

Programación web con **PHP**



Arreglos

Temario

¿Que es un arreglo?.....	4
Creación de un arreglo.....	4
Acceder a contenido de un arreglo.....	4
Editar arreglos.....	6
Agregar elementos.....	6
Eliminar elementos.....	6
Ordenar arreglos.....	7
Unir elementos.....	8
Otras Funciones.....	9
array_flip.....	9
array_key_exist.....	9
array_merge.....	9
array_product.....	10
array_rand.....	10
array_replace.....	10
array_search.....	10
array_sum.....	11
array_unique.....	11
count.....	11
Lista completa de funciones para arreglos.....	12

¿Que es un arreglo?

Un arreglo o array es un grupo de elementos relacionados por medio de índices que permite manejarlos como si se tratara de una sola variable. Los elementos del arreglo pueden ser añadidos, editados y eliminados y pueden ser de cualquier tipo.

Existen dos tipos básicos de arreglos: numerarios y asociativos. Los arreglos numéricos tienen como índices números enteros donde el primer índice es 0, el segundo 1 y así sucesivamente. Los asociativos por otro lado pueden tener como índices cadenas de caracteres.

Creación de un arreglo

Para crear un arreglo utilizamos la función array.

Si el array es numérico solamente es necesario pasar como parámetros el contenido del array

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');
```

Si por el contrario es relacional es necesario especificar tanto el nombre del índice como su contenido.

```
$arreglo=array('nombre'=>'Pedro', 'apellido'=>'Gutierrez', 'edad'=>25, 'sexo'=>'Masculino');
```

Acceder a contenido de un arreglo

Para acceder a un valor de un arreglo debe especificarse el índice al cual queremos acceder, esto lo hacemos colocando el nombre del arreglo seguido por el nombre del índice entre corchetes.

En el caso de un arreglo numérico sera el numero correspondiente al elemento deseado.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
echo $arreglo[0];
```

Esto devolverá “Pedro” que es el elemento en la primera posición

En el caso de un arreglo relacional sera el nombre asignado al índice deseado

```
$arreglo=array('nombre'=>'Pedro','apellido'=>'Gutierrez','edad'=>25,'sexo'=>'Masculino');  
echo $arreglo['edad'];
```

Esto devolverá 25 que es el elemento bajo el índice edad.

Si queremos obtener todos los elementos de un arreglo utilizamos la función foreach(), esta función es muy similar a la estructura while, esta recorrerá cada elemento de el array hasta que no encuentre ninguno mas, para su utilización utilizamos la notación as seguida de una variable auxiliar que tomara el valor de cada elemento tomado del array en cada iteración, por lo tanto se ira reescribiendo con el valor del primer elemento en la primera iteración, con el valor del segundo elemento en la siguiente y así sucesivamente hasta llegar al final del array.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
foreach($arreglo as $elemento)  
{  
echo "Valor: ".$elemento;  
}
```

El resultado de ejecutar el código sera:

Valor:Pedro Valor:Marcos Valor:Maria Valor:Matias

Para un array asociativo donde es posible que nos interese mostrar no solo el valor sino también el nombre del índice en ves de utilizar una variable auxiliar utilizamos un par clave-elemento.

```
$arreglo=array('nombre'=>'Pedro','apellido'=>'Gutierrez','edad'=>25,'sexo'=>'Masculino');  
foreach($arreglo as $clave =>$elemento)  
{  
echo $clave." ":"$elemento;  
}
```

El resultado será:

nombre:Pedro apellido:Gutierrez edad:25 sexo:Masculino

Otra manera de mostrar todos los elementos en un arreglo es a través de la función print_r() pasando como parámetro el array.

Editar arreglos

Para editar el contenido de un arreglo simplemente debemos igualar el índice deseado al nuevo valor que deseamos que tome.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
$arreglo[0]='Pablo';
```

Aquí hemos modificado el valor del primer elemento de 'Pedro' a 'Pablo'

```
$arreglo=array('nombre'=>'Pedro','apellido'=>'Gutierrez','edad'=>25,'sexo'=>'Masculino');  
$arreglo['edad']=20;
```

Aquí hemos modificado el valor del elemento edad de 25 a 20.

Agregar elementos

Si deseamos insertar un elemento al final de un arreglo utilizamos la función `array_push()` pasando como primer parámetro el array y como segundo el dato a insertar.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
array_push($arreglo,'Ana');
```

Si mostráramos el array el resultado seria: Pedro Marcos Maria Matias Ana

Si deseamos insertar el elemento al principio del arreglo utilizamos la función `array_unshift()` pasando como primer parámetro el array y como segundo el dato a insertar.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
array_unshift($arreglo,'Ana');
```

Si mostráramos el array el resultado seria: Ana Pedro Marcos Maria Matias

Eliminar elementos

Para eliminar el ultimo elemento del array utilizamos la función `array_pop()` pasando como parámetro el array.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
array_pop($arreglo);
```

Si mostráramos el array el resultado seria: Pedro Marcos Maria

Para eliminar el primer elemento del array utilizamos la función `array_shift()` pasando como parámetro el array.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
array_shift($arreglo);
```

Si mostráramos el array el resultado seria: Marcos Maria Matias

Ordenar arreglos

Existen diferentes algoritmos de ordenamiento que pueden utilizarse al ordenar arreglos:

`SORT_REGULAR`: compara los elementos normalmente (no cambia los tipos)

`SORT_NUMERIC`: compara los elementos de forma numérica.

