



# PHP | Cadenas de texto

## 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.

Un string o cadena es un tipo de dato que contiene una serie de caracteres alfanuméricos.

**NOTA:** ¡Un string puede almacenar información de hasta 2GB de tamaño!

Se puede definir una cadena en PHP de las siguientes maneras:

- Entrecorillado simple
- Entrecorillado doble
- Sintaxis heredoc y nowdoc (no lo veremos en el curso de nivelación)

## Entrecorillado simple

La manera más sencilla de especificar un string es delimitarlo con corillas simples (el carácter `'`).

```
$texto = 'hola soy una cadena';  
$fraseIngles = 'This is the Luis\'s car';  
echo $texto;  
echo $fraseIngles;
```

El resultado será:

hola soy una cadena  
This is the Luis's car

En este entrecomillado NO se evaluarán/expandirán las variables, ej:

```
$nombre = 'Luis';  
echo 'se llama $nombre'
```



El resultado será:

se llama \$nombre

## Entrecomillado doble

Si un string está delimitado con comillas dobles («), PHP interpretará las siguientes secuencias de escape como caracteres especiales.

Algunos ejemplos de caracteres especiales:

 Secuencia	 Significado
<code>\n</code>	Salto de linea
<code>\t</code>	Tabulador

La característica más importante del entrecomillado doble de un string es el hecho de que se expanden los nombres de las variables. Consulte el análisis de string para más detalles.

```
$texto = "hola soy una cadena";  
$fraseIngles = "This is the Luis's car";  
echo $texto;  
echo $fraseIngles;
```

El resultado será:

```
| hola soy una cadena  
| This is the Luis's car
```

En este entrecomillado se evaluarán/expandirán las variables, ej:

```
$nombre = 'Luis';  
echo "se llama $nombre";
```

El resultado será:

```
| se llama Luis
```

Ejemplo con salto de línea:

```
echo "Salto\nde\nlínea";
```

El resultado será:

```
| Salto  
| de  
| línea
```

## Concatenación (unión ) de cadenas

Se pueden juntar cadenas utilizando el operador '.' (punto)

Se pueden concatenar strings tanto con comillas simples como dobles, en las dobles se evaluarán las variables, y en las simples no.

Ejemplos:

```
$nombre = 'Juan';  
$apellidos = 'Fernandez';  
echo "Se llama " . $nombre . " " . $apellidos;
```

El resultado será:

| Se llama Juan Fernandez

## Funciones con cadenas

Aunque todavía no has visto qué son las funciones, puedes pensar en ellas como una caja negra a la que le puedes pasar valores y te puede devolver otros valores.

En el caso de las cadenas, PHP pone a disposición de los desarrolladores un buen número de funciones para el tratamiento de las mismas.

Es fundamental tener a mano la documentación para saber cual es la manera de utilizar esas funciones. [DOCUMENTACIÓN](#)

Veremos ahora un ejemplo de unas cuantas funciones útiles de tratamiento de cadenas y cómo usarlas:

## strtoupper

Convierte todos los caracteres de una cadena a mayúsculas. Ejemplo:

```
$nombre = 'Santiago';  
echo strtoupper($nombre);
```

El resultado será:

| SANTIAGO

## explode

Esta función divide un string en varios string, se le tiene que indicar un delimitador por el que hacer la división. La función devuelve un array.

La mejor manera de entenderlo es con un ejemplo:

```
$colores = 'rojo azul verde';  
print_r(explode(' ', $colores));
```

La función **print\_r** lo único que hace es imprimir el valor de un array

Lo que estamos haciendo es dividir la cadena colores utilizando como separador un espacio en blanco, lo que genera es un array de 3 elementos con los 3 colores, es decir, el resultado será:

```
Array(
    [0] => "rojo"
    [1] => "azul"
    [2] => "verde"
)
```

## implode

Podemos decir que es la inversa de explode, lo que hace es junta elementos de un array por un 'pegamento', y genera una cadena. El pegamento es otra cadena. Lo mejor es con otro ejemplo:

```
$colores = ["rojo", "azul", "verde"];
echo implode('-', $colores);
```

El resultado será una cadena con el valor:

```
| rojo-azul-verde
```

## strlen

Obtiene la longitud de una cadena. Ejemplo:

```
$nombre = 'Santiago';
echo "La longitud de $nombre es " . strlen($nombre);
```

Aquí vemos un ejemplo con dobles comillas y concatenación de cadenas, el resultado es:

La longitud de Santiago es 8

## substr

Devuelve una parte de una cadena, se le tiene que indicar el comienzo y la longitud de la cadena. Ejemplo:

```
$nombre = 'Don Arturo';  
echo substr($nombre, 4, 6);
```

Lo que está haciendo el ejemplo es obtener la parte de la cadena que empieza en la posición 4 (se salta los 4 primeros caracteres) con una longitud de 6 caracteres.

El resultado será:

Arturo