

PHP I

CSCI 311

Interactive 3D Ascii scenes

<https://yeahpython.github.io/game/game.html>

Objectives

Learn the basics of PHP syntax

See some examples of PHP in action

Working With PHP

PHP runs on the server, not the client

Server must be capable of running the php

- csci department has an apache server
 - edit your files on otter
 - must run them from `wwwstu.csci.viu.ca/~username/...`
 - must be on campus (and likely in the lab), or
 - ssh tunneled in (instructions on course page)

Working With PHP

To run PHP from home

- tunnel in (requires very little setup and experience), or
- install and run apache yourself (more advanced, and not within the scope of this course)
 - You are welcome to do this, but...
 - different versions of php/apache etc can cause differences
 - all code for this class MUST work, and be located on our apache server
 - labs
 - project

PHP Scripting Overview

PHP scripts enclosed by `<?php ... ?>`

Can be placed in files by themselves, or in HTML files

PHP interpreter:

- filters requested php file
- replaces each bracket with output generated by code
- lets static content pass through untouched
- inclusion/exclusion of static parts can be controlled with logic

PHP first example

Create a simple php file that outputs hello world

- basic html template
- add php code
 - start php
 - echo statement
 - end php
- save the file
- **set the permissions*******
- run the php

More complex example

Use library function to get date and output it

PHP example

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
  <head> <meta charset="utf-8"/>
  <title>PHP Hello World</title></head>
  <body>
  <h1>Hello from PHP</h1>
  <p>Hello it is:
  <?php
    if(function_exists("date_default_timezone_set")){
      date_default_timezone_set('US/Eastern');
    }
    echo "<br/>";
    echo date("1 M. d, Y");
    echo "<br/>";
    echo date("2 e");?>,
    <br/>do you know where your project is?</p>
  </body></html>
```

PHP Syntax

statements must (should) end with ‘;’

PHP echo operator when combined with date function

- displays current local time computed by built-in function date

After PHP interpretation line becomes:

- <p>Hello, it is Thursday February 13, 2020,
- (except actual date)

PHP must always produce correctly formatted HTTP responses

PHP Variables

Variable names start with \$

- `$length = 44;`

Data from incoming HTTP request put in predefined super global variables:

- `$_POST` for POST form data
- `$_GET` for GET form data or query string
- `$_REQUEST` for either get or post request data
- `$_SERVER` for info related to web server, http request headers, and PHP script itself
- `$ENV` CGI defined variables

All above are associative arrays

- e.g., if formdata item=hammer & price=4.50
 - `$product = $_POST['item'];`
 - `$cost = $_POST['price'];`

PHP Variables

You can check if a variable is set:

- `isset($a);`
- This will tell you if:
 - A variable has not been declared yet
 - A variable has been declared by not initialized
 - Has been set to null

You can check if a variable is empty:

- `isempty($a);`
 - its value is “empty” (0, “”)
 - Unset variables are also empty

<https://secure.php.net/manual/en/language.variables.basics.php>

Example including files

Template example

- front.php with header information
- back.php with footer information
- template.php

New stuff:

- declaring variables
- setting variables
- accessing variables from other files

template.php

```
<?php
    $page_title="A First Template";
    $page_background="#def";
    require("front.php");
    $aVar  ="dingdong";
?>
<p>Page content here </p>
<?php $company="Super Cool Company";
    require("back.php");
?>
```

front.php

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
lang="en" xml:lang="en">
<head>
<meta charset="utf-8"/>
<title><?php echo $page_title; ?></title>
</head>
<body style="background-color: <?php echo
$page_background; ?>; margin='50px'">
```

back.php

```
<footer><p style="font-size: small">Copyright  
&copy; <?php echo $company; ?> All rights  
reserved</p>  
</footer></body></html>
```


Navbar example

Detailed discussion of products.php example

- front and back files
- css included
- navbar file contains just the nav bar
- products file includes front/back, front includes navbar

PHP Conditionals

PHP supports:

- if
- if else
- if else if

Strings in PHP

In double quoted strings the following are recognized (and interpreted)

- variables
- escape sequences (`\n`, `\t`, `\r`, `\$`, `\"`)

Strings can be concatenated with dot operator:

- `$me = 'It\'s my name ' . $first . " $last\n";`

String functions

[strlen\(str\): returns number of chars in str](#)

trim(str): strips white space from both ends and returns it

substr(str, i, [, len]): returns a substring of str from position i to end, or to length len

strstr(line, word): finds first instance of a word in line, and returns a substring of line that begins with word

strtolower(s), strtoupper(s) returns the lowercase or uppercase version of s

strcmp(str1, str2): returns positive, negative or zero if str1 is greater than, less than or equal to str2

md5(str) returns the MD5 hash of str

urlencode(str): URLencode str and returns a proper query string

Arrays in PHP

PHP arrays support numerical indexing (0 indexed)

They also support string indexing (associative) at the same time!

