

CADENAS

Aquí tenemos algunos ejemplos de las funciones de cadenas:

- strstr

```
<?php
$email = 'name@example.com';
$domain = strstr($email, '@');
echo $domain; // mostrará @example.com

$user = strstr($email, '@', true); // Desde PHP 5.3.0
echo $user; // mostrará name
?>
```

- strrchr

```
<?php
// obtiene el último directorio en $PATH
$dir = substr(strrchr($PATH, ":"), 1);

// obtiene todo lo que está después de la última nueva línea
$text = "Line 1\nLine 2\nLine 3";
$last = substr(strrchr($text, "\n"), 1);
?>
```

- stristr

```
<?php
$email = 'USER@EXAMPLE.com';
echo stristr($email, 'e'); // salida ER@EXAMPLE.com
echo stristr($email, 'e', true); // A partir de PHP 5.3.0, salida US
?>
```

```
<?php
$string = 'Hello World!';
if(stristr($string, 'earth') === FALSE) {
    echo "earth not found in string";
}
// salida: "earth" not found in string
?>
```

```
<?php
$string = 'APPLE';
echo stristr($string, 97); // 97 = a minúscula
// salida: APPLE
?>
```

- strpos

```
<?php
$string = 'abc';
$findme = 'a';
$pos = strpos($string, $findme);

// Nótese el uso de ===. Puesto que == simple no funcionará como se espera
// porque la posición de 'a' está en el 1º (primer) caracter.
if ($pos === false) {
    echo "La cadena '$findme' no fue encontrada en la cadena '$string'";
} else {
    echo "La cadena '$findme' fue encontrada en la cadena '$string'";
    echo " y existe en la posición $pos";
}
?>
```

- strrpos

Ejemplo #1 Verificando si hay una aguja en el pajar

Es fácil confundir el valor devuelto por "caracter encontrado en la posición 0" y "caracter no encontrado". Aquí está la forma de detectar la diferencia:

```
<?php

$pos = strrpos($string, "b");
if ($pos === false) { // nota: tres signos de igual
    // no encontrado...
}

?>
```

- strstr

Encuentra la longitud del segmento inicial de **subject** que contiene *solamente* caracteres de **mask**.

Si **start** y **length** son omitidos, entonces la totalidad de **subject** será examinada. Si son incluidos, entonces el efecto será el mismo que llamando strstr(substr(\$subject, \$start, \$length), \$mask) (ver [substr](#) para más información).

La línea de código:

```
<?php
$var = strstr("42 is the answer to the 128th question.", "1234567890");
?>
```

asignará 2 a **\$var**, debido a que el string "42" es el segmento inicial de **subject** que consiste solamente en caracteres contenidos dentro de "1234567890".

- strcspn

```
<?php
$a = strcspn('abcd', 'apple');
$b = strcspn('abcd', 'banana');
$c = strcspn('hello', 'l');
$d = strcspn('hello', 'world');
$e = strcspn('abcdhelloabcd', 'abcd', -9);
$f = strcspn('abcdhelloabcd', 'abcd', -9, -5);

var_dump($a);
var_dump($b);
var_dump($c);
var_dump($d);
var_dump($e);
var_dump($f);
?>
```

El resultado del ejemplo sería:

El resultado del ejemplo sería:

```
int(0)
int(0)
int(2)
int(2)
int(5)
int(4)
```

- strcmp

```
<?php
$var1 = "Hola";
$var2 = "hola";
if (strcmp($var1, $var2) !== 0) {
    echo '$var1 no es igual a $var2 en una comparación que considera mayúsculas y minúsculas';
}
?>
```

- strcasecmp

Ejemplo #1 Ejemplo de strcasecmp()

```
<?php
$var1 = "Hello";
$var2 = "hello";
if (strcasecmp($var1, $var2) == 0) {
    echo '$var1 is equal to $var2 in a case-insensitive string comparison';
}
?>
```

- strnatcmp

(PHP 4, PHP 5, PHP 7, PHP 8)

strnatcmp — Comparación de strings utilizando un algoritmo de "orden natural"

Descripción

```
strnatcmp(string $str1, string $str2): int
```

Esta función implementa un algoritmo de comparación que ordena strings alfanuméricos de la manera en que un humano lo haría, lo cual se describe como "orden natural". Tener en cuenta que esta comparación es sensible a mayúsculas y minúsculas.

