

# Arrays

- Los arrays o matrices son estructuras que permiten el almacenamiento de un conjunto de datos.
- Definición: un array o matriz es un conjunto ordenado de elementos identificados por un índice (la posición del elemento dentro de esta colección ordenada), de modo que en cada posición marcada por un índice del array, este contiene un valor.
- La longitud del array se modifica de forma dinámica siempre que añadimos un nuevo elemento.

# Arrays: Tipos

Existen 3 tipos de arrays:

- Indexados.- Arrays con un índice numérico.
- Asociativos.- Arrays con claves nominales.
- Multidimensionales.- Arrays que contienen uno o más arrays.

# Arrays: inicialización

```
// Arrays indexados
```

```
$numerico= array(3, 8, 1223, rojo);
```

```
// Arrays asociativos
```

```
$mimatriz= array(clave => array(),...);
```

```
$miarrray=array( 'altura'=>170, "peso"=>70);
```

```
//Arrays múltiples
```

```
$mimatriz(0 => array(0=>2,1=>4), 1  
=>array(0=>1,1=>3));
```

# Arrays: inicialización

```
// Los dos arrays son indexados  
// y se consideran iguales en PHP
```

```
$datos = array(  
    "Gato",  
    "Perro",  
    "Jirafa"  
);  
$datos = array(  
    0 => "Gato",  
    1 => "Perro",  
    2 => "Jirafa"  
);
```

# Arrays: acceso

```
<?php  
$miarray = array();  
$miarray[0] = 3;  
echo "El valor es de" . $miarray[0] . " puntos";  
var_dump($miarray);  
?>
```

# Arrays: recorrer

```
foreach($miarray as $valor){  
    echo $valor;  
}
```

```
foreach ($miarray as $clave =>  
    $valor){  
  
}
```

# PHP - Arrays

Un array en PHP es, actualmente, un mapa ordenado. Un mapa es un tipo de dato que asocia valores a claves.

- Esta estructura puede ser tratada de forma muy flexible pues permite implementar, además de arrays, listas (vector), tablas de hash (una implementación de un mapa), diccionarios, colecciones, pilas, colas and probablemente más.
- La clave puede ser un ***integer*** o un ***string***. El valor puede ser de cualquier tipo (int, float, bool, string), incluso otros arrays, arrays multidimensionales, objetos, etc.

# PHP - Creación de arrays

Puede ser creado con el constructor del lenguaje **array()** y puede contener cualquier número de pares clave=>valor como argumentos.

```
array(  
    key1 => value,  
    key2 => value2,  
    key3 => value3,  
    ...  
)
```



# PHP - Creación de arrays

A partir de PHP 5.4 se permite la sintaxis abreviada cambiando `array()` por `[ ]`:

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
);
```

```
// a partir de PHP 5.4
$array = [
    "foo" => "bar",
    "bar" => "foo",
];
?>
```

Nota: La coma después del último elemento del array es opcional.

# PHP - conversiones de claves

Pueden darse automáticamente las siguientes conversiones de clave:

- Un string que contenga un integer válido será convertido al tipo integer. P.ej. la clave "8" en realidad será almacenada como 8. Por otro lado "08" no será convertido, ya que no es un número integer válido.
- Un float también será convertido a integer, lo que significa que la parte fraccionaria se elimina. P.ej., la clave 8.7 en realidad será almacenada como 8.
- Un booleano será convertido a *integer* también, es decir, la clave *true* en realidad será almacenada como 1 y la clave *false* como 0.
- Un null será amoldado a un string vacío, es decir, la clave null en realidad será almacenada como "".

Los arrays y los objects no pueden utilizarse como claves. Si se hace, dará lugar a una advertencia: ***Illegal offset type***.

# PHP - conversiones de claves

Si varios elementos en la declaración del array usan la misma clave, sólo se utilizará la última, los demás serán sobrescritos.

```
<?php
$array = array(
    1    => "a",
    "1"  => "b",
    1.5  => "c",
    true => "d",
);
var_dump($array);
?>
```

Como todas las claves en el ejemplo anterior se convierten en 1, los valores serán sobrescritos en cada nuevo elemento, por lo que el último valor asignado "d" es el único que queda.

# PHP - claves indexadas o asociativas

Los arrays de PHP pueden contener claves integer y string al mismo tiempo ya que PHP no distingue entre arrays indexados (la clave es un integer) y asociativos (la clave es un string)

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
    100  => -100,
    -100 => 100,
);
var_dump($array);
?>
```

# PHP - clave opcional

La clave es opcional. Si no se especifica, PHP usará la clave de tipo integer mayor utilizada anteriormente, incrementada en 1.

```
<?php  
$array = array("foo", "bar", "hello", "world");  
var_dump($array);  
?>
```

Tomarán los índices 0, 1, 2 y 3 respectivamente.

# PHP - clave sin especificar

Es posible especificar la clave sólo en algunos elementos y excluirla en los demás:

```
<?php
$array = array(
    "a",
    "b",
    6 => "c",
    "d",
);
var_dump($array);
?>
```

El índice del valor “d” es 7. Esto es debido a que la mayor clave integer anterior era 6.

# PHP - accediendo a elementos

Los elementos de array se pueden acceder utilizando la sintaxis `array[key]`.