع للغة PHP تأتى اللغة بدوال قوية للترتيب نستعرضها فيما يلى:

الدالة `Sort()`:

تقوم هذه الدالة بترتيب البيانات فى المصفوفة ابجديا اذا كانت تحتوى على نصوص او عدديا اذا كانت تحتوى على ارقام ولا تقوم هذه الدالة بارجاع اى بيانات عكس مثلتها فى لغة Perl

```
$an_array = array("x","a","f","c");
sort( $an_array);
foreach ( $an_array as $var )
{
```

```
print "$var<BR>";  
}
```

يمكنك ان تقوم بعكس عملية الترتيب (تنازليا) عن طريق الدالة (rsort) بنفس الطريقة السابقة.

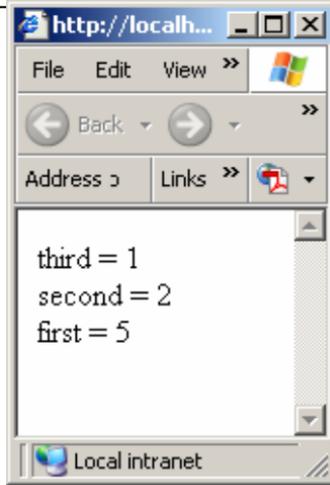
- فهرسة المصفوفات المترابطة:

لاتقوم باستخدام نفس الطريقة السابقة لترتيب البيانات مع المصفوفات المترابطة فسوف تقوم الدالة بالترتيب و لكن سيتم حذف مفاتيح العناصر و استبدالها بارقام تناسب الترتيب الجديد و سنتعرف الان كيفية ترتيب المصفوفات المترابطة بدون حدوث ذلك.

لكي تستطيع ترتيب المصفوفات المترابطة يجب ان تقوم باستخدام الدالة (asort) التي تستطيع ترتيب المصفوفة بدون حذف مفاتيح عناصرها لا المثال الاتي:

```
$first =  
array("first"=>5,"second"=>2,"third"=>1);  
asort( $first );  
foreach ( $first as $key => $val )  
{  
print "$key = $val<BR>";  
}
```

وتكون نتيجة المثال الاتي كما بالشكل:



وبنفس الطريقة يمكنك عكس الترتيب باستخدام الدالة `arsort()`

- ترتيب المصفوفات المترابطة عن طريق المفتاح:

يمكن عمل ذلك عن طريق الدالة `ksort()` التي تستقبل مصفوفة مترابطة ولا تقوم الدالة بارجاع اى قيمة .

مثال:

```
$first = array("x"=>5,"a"=>2,"f"=>1);
ksort( $first );
foreach ( $first as $key => $val )
{
print "$key = $val<BR>";
}
```

تكون نتيجة المثال السابق هو ترتيب العناصر بالشكل الاتى:

