

5. Arrays

5.1 Asignación.

Los arrays en PHP no necesitan de declaración:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$alumnos[0] = 12;
$alumnos[1] = 15;
$alumnos[2] = 17;
$alumnos[3] = 15;
$alumnos[4] = 22;

echo "El número de alumnos que asistieron el miércoles fue de ",$alumnos[2];

?>
</body>
</html>
```

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

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$alumnos =[12, 15, 17, 15, 22];

echo "El número de alumnos que asistieron el miércoles fue de ",$alumnos[2];

?>
</body>
</html>
```

Se puede utilizar **for** o **foreach** para iterar los elementos:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 <?php
8 echo "<b>Números:</b><br>";
9 for ($i = 0; $i < 10; $i++) {
10     // números aleatorios entre 1 y 10
11     $numeros[$i] = rand(1, 10);
12 }
13 foreach ($numeros as $elemento) {
14     echo $elemento, "<br>";
15 }
16 ?>
17 </body>
18 </html>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$dia = [
    "Lunes",
    "Martes",
    "Miércoles",
    "Jueves",
    "Viernes",
    "Sábado",
    "Domingo"
];
echo "Comenzaré a estudiar el ", $dia[rand(0, 6)], ".";
?>
</body>
</html>
```

5.2 Arrays asociativos

En un array asociativo se pueden utilizar índices que no son numéricos, a modo de claves. Veamos un ejemplo de un array asociativo que almacena los dorsales de los jugadores de un equipo de fútbol.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$dorsal= ["Rubén"=> 13, "Néstor"=> 4, "Joseph"=> 15, "Renato"=> 14, "Iago"=> 10];

echo "El dorsal de Iago Aspas es el ", $dorsal['Iago'] ;
?>
</body>
</html>
```

5.3 Arrays bidimensionales

Se pueden crear arrays bidimensionales y cambiarle el tamaño dinámicamente.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$tablero = [

    [ 0 , 1 , 2],
    [ 3 , 4 , 5],
    [ 6 , 7 , 8]

];

echo $tablero[0][1];

$tablero[4][0] = 9;
$tablero[4][1] = 10;
$tablero[4][2] = 11;

echo $tablero[4][1];

?>
</body>
</html>
```

5.4 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.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php

    for ($i=0; $i<10; $i++){
        $aleatorio[$i] = rand(0,100);
    }

    foreach ($aleatorio as $elemento){

        echo $elemento."<br>";
    }

?>
</body>
</html>
```

5.5 Recoger datos a través de la web

Una forma sencilla de recoger datos para un array es ir añadiéndolos en una cadena de caracteres con un separador, por ejemplo, un espacio en blanco. Una vez terminemos haremos un explode, equivalente al split de java, para obtener el array. El siguiente ejemplo recoge 5 números y los muestra:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
$numero = $_GET['numero'];
$contadorNumeros = $_GET['contadorNumeros'];
$numeroTexto = $_GET['numeroTexto'];

if (! isset($numero)) {
    $contadorNumeros = 0;
    $numeroTexto = "";
}
if ($contadorNumeros == 5) {
    $numeroTexto = $numeroTexto . " " . $numero; // añade el último número leído
    $numero = explode(" ", $numeroTexto); //Equivale al split de Java
    echo "Los números introducidos son: ";
    foreach ($numero as $n) {
        echo $n, " ";
    }
}

// Pide número y añade el actual a la cadena
if (($contadorNumeros < 5) || (! isset($n))) {
    ?>
<form action="#" method="get">
    Introduzca un número: <input type="number" name="numero" autofocus>
    <input type="hidden" name="contadorNumeros" value="<?= $contadorNumeros+1 ?>">
    <input type="hidden" name="numeroTexto" value="<?= $numeroTexto . " " . $numero ?>">
    <input type="submit" value="Siguiente">
</form>
<?php
}
?>
</body>
</html>

```

Ejercicio 5.1

Escribe una web que genere 50 números aleatorios entre 1 y 100, los almacene en un array y que luego muestre los números junto con las palabras “máximo” y “mínimo” al lado del máximo y del mínimo respectivamente.

Ejercicio 5.2

Escribe una web que pida 10 números por teclado y que luego muestre los números introducidos junto con las palabras “máximo” y “mínimo” al lado del máximo y del mínimo respectivamente.

Ejercicio 5.3

Escribe un programa que genere 50 números aleatorios del 0 al 20 y que los muestre por pantalla separados por espacios. El programa pedirá entonces dos valores y a continuación cambiará todas las ocurrencias del primer valor por el segundo en la lista generada anteriormente.

Ejercicio 5.4

Escribe un programa que genere 20 números enteros aleatorios entre 0 y 100 y que los almacene en un array. El programa debe ser capaz de pasar todos los números pares a las primeras posiciones del array (del 0 en adelante) y todos los números impares a las celdas restantes. Utiliza arrays auxiliares si es necesario.

Ejercicio 5.5

Crea un mini-diccionario español-inglés que contenga, al menos, 10 palabras (con su traducción). Utiliza un array asociativo para almacenar las parejas de palabras. El programa pedirá una palabra en español y dará la correspondiente traducción en inglés.

Ejercicio 5.6

Realiza un programa que sea capaz de rotar todos los elementos de una matriz cuadrada una posición en el sentido de las agujas del reloj. La matriz debe tener 6 filas por 6 columnas y debe contener números generados al azar entre 1 y 99. Se debe mostrar tanto la matriz original como la matriz resultado, ambas con los números convenientemente alineados.

Ejercicio 5.7

Mejora la web anterior de forma que cada vez que se pulse un botón siguiente muestre la matriz rotada de nuevo.