¿Qué es un array?

Un array es sencillamente una **tabla de valores**.
Cada uno de los elementos de esa tabla se identifica por medio de un nombre (común para todos) y un índice (que diferenciaría a cada uno de ellos).
La sintaxis que permite definir elementos en un array es esta: **$nombre[indice]**

$nombre utiliza exactamente la misma sintaxis empleada para definir variables, con la única particularidad de que ahora deben añadírsele los corchetes y los índices.
El índice puede ser un número (habría que escribirlo dentro del corchete sin comillas), una cadena (que habría que poner en el corchete encerrada entre comillas sencillas), o una variable PHP en cuyo caso tampoco necesitaría ir entre comillas.
Cuando los índices de un array son números se dice que es **escalar** mientras que si fueran cadenas se le llamaría array **asociativo**.

Arrays escalares

Los elementos de un array escalar puede escribirse con una de estas sintaxis: **$a[]=valor** ó **$a[xx]=valor**

En el primero de los casos PHP asigna los índices de forma automática atribuyendo a cada elemento el valor entero siguiente al último asignado. Si es el primero que se define le pondrá índice 0 (CERO). En el segundo de los casos, seremos nosotros quienes pongamos (xx) el número correspondiente al valor del índice.

Si ya existiera un elemento con ese índice, se cambiaría el valor de su contenido, en caso contrario creará un nuevo elemento del array y se le asignaría como valor lo especificado detrás del signo igual, que de la misma forma que ocurría con las variables debería ir entre comillas si fuera una cadena o sin ellas, si se tratara de números.

Arrays asociativos

Los elementos de un array asociativo pueden escribirse usando la siguiente sintaxis: **$a['indice']=valor**
En este caso estamos obligados a escribir el nombre del índice que habrá de ser una cadena y debe ponerse entre comillas. Tanto en este supuesto como en el anterior, es posible y bastante frecuente utilizar como índice el contenido de una variable. El modo de hacerlo sería: **$a[$ind]=valor**
En este caso, sea cual fuere el valor de la variable $ind, el nombre de la variable nunca se pone entre comillas.

Tablas (arrays) unidimensionales

Mediante el uso de arrays podemos utilizar el mismo nombre para varias variables diferenciándolas entre sí mediante índices distintos

Tablas unidimensionales					
Array escalar			Array asociativo		
Variable	**Indice**	**Valor**	**Variable**	**Indice**	**Valor**
$a[0]	0	Domingo	$a['Primero']	Primero	Domingo
$a[1]	1	Lunes	$a['Segundo']	Segundo	Lunes
$a[2]	2	Martes	$a['Tercero']	Tercero	Martes
$a[3]	3	Miércoles	$a['Cuarto']	Cuarto	Miércoles
$a[4]	4	Jueves	$a['Quinto']	Quinto	Jueves
$a[5]	5	Viernes	$a['Sexto']	Sexto	Viernes
$a[6]	6	Sábado	$a['Septimo']	Septimo	Sábado

