INFO2180 - LECTURE 5

PHP

PHP IS A SERVER-SIDE SCRIPTING LANGUAGE, AND A POWERFUL TOOL FOR MAKING DYNAMIC AND INTERACTIVE WEB PAGES.

www.w3schools.com/php/

OVERVIEW

- PHP recursive acronym for PHP: Hypertext Preprocessor.
- The Current Version of PHP is PHP7.
- Free and Open Source and very popular. It is supported by most web hosting providers.
- Just like JavaScript, it too has first-class functions.
- Used primarily for web development, however it can be used for command line applications and desktop applications.
- PHP is executed on the server and returns plain HTML back to your web browser.

OVERVIEW

- The default file extension for PHP files is "**.php**"
- You will need a web server (e.g. Apache or Nginx) and of course you need to install PHP.
- For Windows you can install WAMP and for Mac you can install MAMP. These come with Apache, MySQL (database) and PHP.
- PHP does have a built-in web server but this should never be used in production.

What happens when a browser makes a request?

Your browser takes the URL and looks up the servers IP address using DNS. Your browser then connects to that IP address and ***requests*** the given page. Your web server then finds that page and when found sends back a ***response*** with the contents to your browser where it is then displayed.

If the browser requests a **`.html`** file, that will usually return static content, so the server just sends that file. However, if a browser requests a **`.php`** file, that will execute the code in that file on the server before sending a response back to the browser.

What is the difference between ***client-side*** *and* ***server-side*** *code?*

Client-side code ***runs on your computer (in the web browser)***, while Server-side code ***runs on the web server*** to generate a dynamic web page before it is sent back to the browser to be rendered.

EXAMPLES OF CLIENT SIDE/FRONTEND LANGUAGES/FRAMEWORKS

- HTML
- CSS
- JavaScript/jQuery

EXAMPLES OF SERVER-SIDE/BACKEND LANGUAGES/FRAMEWORKS

- PHP
- Ruby on Rails (Based on Ruby)
- Django, Flask (Based on Python)
- Node.js (Based on JavaScript but runs on the server-side instead of client-side)
- And there are many others.

PHP TAGS

PHP TAGS

```php
<?php
// PHP code goes here
?>
```

If a PHP file only contains PHP code then you don't need the closing tag.

EXAMPLE OF PHP TAGS MIXED IN HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>My PHP Page</title>
  </head>
  <body>
    <h1>Heading</h1>
    <?php
      echo "Hello world!";
    ?>
  </body>
</html>
```

EXAMPLE OF SHORT ECHO TAG

```
<!DOCTYPE html>
<html>
  <head>
    <title>My PHP Page</title>
  </head>
  <body>
    <h1>Heading</h1>
    <?= "Hello world!"; ?>
  </body>
</html>
```

DATA TYPES

PHP DATA TYPES

- String
- Integer (e.g. 12)
- Float (floating point numbers - also called double) (e.g. 3.59)
- Boolean (e.g. true or false)
- Array
- Object
- NULL
- and there are a few others. http://php.net/manual/en/language.types.php

OPERATORS

OPERATORS

- Arithmetic Operators (e.g. `+, -, *, /, %`)
- Comparison Operators (e.g. `==, !=, ===, !==, <, >, <=, >=`)
- Logical Operators (e.g. `and, or, &&, ||, !`)
- And there are others. See http://php.net/manual/en/language.operators.php

COMMENTS

COMMENTS

```
# single-line comment

// single-line comment

/*
multi-line comment
*/
```

VARIABLES

VARIABLE NAMES

- A variable starts with the `$` sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (`A-z`, `0-9`, and `_`)
- Variable names are case-sensitive (`$age` and `$AGE` are two different variables)

EXAMPLE OF VARIABLES

```
$variableName = "some value”;

$amount = 10;
```

STRINGS

EXAMPLE OF STRING CONCATENATION

```
$myString = "Hello " . "World!";
```

```
$aLongString = "Hello World! This";
$aLongString .= "string has lots";
$aLongString .= "more text";
```

EXAMPLE OF STRING INTERPOLATION

```
$myString = "Hello\nWorld!";
```

```
$name = 'John Doe';

$message = "Hello my name is $name";

$message = "Hello my name is {$name}. How are you?";
```

String Interpolation only occurs within a string with double quotes ""

SOME COMMON STRING FUNCTIONS

- **`str_replace`**
- **`strip_tags`**
- **`strlen`**
- **`substr`**
- **`strtoupper, strtolower`**
- And there are more. http://php.net/manual/en/ref.strings.php

ECHO/PRINT

ECHO/PRINT

- There are two ways to output data to the screen: `echo` and `print`
- They are the same for the most part. The only difference between them is that `echo` can accept more than one argument and doesn't have a return value.

EXAMPLES OF ECHO AND PRINT

