

## MATRICES

En múltiples ocasiones existen variables que tienen información similar y se procesan de forma semejante. Para ello PHP posee un elemento denominado array. Un array es un conjunto de variables agrupadas bajo un único nombre. Cada variable dentro de la matriz se denomina elemento. Dentro de la misma matriz pueden existir variables de diferentes tipos y no es necesario que sean todas del mismo tipo.

### Existen dos tipos de matrices:

**Matriz Indexada:** Aquella cuyo acceso a los elementos se realiza por la posición que ocupan dentro de la estructura (se inician siempre desde la posición 0).

Ejemplo: `$amigos['0']`

**Matriz Asociativa:** Es aquella en la que los elementos están formados por pares clave-valor, en donde se accede al valor por su clave.

Ejemplo: `$amigos['edad']`

### Para crear matrices en PHP existen dos formas:

**La forma explícita,** consiste en asignarle un valor al elemento de una matriz ya sea indicando su posición o su clave. En caso de no indicarle una clave o una posición, el array tomara el valor siguiente al último valor introducido. Ejemplo:

```
$m1['nombre']='Javier'; // a la clave 'nombre' de 'matriz1' se asigna 'Javier'
$m1[]='Sonia'; // al siguiente elemento se asigna 'Sonia'
$m2['0']='Matías'; // al 1° elemento de 'matriz2' se asigna 'Matías'
$m2[]='María'; // al siguiente elemento de 'matriz2' se asigna 'María'

echo $m1['nombre'].'<br />'; // escribe 'Javier'
echo $m1['0'].'<br />'; // escribe 'Sonia' por ser el 1° elemento numerado
echo $m2['0'].'<br />'; // escribe 'Matías'
echo $m2['1']; // escribe 'María'
```

**Mediante array():** en el cual le pasamos los elementos como parámetros. En caso de matriz indexada toman la posición que ocupan en la creación de la matriz, mientras que los de la matriz asociativa se les asigna su valor mediante "=>". Ejemplo:

```
$m3=array('elem0','elem1','elem2','elem3','elem4');
$m4=array('c1'=>'elem1','c2'=>'elem2','c3'=>'elem3','c4'=>'elem4');
echo $m3['3'].'<br />'; // escribe 'elem3'
echo $m4['c1']; // escribe 'elem1'
```

### Cómo Recorrer los elementos de una Matriz:

- Para Matrices Indexadas: Basta conocer la cantidad de elementos de la matriz utilizando **count(matriz)** o **sizeof(matriz)** y luego utilizar un bucle **For**:

```
$matriz=array('a','b','c','d','e');
$elementos=count($matriz); // cuenta los elementos de $matriz
for ($i=0;$i<$elementos;$i++){
    echo $matriz[$i]; // imprime 'abcde'
}
```

- Cuando se trata de matrices asociativa no se puede aplicar el mismo tratamiento. Para ello existen funciones que deslazan el puntero dentro de la matriz.

Sintaxis	Acción
<code>reset(matriz)</code>	El puntero interno vuelve a la primera posición
<code>end(matriz)</code>	El puntero interno va a la última posición

<code>next(matriz)</code>	El puntero va al elemento siguiente
<code>prev(matriz)</code>	Accede al elemento anterior
<code>current(matriz)</code>	Devuelve el contenido del elemento actual

```
$masoc=array('c1'=>'elem1','c2'=>'elem2','c3'=>'elem3','c4'=>'elem4');
echo current($masoc).'\n />'; // imprime 'elem1'
echo next($masoc).'\n />'; // imprime 'elem2'
echo end($masoc).'\n />'; // imprime 'elem4'
echo prev($masoc).'\n />'; // imprime 'elem3'
echo reset($masoc); // imprime 'elem1'
```

Utilizando la estructura de control **Foreach()** se simplifica el acceso a las matrices asociativas, por ejemplo para la matriz anterior tendríamos el siguiente caso.

