

1. ARRAYS ASOCIATIVOS

Hemos visto que la diferencia entre una variable y un array es que, en este último caso, se accede a través de un **índice** a los valores que almacenamos en el array.

Indicábamos que ese índice era numérico y que, por defecto, empezaba por cero.

Bien, existe otra forma de establecer los índices de un array. En este caso no se trata de utilizar valores numéricos sino **cadena de texto**, lo que facilita la forma de identificar el elemento de un array que queremos utilizar.

Se conoce como **array asociativo** a este tipo de estructura y vamos a estudiarla en esta lección.

En el código siguiente puedes ver un ejemplo de esto. Se trata de un array de nombre **monedas** con cinco elementos.

```
<?php
$monedas = array("España" => "Euro", "Francia" =>
"Euro", "Reino Unido" => "Libra esterlina",
"USA" => "Dólar", "Argentina" => "Peso");
?>
```

Ahora no se ha creado el array de forma que el primer elemento tenga el índice **0**, el segundo el índice **1**, etc., sino que, en lugar de ello, cada índice coincide con el nombre de un país.

Así, el primer elemento del array es **\$monedas["España"]** en lugar de **\$monedas[0]**.

Esto hace que sea más fácil para nosotros trabajar con los elementos del array. Así, sin necesidad de recordar cómo lo creamos, podremos escribir la expresión **\$monedas["España"]** sabiendo perfectamente que nos estamos refiriendo a la moneda de España.

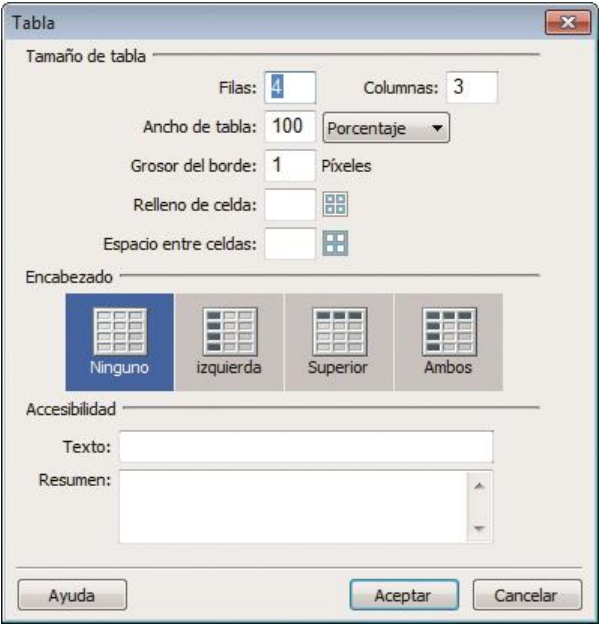
Fíjate que como se tiene que indicar el índice y el valor de cada elemento, se utiliza el operador **=>** (el signo igual seguido del signo mayor que) como si fuera una "flecha" que apunta al valor almacenado.

Así pues la expresión **"España"=> "Euro"** está indicando que en el elemento del array cuyo índice es "España" se guarda el valor "Euro". Pues bien, si has entendido esto, no será muy difícil trabajar con el array asociativo...

Vamos a crear una tabla en esta página web para mostrar el listado de los países y sus correspondientes monedas oficiales.

Podríamos ir escribiendo el código HTML correspondiente a la tabla, pero ya que estamos en Dreamweaver, aprovecharemos sus características visuales.

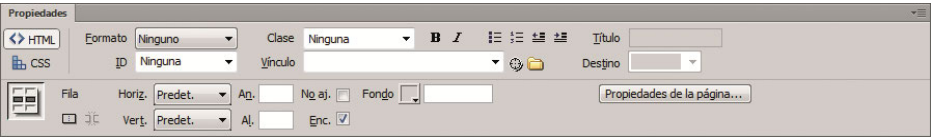
Para ello, trabajaremos en la vista **Diseño** y elegiremos la opción **Tabla** en su menú **Insertar**. Al hacerlo, Dreamweaver presenta un cuadro de diálogo donde se nos pregunta cuántas filas y columnas tendrá la nueva tabla, además de otras opciones de la misma.



