**PHP String Functions: substr, strlen, strtolower, explode, strpos, str\_replace**

**What is a string?**

A string is a collection of characters. String is one of the data types supported by PHP.

The string variables can contain alphanumeric characters. Strings are created when;

* You declare variable and assign string characters to it
* You can directly use them with echo statement.
* String are language construct, it helps capture words.
* Learning how strings work in PHP and how to manipulate them will make you a very effective and productive developer.

**In this tutorial, you will learn-**

* [Create strings](https://www.guru99.com/php-strings.html#2)
* [Create Strings Using Double quotes](https://www.guru99.com/php-strings.html#3)
* [Heredoc](https://www.guru99.com/php-strings.html#4)
* [Nowdoc](https://www.guru99.com/php-strings.html#5)
* [String functions](https://www.guru99.com/php-strings.html#6)

**PHP Create strings**

Let’s now look at the four different ways of creating strings.

Creating Strings Using Single quotes: The simplest way to create a string is to use single quotes.

Let’s look at an example that creates a simple string in PHP.

<?php

    var\_dump("You need to be logged in to view this page");

?>

**Output:**

string(42) "You need to be logged in to view this page"

If the single quote is part of the string value, it can be escaped using the backslash.

The code below illustrates how to escape a single quote.

<?php

echo "I \'ll be back after 20 minutes";

?>

**Output:**

I \'ll be back after 20 minutes

**PHP Create Strings Using Double quotes**

The double quotes are used to create relatively complex strings compared to single quotes.

Variable names can be used inside double quotes and their values will be displayed.

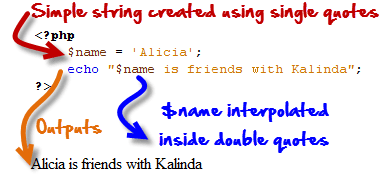
Let’s look at an example.

<?php

$name='Alicia';

echo "$name is friends with kalinda";

?>

[](https://www.guru99.com/images/2013/04/php_variable_interporation.png)

HERE,

* The above example creates a simple string with the value of Alicia.
* The variable name is then used in the string created using double quotes and its value is interpolated at run time.

**Output:**

Alicia is friends with kalinda

In addition to variable interpolations, the double quote string can also escape more special characters such as “\n for a linefeed, \$ dollar for the dollar sign” etc.

More examples Let’s suppose that we have the following code

<?php $pwd = "pas$word"; echo $pwd; ?>

**Output:**

NOTICE : Undefined variable

pas

executing the above codes issues a notice “Notice: Undefined variable”.

This is because $word is treated as a variable.

If we want the dollar sign to be treated as a literal value, we have to escape it.

<?php

$word="word";

$pwd = "pas\$word";

echo $pwd; ?>

**Output:**

pas$word

**PHP Heredoc**

This heredoc methodology is used to create fairly complex strings as compared to double quotes.

The heredoc supports all the features of double quotes and allows creating string values with more than one line without php string concatenation.

Using double quotes to create strings that have multiple lines generates an error.

You can also use double quotes inside without escaping them.

The example below illustrates how the Heredoc method is used to create string values.

<?php

$baby\_name = "Shalon";

echo <<<EOT

    When $baby\_name was a baby,

    She used to look like a "boy".

EOT;

?>

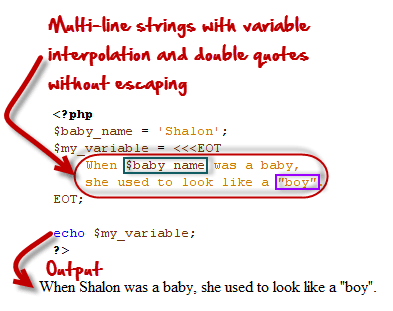
  HERE,

**<<<EOT** is the string delimiter.

EOT is the acronym for end of text.

It should be defined in its on line at the beginning of the string and at the end.

Note: you can use anything you like in place of EOT

[](https://www.guru99.com/images/2013/04/php_eot.png)

**Output:**

When Shalon was a baby, She used to look like a "boy".

**PHP Nowdoc**

The Nowdoc string creation method is similar to the heredoc method but works like the way single quotes work.

No parsing takes place inside the Nowdoc.

Nowdoc is ideal when working with raw data that do not need to be parsed.

The code below shows the Nowdoc implementation

<?php

$baby\_name = "Shalon";

$my\_variable = <<<'EOT'

    When $baby\_name was a baby,

    She used to look like a "boy".

EOT;

echo $my\_variable;

?>

**Output:**

When $baby\_name was a baby, She used to look like a "boy".

**PHP string functions**

PHP string functions are used to manipulate string values.

We are now going to look at some of the commonly used string functions in PHP

| **Function** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- |
| strtolower | Used to convert all string characters to lower case letters | echo strtolower( 'Benjamin'); | outputs benjamin |
| strtoupper | Used to convert all string characters to upper case letters | echo strtoupper('george w bush'); | outputs GEORGE W BUSH |
| strlen | The string length function is used to count the number of character in a string. Spaces in between characters are also counted | echo strlen('united states of america'); | 24 |
| explode | Used to convert strings into an array variable | $settings = explode(';', "host=localhost; db=sales; uid=root; pwd=demo"); print\_r($settings); | Array ( [0] => host=localhost [1] => db=sales [2] => uid=root [3] => pwd=demo ) |
| substr | Used to return part of the string. It accepts three (3) basic parameters. The first one is the string to be shortened, the second parameter is the position of the starting point, and the third parameter is the number of characters to be returned. | $my\_var = 'This is a really long sentence that I wish to cut short';echo substr($my\_var,0, 12).'...'; | This is a re... |
| str\_replace | Used to locate and replace specified string values in a given string. The function accepts three arguments. The first argument is the text to be replaced, the second argument is the replacement text and the third argument is the text that is analyzed. | echo str\_replace ('the', 'that', 'the laptop is very expensive'); | that laptop is very expensive |
| strpos | Used to locate the and return the position of a character(s) within a string. This function accepts two arguments | echo strpos('PHP Programing','Pro'); | 4 |
| sha1 | Used to calculate the SHA-1 hash of a string value | echo sha1('password'); | 5baa61e4c 9b93f3f0 682250b6cf8331b 7ee68fd8 |
| md5 | Used to calculate the md5 hash of a string value | echo md5('password'); | 9f961034ee 4de758 baf4de09ceeb1a75 |
| str\_word\_count | Used to count the number of words in a string. | echo str\_word\_count ('This is a really long sentence that I wish to cut short'); | 12 |
| ucfirst | Make the first character of a string value upper case | echo ucfirst('respect'); | Outputs Respect |
| lcfirst | Make the first character of a string value lower case | echo lcfirst('RESPECT'); | Outputs rESPECT |

For a complete list of PHP strings, check <http://php.net/manual/en/ref.strings.php>

**Summary**

* A string is a set of characters
* single quotes are used to specify simple strings
* double quotes are used to create fairly complex strings
* heredoc is used to create complex strings
* Nowdoc is used to create strings that cannot be parsed.