

`strlen($str)`

This function returns the length of the string or the number of characters in the string including whitespaces.

```
<?php

$str = "Welcome to Studytonight";

// using strlen in echo method
echo "Length of the string is: ". strlen($str);

?>
```

Length of the string is: 23

`str_word_count($str)`

This function returns the number of words in the string. This function comes in handy in form field validation for some simple validations.

```
<?php

$str = "Welcome to Studytonight";
```

```
// using str_word_count in echo method

echo "Number of words in the string are: ".
str_word_count($str);
```

?>

```
Number of words in the string are: 3
```

`strrev($str)`

This function is used to reverse a string.

Let's take an example and see,

```
<?php

$str = "Welcome to Studytonight";

// using strrev in echo method

echo "Reverse: ". strrev($str);
```

?>

```
Reverse: thginotydutS ot emocleW
```

strpos(\$str, \$text)

This function is used to find the position of any text/word in a given string. Just like an [array](#), string also assign index value to the characters stored in it, starting from zero.

```
<?php

$str = "Welcome toStudytonight";

// using strpos in echo method

echo "Position of 'Studytonight' in string: ".
strpos($str, 'Studytonight');

?>
```

Position of 'Studytonight' in string: 11

str_replace(\$replacethis, \$replacewith, \$str)

This function is used to replace a part of the string with some text. While using this function, the first argument is the part of string that you want to replace, second argument is the new text you want to include, and the last argument is the string variable itself.

Let's take an example,

```
<?php
```

```
$str = str_replace("Studytonight", "Studytonight.com",  
"Welcome to Studytonight");
```

```
echo $str;
```

```
?>
```

Copy

```
Welcome to Studytonight.com
```

`ucwords($str)`

This function is used for formatting the string. This function converts first letter/character of every word in the string to uppercase.

Let's take an example and see,

```
<?php
```

```
$str = "welcome to studytonight";
```

```
echo ucwords($str);
```

```
?>
```

Welcome To Studytonight

strtoupper(\$str)

To convert every letter/character of every word of the string to uppercase, one can use `strtoupper()` method.

```
<?php

$str = "welcome to studytonight";

echo strtoupper($str);

?>
```

Copy

WELCOME TO STUDYTONIGHT

strtolower(\$str)

This function is used to convert every letter/character of a string to lowercase.

```
<?php

$str = "WELCOME TO STUDYTONIGHT";
```

```
echo strtolower($str);
```

```
?>
```

Copy

```
welcome to studytonight
```

`str_repeat($str, $counter)`

This function is used to repeat a string a given number of times. The first argument is the string and the second argument is the number of times the string should be repeated.

```
<?php
```

```
$str = "Studytonight";
```

```
echo str_repeat($str, 4);
```

```
?>
```

Copy

```
StudytonightStudytonightStudytonightStudytonight
```

strcmp(\$str1, \$str2)

This function is used to compare two strings. The comparison is done alphabetically. If the first string is greater than second string, the result will be greater than 0, if the first string is equal to the second string, the result will be equal to 0 and if the second string is greater than the first string, then the result will be less than 0.

```
<?php

$str1 = "Studytonight";
$str2 = "Studytonight.com";

// comparing str1 and str2
echo strcmp($str1, $str2);

// comparing str2 and str1
echo strcmp($str2, $str1);

// comparing str1 with str1
echo strcmp($str1, $str1);

?>
```

Copy

-4
4
0

`substr($str, $start, $length)`

This function is used to take out a part of the string(substring), starting from a particular position, of a particular length.

The first argument is the string itself, second argument is the starting index of the substring to be extracted and the third argument is the length of the substring to be extracted.

```
<?php

$str = "Welcome to Studytonight";

echo substr($str, 11, 12);

?>
```

Copy

Studytonight

`trim($str, charlist)`

This function is used to remove extra whitespaces from beginning and the end of a string. The second argument **charlist** is optional. We can provide a list of character, just like a string, as the second argument, to trim/remove those characters from the main string.

```
<?php

$str1 = "    Hello World    ";

echo trim($str1) . "<br/>";

$str2 = "Hello Hello";

echo trim($str2, "Heo");

?>
```

Copy

```
Hello World
llo Hell
```

As you can see in the output, additional spaces from the beginning and end are removed and in the second case, the characters specified are removed from the beginning and the end of the string.

`explode(separator, $str, $limit)`

This function is used to break a string, create an array of the broken parts of the string and return the array. The first argument, **separator** defines where to break the string from. It can be a space, hyphen(-) or any other character.

The second argument of this function is the string itself and the third argument is the **limit**, which specifies the number of array elements to return. the third argument is optional.

Let's have an example,

```
<?php

$str = "Its a beautiful day";

// we can directly print the result of explode
print_r(explode(" ", $str));

?>
```

Copy

```
Array (
    [0] => Its
    [1] => a
    [2] => beautiful
    [3] => day
)
```

In the example above, we have provided **space** as separator to break the string and return an array.

If we provide the third argument **limit** as well, we can limit the number of array elements returned. For example, if we provide **2** as the third argument, then we will only get 2 elements in the array, the first two.

`implode(separator, $arr)`

This function is used to form a string using the array elements from the array provided and join them using the **separator**.

Let's take an example,

```
<?php

$arr = array("Its", "a", "beautiful", "day");

// <br> is used to jump to next line

echo implode(" ", $arr) . "<br>";
echo implode("-", $arr) . "<br>";
echo implode("/", $arr) . "<br>";

?>
```

Copy

```
Its a beautiful day
Its-a-beautiful-day
Its/a/beautiful/day
```

nl2br(\$str)

This function is used to change line break or `\n` to the HTML tag for line break, which is `
`.

This function is very useful to format string data to display on HTML pages because when a multiline form data is submitted, it has `\n` included in the string for line breaks, but when you display it on your HTML page, the line breaks will not get rendered because HTML doesn't understand `\n`.

```
<?php

echo nl2br("Its a\nbeautiful day");

?>
```

Copy

```
Its a
beautiful day
```