

substr_count() substr_count(\$haystack, \$needle, \$offset, \$length)

Returns the no. of times for a substr occurred in a string.

Example:

```
<?php
```

```
$text = 'This is a test';
```

```
echo strlen($text); // 14
```

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

// the string is reduced to 's is a test', so it prints 1

```
echo substr_count($text, 'is', 3);
```

// the text is reduced to 's i', so it prints 0

```
echo substr_count($text, 'is', 3, 3);
```

// generates a warning because $5+10 > 14$

```
echo substr_count($text, 'is', 5, 10);
```

// prints only 1, because it doesn't count overlapped substrings

```
$text2 = 'qcdgdcgd';
```

```
echo substr_count($text2, 'qcdgcd');
```

>

str_replace() str_replace(\$search, \$replace, \$subject[, \$count])

Replace all occurrences of the search string with the replacement string. \$count, if passed, this will hold the

Example:

no. of matched and replaced needles

```
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
```

```
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
```

```
// HLl World f PHP
```

```
$phrase = "You should eat fruits, vegetables, and fiber every  
day.";
```

```
$healthy = array("fruits", "vegetables", "fiber");
```

```
$yummy = array("pizza", "beer", "ice cream");
```

```
// You should eat pizza, beer, and ice cream every day
```

```
str_replace($healthy, $yummy, $phrase);
```

```
$str = str_replace("ll", "l", "good golly miss molly", $count);
```

```
echo $count; //2
```


int strlen()

Returns number of characters in a string.

Example:

Output:

```
<?php
$str = 'abcdef';
echo strlen($str);
$str = ' ab cd ';
echo strlen($str);
?>
```

6
7

string strtolower()

Returns string with all alphabetic characters converted to lowercase.

string strtoupper()

Returns string with all alphabetic characters converted to uppercase.

Example:

```
<?php
$str = "Marry Had A Little Lamb and She LOVED It So";
$str1 = strtolower($str);
echo $str1; echo "\n";
$str2 = strtoupper($str);
echo $str2;
?>
```

Output:

marry had a little lamb and she loved it so

MARRY HAD A LITTLE LAMB AND SHE LOVED IT SO

string trim(string \$str [, string \$charlist])

strip whitespace (or other characters) from the beginning and end of a string.

Example:

```
<?php
```

```
$text = " !+These :) ... ";
```

```
$hello $hello = "Hello World";
```

```
$trimmed = trim($text); //These :) ...
```

```
var_dump($trimmed); ✓
```

```
$trimmed = trim($text, " !+."); //These :) ✓
```

```
var_dump($trimmed); ✓
```

```
$trimmed = trim($hello, "Hdle"); //o Wor
```

```
var_dump($trimmed); ✓
```

```
?>
```

Output:

```
string(12) "These :) ..."
```

```
string(8) "These :)"
```

```
string(5) "o Wor"
```

Date

25/07/2019

string strpos(string \$substring, mixed \$who, string [, \$before_substring = false])

Returns the substring, starting from the first occurrence of the \$substring to the end of the \$whole string and false if no match found. Searching process is case sensitive.

Example:

```
$email = 'name@example.com';
```

```
$domain = strpos($email, '@');
```

```
echo $domain; // @example.com
```


addslashes()

Returns a string with backslashes before characters `single quote (')`, `double quote (")`, `backslash (\)` and `NUL (the NULL byte)`.

Example:

```
<?php
```

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

```
echo addslashes($str);
```

```
?>
```

Output:

Is your name O\'reilly?

mysql_real_escape_string()

Returns a string with backslashes before special characters in a string for use in a SQL statement.

This function uses backslashes for the following characters:

`\eOO`, `"`, `'`, `\r`, `\n`, `\t`, `\b`, `\f`, `\z`

stripslashes()

Un-quote string quoted with `addslashes()`.

Example:

```
<?php
```

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

```
echo stripslashes($str);
```

```
?>
```

Output:

Is your name O'reilly?