# CPSC 2030 – Web Development II

**Lab6: [30 marks] Introduction to PHP**

|  |
| --- |
| **Objectives** |
| * Practice on the basics of PHP |
| **Preparation** |
| 1. Read Chapter 12 of the textbook 2. Create a new folder named **Lab6** under your CPSC2030\Labs folder. |
| **What to do** |

**Install XAMPP to your computer**

* For this lab, you will need to use the XAMPP server environment for creating a local server to run your PHP code. Download the following file and extract it onto your computer:

<https://sourceforge.net/projects/xampp/>

* Unzip the XAMPP folder and run the following application: **xampp-control.** This is a graphical interface for the XAMPP environment. Click the “start” buttons for the Apache and MySQL modules; the module names should turn green. This means your local server has been activated!
* Also in the XAMPP folder, you will find a folder called **htdocs**. This is where you will place the files for your assignment; any PHP files inside this folder can be run on the server. To run a PHP file (stored at htdocs/foldername/filename.php) on the server and view the output in the browser, navigate to the following URL:

**http://localhost/foldername/filename.php**

**Setup:**

* In this lab, we will create a very simple web page using PHP. The page will display some information about yourself and allow the user to change the color of the text.
* Download **Lab6StarterFiles.zip** from D2L and unzip it. You will edit a single PHP file called **lab6.php** inside the folder Lab6.
* See the file **sampleoutput.html** for a demonstration of what the HTML output should be generated by your PHP. The file **screenshot.png** will give you an idea of how the page should look rendered in the browser.

**Task 1: [10 marks] Developer Profile Function**

* Copy the **lab6** folder to your **htdocs** folder. The file **lab6.php** contains a basic HTML shell, but no PHP code.
* At the top of the document (before the doctype tag) create a set of **php tags: <?php ?>**. Inside the tags, declare three variables: **$my\_name, $description**, and **$favorite\_animals**. Set the value of the first two variables to strings representing your name and a short description of yourself. For the third variable, set the value to an array of strings representing your favorite animals (any number is fine).
* Below the three variables you declared, declare a function called **the\_developer\_profile()**. This function should not take any parameters or return anything, but should simply output the values of the previously declared variables to the page, marked up using HTML. See the files **sampleoutput.html** and **screenshot.png** for an idea of what the output of the function should be.
  + The output should include a **div**.
  + The list of favorite animals should be output, separated by commas, into a paragraph tag, and the final animal in the list should have the word “and” before it.
    - There’s more than one way to do this; I used the **foreach()** function.
* Create a set of php tags in the body of the HTML document, and call the aforementioned function.

**Task 2: [10 marks] Color Picker Function**

* After your **the\_developer\_profile()** function, write another function called **the\_color\_form()** that outputs a form to the page, asking the user to select a color. When the form is submitted, the color of the text on the page will change to that color, although for now, we'll just worry about choosing the color without actually applying it. Again, see the files **sampleoutput.html** and **screenshot.png** for an idea of what the output of the function should be.
  + To do this, you will use a **form** and an **input** tag with the type **“color”.**
    - Note that we will process the form using the same page, so you can set the **action** attribute of the form to **“lab6.php”.** We're going to use an HTTP post request to submit the color, so in for the form tag, set **method="POST"**.
  + The output of this function should include a **div**.
* Call this function in the body of the HTML document.

**Task 3: [10 marks] Get Color Function**

* Now that we have a way for the user to choose a color, we want to actually apply that color to the text on the page. To do this, we'll create a new function (declared beneath the **the\_color\_form()** function) called **get\_color().**
  + We're using a post request to send the color to the server. In order to access the color that was sent, we will use the **$\_POST[]** global array.
  + The **get\_color()** function should check the **$\_POST[]** array to see if any color has been set at the appropriate key. The name of the key will be the same as the **name** attribute for the **input.** You check to make sure the user has chosen a color and submitted the form by using the **isset()** function.
  + If the user has submitted a color, **return** that color. If not, just return **black**.
  + In the **<html>** element of the document, add a **style** attribute to set an inline style. To set the color, call the **get\_color()** function. This function will only return a color without outputting it to the HTML page. Find a way to output this color into the value of the style attribute.

|  |
| --- |
| **What to hand in** |
| * Zip your Lab6 folder and upload it to D2L. |
| **When to hand in** |
| * By 11:59pm, Sunday, October 23, 2022 |