# PHP

## EXERCISE 1: STARTING PHP

Here is a page of HTML you should have no trouble in understanding. Copy and paste it into a new document in ***NotePad++*** and then save it with the name **time1.html**. Look at it in a web browser.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Time</title>  <link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>  </head>  <body>  <h1>The time is 10:13!</h1>  </body>  </html> |

It says the time is 10:13. It will of course ALWAYS say the time is 10:13.

PHP allows us to create web pages which re-write themselves whenever someone views them. This allows us to make a page which displays the time it was requested. To turn **time1.html** into a PHP page that does this, the following three things are necessary:

1. Add some PHP code to it.
2. Save a copy of it giving it the name ***time1.php***.
3. Upload it to a webserver.

We need to rename it because if a file is not called something***.php*** it will not be treated as a PHP file. We need to add some PHP code to it because otherwise it won’t do anything different to the plain HTML version. We need to upload it to a web server because it is the web server that executes PHP code, if we don’t view it live on a web server the PHP won’t execute.

PHP code is inserted into a page of HTML using the special tokens **<?php** and **?>**. Change the file to look like this:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Time</title>  <link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>  </head>  <body>  **<?php**  **print "<h1>The time is ";**  **print time();**  **print "!</h1>";**  **?>**  </body>  </html> |

If you view this file locally in a web browser (for example drag and drop the file into a web browser window) the PHP won’t execute and you will see something like this displayed:

The time is "; print time(); print "!"; ?>

You must upload the file to a web browser and view it properly. Then you will see something like this:

**The time is 1442581120!**

The PHP code consists of three **print** commands[[1]](#footnote-1). The first and last print out a tiny bit of HTML just like you would have typed yourself, but the middle one prints the output of the **time()** function. The **time()** function returns the number of seconds elapsed since January 1st 1970. Refresh the page a few times to see that the number of seconds go up each time. Examine the HTML source of the page in the web browser by right-clicking and selecting “view source” or whatever works in your web browser. You will see that the HTML the browser sees is ***just*** HTML the PHP code is not delivered to the browser, only the output it produces.

The time in seconds was not what we wanted. Fortunately, PHP provides a lot of useful functions. The **date()** function converts a time value in seconds into a readable string. Change the middle print statement to:

print date( "r", time());

This passes the number of seconds into the date function. Remember to upload the changed version of the file to the web server and then refresh it in the web browser. You will see something like this:

**The time is Fri, 18 Sep 2015 14:14:23 +0100!**

Look at the date function again. The “r” in it is a string telling it how to format the time. The “r” option tells it to use a predefined standard format (the format is called RFC 2822, which is where the r comes from). Unfortunately that is much more information than we wanted.

Read the documentation on the date() function at <http://php.net/manual/en/function.date.php> and figure out what formatting string is required to print just the time as hours and minutes. Change the file to make it work that way.

### Embedding PHP

PHP can be embedded any number of times in the same file. Try adding another block to print out a the number of seconds in the title using the **time()** function. You just need to change it to look like this:

<title>**<?php print time(); ?>**</title>

Try this code in the main PHP section of your page, after the time is printed out:

$words = array( "happy", "sad", "<i>fruity</i>");

$r = rand(0,2);

print "<h2>Are you feeling ".$words[$r]."?</h2>\n";

Refresh the page a few times to see what it prints out. This code introduces PHP **variables** and **arrays**. Variables in PHP have names which start with a $. The variables in this code are $words and $r. The value assigned to $words is an ***array***, consisting of three elements. This means that $words represents a list of three items. We can refer to items in the list by giving a numerical index in square brackets after $words. That is $words[0] refers to “happy” and $words[1] refers to “sad”. Note that arrays count from zero, not one. The function **rand()** chooses a random number between the limits given 0, 2 means any whole number from 0 (inclusive) up to and including 2. So it can produce 0, 1 or 2. This function is used to set a value for the $r variable. Finally we print a fragment of HTML which includes $words[$r] in it. That will be one of the three elements in the list. Note the use of the full stop between "<h2>Are you feeling " and $words[$r] and again before "?</h2>\n". This joins together the two literal strings with the value of the variable. This lets us use just one print statement instead of having to write:

**print "<h2>Are you feeling ";**

**print $words[$r];**

**print "?</h2>\n";**

The way we did with the time message. Note also the **\n** after the close h2 tag. This is a code meaning “insert a line break character”, but this is not a line break that can be seen in the displayed output, only in the HTML source code. It is not necessary but it makes the HTML source of the web page delivered to the browser easier to read, which can help figure out mistakes when there is a lot of output. View the HTML source of the page in the web browser to see if you can see this line break there.