Parámetros

- strlen

Ejemplo #1 Un ejemplo de strlen()

```
<?php
$str = 'abcdef';
echo strlen($str); // 6

$str = ' ab cd ';
echo strlen($str); // 7
?>
```

- substr

```
<?php
$rest = substr("abcdef", -1);    // devuelve "f"
$rest = substr("abcdef", -2);    // devuelve "ef"
$rest = substr("abcdef", -3, 1); // devuelve "d"
?>
```

- substr_replace

substr_replace

(PHP 4, PHP 5, PHP 7, PHP 8)

substr_replace — Reemplaza el texto dentro de una porción de un string

Descripción

```
<?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original: $var<br />\n";

/* Estos dos ejemplos reemplazan todo $var por 'bob'. */
echo substr_replace($var, 'bob', 0) . "<br />\n";
echo substr_replace($var, 'bob', 0, strlen($var)) . "<br />\n";

/* Inserta 'bob' justo al comienzo de $var. */
echo substr_replace($var, 'bob', 0, 0) . "<br />\n";

/* Estos dos siguientes reemplazan 'MNRPQR' en $var por 'bob'. */
echo substr_replace($var, 'bob', 10, -1) . "<br />\n";
echo substr_replace($var, 'bob', -7, -1) . "<br />\n";

/* Elimina 'MNRPQR' de $var. */
echo substr_replace($var, '', 10, -1) . "<br />\n";
?>
```

Ejemplo #2 Usando substr_replace() para reemplazar múltiples strings de una vez

```
<?php
$input = array('A: XXX', 'B: XXX', 'C: XXX');

// Un caso simple: reemplazar XXX en cada string con YYY.
echo implode('; ', substr_replace($input, 'YYY', 3, 3))."\n";

// Un caso más complicado donde cada reemplazo es diferente.
$replace = array('AAA', 'BBB', 'CCC');
echo implode('; ', substr_replace($input, $replace, 3, 3))."\n";

// Reemplaza un número diferente de caracteres cada vez.
$length = array(1, 2, 3);
echo implode('; ', substr_replace($input, $replace, 3, $length))."\n";
?>
```

El resultado del ejemplo sería:

```
A: YYY; B: YYY; C: YYY
A: AAA; B: BBB; C: CCC
A: AAAXX; B: BBBX; C: CCC
```

- str_replace

str_replace

(PHP 4, PHP 5, PHP 7, PHP 8)

str_replace — Reemplaza todas las apariciones del string buscado con el string de reemplazo

Descripción

```
<?php
// Produce: <body text='black'>
$bodytag = str_replace("%body%", "black", "<body text='%body%'>");

// Produce: Hll Wrld f PHP
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");

// Produce: You should eat pizza, beer, and ice cream every day
$phrase = "You should eat fruits, vegetables, and fiber every day.";
$healthy = array("fruits", "vegetables", "fiber");
$yummy = array("pizza", "beer", "ice cream");

$newphrase = str_replace($healthy, $yummy, $phrase);

// Produce: 2
$str = str_replace("ll", "", "good golly miss molly!", $count);
echo $count;
?>
```

- strtr

(PHP 4, PHP 5, PHP 7, PHP 8)

strtr — Convierte caracteres o reemplaza substrings

```
<?php
//De esta forma, strtr() hace una conversión byte a byte
//Por lo tanto, aquí se asume una codificación de un solo byte:
$addr = strtr($addr, "äö", "ao");
?>
```

```
<?php
$convo = array("h" => "-", "hola" => "hey", "hey" => "hola");
echo strtr("hey, dije hola", $convo);
?>
```

hola, dije hey

```
<?php
echo strtr("baab", "ab", "01"), "\n";

$convo = array("ab" => "01");
echo strtr("baab", $convo);
?>
```

