BU Center for Digital Imaging Arts

WEB DESIGN AND DEVELOPMENT CURRICULUM

WD108 WordPress 1 - PHP

WD108 WordPress 1 - PHP

PHP, a recursive acronym for “PHP: Hypertext Preprocessor”, is a general use scripting language. PHP code is embedded into HTML source code and is interpreted by a web server with a PHP processor module that outputs a webpage document. Thus PHP is used to create dynamic web pages. PHP basically acts as a set of parameters that tell a HTML page how to display in a browser.

Syntax

PHP Can be put anywhere within the HTML of a document. You can even intermingle HMTL and PHP. To denote that you are writing PHP there is a PHP tag:

```
<?php   …   ?>
```

Within the PHP tags you can comment code out the same way you would with JavaScript:

```
//  Line Comment
/*  Defined Region Comment  */
```

Variables:

Variables are “containers” that can store content (strings, numbers, functions, information of any kind). Once you have defined a variable calling it will have the effect of “pasting” that stored value in that spot.

```
$foo = ‘Hello World’;
echo($foo); //Will Print the value “Hello World”
```

Variables can be created and modified

```
$bu = “Boston University”;
echo($bu); //Will print the value “Boston University”
$bu = $bu . “ CDIA”;
echo($bu); //Will print the value “Boston University CDIA”
```

Quotes:

Single Quotes (‘’): Indicate a string in which whatever you write is a literal string value.
Double Quotes (“”): A string with in double quotes will be parsed by PHP! So, if you pass a variable into a string inside of double quotes it will print the variable’s value in its place.

```
$name = “Bob”;

echo “Hi $name”;
echo ‘Hi ’ .  $name
```

Functions:

Functions are ways of carrying out a series of events or changes bound to a specific call:

```
function sayName($var1){            //Define the function
        echo $var1;
};

sayName(‘Betty’);                   //Call the function…
```

Global Variables:
A special set of variables stored as an associative array.

```
$_GET[‘page’]    //This will look in the url and pull the value from the variable called “page”
```

the query string is everything after the “?” in the url…
url: www.mysite.com/index.php**?page=home&sex=male&hair=red**

Arrays:
Arrays can be thought of as “super” variables. Much like variables they store information and can be called to regurgitate that information. Arrays are diffent from variables because they can hold multiple values. So if a variable is a container then an array is a utility box. Arrays thus have “keys” (compartment names) and “values” (what is in that compartment).

```
$myArray = array(
        ‘key’ => ‘value’,
        ‘key’ => ’value’
);
```

There are two basic types of arrays: Associative arrays and numeric arrays. Associative arrays are ones where you assign the key a string value, like “foo” and foo would then have an associated value.

```
$myAssociativeArray = array(
        ‘foo’ => ‘bar’
);

echo $myAssociativeArray[‘foo’]    //Will print “bar”.
```

Numeric Arrays do not have an assigned name to the key, so it defaults to a numeric indexing starting at 0 and incrementing up one integer increment for each time.

```
$myNumericArray = array(‘zero’, ‘one’, ‘two’, ‘three’);

echo $myNumericArray[0];        //Will print “zero”
```

As a note if you ever see the “->”, this is not the same as “=>”! The former, referred to as the Arrow Operator is used to reference a method or property via a defined class. This is much more complicated in terms of programming, so is out of scope. It is however worth mentioning since we will see it in WordPress. Whenever you see it you can basically read it as “from”.

```
$myQuery -> special_function()
```

Conditional Statements:
Conditional statements, much as the name implies, are used to assess if a condition has been met and, dependent upon the answer, execute some code.

```
$sex = ‘male’;
if($sex == ‘male’){
        //Do Something Masculine
}
else{
        //Don’t do something masculine
};
```

Another syntax that is sometimes used is the following:

```
$sex = ‘male’;
if($sex == ‘male’):
        //Do Something Masculine

else:
        //Don’t do something masculine
endif;
```

Sometimes you many see the code written in a short hand like this if there is only one line of code to execute and there is no else condition:

```
if($name == ‘Henry’)
echo “Hi $name”;
```

Read More?

http://php.net/
http://www.w3schools.com/php/