Try adding more words to the array and adjusting the range of random numbers rand() generates to match the length of the array. See if you can use the **count** function to automatically make the rand function choose the right length for the array. (<http://php.net/manual/en/function.count.php>).

Try changing the background colour of the web page by writing PHP code to print out a style sheet between <style>… </style> tags in the head of the page, with the background colour chosen randomly from an array.

### Defining functions

Besides the built in language functions like **time()** it is possible to define your own functions. Add this function to the page in the main PHP section before the time is printed out:

**function timestable( $n) {**

**$text = "<p>\n";**

**for( $i = 1; $i < 13; $i++) {**

**$text .= $i." times ".$n." is ".($i \* $n)."<br>\n";**

**}**

**$text .= "</p>\n";**

**return $text;**

**}**

This function constructs a fragment of HTML containing a times-table when it is called. To call it you must give it a number, and you need to print what it returns to see anything. Add a call to this function as the last line of the main section of PHP giving it the number 7:

**print timestable(7);**

Have a look at the HTML source this produces and compare it to the PHP code inside the function. Can you see the HTML tags and the numbers that go between them there?

By now you should be getting the idea of why PHP is useful. For one thing it can print out web pages that change according to the time. For another it can automate displaying a large amount of data. Now that we have a times table function we can easily do this:

**// code to print out all twelve times tables**

**print "<table class='w3-table-all'>\n";**

**print "<tr>\n";**

**print "<td>".timestable(1)."</td>\n";**

**print "<td>".timestable(2)."</td>\n";**

**print "<td>".timestable(3)."</td>\n";**

**print "<td>".timestable(4)."</td>\n";**

**print "<td>".timestable(5)."</td>\n";**

**print "<td>".timestable(6)."</td>\n";**

**print "</tr>\n";**

**print "<tr>\n";**

**print "<td>".timestable(7)."</td>\n";**

**print "<td>".timestable(8)."</td>\n";**

**print "<td>".timestable(9)."</td>\n";**

**print "<td>".timestable(10)."</td>\n";**

**print "<td>".timestable(11)."</td>\n";**

**print "<td>".timestable(12)."</td>\n";**

**print "</tr>\n";**

**print "</table>\n";**

Do this, and write an explanation of what this code does in your report.

That made it easy to print a lot of times tables. Now try this:

**// code to print out LOTS of times tables**

**print "<table class='w3-table-all'>\n";**

**for( $i = 1; $i < 1000; $i += 6) {**

**print "<tr>\n";**

**for( $j = 0; $j < 7; $j++) {**

**print "<td>".timestable($i + $j)."</td>\n";**

**}**

**print "</tr>\n";**

**}**

**print "</table>\n";**

Try and make sense of what this code does and how it works, and write an explanation in your report.

To help understand this, read up on the **for** control structure in the language reference at <http://www.php.net/manual/en/language.control-structures.php>.

|  |
| --- |
| **EXERCISE 1: STARTING PHP** |
| ***Required in your report (basic exercise).*** |
| * The code of your final version of time.php with all modifications in place. * A clickable link to the page in your Daydream account. * An explanation of the code to print out all twelve times tables. * An explanation of the code that prints out *lots* of times tables. * Reflection on the tasks and all activities involved. |
| ***Extended Tasks*** |
| * Examine the documentation for the function mktime() and use it to add a line to the page that prints out how many seconds there are left in the year (that is, before 00:00 1st January next year). * Define a function that randomly prints out one of the following and call it several times in the body of the page to separate the sections with randomly coloured lines. Strings: "<hr style='color:red;'>", "<hr style='color:green;'>", "<hr style='color:blue;'>", "<hr style='color:blue;'>" |
| *Original additional work:*  *Investigate and experiment with any related subject matter that interests you.* |

1. Note that PHP has another command which is almost identical to “print” that is called “echo”. You may see this in examples in the manual or tutorials. It does the same thing as print. [↑](#footnote-ref-1)