



# PHP | Arrays

## Después de esta lección sabrás:

1. Las diferentes cadenas de texto que hay en PHP.
2. Realizar diferentes acciones con cadenas de texto.
3. Manejarte con la documentación oficial de PHP.

**IMPORTANTE:** La posición de los arrays empieza en cero, ¿qué quiere decir esto? pues que el primer elemento de un array está en la posición 0, el segundo en la 1, el tercero en la 2 y así sucesivamente, ¡recordadlo!

## Añadir elementos a un array

Dada un array, siempre puedes añadir elementos a dicho array. Ejemplo

```
$persona = ['nombre' => 'moises', 'edad' => 36];  
$persona['telefono'] = 666787677;  
print_r($persona);
```

Resultado:

```
Array(  
    [nombre] => moises  
    [edad] => 36  
    [telefono] => 666787677  
)
```

Si se quiere añadir un valor sin ninguna clave, simplemente se añade con los corchetes vacíos. Ejemplo:

```
$profesores = ['kiko', 'jorge', 'moises'];  
$profesores[] = 'roberto';  
print_r($profesores);
```

Dando como resultado:

```
Array(  
    [0] => kiko  
    [1] => jorge  
    [2] => moises  
    [3] => roberto  
)
```

## Actualizar un elemento del array

Para actualizar un elemento de un array hay que utilizar la misma clave y asignarle otro valor.

En caso de los arrays asociativos con claves de tipo string:

```
$persona = ['nombre' => 'moises', 'edad' => 36];  
$persona['edad'] = 40;  
print_r($persona);
```

Resultado:

```
Array(  
    [nombre] => moises  
    [edad] => 40  
)
```

En caso de ser valores numéricos (recordad que arrays simples tienen claves numéricas implícitas), se hace de la misma manera:

```
$profesores = ['kiko', 'jorge', 'moises'];  
$profesores[1] = 'ricky';  
print_r($profesores);
```

Dando como resultado:

```
Array(  
    [0] => kiko  
    [1] => ricky  
    [2] => moises  
)
```

## Iterar por los elementos de un array

La iteración consiste en recorrer los elementos de un array y hacer algo con cada uno de los elementos de dicho array. Ese 'algo' puede ser cualquier cosa, desde procesar el valor aplicándole funciones de transformación, realizar llamadas http con las urls de dicho array, o simplemente mostrarlo por pantalla.

Los arrays que solamente tienen valores, internamente tienen índices numéricos por los que se pueden iterar, veamos un ejemplo de como recorrer un array que solo tiene valores, y cómo internamente las claves son números.

El ejemplo es muy sencillo e imprimirá el nombre en mayúsculas junto con el índice (recordad que el índice puede ser también una cadena).

```
$nombres = ['kiko', 'jorge', 'moises'];  
foreach ($nombres as $clave => $valor) {  
    echo "$clave -> " . $valor . "\n";  
}
```

El resultado será:

```
0 → kiko  
1 → jorge  
2 → moises
```

Si solamente queremos imprimir los valores cambia el bucle foreach

```
$nombres = ['kiko', 'jorge', 'moises'];
foreach ($nombres as $profesor) {
    echo $profesor . "\n";
}
```

Y el resultado será:

```
kiko
jorge
moises
```

Si el array tuviese claves asociativas sería parecido al primer ejemplo.

```
$persona = ['nombre' => 'moises', 'edad' => 36];
foreach ($persona as $clave => $valor) {
    echo "$clave -> " . $valor . "\n";
}
```

Dando como resultado

```
nombre → moises
edad → 36
```

## Funciones con matrices

Al igual que con las cadenas, vamos a ver algunas funciones útiles para tratar con arrays.

Es importante tener en cuenta que en este lenguaje los arrays se usan constantemente y tendrás que tener mucha soltura con ellos.

Como siempre, y es algo que recalcaremos a lo largo del curso, es fundamental leer la documentación oficial del lenguaje, donde podrás ver la sintaxis del lenguaje y de las funciones, en este caso aquí está el enlace al listado de funciones relacionadas con matrices de PHP. [DOCUMENTACIÓN](#)

# in\_array

Comprueba si un valor existe en un array, devuelve el valor booleano TRUE si existe, y FALSE si no existe

Ejemplo:

```
$nombres = ['kiko', 'jorge', 'moises'];  
if (in_array('moises', $nombres)) {  
    echo "Moises está en el array";  
} else {  
    echo "Moises NO se encuentra en el array";  
}
```

Aquí comprobamos si el string 'moises' se encuentra en el array, y como sí que está el resultado será:

| Moises está en el array

# count

Cuenta todos los elementos de un array

Ejemplo

```
$nombres = ['kiko', 'jorge', 'moises'];  
echo "Total de nombres: " . count($nombres);
```

Siendo el resultado:

| Total de nombres: 3