

1 Arrays

Un *array* en PHP é en realidade un mapa ordenado. Un mapa é un tipo de datos que asocia valores con claves. Este tipo é optimizado para varios usos diferentes; pode ser usado como unha matriz real, unha lista (vector), táboa asociativa (caso particular de implementación dun mapa), dicionario, colección, pila, cola e probablemente mais. Xa que os valores dunha matriz poden ser outras matrices, tamén é posible crear árbores e matrices multidimensionais.

1.1 Sumario

- 1 Modos de Crear un array
 - ◆ 1.1 Empregando a sintaxe de corchetes cadrados
 - ◆ 1.2 Empregando a construción **array()**
- 2 Mostrar o contido do array
- 3 Paso de argumentos á funcións
- 4 Algunhas funcións útiles sobre arrays

1.2 Modos de Crear un array

1.2.1 Empregando a sintaxe de corchetes cadrados

Poderemos crear un array de forma rápida (dende a versión de PHP 5.4 empregando corchetes).

```
$datos= ['casa', 'coche', 'gato'];
```

Se queremos facer un **array asociativo** do tipo **chave => valor** con este formato:

```
// chave pode ser un 'integer' ou 'string'  
// valor pode ser calquera valor  
  
$datos= [  
    'propietario' => 'Antonio',  
    'domicilio' => 'Santiago de Compostela',  
    'idade' => 45  
];
```

Se **\$matriz** non existe aínda, será creada, así que este é tamén un xeito de crear unha matriz. Para modificar un certo valor, simplemente hai que asignar un novo valor a un elemento usando a súa clave. Para eliminar unha parella clave/valor, débese empregar a función **unset()** sobre ela.

Vexamos un exemplo:

```
<?php  
//Crear o array $productos:  
$productos[0] = "Azúcar";  
$productos[1] = "Aceite";  
$productos[2] = "Arroz";  
  
?>
```

1.2.2 Empregando a construción **array()**

Un valor array pode ser creado pola construción de linguaxe **array()**.

Esta toma un certo número de parellas **chave => valor** separadas con coma.

- **chave** pode ser un *integer* ou *string*
- **valor** pode ser calqueira valor

```
<?php  
$matriz = array("negocio" => "bar", 12 => true);  
  
echo $matriz["negocio"]; // bar  
echo $matriz[12]; // 1
```

?>

Se non se especifica unha clave para un valor dado, entón úsase o máximo dos índices enteiros, e a nova clave será ese valor máximo mais 1. Se se especifica unha clave que xa ten un valor asignado, ese valor será sobrescrito.

```
<?php

// Esta matriz é a mesma que ...
$datos = array(5 => 43, 32, 56, 'b' => 12, 60);
echo "{$datos[5]}"; // Amosará 43
echo "<br/>{$datos[6]}"; // Amosará 32
echo "<br/>{$datos[7]}"; // Amosará 56
echo "<br/>{$datos[8]}"; // Amosará 60
echo "<br/>{$datos['b']}<hr/>"; // Amosará 12

// ...esta outra matriz
array(5 => 43, "b" => 12, 7 => 56, 6 => 32, 60);
echo "{$datos[5]}"; // Amosará 43
echo "<br/>{$datos[6]}"; // Amosará 32
echo "<br/>{$datos[7]}"; // Amosará 56
echo "<br/>{$datos[8]}"; // Amosará 60
echo "<br/>{$datos['b']}"; // Amosará 12

?>
```

Así, outro xeito de inicializar o array \$productos sería o seguinte:

```
<?php
    $productos = array("Azúcar", "Aceite", "Arroz");
<?>
```

1.3 Mostrar o contido do array

- Mostrar o contido do array \$productos có bucle **for**:

```
// Cando se trata dun array simple que non é asociativo.

for ($i=0; $i<3; $i++)
{
    echo $productos[$i] , "<br />";
}
```

- Mostrar o contido do array \$productos coa estrutura **foreach**.

```
// Para arrays sinxelos
foreach (expresión_array as $valor)
    sentenzas

// Para arrays asociativos evaluando a chave e o seu valor.
foreach (expresión_array as $chave => $valor)
    sentenzas

// A estrutura foreach permítenos iterar sobre os arrays e obxectos evaluando os índices ou claves
// e o seu valor correspondente.

echo "<ul>";
foreach ( $productos as $p )
{
    echo "<li>", $p , "</li>";
}
echo "</ul>";
```

- Se queremos imprimir unha posición dun array asociativo, teremos que **empregar claves** para envolver ese array. Isto farémolo **cando empregemos comillas dobres**, por exemplo:

```
echo "Contido da país con código sp: {$países["sp"]}";
echo "Contido da país con código sp: {$países['sp']}";

// Ou ben concatenando.
```

```
echo "Contido da país con código sp: ", $países['sp'];  
echo "Contido da país con código sp: ".$países['sp'];
```

- Se queremos imprimir o valor dunha variable array para ver toda a estrutura podemos facer:

```
echo '<pre>';  
  
// Amosando o tipo de datos que contén en cada posición:  
var_dump($variableArray);  
  
// ou tamén, visualizando soamente os contidos:  
print_r($variableArray);  
  
echo '</pre>';
```

1.4 Paso de argumentos á funcións

<http://php.net/manual/es/functions.arguments.php>

1.5 Algunhas funcións útiles sobre arrays

- ◊ `count($array)` -> Devolve o número de posicións do vector.
- ◊ `sort($array)` -> Ordena o vector.
- ◊ `shuffle($array)` -> Mistura (desordena) o vector.
- ◊ `range($baixo, $alto)` -> Xera un array, sendo os seus elementos os valores dende o número \$baixo ata o número \$alto.
Exemplo: `$array = range(-4, 7);`