

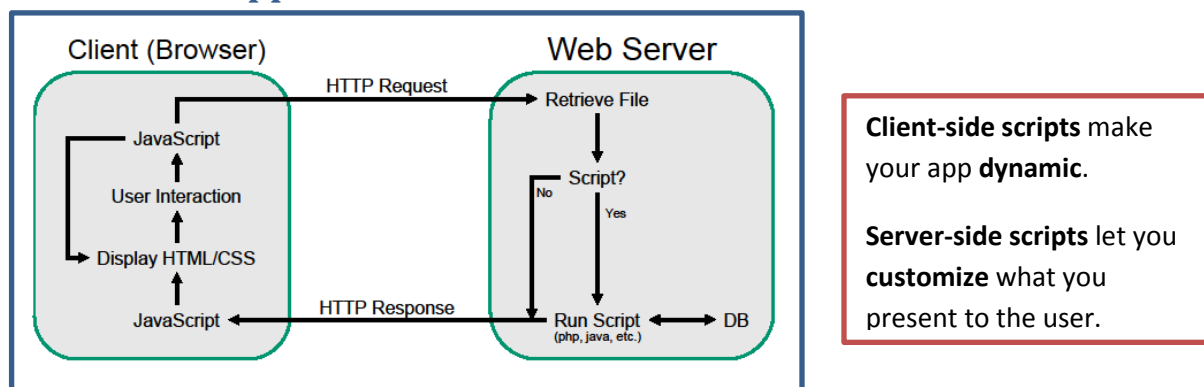
CEMC 2014: Intro to PHP Lesson 1

© 2014 Sam Scott, Sheridan College.

Readings

1. Sam Scott, *PHP for Sheridan Students*, 2014: Sections 1 and 2.
2. (Optional) Sam Scott, *Setting up NetBeans*, 2014
3. For more details on PHP see w3Schools.
<http://www.w3schools.com/php>
4. For even more details, see the complete PHP Manual.
<http://www.php.net/manual/en/>

A Basic Web App Architecture



Setup

To do server-side development, you must have a web server. You can either use `mobile.sheridanc.on.ca` (preferred option), or you can set up XAMPP or WampServer on your local machine to work offline.

If you are using `mobile.sheridanc.on.ca`, then to get full error reporting for your PHP programs you need a `.htaccess` file (note the dot) inside `public_html`. The contents of this file should be:

```
php_flag display_errors 1
php_value error_reporting 32767
php_flag xdebug.default_enable on
```

It would also be a good idea to set up `.htaccess` to provide password protection on your work as well. See the main handout for more details.

PHP Basics

PHP programs run on the server side of the client-server interface. PHP programs respond to HTTP requests by creating HTTP responses. The output of a PHP program is usually a text stream consisting of HTML, CSS and/or JavaScript code (though PHP can be used to create other types of responses as well).

PHP Tag (2.1,2.2): `<?php ... ?>`

Output (2.1,2.2): `echo "<p>Hi</p>";` **Or PHP Expression Tag (2.4.2):** `<?= ... ?>`

Variables (2.4.1): Name with a \$ character: `$varName`. No variable declaration is necessary.

Data Types (2.4.1): Weak typing with aggressive implicit typecasting.

Control Structures (2.4.4): The if, else, while, do, for, and switch statements all work the same as in Java / C / C++ / JavaScript.

Chaining else and if (2.4.4): Use the special `elseif` operator (one word).

Functions (2.4.1) : `function foo($a, $b) { ... return $c; }` (return is optional)

Strings (2.4.3): Use `.` operator for concatenation. Strings can be single or double quoted.

Use double quotes to embed variable names: `echo "hi $name";`

Use curly brackets for array access: `echo "hi {$myArray[$i]}";`

Arrays (2.4.4): `$a=array(); a[0]="first item"; a["make"]="Honda";`

Regular Array (2.4.4): `$a=array("first item", "second item", "third item");`

Associative Array (2.4.4): `$a=array("make" => "honda", "model"=>"civic");`

Variable Scope (2.4.1): Global or function scope only. All variables used in a function are assumed to be local unless `global` keyword is used (`global $x;`)

String and Array Functions (2.4.4): Strings and arrays are not objects. Use PHP's built-in global functions instead of For example, use `strlen($s)` and `count($a)` instead of `$s.length` and `$a.length`.

String Function Reference: http://www.w3schools.com/php/php_ref_string.asp

Array Function Reference: http://www.w3schools.com/php/php_ref_array.asp

Exercises

Try some of the exercises in section 2.6 of *PHP for Sheridan Students*.