

## Chapter two exercise

1. Create two variables, one for the word “Twinkle” and one for the word “star”. Echo the statement to the browser, this time substituting the variables for the relevant words. Change the value of each variable to whatever you like, and echo the statement a third time. Remember to include code to show your statements on different lines.
2. PHP includes all the standard arithmetic operators. For this PHP exercise, you will use them along with variables to print equations to the browser. In your script, create the following variables:

```
$x=10;  
$y=7;
```

Write code to print out the following:

```
10 + 7 = 17  
10 - 7 = 3  
10 * 7 = 70  
10 / 7 = 1.4285714285714  
10 % 7 = 3
```

Use numbers only in the above variable assignments, not in the echo statements. You will need a third variable as well.

Note: this is intended as a simple, beginning exercise, not using arrays or loops. Some of the solutions in comments include these structures. If you don't understand them, just keep learning, and you will.

3. Arithmetic-assignment operators perform an arithmetic operation on the variable at the same time as assigning a new value. For this PHP exercise, write a script to reproduce the output below. Manipulate only one variable using no simple arithmetic operators to produce the values given in the statements. Hint: In the script each statement ends with "Value is now \$variable."

```
Value is now 8.  
Add 2. Value is now 10.  
Subtract 4. Value is now 6.  
Multiply by 5. Value is now 30.  
Divide by 3. Value is now 10.  
Increment value by one. Value is now 11.  
Decrement value by one. Value is now 10.
```

4. When you are writing scripts, you will often need to see exactly what is inside your variables. For this PHP exercise, think of the ways you can do that, then write a script that outputs the following, using the echo statement only for line breaks.

```
string(5) "Harry"  
Harry  
int(28)  
NULL
```

5. For this PHP exercise, write a script using the following variable:  
\$around="around";

Single quotes and double quotes don't work the same way in PHP. Using single quotes ( ' ') and the concatenation operator, echo the following to the browser, using the variable you created:

```
What goes around comes around.
```

6. PHP allows several different types of variables. For this PHP exercise, you will create one variable and assign it different values, then test its type for each value. Write a script using one variable "\$whatsit" to print the following to the browser. Your echo statements may include no words except "Value is". In other words, use the function that will output the variable type to get the requested text. Use simple HTML to print each statement on its own line and add a relevant title to your page. Include line breaks in your code to produce clean, readable HTML.

```
Value is string.
```

```
Value is double.
```

```
Value is boolean.
```

```
Value is integer.
```

```
Value is NULL.
```