|  |
| --- |
| **Activity 2.2.2 Introducing PHP** |

Introduction

|  |  |
| --- | --- |
| Where does data in a web form go when you click “Submit”? How can a website remember who you are so that you don’t have to log back in every time you navigate back to it? How do businesses maintain records of what customers have purchased through their websites?  Looking at a form created in HTML leaves a lot of questions unanswered. One of the most popular answers to many of these questions is PHP and MySQL®, the dynamic duo. |  |

Materials

* Computer with Enthought Canopy distribution of *Python®* and Internet access
* Firefox with FireFTP and FireSSH add-ons
* Credentials for the school's MySQL database artists on pltwcs.org.

Procedure

1. Form pairs as directed by your teacher.
2. Meet or greet your partner and establish team norms.

**Part I: Outline of a PHP page**

1. PHP is the most common **server-side scripting language**. This means that the code is executed by the server rather than the client. This allows a web designer to deliver interactive pages without the client being able to see the code. The acronym originally stood for Personal Home Page, but has since been changed into a self-referencing backronym, PHP: Hypertext Preprocessor. Why might this ability to hide code from the client be preferable sometimes?
2. To understand some of the advantages of using a server-side language like PHP, we will continue to examine a model student art gallery. Get a copy of 2.2.2.A source.zip as directed by your instructor.
3. Open the file login.php , and replace the bolded text with your login information. For the web page to access the database on the server, you will need to insert credentials in the file login.php and upload it to your home directory on the server using FTP.
   1. The code is shown below. You must modify lines 3 – 5 using information provided by your instructor.

The database name is clevenge

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?php  $db\_host = 'localhost';  $db\_database ='**the database name**';  $db\_username = **'your MySQL user name**';  $db\_password = **'your MySQL password**';  ?> |

* 1. Save the changes.
  2. Use FTP to upload the login.php file you modified to your home directory on the server.

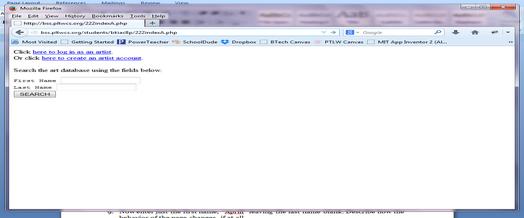
1. Upload the following files to your home directory on the server using FTP.

* 222indexA.php
* 222account\_creationA.php
* 222artist\_portalA.php
* 222logoutA.php
* thumbs.py

1. In a separate web browser tab, visit your page at **http://bluesprings.pltwcs.org/students/bkiac8p/222indexA.php** .

…. Where you see **bkiac8p**, replace with your user name.

You should see a web form with two links and two fields that you can fill in.



1. Enter the first name “Aprill” and the last name “Aronie” and click the SEARCH button. Describe what happens.

1. Now enter just the first name, “Aprill” leaving the last name blank. Describe how the behavior of the page changes, if at all.

1. Enter just the last name, “Aronie”, leaving the first name blank. Describe how the behavior of the page changes if at all.

1. Now enter your own first and last name. How does the behavior of the page differ from Steps 5-7, if at all?

1. Describe how this page behaves differently than the 221indexA.html provided in the previous activity, if at all.

1. PHP makes the index page for our website more responsive. The program **222indexA.php** constructs a table of images similar to **221indexA.html**. However it creates the HMTL based upon the user's query by accessing a MySQL database.
2. Open **222indexA.php** with **Notepad++.** Use Ctrl-F to search for instances of the variable names firstname and lastname as well as other key words where appropriate, and read through the comments in **222indexA.php**. Write down the line number(s) where the code performs each of the following tasks:
   1. PHP code checks to see if both firstname and lastname are non-empty strings (Case 1).
   2. PHP code checks to make sure that at least firstname is non-empty, but allows lastname to be empty (Case 2).
   3. PHP code checks to make sure that at least lastname is non-empty, but allows firstname to be empty.
   4. HTML code creates the form that we see on the page.
   5. A <TABLE> tag opens in HTML.
   6. PHP code makes sure that the table never gets wider than 6 cells
   7. HTML code inserts an image into a cell in the table
3. Our web page responds to data sent by the user to the server by running a PHP program. **Apache**™ passes the user's data to the PHP program, and the PHP code is processing the information that the user typed into the search fields. PHP passes the information to a program in yet another language, MySQL. The SQL program returns search results back to the PHP program, which creates HTML based on the response from MySQL. A MySQL query is submitted using the PHP command mysql\_query($query). In which line numbers is PHP querying MySQL? You may want to use the find function of your text editor as you did in the previous step.

**Part II: PHP Basics**

1. HTML is a client-side language. HTML code is sent to the client machine and is then rendered by the client machine’s web browser. What does PHP use HTML code for?
2. PHP is a server-side scripting language, meaning that it runs on the server instead of on the client’s machine. What can be done with PHP that cannot be done with HTML alone?
3. The syntax of a language includes the rules about what characters can be used in variable names, how key words can put together, and how commands are separated and grouped. Examine 222indexA.php, 222account\_creationA.php, and 222artist\_portalA.php with your partner. Answer 5-10 of these questions, as directed by your teacher.
   1. How, if at all, does the syntax for a multi-line comment in PHP vary from the syntax in *Python*?
   2. What character appears at the end of a statement in PHP?
   3. What are curly braces { and }, used for in PHP?
   4. What symbol do all variables begin with in PHP?
   5. In *Python*, we used + to indicate string concatenation. What is used to concatenate strings in PHP?
   6. How, if at all, does the syntax for a function call in PHP differ from the syntax for a function call in *Python*?
   7. What does the keyword echo do in PHP? Use the Internet to help you find an answer if you do not know.
   8. In *Python*, the assignment operator is =, while the condition equality operator is ==. How are assignment and conditional equality expressed in PHP?
   9. In *Python*, elements of a list can be accessed with *variable*[*index*]. The language PHP uses arrays instead of lists. In an array, all the elements must be of the same data type, but languages index lists and arrays similarly. How, if at all, does the syntax for indexing an array differ in PHP from the way that lists are indexed in *Python*?
   10. How, if at all, does the syntax for defining a function in PHP differ from the syntax in *Python*?

