

FUNCIONES DE CADENA:

Mayúsculas y Minúsculas:

- [strtoupper \(\\$cadena\)](#): Convierte una cadena a mayúsculas.
- [strtolower \(\\$cadena\)](#): Convierte una cadena a minúsculas.
- [ucfirst \(\\$cadena\)](#): Pasa a mayúsculas el primer caracter.
- [ucwords \(\\$cadena\)](#): Pasa a mayúsculas el primer caracter de cada palabra.

Ejemplos:

```
<?php
$cadena="texTo De pRuEbA";
$texto= strtoupper($cadena); // pasa el texto a mayúsculas
echo "$texto<br />";
$texto= strtolower($cadena); // pasa el texto a minúsculas
echo "$texto<br />";
$texto= ucfirst($cadena); // a mayúsculas el 1º caracter
echo "$texto<br />";
$texto= ucwords($cadena); // a mayúsculas el 1º caracter de c/palabra
echo "$texto<br />";
?>
```

Triming:

- [chop\(\\$cadena\)](#): Elimina blancos y saltos de línea al final de una cadena.
- [ltrim\(\\$cadena\)](#): Elimina blancos y saltos de línea al inicio de una cadena.
- [trim\(\\$cadena\)](#): Elimina blancos y saltos de línea a ambos lados de una cadena.

Ejemplos:

```
<?php
$resultado=chop(" Texto de Prueba "); // quita espacios de la derecha
echo "---$resultado---<br />";
$resultado=ltrim(" Texto de Prueba "); // quita espacios de la izquierda
echo "---$resultado---<br />";
$resultado=trim(" Texto de Prueba "); // quita espacios de ambos lados
echo "---$resultado---<br />";
?>
```

Nota que el resultado de cada función se imprime con 3 guiones a ambos lados, esto lo hice con intención de mostrar que realmente se quitan los espacios.

Comparaciones y subcadenas:

- [strpos\(\\$cadena1,\\$cadena2\)](#): Devuelve la posición de la primera ocurrencia de cadena2 dentro de cadena1, **tener en cuenta que la cadenas se numeran desde cero**. esta función devuelve falso si no se encuentra la cadena.
- [strlen\(\\$cadena1,\\$cadena2\)](#): Devuelve la long. de cadena1 desde el principio hasta hallar un caracter que pertenece a cadena2.
- [substr\(\\$cadena1,\\$cadena2\)](#): Devuelve los caracteres de cadena1 desde la primera ocurrencia de cadena2 hasta el final.

Ejemplos:

```
<?php
$pajar="ablqsdcbiej";
$aguja="dc";

$resultado=strpos($pajar,$aguja); // busca $aguja en $pajar
echo "$resultado<br />"; // devuelve 5 (posic. en que comienza "dc")
```

```

$resultado=strcspn($pajar,$aguja); // cuenta caracteres en $pajar antes que
// aparece $aguja
echo "$resultado<br />"; // devuelve 5 (cantidad de caracteres antes de "dc")

$resultado= strstr($pajar,$aguja); // toma caracteres de $pajar desde $aguja
// hasta el final
echo "$resultado<br />"; // devuelve 4 (cantidad de caracteres antes de "o")
?>

```

- [substr\(\\$cadena,offset,longitud\)](#): Devuelve la subcadena correspondiente a \$cadena desde la posición indicada por offset y de la longitud indicada como tercer parámetro, si no se pasa el tercer parámetro, toma todos hasta el final.
- [substr_replace\(\\$cadena,\\$cadena_reemplazo,offset,longitud\)](#): igual al anterior, pero la subcadena es reemplazada por \$cadena_reemplazo, si esta última es "" entonces sirve para eliminar una porción de \$cadena.
- [str_replace\(\\$cadena1,\\$cadena2,\\$cadena3\)](#): Busca \$cadena1 en \$cadena3 y la reemplaza por \$cadena2.

Ejemplos:

```

<?php
$cadena1="Texto de Prueba 1";
$cadena2="reemplazado ";
$cadena3="Texto de Prueba 1 y mucho más";
$offset=6;
$long=3;
$resultado = substr($cadena1,$offset,$long); // Toma $long caracteres de
// $cadena1 desde $offset
// devuelve "de "
echo "$resultado<br />";
$resultado = substr_replace($cadena1,$cadena2,$offset,$long); // Toma $long
// caracteres de
// $cadena1 desde $offset y lo
// reemplaza por $cadena2
echo "$resultado<br />";
$resultado = str_replace($cadena1,$cadena2,$cadena3); // busca cadena1
// en cadena3 y lo
// reemplaza por cadena2
echo "$resultado<br />";
?>

```

Parsing:

- [explode\(separador,\\$cadena\)](#): Devuelve un vector donde cada elemento del vector es una subcadena de la \$cadena particionado por el caracter separador del primer parámetro.
- [implode\(separador,array\)](#): Genera una cadena concatenando todos los elementos del array intercalando el caracter separador que se pasa como parámetro.
- [parse_str\(\\$cadena\)](#): Dada una cadena de la forma "nombre=valor&nombre2=valor2&nombre3=valor3", separa las variables correspondientes con los valores indicados.
- [chr\(\\$entero\)](#): Devuelve un caracter dado su número ASCII.
- [ord\(\\$caracter\)](#): Devuelve su código ASCII, dado un caracter.
- [base64_decode\(\\$cadena\)](#): Decodifica una cadena encodeada en base64.
- [base64_encode\(\\$cadena\)](#): Codifica una cadena a base64.
- [urlencode\(\\$cadena\)](#): Codifica una cadena reemplazando los caracteres que no sean alfanuméricos por un "%" seguido de una representación hexadecimal de dicho caracter (formato seguro para pasarlo por URL).
- [urldecode\(\\$cadena\)](#): Decodifica una cadena encodeado con urlencode.

Hay muchas más, que aquí no se citaron, sólo es cuestión de ponerlas a prueba y ver si aún están disponibles en las nuevas versiones.