



Trabajar con arrays

Ordenar arrays

Podemos ordenar los elementos de un [array](#), tanto en orden ascendente (de menor a mayor), como en orden descendente (de mayor a menor). Si el array contiene elementos numéricos esto es fácil de entender, pero si los elementos son cadenas de texto, la ordenación se hará por orden alfabético.

Para ello utilizaremos las siguientes funciones, en todas ellas dentro del paréntesis (parámetro) escribiremos la variable que contiene el array:

En arrays indexados

- `sort($array)`: Ordena el array en orden ascendente
- `rsort($array)`: Ordena el array en orden descendente

En arrays asociativos

- `asort($array)`: Ordena el array en orden ascendente por valor.
- `arsort($array)`: Ordena el array en orden descendente por valor.
- `ksort($array)`: Ordena el array en orden ascendente por clave.
- `krsort($array)`: Ordena el array en orden descendente por clave.

Insertar elementos

Insertar un elemento en un array asociativo es lo mismo que asignarle al array un nuevo elemento con una nueva clave.

Ejemplo:

```
$cap=array ("España"=>"Madrid",           "Francia"=>"París",  
           "Inglaterra"=>"Londres",         "Italia"=>"Roma",  
           "Portugal"=>"Lisboa", "Alemania"=>"Berlín");
```

Si queremos insertar un nuevo elemento lo haremos de la siguiente manera:

```
$cap["Bélgica"]="Bruselas";
```

Añadir elementos en un array indexado es algo más complicado, ya que, si lo hacemos como en los arrays asociativos, en el corchete debemos escribir el número del elemento, en este caso, el valor dado sustituye al del elemento que ocupa ese lugar. Existen sin embargo una serie de funciones que añaden elementos a un array indexado. Estas son:

- `array_push($array, "elemento1", "elemento2");`
`array_unshift($array, "elemento1", "elemento2");`
- `$array2=array_pad($array, "tamaño", "elemento_relleno");`

Eliminar elementos

Existen también varias funciones para eliminar elementos de un array. Estas son:

- `$elem = array_shift($array)`: Elimina el primer elemento del array y lo devuelve en la variable `$elem`.
- `$elem =array_pop($array)` : Elimina el último elemento del array y lo devuelve en la variable `$elem`.
- `$array2 = array_unique($array)` : Devuelve un array en el que se eliminan los elementos repetidos que haya en el array original `$array`, el cual se conserva como estaba.

Arrays multidimensionales

Como hemos visto anteriormente, un array es una lista en la que se guardan varios datos o elementos. Sin embargo, muchas veces una lista no es suficiente para guardar los datos, y estos quedan mejor si los guardamos en una tabla.

La solución para guardar una tabla en un array es crear un array multidimensional, es decir un array en el que cada uno de sus elementos es a su vez otro array. En el array principal guardaremos la tabla. mientras que en cada uno de sus elementos o arrays secundarios guardaremos una fila de la tabla.

Es conveniente que para llevar un cierto orden en la tabla que queremos mostrar, cada uno de los arrays secundarios (filas) tenga el mismo número de elementos, así facilitaremos el trabajo de visualización en pantalla.

Si quieres aprender más acerca de los Arrays multidimensionales, te invitamos a ver el siguiente video: <https://www.youtube.com/watch?v=pKefSEbKBW0>