El resultado del ejemplo sería:

```
1001
ba01
```

- substr_count

substr_count

(PHP 4, PHP 5, PHP 7, PHP 8)

substr_count — Cuenta el número de apariciones del substring

```
<?php
$text = 'This is a test';
echo strlen($text); // 14

echo substr_count($text, 'is'); // 2

// el string es reducido a 's is a test', así que muestra 1
echo substr_count($text, 'is', 3);

// el texto es reducido a 's i', así que muestra 0
echo substr_count($text, 'is', 3, 3);

// genera una advertencia debido a que 5+10 > 14
echo substr_count($text, 'is', 5, 10);

// muestra sólo 1, debido a que no cuenta subcadenas traslapadas.
$text2 = 'gcdgcdgcd';
echo substr_count($text2, 'gcdgcd');
?>
```

- mixed count_chars

```
<?php
$data = "Two Ts and one F.";

foreach (count_chars($data, 1) as $i => $val) {
    echo "There were $val instance(s) of \"", chr($i), "\" in the string.\n";
}
?>
```

The above example will output:

```
There were 4 instance(s) of " " in the string.
There were 1 instance(s) of "." in the string.
There were 1 instance(s) of "F" in the string.
There were 2 instance(s) of "T" in the string.
There were 1 instance(s) of "a" in the string.
There were 1 instance(s) of "d" in the string.
There were 1 instance(s) of "e" in the string.
There were 2 instance(s) of "n" in the string.
There were 2 instance(s) of "o" in the string.
There were 1 instance(s) of "s" in the string.
There were 1 instance(s) of "w" in the string.
```

- rtrim

rtrim

(PHP 4, PHP 5, PHP 7, PHP 8)

rtrim — Retira los espacios en blanco (u otros caracteres) del final de un string

- rtrim , trim

ltrim

(PHP 4, PHP 5, PHP 7, PHP 8)

ltrim — Retira espacios en blanco (u otros caracteres) del inicio de un string

trim

(PHP 4, PHP 5, PHP 7, PHP 8)

trim — Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena

- str_pad

str_pad

(PHP 4 >= 4.0.1, PHP 5, PHP 7, PHP 8)

str_pad — Rellena un string hasta una longitud determinada con otro string

Ejemplo #1 Ejemplo de str_pad()

```
<?php
$input = "Alien";
echo str_pad($input, 10);           // produce "Alien    "
echo str_pad($input, 10, "-", STR_PAD_LEFT); // produce "-----Alien"
echo str_pad($input, 10, "_", STR_PAD_BOTH);  // produce "__Alien__"
echo str_pad($input, 6, "___");           // produce "Alien_"
echo str_pad($input, 3, "*");             // produce "Alien"
?>
```

- strtolower

(PHP 4, PHP 5, PHP 7, PHP 8)

`strtolower` — Convierte un string a minúsculas

- `ucfirst`

ucfirst

(PHP 4, PHP 5, PHP 7, PHP 8)

`ucfirst` — Convierte el primer caracter de una cadena a mayúsculas

- `ucwords`

ucwords

(PHP 4, PHP 5, PHP 7, PHP 8)

`ucwords` — Convierte a mayúsculas el primer caracter de cada palabra de una cadena

- `addslashes`

addslashes

(PHP 4, PHP 5, PHP 7, PHP 8)

`addslashes` — Escapa un string con barras invertidas

```
<?php
$str = "Is your name O'Reilly?";

// Outputs: Is your name O\'Reilly?
echo addslashes($str);
?>
```

- `stripslashes`

stripslashes

(PHP 4, PHP 5, PHP 7, PHP 8)

stripslashes — Quita las barras de un string con comillas escapadas

Ejemplo #1 Un ejemplo de stripslashes()

```
<?php
$str = "Is your name O'reilly?";

// Salida: Is your name O'reilly?
echo stripslashes($str);
?>
```

- nl2br

nl2br

(PHP 4, PHP 5, PHP 7, PHP 8)

nl2br — Inserta saltos de línea HTML antes de todas las nuevas líneas de un string

Ejemplos

Ejemplo #1 Usar nl2br()

```
<?php
echo nl2br("foo no es\n bar");
?>
```

El resultado del ejemplo sería:

```
foo no es<br />
bar
```

- addslashes

addslashes

(PHP 4, PHP 5, PHP 7, PHP 8)

addslashes — Escapa una cadena al estilo de C

```
<?php
echo addslashes('foo[ ]', 'A..z');
// salida:  \f\o\o\[ \]
// Serán escapadas todas las letras mayúsculas y minúsculas
// ... pero también [\]^_`
?>
```