Una vez creada la tabla de esta forma, es sencillo establecer propiedades de alguna celda, de alguna fila o de la propia tabla en sí. Para ello, utilizaremos el **Inspector de propiedades** que aparece en la parte inferior de la ventana de Dreamweaver.

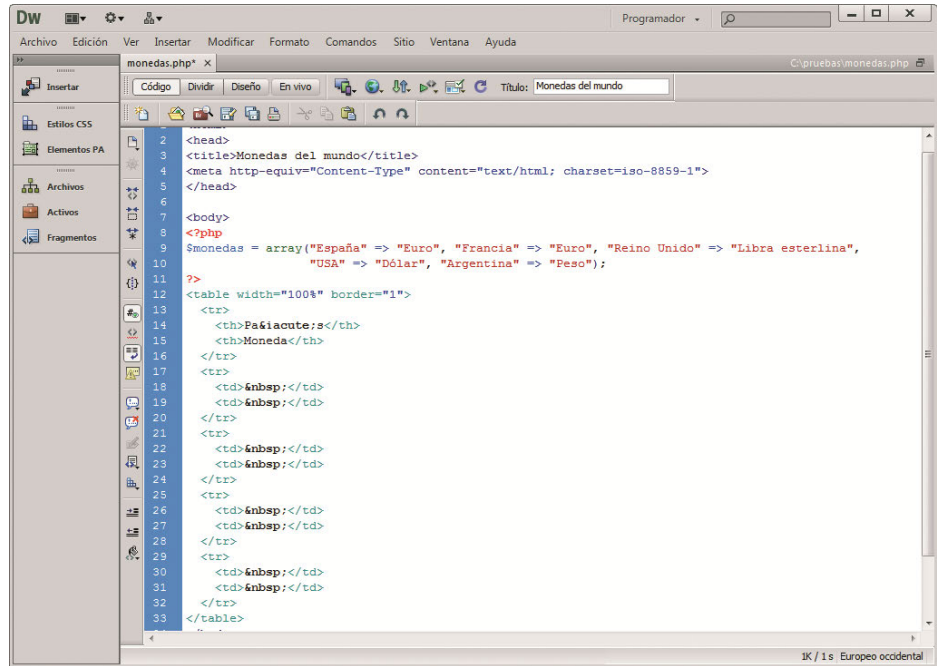
Puedes mostrar este panel bien con el comando **Propiedades** del menú contextual de la selección realizada previamente o bien a través del menú **Ventana** de Dreamweaver.

Por ejemplo, en la figura siguiente puedes ver las propiedades de una fila de la tabla, previamente seleccionada. Vemos que hemos activado la casilla **Enc.** para indicar que las celdas de dicha fila son de encabezado.



Dreamweaver genera el código HTML necesario para representar la tabla que hemos creado en la vista **Diseño**. Por ejemplo, vemos que se crea la tabla con la etiqueta `<table>`, las filas con `<tr>`, las celdas con `<th>` si son de encabezado o `<td>` si son de datos, etc.

Ahora podremos ir completando la tabla para que muestre los valores almacenados en el array asociativo **\$monedas**.



2. EL BUCLE FOREACH

Acceder a los elementos de un array asociativo es prácticamente igual que con cualquier otro array. La diferencia se encuentra en los índices que tendremos que utilizar.

Podemos utilizar un bucle de repetición muy útil para estos casos. Así, en lugar de escribir línea a línea el código necesario para cada celda de la tabla, ¿por qué no recorremos el array y escribimos una única línea para ello?

El problema es que para recorrer el array con un bucle, necesitaríamos utilizar un índice numérico. Sin embargo, hay otro bucle que nos sirve muy bien: el bucle **foreach**.

