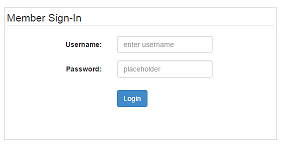
**Introduction to PHP: Playing Catch with HTML**

**Introduction**

In the early days of the Web, the advent of HTML forms marked a significant step forward in a progression that eventually led to data-driven websites. HTML forms allow for the collection of data on the client side. In this activity, you will learn how to move the information you collect from the client side to the server side. Once on the server side, a host of things can be done with that data. For this reason, knowing how to pass data to a server after collecting it in a web browser is an essential skill in the development of data-driven websites.

In this activity, we will build an HTML form and send the data collected to a server running PHP. A server-side script will collect the data and report what was sent back to the client using a PHP script.

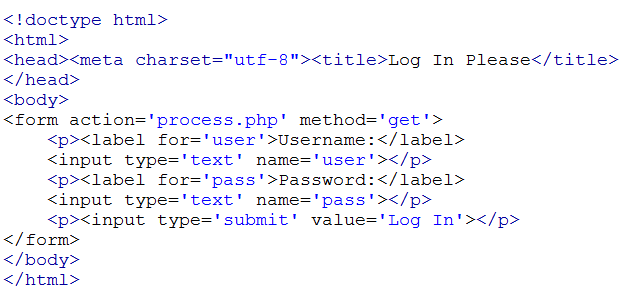
Materials

* Computer
* Text Editor, such as Notepad++
* Server environment, local or remote

Procedure

**Part I: Create an HTML form**

1. We will start with a boilerplate a basic HTML page: one having Doctype, html, head and body tags. Between the <body> tags, create the following form:



Save the file as login.html it in the htdocs folder of your server.

1. Load the form in your browser (If using a local server, the address will be http://localhost:81/login.html. Note you may have a different port—here I’ve used 81).

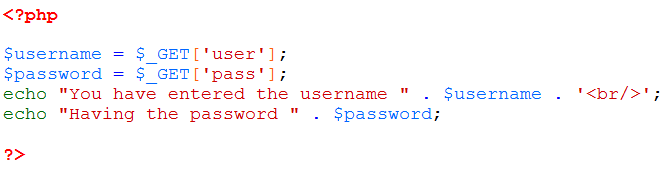
Fill out the requested information with appropriate, albeit fictitious, inputs. Click the SUBMIT button. You will receive an error (we will find out why in a moment), but note what you see in the URL. Record that information below:

URL after submitting the form: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part II: Create the server-side PHP file**

You received an error upon submission because, as indicated by the form’s action attribute, you attempted to send the form information to a file on the server called *process.php*. Unfortunately, no such file by that name exists on the server. We will fix that problem now.

1. Create a new text file using the code below, and save it in the htdocs folder of your server (local or remote) using the name *process.php*.



Now that *process.php*  exists, it can catch the data thrown to it by the HTML form. Run the login.html form, fill out the requested information, and click the submit button. If everything was typed correctly, the browser should report back the information received by the server.

Congratulations! You have successfully played catch with the client and server using HTML forms (Throw) and PHP (Catch).