```
$x = 10;

$y = 5;

echo "Hello World"

echo $x + $y

echo "Hello", "World"

print "Another String"
```

CONTROL STRUCTURES

IF/ELSE STATEMENTS

```php
$t = date("H");
if ($t < "10") {
   echo "Good Morning!";
} elseif ($t == "20") {
   echo "Have a good day!";
} else {
   echo "Have a good night!";
}
```

WHILE LOOP

```php
$x = 1;

while ($x <= 5) {
   echo "The number is: $x <br>";
   $x++;
}
```

FOR LOOP

```php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is $x <br>";
}
```

FOREACH LOOP

```
$courses = ["INFO2180", "COMP2112",
"INFO3180"];


foreach($courses as $course) {
  echo "This course is $course <br>";
}
```

SWITCH STATEMENTS

```
$favcolor = "red";
switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
```

FUNCTIONS

FUNCTIONS

- Function names start with a letter or underscore, followed by any number of letters, numbers, or underscores.
- Function names are case-insensitive, though it is usually good form to call functions as they appear in their declaration.

EXAMPLE PHP FUNCTION

```php
function helloWorld($x, $y) {
  $z = $x + $y;
  return $z;
};


echo helloWorld(4, 2);
```

ARRAYS

EXAMPLE OF AN ARRAY

```
$courses = array(
    "INFO2180",
    "COMP1126",
    "COMP1161"
);

$courses = [
    "INFO2180",
    "COMP1126",
    "COMP1161"
];
```

Shorthand syntax added in PHP 5.4

EXAMPLE OF GETTING/SETTING A VALUE FROM AN ARRAY

```
$courses[0]
// INFO2180
```

```
$courses[] = "COMP1161";
```

EXAMPLE OF AN ASSOCIATIVE ARRAY

```php
$courses = array(
    "INFO2180" => "Web Dev 1",
    "COMP1126" => "Intro to Computing",
    "COMP1161" => "Object Oriented Programming"
);


$courses = [
    "INFO2180" => "Web Dev 1",
    "COMP1126" => "Intro to Computing",
    "COMP1161" => "Object Oriented Programming"
];
```

EXAMPLE OF GETTING/SETTING A VALUE FROM AN ARRAY

```
$courses["INFO2180"]
// Web Dev 1
```

```
$courses["COMP1121"] = "My cool course";
```

LENGTH OF AN ARRAY

```
$cars = array("Volvo", "BMW", "Toyota");

echo count($cars);

// 3
```

SOME COMMON ARRAY FUNCTIONS

- `in_array`
- `array_pop`
- `array_push`
- `sort`
- `array_reverse`
- And there are many more. http://php.net/manual/en/ref.array.php

SUPER GLOBALS

SUPERGLOBALS ARE BUILT-IN VARIABLES THAT ARE ALWAYS AVAILABLE IN ALL SCOPES.

http://php.net/manual/en/language.variables.superglobals.php

PHP SUPER GLOBALS

- **`$GLOBALS`** References all variables available in global scope
- **`$_SERVER`** an array containing information such as headers, paths, and script locations.
- **`$_REQUEST`** an associative array containing **`$_POST`**, **`$_GET`** and **`$_COOKIE`**
- **`$_POST`** an associative array of variables passed to the current script via the HTTP POST method.
- **`$_GET`** an associative array of variables passed to the current script via the URL parameters.

PHP SUPER GLOBALS

- **`$_FILES`** an associative array of items uploaded to the current script via the HTTP POST method.
- **`$_ENV`** an associative array of variables passed to the current script via the environment method.
- **`$_COOKIE`** an associative array of variables passed to the current script via HTTP Cookies.
- **`$_SESSION`** an associative array containing session variables available to the current script.

PHP INCLUDES

PHP INCLUDES

- This takes code that exists in a file and copies it into the file that uses the `include`/`require` statement.
- It's useful when you want to include the same PHP, HTML or text on multiple pages.
- `require` will produce a fatal error and stop the script if it cannot find the file.
- while `include` will only produce a warning and the script will continue.

PHP INCLUDES

- You can also use `include_once`/`require_once` statements.
- These are similar to `include` and `require` with the only difference being that the file will be included once.

PHP INCLUDES

```
include 'filename.php';
```

or

```
require 'filename.php';
```

PHP INCLUDES

```
include_once 'filename.php';
```

or

```
require_once 'filename.php';
```

EXAMPLE OF PHP INCLUDES

```php
<?php
echo "<p>Copyright &copy; 1999-" . date("Y") .
" The University of the West Indies.</p>";
?>
```

Let's call this footer.php

EXAMPLE OF PHP INCLUDES

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>

</body>
</html>
```

