

# PHP - Heredoc & Nowdoc

PHP provides two alternatives for declaring single or double quoted strings in the form of **heredoc** and **nowdoc** syntax.

- The single quoted string doesn't interpret the escape characters and doesn't expand the variables.
- On the other hand, if you declare a double quoted string that contains a double quote character itself, you need to escape it by the "\" symbol. The **heredoc** syntax provides a convenient method.

## Heredoc Strings in PHP

The heredoc strings in PHP are much like double-quoted strings, without the double-quotes. It means that they don't need to escape quotes and expand variables.

## Heredoc Syntax

```
$str = <<<IDENTIFIER  
place a string here  
it can span multiple lines  
and include single quote ' and double quotes "  
IDENTIFIER;
```

First, start with the "<<<" operator. After this operator, an identifier is provided, then a newline. The string itself follows, and then the same identifier again to close the quotation. The string can span multiple lines and includes single quotes (') or double quotes (").

The closing identifier may be indented by space or tab, in which case the indentation will be stripped from all lines in the doc string.

## Example

The identifier must contain only alphanumeric characters and underscores and start with an underscore or a non-digit character. The closing identifier should not contain any other characters except a semicolon (;). Furthermore, the character before and after the closing identifier must be a newline character only.

Take a look at the following example –

&lt;/&gt;

Open Compiler

```
<?php
$str1 = <<<STRING
Hello World
    PHP Tutorial
        by Tutorialspoint
    STRING;

echo $str1;
?>
```

It will produce the following **output** –

```
Hello World
    PHP Tutorial
        by Tutorialspoint
```

## Example

The closing identifier may or may not contain indentation after the first column in the editor. Indentation, if any, will be stripped off. However, the closing identifier must not be indented further than any lines of the body. Otherwise, a ParseError will be raised. Take a look at the following example and its output –

&lt;/&gt;

Open Compiler

```
<?php
$str1 = <<<STRING
Hello World
    PHP Tutorial
by Tutorialspoint
    STRING;

echo $str1;
?>
```

It will produce the following **output** –

PHP Parse error: Invalid body indentation level  
(expecting an indentation level of at least 16) in hello.php on line 3

## Example

The quotes in a heredoc do not need to be escaped, but the PHP escape sequences can still be used. Heredoc syntax also expands the variables.

&lt;/&gt;

Open Compiler

```
<?php
$lang="PHP";
echo <<<EOS
Heredoc strings in $lang expand variables.
The escape sequences are also interpreted.
Here, the hexadecimal ASCII characters produce \x50\x48\x50
EOS;
?>
```

It will produce the following **output** –

Heredoc strings in PHP expand variables.  
The escape sequences are also interpreted.  
Here, the hexadecimal ASCII characters produce PHP

## Nowdoc Strings in PHP

A nowdoc string in PHP is similar to a heredoc string except that it doesn't expand the variables, neither does it interpret the escape sequences.

&lt;/&gt;

Open Compiler

```
<?php
$lang="PHP";

$str = <<<'IDENTIFIER'
This is an example of Nowdoc string.
it can span multiple lines
and include single quote ' and double quotes "
```

```
IT doesn't expand the value of $lang variable  
IDENTIFIER;  
  
echo $str;  
?>
```

It will produce the following **output** –

This is an example of Nowdoc string.  
it can span multiple lines  
and include single quote ' and double quotes "  
IT doesn't expand the value of \$lang variable

The nowdoc's syntax is similar to the heredoc's syntax except that the identifier which follows the "<<<" operator needs to be enclosed in single quotes. The nowdoc's identifier also follows the rules for the heredoc identifier.

Heredoc strings are like double-quoted strings without escaping. Nowdoc strings are like single-quoted strings without escaping.