***<https://github.com/JessicaAlexander/week9>***

1. The difference between include() and require() is that if the include() function fails it generates a warning and continues to run while the require() function terminates the program.
2. PHP files are just like HTML files, but they can include both HTML and PHP code. The PHP code is parsed (or executed) by the Web server when the page is accessed and the resulting output is written as HTML within the Web page. When a user accesses a PHP page, his Web browser only gets sent the HTML code, since the Web server has processed the PHP code in the background. Most PHP pages are processed so quickly that it does not noticeably slow down the loading of the Web page. The .php extension is important, since it tells the Web server that the page may include PHP code. Therefore, it must be run through the server's PHP engine before being sent to a client's Web browser. This allows dynamic content to be generated each time the Web page is loaded, based on the variables included in the PHP code. For example, PHP pages may load objects such as the current date and time, data from form fields submitted by a user, or information from a database. Still, once the page reaches the user's Web browser, everything is formatted as HTML
3. Relative Reference indicates where the included file is in relation to the parent file. While theAbsolute Reference is a complete, specific address.
4. You define constants using define() function. Constant names are case-sensitive. To check if a constant has been defined use the defined() function.
5. A *Timestamp* is a number representing how many seconds have passed since midnight on January 1, 1970 – this moment is known as the *Epoch*.
6. $\_SERVER['REQUEST\_METHOD'] returns the request method used to access the page. Request methods are 'GET', 'HEAD', 'POST', 'PUT'.
7. You should make your form sticky so that it “remembers“ previously submitted values.
8. To see the php you need to enable the display error turned on, in php.ini.

display\_errors = on

To initiate the errors

ini\_set('display\_errors', 1); \*\*\* Error reporting.

ini\_set('display\_startup\_errors', 1);

error\_reporting()

1. The *“headers already sent”* error means that you have used header(), setcookie(), or session\_start after the browser has already received some text, HTML, or even a blank space. The *“headers already sent”* error can be prevented by making use of the output buffering.