Fíjate en el código de la página siguiente.

```
<?php
$monedas = array("España" => "Euro", "Francia" =>
"Euro", "Reino Unido" => "Libra esterlina",
                  "USA"    => "Dólar",  "Argentina" =>
"Peso");
?>
<table width="100%" border="1">
  <tr>
    <th>País</th>
    <th>Moneda</th>
  </tr>
  <?php
  foreach ($monedas as $clave => $valor)
  {
    echo "<tr>";
    echo "<td>$clave</td>";
    echo "<td>$valor</td>";
    echo "</tr>";
  }
  ?>
</table>
```

Repasemos un poco el nuevo código.

- Se utiliza el bucle **foreach** para iterar por todos los elementos del array **monedas**. Lo que hace este bucle es empezar por el primer elemento y moverse en cada iteración al siguiente hasta que llegue al final.

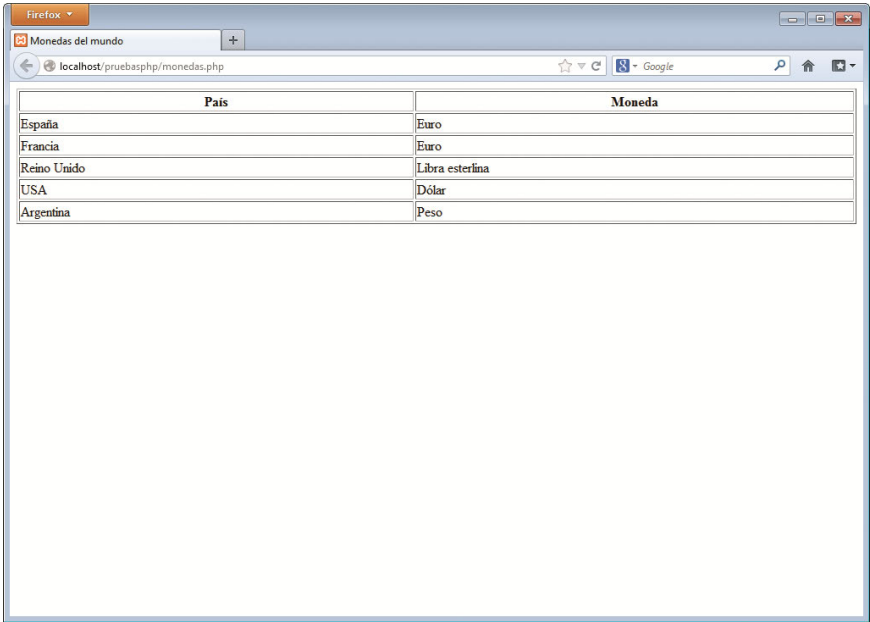
Para eso vemos que se indica el nombre del array (**\$monedas**), después la palabra **as** y una variable que sirve para poder acceder al valor almacenado en cada elemento del array.

En nuestro caso, como nos interesa también el propio índice, utilizamos la expresión **\$clave => \$valor**, de forma que obtenemos tanto el índice (**\$clave**) como el valor almacenado (**\$valor**) en el elemento.

- El resto del código completa el código HTML de la tabla escribiendo, en cada iteración, una fila con dos celdas en las que se incluye el nombre del índice y el valor almacenado.

No es difícil, simplemente hay que pensar que el bucle repetirá este código HTML para cada uno de los elementos del array **monedas**.

Vemos que con mucho menos código hemos conseguido la tabla completa con cada uno de los países y sus correspondientes monedas.

A screenshot of a web browser window titled "Monedas del mundo". The address bar shows "localhost/pruebasphp/monedas.php". The browser displays a table with two columns: "País" and "Moneda". The table contains five rows of data: España (Euro), Francia (Euro), Reino Unido (Libra esterlina), USA (Dólar), and Argentina (Peso).