`SORT_STRING`: compara elementos como cadenas

`SORT_LOCALE_STRING`: compara elementos como cadenas, basándose en la configuración regional en uso.

`SORT_NATURAL`: compara los elementos como cadenas usando el orden natural, es decir, como lo haría un humano, un ejemplo donde puede verse la diferencia es nombres de archivos del tipo `img1`, `img2`, etc. Un orden regular ordenaría los nombres `img1`, `img10`, `img11`, `img12`, `img2`, `img3` mientras que el orden natural ordenara `img1`, `img2`, `img2`,..., `img10`, `img11`.

Para ordenar un arreglo numérico utilizaos la función `sort()` pasándole por parámetro el array y el tipo de algoritmo de ordenamiento que se va a utilizar.

```
$arreglo = array('Pedro', 'Marcos', 'María', 'Matias');  
$arreglo=sort($arreglo,SORT_STRING);
```

Si mostráramos el array el resultado seria: Marcos Maria Matias Pedro

Para ordenar un array asociativo por sus claves utilizamos la función `ksort()` pasando como

parámetros el array y el algoritmo de ordenamiento.

```
$arreglo=array('nombre'=>'Pedro','apellido'=>'Gutierrez','edad'=>25,'sexo'=>'Masculino');  
$arreglo=ksort($arreglo,SORT_STRING);
```

Si mostráramos el array el resultado seria: apellido=>Gutierrez edad=>25 nombre=>Pedro sexo=>Masculino

Para ordenar un array asociativo por sus valores utilizamos la función asort() pasando como parámetros el array y el algoritmo de ordenamiento.

```
$arreglo=array('nombre'=>'Pedro','apellido'=>'Gutierrez','edad'=>25,'sexo'=>'Masculino');  
$arreglo=asort($arreglo,SORT_STRING);
```

Si mostráramos el array el resultado seria: edad=>25 apellido=>Gutierrez sexo=>Masculino nombre=>Pedro

Unir elementos

Para unir los elementos de un array en un solo string utilizamos la función implode() pasando como parámetro el valor que separará los elementos y el array.

```
$arreglo = array('Pedro','Marcos','María','Matias');  
$listaNombres= implode(",",$arreglo);
```

Si mostráramos el array el resultado seria: Pedro,Marcos,Maria,Matias

Otras Funciones

array_flip

Intercambia las claves por sus valores y viceversa

```
//Array ( [nombre] => Pedro [apellido] => Gonzales [edad] => 30 )
$arreglo=array_flip($arreglo);
//Array ( [Pedro] => nombre [Gonzales] => apellido [30] => edad )
```

array_key_exist

Busca la clave dentro del array, si existe devuelve verdadero, en caso contrario devuelve falso

```
//Array ( [nombre] => Pedro [apellido] => Gonzales [edad] => 30 )
if (array_key_exists("nombre", $arreglo)) {
    echo "La clave existe";
}
else
{
    echo "La clave no existe";
}
```

array_merge

Combina dos o mas array creando un array con los elementos de los arrays pasados por parámetro.

```
$arreglo=array("nombre"=>"Pedro","apellido"=>"Gonzales","edad"=>30);
$arreglo2=array("telefono"=>"12312312");
$arreglo=array_merge($arreglo,$arreglo2);
//Array ( [nombre] => Pedro [apellido] => Gonzales [edad] => 30
[telefono] => 12312312 )
```

array_product

Calcula el producto de los valores dentro de un array

```
$arreglo=array(1,4,34,5,7);  
$producto=array_product($arreglo);  
// $producto=4760
```

array_rand

Selecciona una elemento del array de manera aleatorias

```
$arreglo=array(1,4,34,5,7);  
$valor=array_rand($arreglo);
```

array_replace

Reemplaza los elementos de un array por los del arreglo pasado como parámetro.

```
$arreglo = array("rojo", "amarillo", "azul", "verde");  
$reemplazos = array(0 => "rosado", 2 => "celeste");  
$arreglo = array_replace($arreglo, $reemplazos);  
// Array ( [0] => rosado [1] => amarillo [2] => celeste [3] => verde )
```

array_search

Busca un elemento en un array, si lo encuentra devuelve el índice o clave donde se encuentra, en caso contrario retorna falso.

```
$arreglo = array("rojo", "amarillo", "azul", "verde");
$clave = array_search('verde', $arreglo);
if($clave)
{
    echo "El elemento se encuentra en la ".$clave."
    posición";
}
else
{
    echo "El elemento no existe en el array";
}
```

array_sum

Calcula la suma de los elementos de un array

```
$arreglo=array(1,4,34,5,7);
$suma=array_sum($arreglo);
```

array_unique

Elimina los valores del array que estén duplicados

```
arreglo=array(1,4,34,5,7,4,6,8,1,2,3);
$arreglo=array_unique($arreglo);
//Array ( [0] => 1 [1] => 4 [2] => 34 [3] => 5 [4] => 7 [6] => 6 [7] =>
8 [9] => 2 [10] => 3 )
```

count

Devuelve la cantidad de elementos que contiene un array

```
$arreglo=array(1,4,34,5,7,4,6,8,1,2,3);
$cant=count($arreglo);
//$cant=11
```

Lista completa de funciones para arreglos

La lista completa se puede encontrar en la página oficial de php

<http://php.net/manual/es/ref.array.php>