All PHP array elements are key-value pairs

example:

```
<?php
    $a = array(2, 3, 5, 7);
    // $a is the same as: array(0=>2, 1=>3, 2=>5, 3=>7)
    $b = array("first_name" => "Sarah", "last_name" => "Carruthers");
    $c = array(5 => "red", "fox");
?>
```

Arrays in PHP

Actually Ordered Maps

Can treat them like indexed arrays

Declaring an array:

- `$arr = array("name" => "Bob", "Age" => 12);`
- `$arr = ["name" => "Bob", "Age" => 12];`
- `$arr = array("red", "green", "hamster", 12);`
- `$arr = ["colour"=>"red", 100=>33, "age"=>42, "cats"];`//yes, this is ok!

More info:

<https://secure.php.net/manual/en/language.types.array.php>

Arrays in PHP

Assigning to arrays:

- `$a[3] = 5*$a[2];`
- `$a[5] = 100;` //ok that element 5 didn't exist yet!
- `$b['email'] = "this@that.com";`

Array functions:

- `sort($ar, [flag])` sort in increasing order
- `rsort($ar, [flag])` sort in decreasing order
 - flag can be: `SORT_NUMERIC`, `SORT_STRING`
- `count($ar)`
- `empty($ar)`
- ...
- <http://php.net/manual/en/ref.array.php>

Basic form processing

Example: FormAction

collects email and name from incoming formdata and responds with welcome message

generates a new html page on success

welcome.php

```
<?php
    $title="A Warm Welcome";
    if ( empty($_POST['client_name']) ||
        empty($_POST['client_email']) )
    { $error=TRUE;
      $title="Please Go Back";
    }
?>
```

welcome.php

```
<head>  
    <meta charset="utf-8"/>  
    <title><?php echo $title; ?></title>  
</head>
```

welcome.php

```
<body style="margin:50px">
  <h1><?php echo $title; ?></h1>
  <?php if ( isset($error) ) {?>
  <p>Sorry, the form is incomplete.</p>
  <p>Please go back and fill out all the required entries. Thank you.</p>
  <?php } else { ?>
  <p>Hello <span style="color: blue">
  <?php echo $_POST['client_name']; ?></span>, it
      is our great pleasure to welcome you to our site.</p>
  <p>We have your email address, <code style="color: blue">
  <?php echo $_POST['client_email']; ?></code>,
  and we will contact you shortly.</p>
  <?php } ?>
</body>
```

PHP Errors

We can tell PHP what errors we care about:

- `error_reporting(E_WARNING | E_ERROR);`
- `error_reporting(E_ALL);`

When to show errors?

- in development
- not in deployment
 - errors confuse the users

On our server, to view logs/errors/access

- `ssh wwwstu`

PHP variable types

Boolean (0 or 1)

Integers

Floating point

Strings

Arrays

Objects

Resources

NULL

Callbacks

- used to reference functions

Built in PHP Variables

Containing formdata:

- `$_GET`
- `$_POST`
- `$_FILES`

Containing persistent data during user's visit:

- `$_COOKIE`
- `$_SESSION`

Other stuff:

- `$_ENV`
- `$_REQUEST`

Constants in PHP

We can declare constants as follows:

- `define("CONSTANTNAME", "a value");`

Use it:

- `echo CONSTANTNAME;`

Note:

- no \$
- cannot be changed
- use all caps (convention)

Built-in constants:

- `PHP_OS`
- `PHP_VERSION`

Cookies vs. Sessions

Gives us a way to store data without having to pass it along between pages

Cookies:

- stored on user's computer
- call `setcookie()` method to create a cookie
 - give it a name, and value
 - can set expiration time, plus more
 - `setcookie(name, value, expire, path, domain, secure);`
 - `<?php setcookie("user", "4321", time()+60*60*24, , admin.oursite.com);?>`
 - use `$_COOKIE["user"]` to get "4321"

Cookies and Sessions

Session:

- stored on the server
- call `session_start()` to start a new session (at top of page)
- to set a session variable:
 - `<?php session_start(); $_SESSION["user"]="4321"; ?>` `<html...`
 - session active for 15 minutes, on this page
 - can access on other pages by adding `session_start` call before html
- to remove items:
 - `<?php session_start(); unset($_SESSION['user']);?>`
- to end a session:
 - `<?php session_destroy();?>`

More examples

CheckBoxes.html:

- simple form with checkboxes
- css to format it
- show post data using php

Examples

pulldown: OptGroup.html

- build html using front and back php files
- select element with optgroups
- process data with showdata.php
- use the `$_REQUEST` variable

Login demo

Demonstration of how to add login functionality (simple!!) to any of your files

Will need to replace XXX.html files with XXX.php file

- Add php code to check if logged in
- Redirect to an error page if login is unsuccessful or if they're not logged in

Login.php

- Handles the login process

Logout.php

- Ends the session (logs user out)

Basiclogin.php

- A password protected page

Where to learn more?

[W3Schools PHP Tutorials](#)

[PHP Manual](#)

Homework

<https://eev.ee/blog/2012/04/09/php-a-fractal-of-bad-design/>

<http://blog.ircmaxell.com/2012/04/php-sucks-but-i-like-it.html>