

PHP Basics

...

Fall 2016

Topic Area	Dates	Goals	Assignments
Introduction and Infrastructure	8/24 – 8/26	1-5	All
HTML & Structure	8/29 – 9/9	1,2	Brief Assignments
CSS & Presentation	9/12 – 9/30	2	
Client-Side Scripting: JavaScript & jQuery	10/3 – 10/21	3	
Server-Side Scripting: PHP	10/24 – 11/4	3	
Basic User Testing	11/7 – 11/18	4	Final Project
Taking a Site Live	11/21 – 12/2	1-5	

INFO/CS 1300 : Week 10 Monday : PHP Introduction

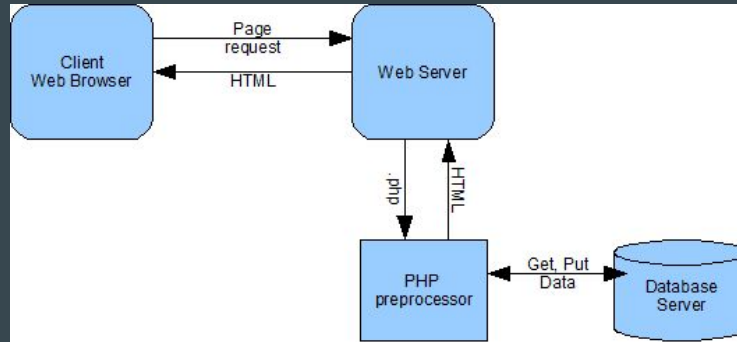
We are here!

What is PHP?

PHP is an acronym for hypertext preprocessor

PHP code is executed on the server and returned to the browser as a plain HTML page.

Extension: .php



http://w3epic.com/wp-content/uploads/2012/10/PHP_Client_Server_Messaging.gif

Syntax

In order to begin coding in php, you must start your PHP code block with

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

HTML page example:

```
<body>
    <?php
        // put PHP code here
    ?>
</body>
```

Basic Logical Operators

Name	Operator	Example	Explanation
and	&&	\$a == 3 and \$b == 9	
	and		
or		\$a == 1 \$a == 2	
	or		
not	!	!false returns true	
inequality	!=	1 != 2	
equal	==	1 == "1" evaluates to true	same number, but different type
identical	===	1 === "1" evaluates to false	same number and same type

Comments

Single line comments

// this is a single line comment

Multiple line comments

/* this is a
multiple line comment */

Variables

- All variables are preceded with a \$ sign.
- Variables must start with a letter or underscore.
- Variables must contain only alpha numeric characters a-z, A-Z, 0-9 or underscores _

`$_int = 3;` correct

`$state = true;` correct

`$1string = "hello world";` incorrect

`$@ge = 3;` incorrect

String concatenation

Join strings using a period .

```
$string1 = "hello ";  
$string2 = "world";  
$result = $string1 . $string2;
```

Input:
echo \$result;

Output:
hello world

Control Statements

if

```
if (condition) {  
    // execute this code if condition is true  
}
```

if ... else

```
if (condition) {  
    // do this if true  
} else {  
    // do this if false  
}
```

elseif

```
if (condition) {  
    // execute code if condition is true  
} elseif (condition) {  
    // execute this code if this condition is  
    true  
} else {  
    // execute this code if all conditions are  
    false. The else must be present in an elseif  
}
```


Arrays

- Good for storing multiple values in one variable.
- Use the array() function created for PHP.

Before:

```
$major1 = "Information Science";  
$major2 = "Architecture";  
$major3 = "History";
```

With arrays: all indexes start at 0.

Indexed array:

```
$major = array("Information Science", "Architecture",  
"History");
```

```
$major[0] returns Information Science  
$major[1] returns Architecture  
$major[2] returns History
```

Associative Arrays:

```
$major = array("Information Science" => "Arts and Sciences",  
"Architecture" => "AAP", "History" => "Arts and Sciences");
```

```
$major['Information Science'] returns Arts and Sciences  
$major['Architecture'] returns AAP  
$major['History'] returns Arts and Sciences
```

Input:

```
echo "Information Science is in " . $major['Information  
Science'];
```

Output:

Information Science is in Arts and Sciences

Functions

Creating a function syntax

```
function functionName(parameters if any) {  
    // do stuff  
}
```

Example:

```
function addition(num1, num2) {  
    return num1 + num2;  
}
```

Calling a function

This is how you make your function work.
If you don't call your function, the code
inside it won't run.

