

ARRAYS PHP

ARRAYS PHP

[Arrays asociativos](#)

[Arrays bidimensionales](#)

[Iterador foreach\(\)](#)

Un array es un tipo de dato capaz de almacenar múltiples valores. Se utiliza cuando tenemos muchos datos parecidos, por ejemplo, para almacenar la temperatura media diaria en Málaga durante el último año podríamos utilizar las variables *temp0*, *temp1*, *temp2*, *temp3*, *temp4*, ... *y así hasta 365 variables distintas pero serí apocoprá ctico; es mejor utilizar un array denombre temp* y usar un índice para referirnos a una temperatura concreta.

Ejemplo de Array:

```
<?php
$temp[0] = 16;
$temp[1] = 15;
$temp[2] = 17;
$temp[3] = 15;
$temp[4] = 16;
echo "La temperatura en Algeciras el cuarto día del año fue de "; echo
$temp[3], "oC";
?>
```

- Los valores de un array se pueden asignar directamente en una línea. El índice comienza en 0.

```
<?php
$temp = array(16, 15, 17, 15, 16);
echo "La temperatura en Algeciras el cuarto día del año fue de "; echo
$temp[3], "oC";
?>
```

- A partir de PHP 5.4 se puede utilizar la **sintaxis abreviada** (utilizando corchetes) para definir un array.

```
<?php
    $color = ["verde", "amarillo", "rojo", "azul", "blanco", "gris"];
    echo "Mañana me pongo una camiseta de color ", $color[rand(0, 5)], ".";
?>
```

- Para hacer referencia a los valores individuales de la matriz, se deben utilizar índices, que se escriben entre corchetes ([]). Si al crear la matriz no se han indicado otros valores de índices, el primer término tiene el índice [0]

```
<?php
    $nombres = ["Ana", "Bernardo", "Carmen"];

    print "<p>$nombres[1]</p>\n";
    print "<p>$nombres[0]</p>\n";
?>
```

- Si se solicita un valor no definido de una matriz, se produce un aviso (undefined offset). Los avisos no interrumpen la ejecución del programa, pero se deben corregir porque el programa seguramente no tendrá el comportamiento esperado:

```
<?php
    $nombres = ["Ana", "Bernardo", "Carmen"];
    print "<p>$nombres[3]<p>\n";

?>

//Notice: Undefined offset: 3 in ejemplo.php on line 4
```

- Se puede crear una matriz vacía (para añadirle posteriormente elementos).

```
<?php
    $nombres = [ ];

?>
```

- Se pueden crear arrays de tamaño fijo con **SplFixedArray** que son más eficientes en cuanto a uso de la memoria y más rápidas en las operaciones de lectura y escritura.
- La función **var_dump()** se utiliza para mostrar el tipo y el valor de un dato, en este caso muestra los tipos y valores de cada uno de los elementos del array.

```
<?php
$a = new SplFixedArray(10);
    $a[0] = 55;
    $a[2] = 6;
    $a[6] = 2333;
    // Los valores del array que no se han inicializado son NULL
foreach ($a as $elemento) {
    var_dump($elemento);
    echo "<br>";
}
?>
```

Arrays asociativos

En un array asociativo se pueden utilizar índices que no son numéricos, a modo de claves, ni ser correlativos. Veamos un ejemplo de un array asociativo que almacena alturas (en centímetros) y que como índice o clave utiliza nombres

Al crear matrices asociativas, debemos indicar el valor de los índices, utilizando la notación

índice => valor:

```
<?php
$estatura = array("Rosa" => 168, "Ignacio" => 175, "Daniel" => 172, "Rubén" =>
182);
echo "La estatura de Daniel es ", $estatura['Daniel'] , " cm"; ?>
```

Con los arrays asociativos se puede usar también la sintaxis abreviada que emplea corchetes.

```
<?php
$estatura = [
    "Rosa" => 168,
    "Ignacio" => 175,
    "Daniel" => 172,
    "Rubén" => 182,
];
echo "La estatura de Daniel es ", $estatura['Daniel'] , " cm"; ?>
```

Arrays bidimensionales

Un array bidimensional utiliza dos índices para localizar cada elemento. Podemos ver este tipo de datos como un array que, a su vez, contiene otros arrays. En el siguiente ejemplo se define un array con 4 elementos que, a su vez, son un array asociativo cada uno de ellos.

```
<?php
    $persona = array (
        array( "nombre" => "Rosa", "estatura" => 168, "sexo" => "F"),
        array( "nombre" => "Ignacio", "estatura" => 175, "sexo" => "M"),
        array( "nombre" => "Daniel", "estatura" => 172, "sexo" => "M"),
        array( "nombre" => "Rubén", "estatura" => 182, "sexo" => "M")
    );
    echo "<b>DATOS SOBRE EL PERSONAL<b><br><hr>";

?>
```

Iterador foreach()

El iterador foreach se utiliza para recorrer todos los elementos de un array sin que tengamos que preocuparnos por los índices ni por el tamaño del array.

Es común cometer errores al utilizar arrays por no establecer bien el valor inicial o el valor final en el bucle que la recorre, o por no determinar bien el tamaño. Con foreach nos podemos despreocupar de todo eso, simplemente recorreremos todo el array de principio a fin. Veamos un ejemplo.

Ejemplo:

```
<?php
    $cajonDeSastre = [30, -7, "Me gusta el queso", 56, "¡eh!", 237];
    foreach ($cajonDeSastre as $cosa) {
        echo "$cosa<br>";
    }

?>
```

Continúa con: (Matrices asociativas)

<https://www.mclibre.org/consultar/php/lecciones/php-matrices.html>