Uso de arrays

```
<?
# Crearemos un array escalar (basta con definir un elemento)
$a[2]="Este elemento es el segundo del array";
# creemos un nuevo elemento de ese array
# esta vez de forma automática
# si ponemos corchetes vacíos va añadiendo índices automaticamente
$a[]="¿Será este tercero?";
# comprobemos que le ha puesto índice 3
echo "El elemento ".$a[3]." tiene indice 3 (siguiente a 2) <br>";
# ahora insertemos un nuevo elemento con indice 32
$a[32]="Mi indice es 32";
# insertemos otro elemento de forma automática
$a[]= "¿Irá a parar al indice 33 este elemento?";
#  la inserción se hará con indice 33, comprobémoslo
print "Vemos que contiene el elemento de indice 33 ...".$a[33]."<br>";
# ¿qué ocurrirá si pido que imprima el elemento 21 que nadie ha definido
# seguramente estará vacio, ¡¡comprobémoslo!!
print ("Aqui--> ". $a[21]. "<--- si es que hay algo<br>");
# ahora crearemos un nuevo array llamado $b
# insertémosle de forma automatica su PRIMER elemento
$b[]="Estoy empezando con el array b y mi indice será cero";
# comprobemos que efectivamente ha empezado con indice CERO
print ($b[0]."<br>");
# veamos ahora eso de los arrays asociativos
# creemos uno llamado $c con varios elementos
$c["objeto"]="coche";
$c["color"]="rojo";
$c["tamaño"]="ideal";
$c["marca"]= "Ferrari";
$c["precio"]="prohibitivo para un humilde docente";
#encadenemos variables para hacer una salida
# pondremos cadenas " " para que no aparezcan los textos
# pegados unos a otros..
$salida="<H2> El ". $c["objeto"] ." ".$c["marca"]." ".$c["color"];
$salida .=" tiene el tamaño ideal ".$c["tamaño"];
$salida .=" y su precio es ".$c["precio"];
$salida .="</H2>";
print $salida;
# sigamos experimentando ahora
# ¿qué ocurriría si nos olvidamos de poner nombre al indice
# e insertamos un corchete vacio ¿lo crearía?¿que indice pondria?
# probemos ....
$c[]="¿creará un array escalar nuevo y le pondrá indice cero?";
# tratemos ahora de visualizar esa variable
# probemos a escribir $c[0] porque PHP
# habrá entendido que queremos un array escalar
# y como no existe ninguno con ese nombre empezará por cero
# comprobémoslo
echo $c[0];
?>
```

```
// array numérico
$modulos1 = array(0 => "Programación", 1 => "Bases de datos", ..., 9 => "Desarrollo web en entorno servidor");
// array asociativo
$modulos2 = array("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor");
```

Para hacer referencia a los elementos almacenados en un array, tienes que utilizar el valor clave entre corchetes:

```
$modulos1 [9]
$modulos2 ["DWES"]
```

En PHP no es necesario que indiques el tamaño del array antes de crearlo. Ni siquiera es necesario indicar que una variable concreta es de tipo array. Simplemente puedes comenzar a asignarle valores:

```
// array numérico
$modulos1 [0] = "Programación";
$modulos1 [1] = "Bases de datos";
...
$modulos1 [9] = "Desarrollo web en entorno servidor";
// array asociativo
$modulos2 ["PR"] = "Programación";
$modulos2 ["BD"] = "Bases de datos";
...
$modulos2 ["DWES"] = "Desarrollo web en entorno servidor";
```

Ni siquiera es necesario que especifiques el valor de la clave. Si la omites, el array se irá llenando a partir de la última clave numérica existente, o de la posición 0 si no existe ninguna:

```
$modulos1 [ ] = "Programación";
$modulos1 [ ] = "Bases de datos";
...
$modulos1 [ ] = "Desarrollo web en entorno servidor";
```

Recorrer arrays (I).

Las cadenas de texto o strings se pueden tratar como arrays en los que se almacena una letra en cada posición, siendo 0 el índice correspondiente a la primera letra, 1 el de la segunda, etc.

```
// cadena de texto
$modulo = "Desarrollo web en entorno servidor";
// $modulo[3] == "a";
```

Para recorrer los elementos de un arrays, en PHP puedes usar un bucle específico: foreach . Utiliza una variable temporal para asignarle en cada iteración el valor de cada uno de los elementos del arrays. Puedes usarlo de dos formas. Recorriendo sólo los elementos:

```
$modulos = arrays("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor");
foreach ($modulos as $modulo) {
print "Módulo: ".$modulo. "<br />"
}
```

Página PHP que utilice foreach para mostrar todos los valores del array $_SERVER en una tabla con dos columnas. La primera columna debe contener el nombre de la variable, y la segunda su valor.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
http://www.w3.org/TR/html4/loose.dtd">
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Tabla con los valores del array $_SERVER utilizando foreach -->
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>Tabla</title>
<style type="text/css">
td, th {border: 1px solid grey; padding: 4px;}
th {text-align:center;}
table {border: 1px solid black;}
</style>
</head>
<body>
<table>
<tbody>
<tr>
<th>Variable</th>
<th>Valor</th>
</tr>
        <?php
                foreach ($_SERVER as $variable => $valor) {
                print "<tr>";
                print "<td>".$variable."</td>";
                print "<td>".$valor."</td>";
                print "</tr>";
                }
        ?>
</tbody>
</table>
</body>
</html>
```

La inicialización del array en el código siguiente,

```
$a[0] = 0;
$a[1] = "uno";
$a["tres"] = 3;
```

No es necesario que la clave de un array sea siempre numérica o texto. Pueden mezclarse ambos tipos de valores.