



4.2. Estructuras de datos en PHP

1º DAW - Programación
David Soler Talens



- **Arrays indexados**

- Los arrays empiezan siempre por índice 0
- Se declaran

```
$array_name =[ value1, value 2, value3, ... ];
```

- **Arrays asociativos**

- Se declaran especificando cada clave

```
$array_name =[ 'clave1' => value1, 'clave2' => value 2, ... ]
```



Este símbolo es conocido
como «**fat arrow**»



Escribir/leer datos a un array (I)

- **Arrays indexados**

```
1  <?php
2  $arr = []; // crea un array sin elementos
3
4  $arr[0] = 15; // añade 15 al primer elemento
5  $arr[3] = 25; /* OJO! añade 25 a un elemento nuevo con índice 3, el índice
6  |         |         |         3 ya no representa el 4º elemento!!! */
7
8  $arr[3] = 30; /* modifica el elemento con índice 3 */
9  $arr2 = [10, 20, 30, 40]; /* crea un array y lo puebla con valores */
10
11 var_dump($arr); /* muestra el array con su estructura */
12 echo $arr2[3]; /* muestra el elemento con índice 3 */
```

Escribir/leer datos a un array (II)



• Arrays Asociativos

```
1  <?php
2  $arr = [
3      'nombre' => "David",
4      'DNI' => "123456789Z"
5  ];
6
7  $arr['Apellido'] = 'Soler'; // añade el elemento 'apellido'
8  $arr['Apellido'] = 'Pérez'; /* modifica el elemento con
9      |         |         |         |         |         |
10     clave 'Apellido' */
11
12 var_dump($arr); /* muestra el array con su estructura */
13 echo $arr['nombre']; /* muestra el elemento con
14     |         |         |         |         |
15     clave 'nombre' */
```



Arrays multi-dimensionales

```
1  <?php
2  $arr = [
3      'temperatura' => [10, 12, 14, 13],
4      'Humedad' => [90, 87, 75, 60]
5  ];
6
7  $arr['presion'] = [714, 825, 904, 1024]; // añade el elemento 'presion'
8  $arr['presion'][1] = 920; /* modifica el elemento con
9  clave 'presion', y dentro de este elemento, el dato 825, es decir, la
10 celda 1, con el nuevo valor 920 */
11
12 var_dump($arr); /* muestra el array con su estructura */
13 echo $arr['Humedad'][3]; /* muestra el elemento con clave 'Humedad', y
14 dentro de éste, el elemento con índice 3, es decir, el dato 60*/
15
```



Funciones de array (I)

```
1  <?php
2  $fruits = ["Bannana", "Orange", "Apple", "Mango"];
3  echo count($fruits);
4  //4
5  $var = implode("-", $fruits); echo $var;
6  //"Bannana-Orange-Apple-Mango"
7  var_dump(explode("-", $var));
8  //["Bannana", "Orange", "Apple", "Mango"]
9  array_pop($fruits); var_dump($fruits);
10 //["Bannana", "Orange", "Apple"]
11 array_push($fruits, "kiwi"); var_dump($fruits);
12 //["Bannana", "Orange", "Apple", "kiwi"]
13 array_shift($fruits); var_dump($fruits);
14 //["Orange", "Apple", "kiwi"]
15 array_unshift($fruits, "Lemon"); var_dump($fruits);
16 //["Lemon", "Orange", "Apple", "kiwi"]
```



Funciones de array (II)

```
1  <?php
2  $fruits = ["Bannana", "Orange", "Apple", "Mango"];
3  array_splice($fruits,2,0,["Lemon","Kiwi"]); var_dump($fruits);
4  /*
5   en la posición 2, es decir 'Apple' elimina 0 elementos y añade
6   "Lemon" y "Kiwi", resultando en:
7   ["Bannana", "Orange", "Lemon", "Kiwi", "Apple", "Mango"]
8   */
9  array_splice($fruits,1,2); var_dump($fruits);
10 /*
11 desde la posición 1, es decir, 'Orange', elimina 2 elementos y
12 no añade nada mas, resultando en:
13 ["Bannana", "Kiwi", "Apple", "Mango"]
14 */
```



Funciones de array (III)

```
1  <?php
2  $myGirls = ["Cecile", "Lone"];
3  $myBoys = ["Emil", "Tobias", "Linus"];
4  $myChildren = array_merge($myGirls, $myBoys);
5  var_dump($myChildren);
6  /* Nuevo array $myChildren cuyo contenido es
7  ["Cecile", "Lone", "Emil", "Tobias", "Linus"]
8  */
9  $fruits = ["Bannana", "Orange", "Apple", "Mango"];
10 $var = array_slice($fruits, 1);
11 var_dump($var);
12 /*
13 Corta desde la posición 1 incluido hasta el final del
14 array, y genera un nuevo array, resultando en:
15 ["Orange", "Apple", "Mango"]
16 */
17 $var = array_slice($fruits, 1, 2);
18 var_dump($var);
19 /*
20 corta desde la posición 1 incluida, 2 elementos
21 y genera un nuevo array, resultando en:
22 ["Orange", "Apple"]
23 */
```




Recorrer con FOREACH (I)

```
1  <?php
2  $colors = ["red", "green", "blue", "yellow"];
3
4  foreach ($colors as $value) {
5      echo "$value <br>";
6  }
7
8  $age = ["Peter"=>"35", "Ben"=>"37", "Joe"=>"43"];
9
10 foreach($age as $key => $value) {
11     echo "$key = $value<br>";
12 }
```

\$value contiene el valor
de la celda en cada iteración

Contiene los valores
de cada elemento

Contiene la clave de cada elemento

Recorrer con FOREACH (II)



```
1  <?php
2  $people = [
3      "Peter" => [
4          'age' => "35",
5          'gender' => "Male",
6      ],
7      "Ben" => [
8          'age' => "37",
9          'gender' => "Male",
10     ],
11     "Mary" => [
12         'age' => "43",
13         'gender' => "Female",
14     ],
15 ];
16
17 foreach ($people as $person => $data) {
18     echo "$person: <br>":
19     foreach ($data as $key => $value) {
20         echo "$key = $value<br>";
21     }
22 }
```

SALIDA

Peter:
age = 35
gender = Male
Ben:
age = 37
gender = Male
Mary:
age = 43
gender = Female

Claves del array \$people

\$data contiene el array que
representa la 2ª dimensión
de \$people.

Clave y valor del sub-array