

# CEMC 2014: Intro to PHP, Next Steps

---

© 2014 Sam Scott, Sheridan College.

## Readings

1. See w3Schools (sections on sessions, file open/read, file create/write and PHP superglobals)  
<http://www.w3schools.com/php>
2. For more details, see the complete PHP Manual.  
<http://www.php.net/manual/en/>

## Session Management

Most interactions with web apps involve the same user sending multiple requests (adding to a shopping cart, filling in information on a multi-page form, etc.) Sessions help you keep track of who you are talking to across all these multiple requests from multiple users. You can store information about a client in one PHP script and then retrieve it from another.

```
session_start();           // starts or resumes a session
                           // must happen at the VERY TOP of the PHP file

$_SESSION[]                // an associative array (like $_GET) to store and
                           // retrieve data unique to the current session

session_destroy();         // clears the data stored in $_SESSION[]
```

See the session management examples in the example pack for an example.

## File Management

Most web apps store and retrieve data from the server to present custom views to the user. This is best done with a proper database, but for the high school curriculum, you could also make use of text files.

Storing and retrieving data from text files is not as secure as using a database, especially for high traffic applications, but it should work just fine for small-scale apps that most high school students will build.

```
$myfile = fopen("myfile.txt", "r"); // opens a file for reading
fgets($myfile);                     // returns the next line
fgetc($myfile);                     // returns the next char
feof($myfile);                      // returns true if end of file
fclose($myfile);                    // closes the file

$myfile = fopen("newfile.txt", "w"); // opens a file for writing
fwrite($myfile, "Hello, World!");    // writes a line into the file
fclose($myfile);                    // closes the file
```

## POST Parameters

POST parameters are just like GET parameters but they are sent in the body of the HTTP request, so they can't be seen or modified in the address bar of the browser.

To change a form to send POST parameters, just change the `method` attribute:

```
<form action="myphpfile.php" method="POST"> ...
```

PHP programs can access them in the `$_POST` array, which works in exactly the same way as `$_GET`.

GET parameters are length limited (to a total of about 1K) but POST parameters are not limited.

POST parameters should be used to hide sensitive info (passwords, credit card numbers, etc.).

GET Parameters can be bookmarked and shared with the URL. Post parameters cannot. So POST parameters are safer to use for requests that make changes on the server side (e.g. voting in a poll, posting a blog entry, etc.).

**WARNING:** POST parameters are not secure from hackers. For proper security, your server has to be configured to use HTTPS

## Exercise Ideas

1. **Sessions:** Write a PHP app that allows the user to build a set of information across multiple forms (e.g. registering for a food delivery service, building a character for an RPG, etc.). Store all the info in `$_SESSION`. When the final form is submitted, display all the information entered, nicely formatted.
2. **Sessions:** Write a one-page PHP app that lets the user build a shopping list and then print it. The app should have a form at the top to add a new item and should display the list of items entered so far below the form.
3. **Sessions:** Write a PHP app that will play a simple game against the user (NIM, Pig, Tic Tac toe etc.). Store the state of the game in `$_SESSION`.
4. **File Management:** Same as exercise 1, but store the information in a file on the server (file name chosen by the user). Write an extra page that lets the user pick a file name and then opens and displays the data stored in that file.
5. **File Management:** Write a page that displays a poll in an HTML form (e.g. "Who will you vote for in the upcoming election?"). When the user votes (i.e. submits the form), store their vote in a text file. Also make a results page that reads the text file of votes and shows the number of votes for each of the poll options as percentages.

