

Managing strings

A “string” of text can be stored in a variable in much the same way as a numeric value, but the assignment must surround the string with quote marks to denote its beginning and end. Both single and double quote marks can be used for this purpose but you must use the same type of quote marks to denote the beginning and end of the text string. For example, both of these statements make valid string assignments:

```
$song_title = "Summertime Blues" ;
```

```
$song_title = 'Summertime Blues' ;
```

Where you wish to store a string of text that itself includes quote marks, you can “escape” the included quote marks by preceding them with a \ backslash character, or use the alternative type of quote mark within the string. For example, both of these statements assign strings that include quote marks:

```
$song_title = "the \"Summertime\" aria by George Gershwin" ;
```

```
$song_title = 'the "Summertime" aria by George Gershwin' ;
```

The second technique, using double quote marks within the string, is easier to read and is preferred throughout this book.

String values can be displayed as part of a mixed string by enclosing the mixed string in double quotes, like this:

```
echo "Many regard $song_title as a popular classic" ;
```

The double quotes ensure that PHP will evaluate the mixed string and substitute named variables with their stored values. This feature does not work if the string is enclosed in single quotes!

String values can be joined together (“concatenated”) into a single string using the . period concatenation operator, like this:

```
$hi = 'Hello' ;  
$bye = 'Goodbye' ;  
$song_title = $hi . $bye ;          # 'HelloGoodbye'
```

Additionally, spaces and punctuation can usefully be inserted when concatenating strings, so the title assignment above could be modified to include a comma and a space:

```
$song_title = $hi . ', ' . $bye ;    # 'Hello, Goodbye'
```



Variable names cannot contain spaces – but the underscore character is often used in their place.

...cont'd

- 1 Create a valid HTML document, like the one listed on page 16, then insert PHP tags into the body section


```
<?php
  # Statements to be inserted here.
?>
```
- 2 Now, between the PHP tags insert a statement to create and initialize two variables

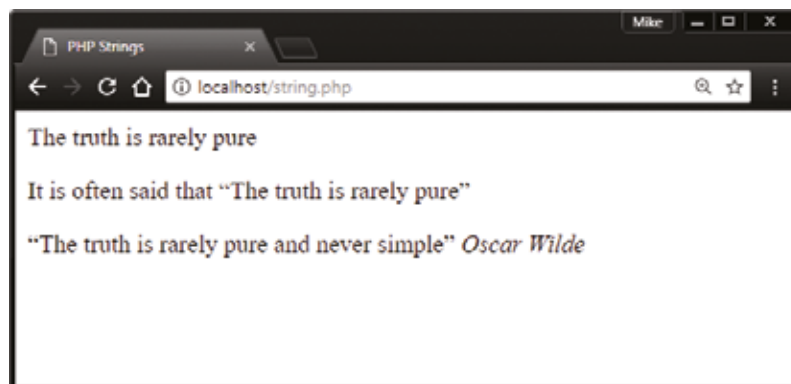

```
$phrase = 'The truth is rarely pure' ;
$author = 'Oscar Wilde' ;
```
- 3 Next, insert a statement to display a variable value alone


```
echo $phrase ;
```
- 4 Then, insert a statement to display the variable value substituted in a mixed string – assigned in double quotes


```
echo "<p>It is often said that <q>$phrase</q> </p>" ;
```
- 5 Insert a statement to concatenate a string to a variable


```
$phrase = $phrase . ' and never simple' ;
```
- 6 Finally, insert a statement to display both current variable values substituted in a mixed string


```
echo "<p><q>$phrase</q><cite>$author</cite></p>" ;
```
- 7 Save the document in your web browser's `/htdocs` directory as `string.php`, then open the page via HTTP to see the variable values get displayed



You can also use a shorthand concatenate assignment operator `.=` to join two strings – so that `$a = $a . $b` can be simply written `$a .= $b`.



Adopt a consistent naming style for your variables – such as all lowercase `$my_var`, or “camelcase” `$myVar`.