Además, si el primer carácter del rango tiene un valor ASCII superior al segundo carácter del rango, este no se podrá construir. Solamente serán escapados los caracteres de inicio, fin y el punto. Use la función [ord\(\)](#) para encontrar el valor ASCII de un carácter.

```
<?php
echo addslashes("zoo['.']", 'z..A');
// salida:  \zoo['\.\']
?>
```

- [stripslashes](#)

stripslashes

(PHP 4, PHP 5, PHP 7, PHP 8)

stripslashes — Desmarca la cadena marcada con [addslashes\(\)](#)

Descripción

- [quotemeta](#)

quotemeta

(PHP 4, PHP 5, PHP 7, PHP 8)

quotemeta — Escapa meta caracteres

▲ 13 ▼ kumarkulandai at gmail dot com

13 years ago

```
<?php
$str = "Hello world. (can you hear me?);
echo quotemeta($str);
?>
```

The output of the code above will be:
Hello world\. \ (can you hear me\?\)

- htmlspecialchars

htmlspecialchars

(PHP 4, PHP 5, PHP 7, PHP 8)

htmlspecialchars — Convierte caracteres especiales en entidades HTML

- htmlentities

Ejemplo #1 Ejemplo de htmlspecialchars()

```
<?php
$nuevo = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
echo $nuevo; // &lt;a href=&#039;test&#039;&gt;Test&lt;/a&gt;
?>
```

- htmlspecialchars_decode

htmlspecialchars_decode

(PHP 5 >= 5.1.0, PHP 7, PHP 8)

htmlspecialchars_decode — Convierte entidades HTML especiales de nuevo en caracteres

Contantes disponibles para flags

Nombre de la Constante	Descripción
ENT_COMPAT	Convertirá las comillas dobles y dejará solamente las comillas sencillas.
ENT_QUOTES	Convertirá tanto las comillas dobles como las sencillas.
ENT_NOQUOTES	Dejará tanto las comillas dobles como las sencillas sin convertir.
ENT_HTML401	Maneja el código como HTML 4.01.
ENT_XML1	Maneja el código como XML 1.
ENT_XHTML	Maneja el código como XHTML.
ENT_HTML5	Maneja el código como HTML 5.

```
<?php
$str = "<p>this -&gt; &quot;</p>\n";

echo htmlspecialchars_decode($str);

// tenga en cuenta que aquí las comillas no se convierten
echo htmlspecialchars_decode($str, ENT_NOQUOTES);
?>
```

El resultado del ejemplo sería:

```
<p>this -> "</p>
<p>this -> &quot;</p>
```

- html_entity_decode

html_entity_decode

(PHP 4 >= 4.3.0, PHP 5, PHP 7, PHP 8)

html_entity_decode — Convierte todas las entidades HTML a sus caracteres correspondientes

Ejemplo #1 Decodificación de entidades HTML

```
<?php
$orig = "I'll \"walk\" the <b>dog</b> now";

$a = htmlentities($orig);

$b = html_entity_decode($a);

echo $a; // I'll &quot;walk&quot; the &lt;b&gt;dog&lt;/b&gt; now

echo $b; // I'll "walk" the <b>dog</b> now
?>
```

- strtok

strtok() divide un string (**str**) en strings más pequeños (tokens), con cada token delimitado por cualquier caracter de **token**. Es decir, si se tiene un string como "Este es un string de ejemplo", se puede tokenizar en sus palabras individuales utilizando el caracter de espacio como el token.

Ejemplo #1 Ejemplo de strtok()

```
<?php
$string = "This is\tan example\nstring";
/* Utiliza tabulador y nueva línea como caracteres de tokenización, así */
$tok = strtok($string, " \n\t");

while ($tok !== false) {
    echo "Word=$tok<br />";
    $tok = strtok(" \n\t");
}
?>
```

El comportamiento cuando se encuentra una parte vacía, cambió con PHP 4.1.0. El comportamiento anterior devolvía una cadena vacía, mientras que el comportamiento nuevo y correcto simplemente se salta esa parte de string:

