

¿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[índice]

\$nombre utiliza exactamente la misma sintaxis empleada para definir variables, con la única particularidad de que ahora deben añadirse 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 *cadena*s se le llamaría array **asociativo**.

Arrays escalares

Los elementos de un array escalar puede escribirse con una de estas sintaxis:

\$a[]=valor

0

\$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 cada vez

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	Índice	Valor	Variable	Índice	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";
# creamos un nuevo elemento de ese array
# esta vez de forma automática
# si ponemos corchetes vacíos va añadiendo índices automáticamente
$a[]="¿Será este tercero?";
# comprobemos que le ha puesto índice 3
echo "El elemento ".$a[3]." tiene índice 3 (siguiente a 2) <br>";
# ahora insertemos un nuevo elemento con índice 32
$a[32]="Mi índice es 32";
# insertemos otro elemento de forma automática
$a[]="¿Irás a parar al índice 33 este elemento?";
# la inserción se hará con índice 33, comprobémoslo
print "Vemos que contiene el elemento de índice 33 ...".$a[33]."<br>";
# ¿qué ocurrirá si pido que imprima el elemento 21 que nadie ha definido
# seguramente estará vacío, ¿comprobémoslo!!
print ("Aquí--> ". $a[21]. "<--- si es que hay algo<br>");
# ahora crearemos un nuevo array llamado $b
# insertémosle de forma automática su PRIMER elemento
$b[]="Estoy empezando con el array b y mi índice será cero";
# comprobemos que efectivamente ha empezado con índice CERO
print ($b[0]."<br>");
# veamos ahora eso de los arrays asociativos
# creamos 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 índice
# e insertamos un corchete vacío ¿lo crearía? ¿qué índice pondría?
# probemos ....
$c[]="¿creará un array escalar nuevo y le pondrá índice cero?";
# tratemos ahora de visualizar esa variable
```

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['índice']=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:

`$a[$índ]=valor`

En este caso, sea cual fuere el valor de la variable *\$índ*, el nombre de la variable **nunca** se pone entre comillas.

```
# 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];
?>
```

ejemplo19.php

Ejercicio nº 14

Crea un formulario -puedes llamarlo **formulario14.php**- en el que se permita introducir un número cualquiera en una caja de texto. Al enviar el formulario deberá aparecer -un mensaje **en letra**- indicando **el resto de dividir entre doce** el valor transferido.

El procedimiento podría ser el siguiente:

- Crear un array conteniendo los nombres de todos restos posibles (división exacta, uno, dos, tres... hasta once).
- Asignar como índices los valores numéricos correspondientes a los literales que contiene cada elemento.
- Comprobar el resto de la división e imprimir el valor del elemento del array cuyo índice coincide con ese resto.

Anterior



Índice



Siguiente