1. Explain the purpose of the function display\_table in 222indexA.php.
2. What do you think is the purpose of the function popen, called on line 131 of 222artist\_portalA.php, in the uploading of images?
3. Examine the official documentation for the PHP function mkdir found at <http://www.php.net/mkdir>. What do the different parameters of mkdir represent?
   1. pathname
   2. mode
   3. recursive
4. Use Firebug to examine the cookies being used by this website when you access the different PHP files. Which pages produce cookies and what values are those cookies storing?

**Part III: Evaluating Efficiency**

1. In this part of the activity you'll look at some JavaScript™ code that interacts with PHP to dynamically create a popout containing information about the artist and the image that a user mouses over. An alternative design decision would be to skip the use of JavaScript and popouts and instead provide the same information using only PHP. This can be done by dynamically changing the code of the index page to display the image that a user clicks on in its full form as well as information about it and its artist instead of the table full of thumbnails. Discuss with your partner the merits of each approach and then describe which design you would like the best and why.
2. Upload the following six files to your home directory using FTP.

|  |  |
| --- | --- |
| * 222popouts.js * 222indexB.php * 222account\_creationB.php | * 222artist\_portalB.php * 222logoutB.php * thumbs.py |

]=rectory using FTP filess.ons, as directed by your teacher.age PHP uses arrays instead of lists. In an array, all the elements

1. Open 222indexB.php with **Notepad++**. You can also ope the file in a web browser and press **Ctrl + U** to display the code. Explain in your own words what each of the following sections of code do.
   1. Line 20:
   2. Lines 21-33:
2. Lines 156 – 157 of 222indexB.php are the lines responsible for calling the JavaScript functions that create popout functionality. The code for those functions can be found in 222popouts.js. Use these lines of code as well as the JavaScript file to help you determine the purpose of each of the arguments in the call to showDetailedView.

a. $div\_id:

b. $id\_name:

c. $firstname:

d. $image\_name:

e. $image\_row[0]:

1. View 222indexB.php in a browser. Do a search for first name “aprill” and mouse over the images that are rendered. What problems do you notice with this implementation of popouts, and how would you want it to behave differently?
2. Use the code that shows the artist's first name in the JavaScript popout as a guide to help you add in your own code to also display their last name in the popout. As you work to produce this code you'll need to follow a different workflow than you have for developing static web pages or other programs. Because all the information in the database is contained on the server you must upload your modified files to the server and then open them in a web browser that accesses the server in order to determine if your program is working correctly.
3. Knowing how long it will take a program to achieve its goals is an important part of the analysis of algorithms. Computer scientists use a descriptive tool called **Big-O notation** to explain the worst case run time of an algorithm. For example: Imagine that you have written a program that changes all of the letters in a word from lowercase to uppercase. The basic computational unit of a program like this would be analyzing a single character and deciding if it must be changed from lower case to upper case. The worst case scenario is if the algorithm had to change all of the characters, and the best case scenario is if the algorithm did not have to change any. To run the program on the input word “Hello” would take five computations, four if we could know in advance that the letter “H” was already capitalized. What would the worst computational case in this problem be for the input word “Stegosaurus”?
4. An important part of the use of Big-O notation is generalization. In order to write the worst case run time of the algorithm discussed in the previous step we would need to first determine the characteristic of the input that has the biggest impact on run time, in this case, the number of letters in the input word. We refer to this character by the variable name “n”. The run time of this algorithm could be described as linear because, as the size of “n” increase, the run time of the solution increases proportionately to “n”. We say that this algorithm is “Big-O of n.” This is written as O(n). When you search for images by a given artist on our website, what is the computational factor that determines how long it will take the PHP script to produce the table of images resulting from our query?
5. What is the PHP variable that represents this factor?
6. Imagine a program that turns all lower case letters in a word to uppercase and then takes the resulting word and converts all of the letters in it back to lower case. Now there are two computations occurring for every letter in the input word. Although the amount of computation being done in this new program doubles over the old one (to 2n), we still say that the program has a linear run time and still expresses its run time as O(n) because its run time is being multiplied by a constant number, in this case 2. How many different statements must execute for every image displayed by our website (the number of statements between line 135 and line 163 of 222indexB.php)?
7. Express the worst-case run time of a rendering of our index page using Big-O notation.

**Conclusion**

1. In Activity 2.2.1, you used a version of this site that was constructed entirely from HTML, JavaScript, and CSS. What do you think are the most important improvements made by using PHP and MySQL?
2. What language would you use if you wanted to access information on a server?
3. What language would you use to create a zoom feature for images on your website without putting any additional strain on the server?
4. What language would you use to put a pretty frame around a table on your website?
5. What purpose did the comments in 222indexB.php serve for you as you worked through this activity?