

Introducción a la programación PHP

2.1.2 - Estructuras de datos - “Arreglos/Arrays”
Prof. Luis Eduardo Fagúndez

Estructuras de datos

- Son formas de organizar y almacenar datos en memoria. Algunos ejemplos son arrays, listas enlazadas, pilas, colas, árboles, grafos, entre otros.

Arrays

- **Un array en PHP es un mapa ordenado.**
- **Un mapa es un tipo de datos que asocia valores con claves.**
- **Este tipo permite implementar las siguientes estructuras:**
 - Array simples y multidimensionales
 - Lista
 - Pila
 - Cola
 - Árboles

Clase Array

- Se puede crear un arreglo usando el **constructor de la clase Array**, este constructor es **Array()** o usando el formato simplificado.
- Este toma cualquier número de pareja de datos
 - Clave => Valor como argumentos
- **Array(**
 - Clave1 => valor1,
 - Clave2 => valor2,
 -
- **)**

Arreglos indexados

- **Estos son arreglos donde cada elemento está asociado a un índice numérico único.**
- **El primer elemento tiene el índice 0, el segundo el índice 1, y así sucesivamente. Se pueden acceder a los elementos utilizando su índice.**

Formas de declarar un arreglo



```
<?php
//Forma 1: Utilizando la función array() y una lista de valores:
$arreglo = array(10, 20, 30, 40, 50);
//Forma 2: Utilizando corchetes vacíos [] (es la que usaremos normalmente):
$arreglo = [10, 20, 30, 40, 50];
//Forma 3: Declarando el arreglo vacío y luego asignando valores:
$arreglo = [];
$arreglo[] = 10;
$arreglo[] = 20;
$arreglo[] = 30;
$arreglo[] = 40;
$arreglo[] = 50;
//Forma 4: Especificando los índices explícitamente:
$arreglo[0] = 10;
$arreglo[1] = 20;
$arreglo[2] = 30;
$arreglo[3] = 40;
$arreglo[4] = 50;
?>
```

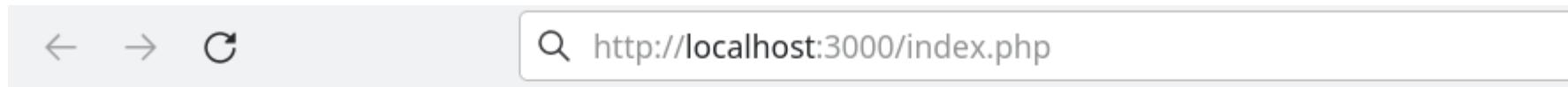
Arreglo indexado

- Veamos un ejemplo con la sintaxis simplificada:

```
<?php  
$edades = [18, 25, 11, 33, 42, 15];  
var_dump($edades)  
?>
```

Arrays

- **Resultado:**



- **Array(6) → El array tiene 6 elementos.**
- **{ ... } → Los elementos del array.**

Arrays

- **Resultado:**



- **[0]=> int(18) [1] → el 0 indica la clave y el [1] el valor**
- **Al no indicar el valor de la clave por defecto empiezan con el 0 y van por orden secuencial [0], [1],[2].**

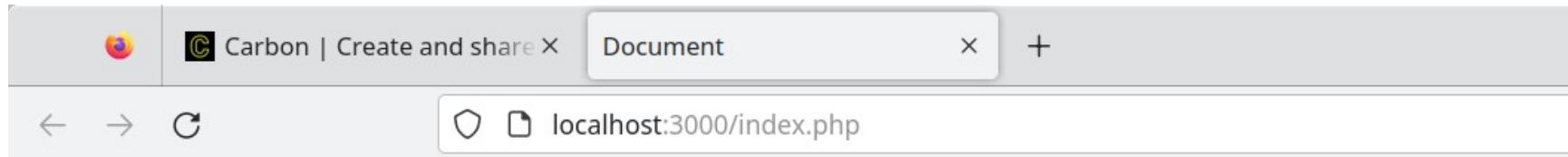
Arrays

- Se pueden almacenar diferentes tipos de datos en un mismo arreglo.

```
<?php
    $arreglito = [30, "Luis", 'Eduardo', 31, true, 3.1415];
    var_dump($arreglito);
?>
```

