



PHP Basics

HFCC – CIS222

18 January 2012

Announcements:

✓ HW 1 Due Today!

- You should turn that in

✓ Quiz 2 on 23 Jan (Monday)

- On this stuff...

Embedded Languages

- Language code is embedded directly in the markup
- Use tags to indicate which code is PHP code

`<?php` `?>`

- End most statements with semi-colons

```
1 <html>
2 <head>
3   <title>Example Page</title>
4 </head>
5 <body>
6   <div id="title">This is the page title</div>
7   <?php
8     echo "<p>Here is some text to display on the page.</p>";
9   ?>
10 </body>
11 </html>
```

Syntax Rules

➤ Use the **.php** file extension

➤ Provide output with **echo**

```
echo "Hello world!";
```

➤ Variables must begin with a dollar sign (\$)

```
$studentName = "Jer";
```

- Dollar sign must be followed by a letter or underscore

➤ Comment lines with **//** or **#**

```
$var = 'This is seen by the parser'; // This is not
```

➤ Comment blocks between **/*** and ***/**

```
/*
```

```
    This is all ignored by the parser
```

```
*/
```

➤ Assign values with the equal sign

```
$sum = 7 + 3;
```

Outputting

- Provide output with *echo*
 - Echo outputs into the markup (the page's source)
 - Output can contain tags as well as text
- Echoing with quotes
 - Single and double quotes are similar with important differences
 - Single quotes do not parse variables and special characters
 - Double quotes DO parse variables and special characters

```
$name = 'Jer';
```

```
echo "Hello, $name";
```

```
echo 'Hello, $name';
```

Hello, Jer

Hello, \$name

Questions?



Let's Try It!

- **Make a basic PHP page**
- Create a basic webpage called *InClass-PHP.php*
 - Create a variables to store your name, and the course number
 - Store your name and course number in the variables
 - Output your variables so that it looks like this
 - My name is Jer
 - I am enrolled in CIS222 this semester
 - Upload both files to the server and test
- **Raise your hand if you have finished (or if you are stuck)**

Additional Thoughts

- You have to use the .php extension in order to ensure that the script will be parsed
- All formatting (including character returns and/or line feeds) must be done with markup
- Programs will not run locally, you must upload the files to the server and view with a browser to see the results

Questions?



Data Types (1/2)

➤ PHP is loosely typed

- No declarations required
- Will change type based on the data you put in them

➤ Types

- **Integer** – positive or negative whole numbers
- **Float** – whole or partial numbers
- **String** – a series of characters stored together
- **Boolean** – value of true or false
- Arrays, Objects, Resources

Data Types (2/2)

\$var1 = 5;

5 (int)

\$var1 = "5";

5 (str)

\$var2 = \$var1 + 5;

10 (int)

\$var3 = \$var2 + 2.2;

12.2 (flt)

\$var4 = \$var3 + "words";

12.2 (flt)

\$var5 = \$var3 + "5 words";

17.2 (flt)

Questions?



`include()` & `require()`

- Add the contents of an external file to the current file
 - Make code more readable
 - Add security
 - Reuse old code
 - Increase portability
- In PHP, `include()` and `require()` drag in a file almost like a copy/paste operation
 - If `include()` fails, the program continues but displays an error
 - If `require()` fails, the program halts and an error is displayed

Using `include()` & `require()`

Here is my `index.php`

```
<html>
<head>
  <title>Title of page</title>
</head>
<body>
  <div id="pageTitle">My Web Page!</div>
  <p>This is the content of my page.</p>
  <?php
    include('otherPage.php');
  ?>
  <p>And here is the end of my page.</p>
</body>
</html>
```

Here is how it looks when the server sends it to the browser:

```
<html>
<head>
  <title>Title of page</title>
</head>
<body>
  <div id="pageTitle">My Web Page!</div>
  <p>This is the content of my page.</p>
  <p>More content...</p>
  <p>And here is the end of my page.</p>
</body>
</html>
```

Here is `otherPage.php`

```
<?php
  echo "<p>More content...</p>";
?>
```

Another Example...

index.php

```
<html>
<head>
    <title>Title of page</title>
</head>
<body>
    <?php include('header.php'); ?>
    <?php include('content.php'); ?>
    <?php include('footer.php'); ?>
</body>
</html>
```

Creates this:

```
<html>
<head>
    <title>Title of page</title>
</head>
<body>
<h1>The Amaze-o-Tron</h1>
<h3>Something amazing you should
buy</h3>
<p>The Amaze-o-Tron is indispensable.
You simply cannot do without it. You
would pay billions for it, but we're
only asking for millions. Altruism!</p>
<div class="copyright">
    &copy; Jer Lance
</div>
</body>
</html>
```

header.php

```
<h1>The Amaze-o-Tron</h1>
<h3>Something amazing you should buy</h3>
```

content.php

```
<p>The Amaze-o-Tron is indispensable. You
simply cannot do without it. You would pay
billions for it, but we're only asking for
millions. Altruism!</p>
```

footer.php

```
<div class="copyright">
    &copy; Jer Lance
</div>
```

Frequent Uses

- Some common uses for `include( )` and `require( )`
 - Include .php files that contain sensitive data (like passwords)
 - Allows the sensitive files to be stored in non-web accessible locations
 - Store all functions in separate files for readability and portability
 - Store pieces of a site that are used frequently in a single separate file for re-usability and improved editing
 - Enable single driver-file encoding of a site

Next Session

- Quiz on PHP Basics
- User Input and Forms
 - (Read CH4, pp188-194)