Chapter 5: Functions

Objectives

- Create functions that build a web page.
- Create a function that displays a string.
- Create functions that use the return statement.
- Creating a function that receives an array.
- Returning more than one value from a function.
- Create an external function file.

What to know in chapter 5

Chapter 5 covers writing your own functions. It's a topic that is introduced in most sections CIS 115.

The basic rule in programming is to "do it once" and that is what functions allow programmers to do.

A note about the **parameters** section: In the section on using parameters it may be a little confusing because the author's example uses the r**eset()** and **current()** functions. The important part in this description is how to use parameters (this will bring you back to you math days.)

Read the entire chapter even though we will not cover the following topics (you can read these lightly so you have an idea of what more advanced programming has to offer)

Using **auto_prepend_file** and **auto_append_file**. This is a nice idea but it really adds limits to your program that you will probably never use.

Passing by reference vs. passing by value. I may include a simple tutorial so you understand why you usually want to pass by value but generally speaking you can get into trouble passing by reference unless you're comfortable working with functions. In larger, more difficult programs you may want to use it in order to pass a memory reference to an object or a large array (it goes on from there).

Recursion. This is an advanced topic. It is good to know that it means "a function that calls itself" however, even the author notes that there are few web-based applications that use recursion.

Implementing Anonymous Functions. This too is an advanced topic and outside the scope of this class.

Even though it may seem like we are skipping a lot of content there is plenty to do with functions from the first part of the chapter.

Folder Structure

Create a folder named **ch05** and save it inside your **web182** folder.

Create a folder called **functions** and save it inside your **ch05** folder.

Create an **index.php** file and save it inside your **ch05** folder.

Copy your **index.php** file from chapter 4 that contained your bootstrap code and paste it into a new file named **html-template-function.php**

Update your **my-code\index.php** file so that it references the **ch05** folder.

Web page creation using functions

(20 points)

In this section we are going to follow the book's model of taking basic HTML code and putting it into functions that we can reuse to create web pages.

Open your **html-template-function.php** in an editor and modify it or your can use the generic code below. Note that my links do not work yet and I've placed a pound sign **#** for the value in the **href** attribute

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <title>WEB 182</title>

   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css
"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">

</head>

<body>

<div class="container-fluid">

   <h1>Building a page with functions</h1>

   <nav class="navbar navbar-default">

       <div class="container-fluid">

           <ul class="nav navbar-nav">

               <!--These links are left blank for now-->

              <li ><a href = "../../" > WEB 182 </a ></li >

             <li ><a href = "../" > My Code </a ></li >

           </ul>

       </div>
   </nav>

   <div class="container">
       <div class="row">
           <div class="col-sm-12">
               <ul>
                   <li><a href="#"></a>No link yet</li>
                   <li><a href="#"></a>Still no link</li>
               </ul>
           </div> <!--col-sm-12-->
       </div> <!--row-->
   </div> <!--container-->
</div> <!--container-fluid-->
```

Test your code to see if it has a heading and a navigation bar.

Break it up using include

Now let's start breaking up the file using the `include()` function.

Make a new folder called **functions** and store it in your **ch05** folder. This is where we will store our functions for this chapter.

Make a new file named **html-builder.php** and save it in the functions folder.

Cut the following code from your **html-template-function.php** file and create the following function in your **functions/html-builder.php** file. We will paste it in that file next.

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <title>WEB 182</title>

   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css
"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">

</head>
```

Make a function to display the header

Now let’s make it into a function and paste the code into the function

```
/*
* Functions to build a Web Page
*/


/*
*  Name:          makeHeader()
*  Desc:          Function to create a generic HTML header
*  Arguments:     None
*  Returns:       None
*/
function makeHeader() {
   echo '<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>WEB 182</title>
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
</head>';
}
```

As you can see this function has an empty parameter list and its only purpose is to display the head section of our HTML file. The nifty thing is that you can reuse this over and over.

Next we have to go to the main program and call the function so it will work. Otherwise, you can think of functions like books in a library. They just sit there until someone checks them out.

Add the following code at the top of your your **html-template-function.php** file

```
<?php

   include 'functions/html-builder.php';

   makeHeader();

?>
```

The include function tells PHP which file you want to include. `makeHeader()` calls the function and runs it.

Make a function to display the footer

Next we will do basically the same thing to create the footer of the HTML file. Start by opening the **html-builder.php** file and create a new function. Place the following code under the last part of the function you just created.

```
/*

*   Name:            makeFooter()

*   Desc:            Function to create a generic HTML footer

*   Arguments:       None

*   Returns:         None

*/



function makeFooter() {

   // paste footer code here next

}
```

Next go back to your html-template.php file and copy the following code and paste it so it looks like

```
function makeFooter() {

   echo '</body>

</html>';

}
```

Now we need to go back to the **html-template-function.php** file and add the following code at the very bottom of the file to call the function. Notice that we have to wrap the function in PHP tags because we have first started the file with PHP then closed the PHP before the HTML code. It is common to intersperse HTML and PHP throughout your files once your programs become larger.

```
<?php makeFooter(); ?>
```

Save both files then test it to see if it runs as expected.

Create a function for the navbar

This one is very similar to the others with one exception. The links in the navbar will change if you were to use this over an entire site. Since we are just getting started we will keep it simple so you can use it for web pages that are just in one folder otherwise we would need to pass arguments to the parameter list that show a relative path to where we want to go.

In keeping with the theme start by opening the **html-builder.php** file and create a new function with the following code.

```
/*
 *  Name:           makeNavBar()
 *  Desc:           Function to create a generic HTML navBar
 *  Arguments:      None
 *  Returns:        None
 */

function makeNavBar()
{
   echo '<nav class="navbar navbar-default" >
       <div class="container-fluid" >
           <ul class="nav navbar-nav" >
              <!--The following line uses a relative path to find
               the web182 and my - code folders . The two dots mean
               go up one level in the directory tree . So the ../../
               means to go up two levels in the directory tree . -->
               <li ><a href = "../../" > WEB 182 </a ></li >
              <li ><a href = "../" > My Code </a ></li >

           </ul >
```

```
        </div >

    </nav >';

}
```

Now we need to update the **html-template-function.php** file. Add the following code just beneath your

```
<h1>Building a page with functions</h1>

<?php makeNavBar(); ?>
```

At this point your code in the **html-template-function.php** file should look like

```
<?php
    include 'functions/html-builder.php';
    makeHeader();
?>
<body>
<div class="container-fluid">
    <h1>Building a page with functions</h1>
<?php makeNavBar(); ?>
    <div class="container">
        <div class="row">
            <div class="col-sm-12">
                <ul>
                    <li><a href="#"></a>No link yet</li>
                    <li><a href="#"></a>Still no link</li>
                </ul>
            </div> <!--col-sm-12-->
        </div> <!--row-->
    </div> <!--container-->
</div> <!--container-fluid-->
```

```
<?php makeFooter(); ?>
```

A little bit of cleaning up

I just noticed that the very last div tag really doesn't belong here. No problem, we can add it to the `makeFooter()` function. Select and cut the last div tag and add it to the function so the function now looks like

```
function makeFooter() {

   echo '</div> <!--container-fluid-->

</body>

</html>';

}
```

Submitting Your Work

Upload your files to your web host

Your file paths may be a little different than mine so your relative paths in the hypertext links will be different. That is okay. Upload the ch05 folder to your webhost and see how it runs.

Submit your files in Moodle

1. Zip up the chapter 5 folder and name it ch05_yourLastName.zip and submit it in Moodle.
2. Add the URL to your chapter 5 folder on your host the Notes section so I can go to it directly.