Arrays

- **Resultado:**



A screenshot of a web browser window. The title bar says "Carbon | Create and share > Document". The address bar shows "localhost:3000/index.php". The main content area displays the following PHP array output:

```
array(5) { [0]=> int(30) [1]=> string(4) "Luis" [2]=> string(7) "Eduardo" [3]=> bool(true) [4]=> float(3.1415) }
```

Arreglos asociativos

- En estos arreglos, cada elemento está asociado a una clave en lugar de un índice numérico.
- Las claves pueden ser cadenas de caracteres o números.
- Se utilizan principalmente para asociar valores con nombres descriptivos.

Arreglos asociativos

Los elementos se almacenan como pares clave-valor.

- **Cada elemento tiene una clave única que se utiliza para acceder a su valor asociado.**
- **Se utilizan principalmente para almacenar datos que se relacionan con un nombre o una etiqueta específica.**

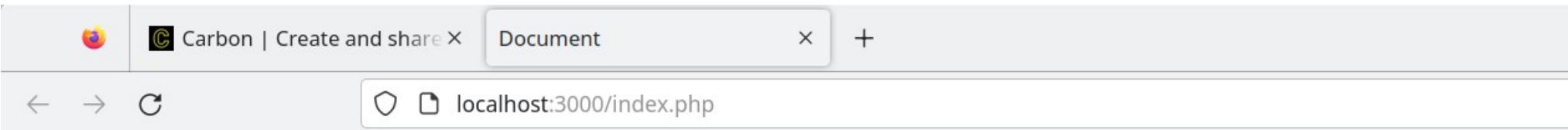
Arreglos asociativos

- Se pueden definir las claves del arreglo sin necesidad de que sean consecutivas:

```
<?php
    $arreglito = array(
        1 => 30,
        2 => "Luis",
        4 => 'Eduardo',
        8 => true,
        9 => 3.1415,
        "José"
    );
    var_dump($arreglito);
?>
```

Arreglos asociativos

- **Resultado:**



A screenshot of a web browser window. The title bar shows the Firefox logo, the Carbon logo, and the word "Document". Below the title bar is a toolbar with back, forward, and refresh buttons. The address bar shows "localhost:3000/index.php". The main content area displays the following PHP array output:

```
array(6) { [1]=> int(30) [2]=> string(4) "Luis" [4]=> string(7) "Eduardo" [8]=> bool(true) [9]=> float(3.1415) [10]=> string(5) "José" }
```

- **Si observamos, “José” no tenía clave definida, continua con el siguiente valor para la clave.**

Otro ejemplo:

```
<?php
$persona = [
    "nombre" => "Luis",
    "edad" => 31,
    "ciudad" => "Tacuarembó"
];
echo $persona["nombre"]; // Imprime el nombre Luis
?>
```

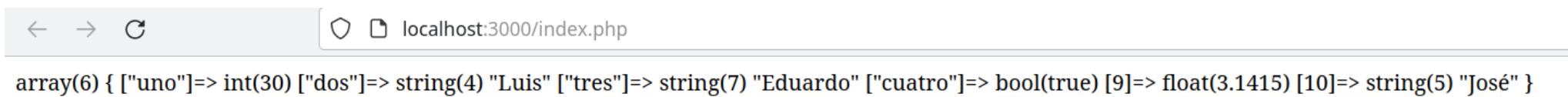
Otro ejemplo más..