Ejemplo #2 Comportamiento anterior de strtok()

```
<?php
$first_token = strtok('/something', '/');
$second_token = strtok('/');
var_dump($first_token, $second_token);
?>
```

El resultado del ejemplo sería:

```
string(0) ""
string(9) "something"
```

- chunk_split

chunk_split

(PHP 4, PHP 5, PHP 7, PHP 8)

chunk_split — Divide una cadena en trozos más pequeños

Descripción

```
chunk_split(string $body, int $chunklen = 76, string $end = "\r\n"): string
```

Ejemplo #1 Ejemplo de chunk_split()

```
<?php
// formatea $datos usando la semántica del RFC 2045
$nueva_cadena = chunk_split(base64_encode($datos));
?>
```

- str_split

str_split

(PHP 5, PHP 7, PHP 8)

str_split — Convierte un string en un array

Ejemplo #1 Ejemplos de uso de str_split()

```
<?php

$str = "Hello Friend";

$arr1 = str_split($str);
$arr2 = str_split($str, 3);

print_r($arr1);
print_r($arr2);

?>
```

El resultado del ejemplo sería:

```
Array
(
    [0] => H
    [1] => e
    [2] => l
    [3] => l
    [4] => o
    [5] =>
    [6] => F
    [7] => r
    [8] => i
    [9] => e
    [10] => n
    [11] => d
)
```

Array

```
Array
(
    [0] => Hel
    [1] => lo
    [2] => Fri
    [3] => end
)
```

- array explode

explode

(PHP 4, PHP 5, PHP 7, PHP 8)

explode — Divide un string en varios string

Descripción

```
explode(string $delimiter, string $string, int $limit = PHP_INT_MAX): array
```

Devuelve un array de string, siendo cada uno un substring del parámetro **string** formado por la división realizada por los delimitadores indicados en el parámetro string **separator**.

Ejemplos

Ejemplo #1 Ejemplo de explode()

```
<?php
// Ejemplo 1
$pizza = "porción1 porción2 porción3 porción4 porción5 porción6";
$porciones = explode(" ", $pizza);
echo $porciones[0]; // porción1
echo $porciones[1]; // porción2

// Ejemplo 2
$datos = "foo:*:1023:1000::/home/foo:/bin/sh";
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $datos);
echo $user; // foo
echo $pass; // *

?>
```

- string implode

implode

(PHP 4, PHP 5, PHP 7, PHP 8)

implode — Une elementos de un array en un string

```
<?php

$array = ['lastname', 'email', 'phone'];
var_dump(implode(",", $array)); // string(20) "lastname,email,phone"

// Devuelve un string vacío si se usa un array vacío:
var_dump(implode('hello', [])); // string(0) ""

// El separador es opcional:
var_dump(implode(['a', 'b', 'c'])); // string(3) "abc"

?>
```

- filter_var

filter_var

(PHP 5 >= 5.2.0, PHP 7, PHP 8)

filter_var — Filtra una variable con el filtro que se indique

```
<?php
// Para filtros que acepten opciones, usar este formato
$opciones = array(
    'options' => array(
        'default' => 3, // valor a retornar si el filtro falla
        // más opciones aquí
        'min_range' => 0
    ),
    'flags' => FILTER_FLAG_ALLOW_OCTAL,
);
$var = filter_var('0755', FILTER_VALIDATE_INT, $opciones);

// Para filtros que únicamente acepten flags, se pueden pasar directamete
$var = filter_var('oops', FILTER_VALIDATE_BOOLEAN, FILTER_NULL_ON_FAILURE);

// Para filtros que únicamente acepten flags, también pueden pasarse como
// array
$var = filter_var('oops', FILTER_VALIDATE_BOOLEAN,
    array('flags' => FILTER_NULL_ON_FAILURE));

// Filtro "callback2"
function foo($value)
{
    // Formato esperado: Apellido, Nombre
```

Manual cadenas PHP

Diego Casero Martin

```
<?php
var_dump(filter_var('bob@example.com', FILTER_VALIDATE_EMAIL));
var_dump(filter_var('http://example.com', FILTER_VALIDATE_URL, FILTER_FLAG_PATH_REQUIRED));
?>
```

El resultado del ejemplo sería:

```
string(15) "bob@example.com"
bool(false)
```