- Imprimir solo sus valores:

```
foreach ( $matrizasociativa as $valor ) {
    echo "$valor,"; // imprime 'elem1,elem2,elem3,elem4,'
}
```
- Imprimir sus claves y sus valores:

```
foreach ( $matrizasociativa as $clave=>$valor ) {
    echo "$clave => $valor,";
    // muestra: c1 => elem1,c2 => elem2,c3 => elem3,c4 => elem4,
}
```

## Manipulación masiva de matrices

### Inserción de elementos:

- `array_unshift(matriz, valor)`: Inserta un elemento al principio
- `array_push(matriz, valor)`: Inserta un elemento al final

```
$matriz = array('e','i','o');
array_unshift ($matriz,'a'); // agrega 'a' al principio
array_push ($matriz,'u'); // agrega 'u' al final
foreach ($matriz as $clave=>$valor){
    echo "$clave = $valor\n />"; // lo mostramos
}
```

### Eliminación de elementos

- `array_shift(matriz)`: elimina el primer elemento
- `array_pop(matriz)`: elimina el último elemento

```
$matriz = array('a','e','i','o','u');
array_shift ($matriz); // quita 'a'
array_pop ($matriz); // quita 'u'
foreach ($matriz as $clave=>$valor){
    echo "$clave = $valor\n />"; // lo mostramos
}
```

### Obtener el índice o la clave de un elemento:

- `array_keys (array,valor)`: Devuelve en un array la clave de un **valor** Buscado, si se omite **valor** devolverá todas las claves.

```
$vocales = array('c1'=>'a','c2'=>'e','c3'=>'i','c4'=>'o','c5'=>'u');
$key_de_i = array_keys ($vocales,'i'); // devuelve c3, posición de 'i'
echo $key_de_i[0].'\n />'; // en la matriz $key_de_i[0]
echo "---\n />";

$array_de_key = array_keys ($vocales); // devuelve (c1,c2,c3,c4,c5)
foreach ($array_de_key as $valor){ // en la matriz $array_de_key
    echo "$valor\n />"; // imprime todas las keys
}
```

- `array_search(valor,array)`: devuelve la clave de un **valor** Buscado, pero esta vez en una variable.

```
$vocales = array('c1'=>'a','c2'=>'e','c3'=>'i','c4'=>'o','c5'=>'u');
$key_d_i = array_search ('i',$vocales); // devuelve c3, posición de 'i'
echo $key_d_i; // en la variable $key_d_i
```

### Consultar si existe un elemento dentro de una matriz:

- `in_array`: busca un valor dentro de la matriz, devuelve 1 si lo encuentra. Se debe tener en cuenta que es sensible a mayúsculas y minúsculas.

```
$vocales = array('A','E','I','O','U');
if (in_array('A',$vocales)){ // busca 'A' en la matriz
    echo "Si existe"; // mensaje si la encuentra
}else{
    echo "No existe"; // mensaje si no la encuentra
}
```

### Pasar un array a un conjunto de variables:

- `list (var1,var2,...)`: los elementos de una array pasan a las variables indicadas por `list`, si el array tiene más elementos serán ignorados.

```
$color=array('azul','rojo','verde','violeta');
list($a1,$a2,$a3,$a4)=$color;
echo "$a1,$a2,$a3,$a4"; // imprime: azul,rojo,verde,violeta
```

### Extraer Clave y Valor de una matriz:

- `extract()`: extrae las claves de una matriz asociativa como variables con sus respectivos valores.

```
$vocales = array('c1'=>'a','c2'=>'e','c3'=>'i','c4'=>'o','c5'=>'u');
extract ($vocales);
echo "$c1,$c2,$c3,$c4,$c5"; // imprime: a,e,i,o,u
```

### Unión de dos matrices en una sola:

- `array_merge(primer_a_matriz,segunda_matriz)`: une dos matrices, el orden de los parámetros indica el orden en que se crea la matriz.

```
$vocMay = array('A','E','I','O','U');
$vocMin = array('a','e','i','o','u');
$final = array_merge($vocMay,$vocMin); // 1º mayúsculas luego minúsculas
foreach ($final as $clave=>$valor){
    echo "$clave => $valor<br />";
}
```

### Ordenación de matrices

Función	Criterio
<code>sort(\$matriz)</code>	Ordena una matriz en forma ascendente(matriz indexada)
<code>rsort(\$matriz)</code>	Ordena una matriz en forma descendente(matriz indexada)
<code>asort(\$matriz)</code>	Ordena una matriz en forma ascendente por valor(matriz asociativa)
<code>arsort(\$matriz)</code>	Ordena una matriz en forma descendente por valor(matriz asociativa)
<code>krsort(\$matriz)</code>	Ordena una matriz en forma ascendente por clave(matriz asociativa)
<code>krsort(\$matriz)</code>	Ordena una matriz en forma descendente por clave(matriz asociativa)