País	Moneda
España	Euro
Francia	Euro
Reino Unido	Libra esterlina
USA	Dólar
Argentina	Peso

Es muy frecuente utilizar este tipo de bucles para generar, en cada iteración, el código HTML correspondiente y no tener que escribirlo una y otra vez.

Lo bueno del bucle **foreach** es que se encarga de acceder a cada elemento del array, sin que nosotros tengamos que preocuparnos de cómo indicar cada uno de sus índices.

3. ARRAYS MULTIDIMENSIONALES

Una característica muy potente de los arrays es que cada uno de sus elementos puede ser, a su vez, otro array, en lo que se conoce como **arrays multidimensionales**.

Esto permite representar estructuras más complejas de datos, al estilo de tablas o bases de datos.

Veamos un ejemplo para entenderlo mejor. Vamos a representar, mediante un array, un listado de productos. Para ello, tendremos de cada uno de ellos un código o identificador de producto, su nombre y precio.

```
<?php
$productos = array(array(1, "Figurita cisne", 421.0),
                    array(2, "Despertador", 79.78),
                    array(3, "Reloj de pared", 625.5));
?>
```



Debes utilizar el punto para separar la parte decimal de valores numéricos en el código PHP, aunque la configuración regional utilice una coma para estos casos.

Como se ha indicado, utilizaremos un array multidimensional (en este caso solo de dos dimensiones) para representar una lista de productos.

Por lo tanto vemos que el primer elemento del array **productos** tiene que ser, a su vez, otro array con los detalles del primer producto.

O sea, que el primer elemento del array **productos** es un array con los elementos **1**, **"Figurita cisne"** y **421.0**, representando el identificador, nombre y precio del producto, respectivamente. Lo mismo para los otros dos productos. Fíjate en la flexibilidad que proporciona el hecho de que cada uno de los elementos de un array puede ser de un tipo de datos distinto.

Así vemos que hemos utilizado el tipo entero, cadena de texto y un valor numérico con parte decimal para representar el primer producto de la lista que estamos creando. De la misma forma que hemos creado un array de dos dimensiones, ya que cada uno de sus elementos es, a su vez, un array, podríamos hacer lo mismo para crear más dimensiones.

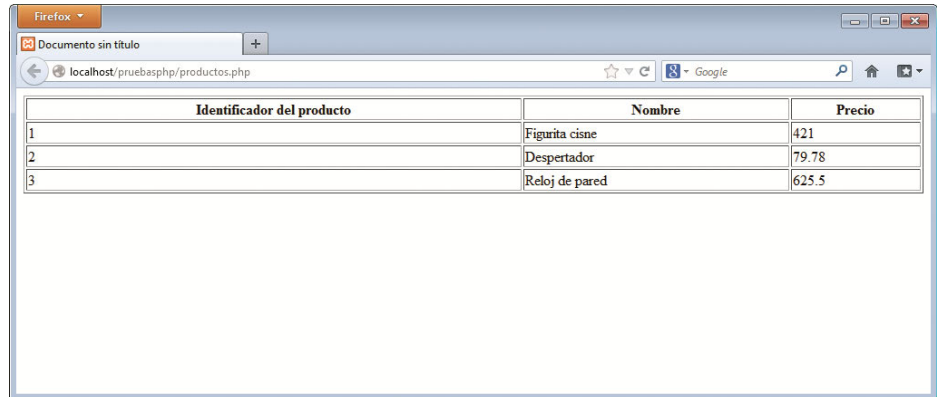
Así, si los elementos del array interno fueran, a su vez, otro array, tendríamos tres dimensiones. Por ejemplo, esto podría tener sentido para representar que cada uno de los productos pertenece a un tipo o categoría de mercadería.