Input:

```
addition(2, 3);
```

Output:

```
5
```

For Loop

Syntax

```
for (init counter; condition; increment counter) {  
    // do stuff  
}
```

1. **init counter:** initializes the counter in the loop
2. **condition:** the loop will run while this condition is satisfied. In other words, if the condition is true, the loop will continue, else it will stop.
 - the condition is evaluated each time it goes through the loop
3. **increment counter:** increases the counter

Example

Print out the numbers from 1 to 5.

```
for ($x = 1; $x < 6; $x++) {  
    echo $x . "<br>"  
}
```

Output:

1
2
3
4
5

Note: `$x++` increments `x`'s value by 1.

Foreach Loop

Syntax

```
$ages = array(18, 19, 20, 21);  
  
foreach ($ages as $value) {  
    // do stuff  
}
```

Output:

```
18  
19  
20  
21
```

Example

Print out all the ages in the array.

```
foreach ($ages as $value) {  
    echo "$value";  
}
```

include

Your footer.php page contains only the code for the footer of the page.

PHP footer.php

```
<div class = "footer">
    <img src = "logo.jpg" alt = "logo">
    <p>Copyright &copy; 2016. John Doe.</p>
</div>
```

In your **index.php** you would include the footer the following way:

```
.
.
.<?php include 'footer.php'; ?>
</body>
</html>
```

Why use include?

1. Imagine your webpage has 5 pages and you want your footer to appear on every page.
2. Instead of copying all that code block on every page, you save it to ONE file and ONLY INCLUDE that SINGLE file on each page.
3. This saves the pain of having to edit the code for the footer on all 5 pages if you wish to make a change to it.

Steps

1. Save your footer code on a separate file using the extension .php (eg. footer.php).
2. Decide where you want it to be on the page.
3. Place the following code in the chosen line:
 <?php include 'footer.php'; ?>

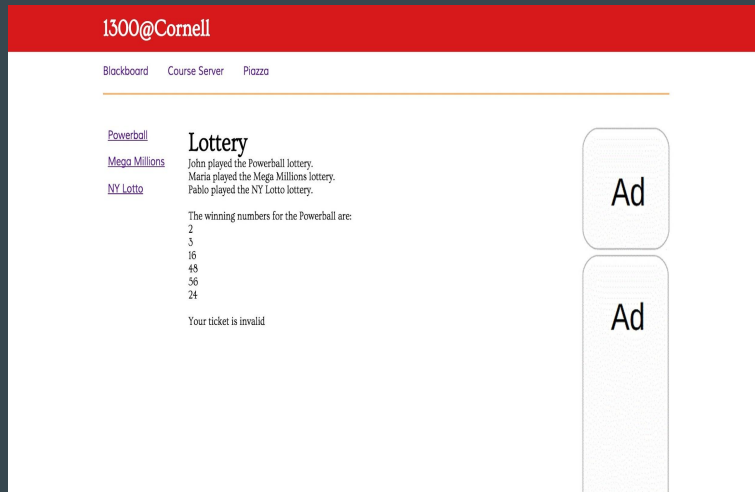
EXERCISE

Download the zip file for the lab on Blackboard.

1. Complete all the **TO DO's** found in **php_basics.php**.
2. Complete all the **TO DO's** found in **exercise.php**.

Solutions will be posted later today.

Expected Output:



MAMP Workshop

Brandon Giraldo will be hosting a workshop on MAMP, the local server used to process your PHP.

VERY USEFUL FOR WORKING AND DEBUGGING IN PHP!

MONDAY, OCTOBER 31 @5PM IN GATES 122

More info in this piazza [post](#)