- Se pueden utilizar claves de tipo String:

```
<?php
    $arreglito = array(
        "uno" => 30,
        "dos" => "Luis",
        "tres" => 'Eduardo',
        "cuatro" => true,
        9 => 3.1415,
        "José"
    );
    var_dump($arreglito);
?>
```

Arreglos asociativos

- En el siguiente ejemplo podemos observar que:



A screenshot of a web browser window. The address bar shows 'localhost:3000/index.php'. The main content area displays the following PHP array output:

```
array(6) { ["uno"]=> int(30) ["dos"]=> string(4) "Luis" ["tres"]=> string(7) "Eduardo" ["cuatro"]=> bool(true) [9]=> float(3.1415) [10]=> string(5) "José" }
```

- **["uno"]=> int(30)** → “uno” es la clave y 30 es el valor
- **["dos"]=> string(4) "Luis"** → “dos” es la clave y “luis” el valor

Arrays

- **Se pueden construir arrays utilizando el constructor del lenguaje [].**
- **Todos los elementos anteriores se pueden utilizar con el constructor**
- **Este toma cualquier número de parejas**
 - Clave1 → valor con argumentos

Arrays

- **Ejemplo:**
- **Array [**
 - Clave1 => valor1,
 - Clave2 => valor2,
 -
- **]**

Ejemplo



```
<?php
$indexado = [10, 20, 30, 40, 50];
$asociativo = ["nombre" => "Luis", "edad" => 30, "ciudad" => "Salto"];
$combinado = ["nombre" => "Eduardo", "edad" => 30, 100, "ciudad" => "Rivera"];
?>
```

Arrays

- **Para acceder al elemento de un array debemos hacerlo mediante su clave:**
 - Sintaxis general:
 - \$nombreDelArray[clave];
 - \$nombreDelArray{clave}; //Solo disponible de php 7.4 y anteriores

Arrays

- Ejemplo de código:

```
<?php
$arreglito = array(
    "uno" => 30,
    "dos" => "Luis",
    "tres" => 'Eduardo',
    "cuatro" => true,
    9 => 3.1415,
    "José"
);
echo $arreglito['uno'];
echo "El valor de la clave dos en el array es: ". $arreglito['dos'];
?>
```

Resultado

- **Realizar el anterior código y ver el resultado ustedes mismos.**

Arrays

- **Para modificar el valor de un elemento se utiliza la siguiente forma:**
 - \$arreglo[clave] = contenido;
- **Si \$arreglo no existe se creará con ese contenido, siendo esta una forma de crear un arreglo.**

Cómo recorrer un arreglo?

- **Bucle foreach:**

- Es la forma más común de recorrer arreglos en PHP. Se utiliza principalmente para recorrer arreglos asociativos, pero también se puede utilizar con arreglos indexados.

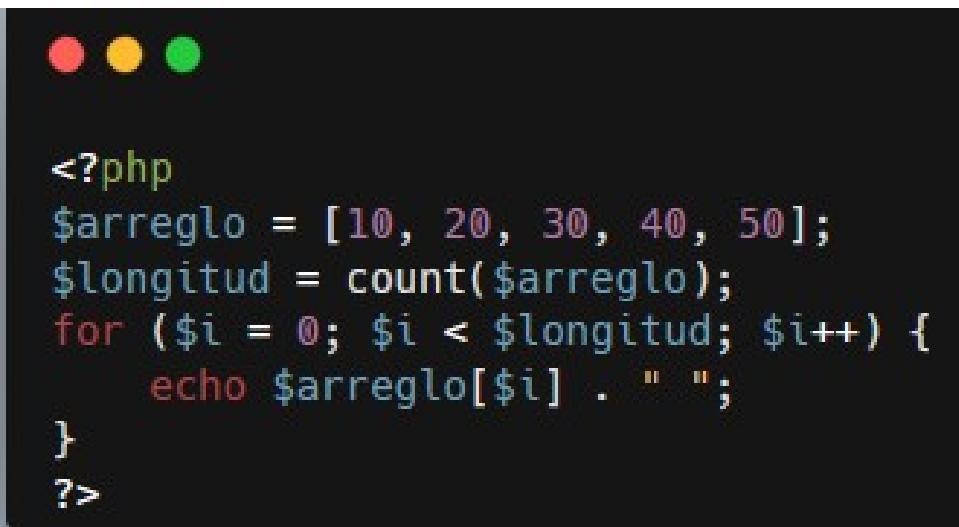


```
<?php
$arreglo = ["a" => 1, "b" => 2, "c" => 3];
foreach ($arreglo as $clave => $valor) {
    echo "Clave: $clave, Valor: $valor <br>";
}
?>
```

Cómo recorrer un arreglo?

