11 Most Important String Functions PHP a programmer should know

Note : In the below mentioned functions the parameters marked in < and > tags are mandatory and the parameters marked in [< and >] are optional

substr()
This function returns the part of the string as an output.

Syntax :
substr(<string>,<start>,[<length>]);

Explanation :
String : It is mandatory parameter. The string from which the part is to be extracted is mentioned here.
Start : The start in the string from which the characters are to be extracted

- Positive number - Start at a specified position in the string
- Negative number - Start at a specified position from the end of the string
- 0 - Start at the first character in string

Length : It is an optional parameter. It specifies the length of the string which is to be extracted.

- Positive number - The length to be returned from the start parameter
- Negative number - The length to be returned from the end of the string

Example 1:
<?php echo substr("Hello world",6); ?> //Returns world

Example 2 :
<?php echo substr("Hello world",6,4); ?> // Returns worl

Example 3 :
<?php echo substr("Hello world", -1); ?> // Returns d

Example 4:
<?php echo substr("Hello world", -3, -1); ?> // Returns rl

strlen()
This function returns the length of the string

Syntax :
strlen(<string>);

Explanation:
String : It is mandatory field. The string whose length is to be found out is mentioned here.

Example 1:

```
<?php echo strlen("Hello world");  ?> // Returns 11
```

trim()
This function removes the whitespaces from both start and the end of the string.

Syntax :
trim(<string>);

Explanation :
String : It is mandatory field. The string of which the whitespaces are to be removed is passed as parameter.

Example 1:
<?php echo trim(" Hello World "); ?> // returns Hello World. If you go view source then you can see that there are no whitespaces.

ltrim()
This function removes the whitespaces from the left part of the string.

Syntax :
ltrim(<string>);

Explanation :
String : It is mandatory field. The string of which the whitespaces are to be removed from left side is passed as parameter.

Example 1:
<?php echo ltrim(" Hello World "); ?> // returns Hello World. If you go view source then you can see that there are no whitespaces on left side but there are spaces on right side.

rtrim()
This function removes the whitespaces from the right part of the string.

Syntax :
rtrim(<string>);

Explanation :
String : It is mandatory field. The string of which the whitespaces are to be removed from right side is passed as parameter.

Example 1:
<?php echo rtrim(" Hello World "); ?> // returns Hello World. If you go view source then you can see that there are no whitespaces on right side but there are spaces on left side

strtolower()
This function converts the string to lower case

Syntax :
strtolower(<string>);

Explanation :
String : It is mandatory field. The string which is to be converted to lower case is passed here.

Example 1:
<?php echo strtolower("HELLO WORLD"); ?> // Returns hello world

strtoupper()
This function converts the string to upper case

Syntax :
strtoupper(<string>);

Explanation :
String : It is mandatory field. The string which is to be converted to upper case is passed here.

Example 1:
<?php echo strtoupper("hello world"); ?> // Returns HELLO WORLD

str_replace()
The str_replace() function replaces some characters with some other characters in a string.
This function works by the following rules:

- If the string to be searched is an array, it returns an array
- If the string to be searched is an array, find and replace is performed with every array element
- If both find and replace are arrays, and replace has fewer elements than find, an empty string will be used as replace
- If find is an array and replace is a string, the replace string will be used for every find value

Syntax :
str_replace(<search>,<replace>,<string/array>,[<count>]);

Explanation :
Search : It is mandatory . The string or value to be searched comes here.
Replace : It is mandatory. The string or value to be replaced comes here.
String/Array : It is mandatory. The string or array in which the value is to be found out comes here.
Count : It is optional. It counts the number of replacements to be done.

Example 1:
<?php echo str_replace("world","Peter","Hello world"); ?>// Returns Hello Peter

Example 2:
<?php
$arr = array("blue","red","green","yellow");

```
print_r(str_replace("red","pink",$arr,$i));
echo "Replacements: $i";
?>
/*
Output :
Array
(
[0] => blue
[1] => pink
[2] => green
[3] => yellow
)
Replacements: 1
*/

Example 3:
<?php
$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);
?>
/*
Output :
You should eat pizza, beer, and ice cream every day
*/
```

strcmp()
The strcmp() function compares two strings.
This function returns:

- 0 - if the two strings are equal
- <0 - if string1 is less than string2
- >0 - if string1 is greater than string2

Syntax :
strcmp(<string1>,<string2>);

Explanation :
String1 : It is mandatory. The first string comes here.
String 2 : It is mandatory. The Second string comes here.

Example 1:
<?php echo strcmp("Hello world!","Hello world!"); ?> //Returns 0

Note: The strcmp() function is binary safe and case-sensitive. For case insensitive comparison you can use strcasecmp(<string1>,<string2>); function. It is similar to strcmp() function.

explode()
This function breaks the string into array on the basis of delimiter passed.

Syntax:
explode(<delimeter>,<string>,[<limit>]);

Explanation:
Delimeter: It is mandatory field. It specifies where to break the string.
String: It is mandatory. It specifies the string to split.
Limit : It is optional. It specifies the maximum number of array elements to return.

Example 1:
```
<?php
$str = "Hello world. It's a beautiful day.";
print_r (explode(" ",$str));
?>
/* Output :
Array
(
[0] => Hello
[1] => world.
[2] => It's
[3] => a
[4] => beautiful
[5] => day.
)
*/
```

implode()
This function join array elements with a string on the basis of delimiter passed.

Syntax:
implode(<delim>,<array>);

Explanation:
Delimiter: It is mandatory field. It specifies what to put between the array elements. Default is "" (an empty string).
Array: It is mandatory field. It specifies the array to join to a string.

Example 1:
```
<?php
$arr = array('Hello','World!','Beautiful','Day!');
echo implode(" ",$arr);
?>
/*
Output:
Hello World! Beautiful Day!
*/
```