Creating variables

A "variable" is a named container in a PHP script in which a data value can be stored. The stored value can be referenced using the variable's name and changed (varied) as the script proceeds. The script author can choose any name for a variable providing it adheres to these three naming conventions:

- Names must begin with a $ dollar sign – for example **$name**.
- Names can comprise letters, numbers, and underscore characters, but not spaces – for example **$subtotal_1**.
- The first character after the $ dollar sign must be a letter or an underscore character – it cannot be a number.

Note that variable names in PHP are case-sensitive so **$name**, **$Name**, and **$NAME** are three separate individual variables.

PHP variables are "loosely typed" meaning they can contain data of any type, unlike "strongly typed" variables in some languages where the data type must be specified when the variable is created. So, a PHP variable may happily contain an integer number, or a floating-point number, or a string of text characters, or a Boolean value of **TRUE** or **FALSE**, or an object, or a **NULL** empty value.

A variable is created in a PHP script simply by stating its name. The variable can then be assigned an initial value (initialized) by using the = assignment operator to state its value. This statement, and all others in PHP, must end with a semi-colon, like this:

```
$body_temp = 98.6 ;
```

The value contained within the variable can then be displayed by referencing it using the variable name, like this:

```
echo $body_temp ;
```

Usefully, the variable's value can be displayed as part of a mixed string by enclosing the string and variable name in double quotes:

```
echo "Body temperature is $body_temp degrees Fahrenheit" ;
```

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

Do not confuse the purpose of double and single quotes. Remember that PHP only makes variable substitutions for mixed strings enclosed within double quotes.

…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.
?>
```

variable.php

2. Now, between the PHP tags insert a statement to create and initialize a variable

```
$body_temp = 98.6 ;
```

3. Next, insert a statement to display the variable value alone

```
echo $body_temp ;
```

Hot tip

Notice that variables created in the main body of a script, like the one in this example, are accessible "globally" throughout the entire PHP script.

4. Then, insert a statement to display the variable value substituted in a mixed string – assigned in double quotes

```
echo "<p>Body temperature is $body_temp
                              degrees Fahrenheit " ;
```

5. Insert a statement to assign a new value to the variable

```
$body_temp = 37.0 ;
```

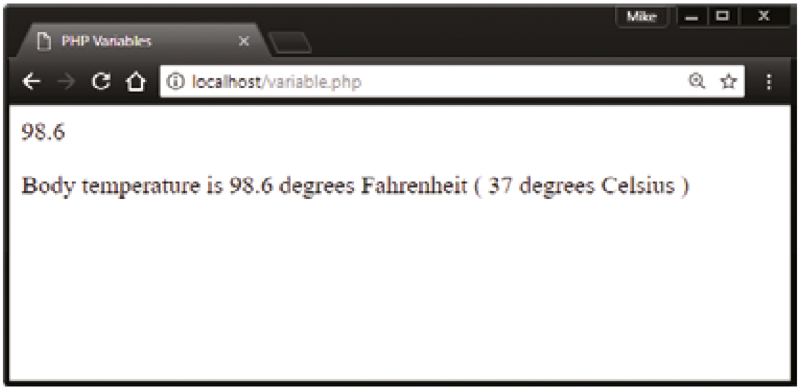
6. Finally, insert a statement to display the new variable value substituted in a mixed string

```
echo "( $body_temp degrees Celsius )</p>" ;
```

7. Save the document in your web browser's **/htdocs** directory as **variable.php**, then open the page via HTTP to see the variable values get displayed

Don't forget

Each statement in the PHP language must be terminated by a semi-colon – just as each statement in the English language must be terminated by a period.



```
98.6

Body temperature is 98.6 degrees Fahrenheit ( 37 degrees Celsius )
```