- **Bucle for:**

- Es útil para recorrer arreglos indexados, ya que permite acceder a los elementos utilizando sus índices.



```
<?php
$arreglo = [10, 20, 30, 40, 50];
$longitud = count($arreglo);
for ($i = 0; $i < $longitud; $i++) {
    echo $arreglo[$i] . " ";
}
?>
```

Otro ejemplo

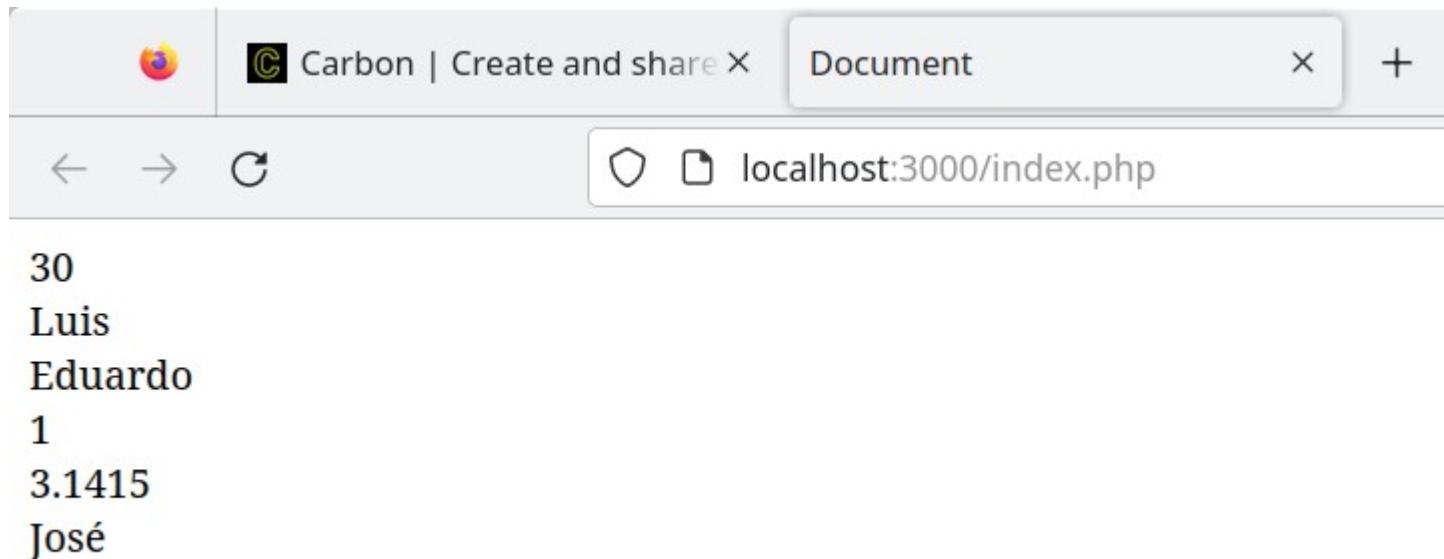
- **Veamos el siguiente ejemplo**
- **Dado un arreglo asociativo**
- **Lo recorremos usando**
- **La estructura foreach:**



```
<?php
    $arreglito = array(
        "uno" => 30,
        "dos" => "Luis",
        "tres" => 'Eduardo',
        "cuatro" => true,
        9 => 3.1415,
        "José"
    );
    foreach($arreglito as $arre){
        echo $arre . "<br >";
    }
?>
```

Arreglos

- **Resultado:**



Arreglos

- **Para eliminar un elemento de un arreglo se utiliza la siguiente función:**
 - `unset($arreglo['clave']);`
- **Problema: no se reasignan las claves.**
- **Para eliminar el array completo**
 - `unset($arreglo);`