OOP

PHP HAS A VERY COMPLETE SET OF OBJECT-ORIENTED PROGRAMMING FEATURES INCLUDING SUPPORT FOR CLASSES, ABSTRACT CLASSES, INTERFACES, INHERITANCE, CONSTRUCTORS, CLONING, EXCEPTIONS, AND MORE.

http://www.phptherightway.com/#programming_paradigms

OBJECT ORIENTED PHP

```php
class SimpleClass {
    // property declaration
    public $var = 'a default value';

    // method declaration
    public function displayVar() {
        echo $this->var;
    }
}

$a = new SimpleClass();
```

VISIBILITY OF PROPERTIES AND METHODS

- There are three (3) types
  - **`public`** - can be accessed everywhere.
  - **`private`** - only be accessed by the class that defines the member.
  - **`protected`** - can be accessed only within the class itself and by inherited classes.

EXAMPLES OF VISIBILITY OF PROPERTIES AND METHODS

```php
public $public = 'Public';
protected $protected = 'Protected';
private $private = 'Private';
```

PHP FORM PROCESSING

A SIMPLE FORM

```
<form action="action.php" method="post">
  <p>Your name: <input type="text" name="name" /></p>
  <p>Your age: <input type="text" name="age" /></p>
  <p><input type="submit" value="Submit" /></p>
</form>
```

A SIMPLE FORM

```
Hi <?php echo htmlspecialchars($_POST['name']); ?>.

You are <?php echo (int)$_POST['age']; ?> years old.
```

PHP FILE HANDLING

PHP FILE HANDLING

- With PHP we are also able to Open/Read and Create/Write files.
- We can Open/Read using the `fopen()` and `fread()` functions.
- We can Create/Write files using the `fopen()` and `fwrite()` functions.
- It's good practice to always close a file using the `fclose()` function after you have finished with them.

EXAMPLE OF OPENING AND READING A FILE

```php
<?php
   $myfile = fopen("somefile.txt", "r") or
die("Unable to open file!");

   echo fread($myfile, filesize("somefile.txt"));
fclose($myfile);
?>
```

EXAMPLE OF OPENING AND WRITING TO A FILE

```
<?php
  $myfile = fopen("newfile.txt", "w") or
die("Unable to open file!");

  $sometext = "This is some text";
  fwrite($myfile, $sometext);
  fclose($myfile);
?>
```

PHP SESSIONS

A SESSION IS A WAY TO STORE INFORMATION (IN VARIABLES) TO BE USED ACROSS MULTIPLE PAGES.

https://www.w3schools.com/php/php_sessions.asp

Usually when you work with an application, you are able to open it, make some changes and then you would close it. This application knows who you are.

By default when using a website the web server typically does not know who you are or what you do because HTTP does not maintain state.

Sessions help to solve this by allowing you to store data (e.g. username, favourite colour, etc.) between requests to multiple pages within your website.

Despite many different users visiting your site. PHP can generate unique Session ID's for each of those users and store session information on the server.

A good use for Sessions is for a login system.

PHP SESSIONS

- You start a session by using the `session_start()` function.
- Session variables are then set using the `$_SESSION` super global. e.g. `$_SESSION['fav_colour'] = 'blue';`.
- You then reference that session variable on another page.
- If you need to remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()`

EXAMPLE OF SESSIONS

```php
<?php
   session_start();
   $_SESSION['username'] = 'jdoe';
   $_SESSION['fav_colour'] = 'blue';
?>
```

EXAMPLE OF SESSIONS

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["fav_colour"] . ".<br>";
echo "My username is " . $_SESSION["username"] . ".";
?>

</body>
</html>
```

DEBUGGING PHP

DEBUGGING YOUR PHP CODE

- When debugging your PHP code it is often good to turn on `error_reporting` and to set the `display_errors` option **`On`** at the top of your PHP file.

```php
<?php
ini_set('display_errors', 'On');
error_reporting(E_ALL | E_STRICT);
```

You can also change this in your php.ini configuration file.

DEBUGGING YOUR PHP CODE

- You can also use the `var_dump` function to display information about a variable.

```php
$someVariable = ["Foo", "bar", 12];
echo '<pre>';
var_dump($someVariable);
echo '</pre>';
```

DEBUGGING YOUR PHP CODE

- You can also use Xdebug which is a PHP extension which helps to display stack traces on error conditions and also allows for remote debugging among other additional features.
- https://xdebug.org/

RESOURCES

- Official Website - http://php.net
- PHP Docs - http://php.net/docs
- W3Schools PHP - http://www.w3schools.com/php/
- PHP The Right Way - http://www.phptherightway.com/
- WAMP - http://www.wampserver.com/en/
- MAMP - https://www.mamp.info/en/
- XAMPP - https://www.apachefriends.org/index.html