Bien, una vez hemos creado el array multidimensional, acceder a sus elementos es prácticamente igual que en el caso de un array de una dimensión.

```
<table width="100%" border="1">
  <tr>
    <th>Identificador del producto</th>
    <th>Nombre</th>
    <th>Precio</th>
  </tr>
  <tr>
    <td><?php echo $productos[0][0]; ?></td>
    <td><?php echo $productos[0][1]; ?></td>
    <td><?php echo $productos[0][2]; ?></td>
  </tr>
  <tr>
    <td><?php echo $productos[1][0]; ?></td>
    <td><?php echo $productos[1][1]; ?></td>
    <td><?php echo $productos[1][2]; ?></td>
  </tr>
  <tr>
    <td><?php echo $productos[2][0]; ?></td>
    <td><?php echo $productos[2][1]; ?></td>
    <td><?php echo $productos[2][2]; ?></td>
  </tr>
</table>
```

Por ejemplo, para acceder al primer elemento del array, utilizamos **\$productos[0]**, pero como este es, a su vez, otro array, tendremos que volver a utilizar los corchetes para indicar qué elemento queremos obtener o establecer del array interno: **\$productos[0][0]**.

La expresión anterior se evaluaría a 1, que es el identificador del primer producto.



Identificador del producto	Nombre	Precio
1	Figurita cisne	421
2	Despertador	79.78
3	Reloj de pared	625.5

La expresión **\$productos[0]** representa el primer producto almacenado en el array; mientras que **\$productos[0][1]** representa el nombre de ese producto, ya que sí está almacenado en el array **productos**. Lo mismo para el precio **\$productos[0][2]** y para el resto de productos.

Podremos utilizar arrays asociativos incluso al crear un array de más de una dimensión. El funcionamiento será exactamente el mismo que ya hemos visto.

Por ejemplo, tal como hemos escrito nuestro código, tenemos que saber que en la posición 0 de cada elemento está el identificador del producto; en la posición 1, su nombre; y en la posición 2, su precio ¿No sería mejor expresarlo justamente con esas palabras?

```
<?php
$productos = array(
    array("Identificador" => 1, "Nombre" => "Figurita
    cisne", "Precio" => 421.0),
    array("Identificador" => 2, "Nombre" =>
    "Despertador", "Precio" => 79.78),
    array("Identificador" => 3, "Nombre" => "Reloj de
    pared", "Precio" => 625.5));
?>
```

También tendremos que cambiar la forma de acceder a estos detalles de cada uno de los productos.

Por ejemplo, la expresión **\$productos[0][0]** hace referencia al identificador del primer producto. Pues utilicemos la sintaxis más intuitiva: **\$productos[0]["Identificador"]...**

```
<table width="100%" border="1">
  <tr>
    <th>Identificador del producto</th>
    <th>Nombre</th>
    <th>Precio</th>
  </tr>
  <tr>
    <td><?php echo $productos[0]["Identificador"]; ?></td>
    <td><?php echo $productos[0]["Nombre"]; ?></td>
    <td><?php echo $productos[0]["Precio"]; ?></td>
  </tr>
  <tr>
    <td><?php echo $productos[1]["Identificador"]; ?></td>
    <td><?php echo $productos[1]["Nombre"]; ?></td>
    <td><?php echo $productos[1]["Precio"]; ?></td>
  </tr>
  <tr>
    <td><?php echo $productos[2]["Identificador"]; ?></td>
    <td><?php echo $productos[2]["Nombre"]; ?></td>
    <td><?php echo $productos[2]["Precio"]; ?></td>
  </tr>
</table>
```

Por lo tanto, ahora es mucho más fácil hacer referencia al identificador, nombre y precio de los distintos productos.

Los arrays son muy utilizados en las páginas php, por lo que el lenguaje proporciona mucha funcionalidad para trabajar con estas estructuras de datos.

Podremos ordenar, mezclar, modificar, eliminar, etc. los elementos de los arrays mediante funciones de lenguaje PHP.

Pero esto lo veremos a medida que lo necesitemos. Lo importante es que ahora ya sabes qué es un array y los conocimientos básicos para poder entender el código que los utiliza.