ITMD 462/562
Web Site Application Development

Lecture 3

Fall 2015 – September 9, 2015

Tonight's Agenda

- PHP Built-in Functions
- Custom Functions
- Form Processing

PHP Built-in Functions

String Manipulation Functions

- Functions to use and modify strings and their contents
- Very necessary to most applications. Most form data comes in as strings.
- http://php.net/manual/en/ref.strings.php

```
echo
print
sprintf, fprintf, printf
explode, implode
trim
htmlspecialchars
nl2br
strlen
substr
```

Math Functions

- Functions and constants for numeric manipulation, cleanup, and display
- http://php.net/manual/en/ref.math.php
- http://php.net/manual/en/math.constants.php

```
rand
mt_rand
min
max
ceil
floor
```

Variable Handling

- Ways to test for and convert variable values
- Since PHP doesn't require a specific declaration of a variable you need to check for them before using them
- http://php.net/manual/en/ref.var.php

```
isset
empty
is_numeric
is_string
intval
floatval
strval
serialize, unserialize
```

Array Functions

- Functions used to count, manipulate, filter, merge, sort, and more
- http://php.net/manual/en/ref.array.php

```
count, sizeof
in_array
key_exists
array_pop
array_push
array_merge
sort
array_slice
```

Loop over elements in arrays:

```
foreach($array as $value){

}

foreach($array as $key=>$value){

}
```

http://php.net/manual/en/control-structures.foreach.php

Date / Time functions

- There is a set of functions for dealing with dates and times
- PHP also has an object-oriented interface for DateTime but we will look at that later. Better to use the classes for date time than the functional code ultimately
- http://php.net/manual/en/ref.datetime.php

```
date
getdate
time
date_default_timezone_set
strtotime
date_parse
```

Custom Functions

User Functions

- A function is a block of statements that can be reused in a program.
- Functions are executed by calling the function by adding parenthesis after the name
- User functions are declared with the `function` keyword

```
function functionName() {
    code statements to execute;
}
```

User Functions

- Any valid PHP code can be inside the function, including function and class definitions
- Same naming rules as variables. Start with a letter or underscore, then followed by letters, underscores, or numbers
- Functions don't necessarily have to be declared before they are used but must exist in the scope of the script before they are used.
- http://php.net/manual/en/functions.user-defined.php

Function Arguments

- You can pass information to the function via an argument.
- Arguments are comma-delimited lists of expressions which evaluate left to right

```
function echoString($aString) {
     echo $aString;
}

echoString('test');    // outputs test
```

- Arguments can have default values
- http://php.net/manual/en/functions.arguments.php

Function Return Values

- You can return a value from a function by using the `return` statement.
- Any data type can be returned by the function.
- The `return` statement causes the function to end immediately and pass back the value.
- If there is no `return` statement NULL will be returned

```
function squareNum($aNum) {
    return $aNum * $aNum;
}

$val = squareNum(4);
echo $var;  // outputs 16
echo squareNum(5); // outputs 25
```

http://php.net/manual/en/functions.returning-values.php

Anonymous Functions

- PHP does support anonymous functions, or closures.
- These are functions without a specific name defined.
- Most useful for values of callback parameters.
- http://php.net/manual/en/functions.anonymous.php

```
$helloWorld = function($aName) {
    echo ‘Hello’ . $aName;
}

$helloWorld(‘Brian’); // outputs Hello Brian
```

Forms

Processing user input

Structure

- Forms are the way for users to enter information into a web page and send it to the server for processing
- Doesn't always have to be a strict 'form' in the display. It can be hidden or disguised as a button, field, or other item
- HTML Form Tag

```
<form action="somescript.php" method="POST">
   <input>
    …
</form>
```

Form Action

- The action is the name or URL of the resource/script that you want to process the form input
- Often can be the same page
- If we detect if the page was called by a GET or POST request we could conditionally do different things or display different pieces of content using a conditional like an `if` statement

Form Method

- This will be the method the browser sends the request and form data to the server.
- Forms typically use POST, URLs typically use GET
- GET request has the form data or parameters as components of the URL
  - http://www.iit.edu/form.php?name=brian&school=sat
  - Need to be careful to URL encode your parameters
- POST requests encode the form data or parameters in the body of the request

Form enctype

- There is an HTML attribute on the form tag called enctype
- It specifies how the form data will be encoded when submitted to the server
- Only used for POST method
- Default Value – Ensures that all characters are encoded before they're sent to the server. This is what is used if you leave it off the form tag.
  - `application/x-www-form-urlencoded`
- If your form has a file upload control you must use a different enctype. This is required and not optional. It ensures no character conversions take place and transfers the form data as a compound MIME doc.
  - `multipart/form-data`

Input tags

- There are various form input controls to submit data such as text, password, radio, checkboxes, select lists, and others.
- The `name` attribute on the form control will be the variable name in PHP

```
<input type="text" name="firstName">
```

- These should be review from your HTML class
- https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms
- http://www.tutorialspoint.com/html/html_forms.htm
- http://www.w3schools.com/html/html_forms.asp
- http://www.w3schools.com/html/html_form_input_types.asp

Handle form input in PHP

Global Input Arrays

- The users input is placed in these arrays based on request method

`$_GET[]`

`$_POST[]`

`$_REQUEST[]`

- Request is the combination of the two other arrays, GET and POST. It is a copy of the data and not a reference to the data.
- They are global variables
- Files come in via the `$_FILES[]` array
- http://php.net/manual/en/reserved.variables.php

Type Conversion

- All form data comes in to the server in string form or arrays of strings.
- You needs some of the type conversion and variable functions we looked at earlier.
- Things to do
  - Check if the variable exists
  - Validate that the variable is in the correct format
  - Process the data to protect your site from security perspective
  - Things like isset, empty, strlen, is_numeric, floatval, ==, all help you

Input Safety and Security

- The User can input anything in the form controls. There is no guarantee that the user input didn't contain SQL, HTML, JavaScript
- JavaScript validation is commonly used to prevent the submission but it can be disabled in the browser and doesn't exist in CLI. YOU MUST DO SERVER SIDE VALIDATION TO BE SAFE
- You must clean and sanitize user input depending how it will be used.

`htmlspecialchars`

`addslashes`

`str_replace`

- Database specific functions exist and more

Testing for POST

- Best way to test if the request is POST is to use the SERVER array
- https://secure.php.net/manual/en/reserved.variables.php

```
if ($_SERVER['REQUEST_METHOD'] == 'POST')
```

- Other options include
- Adding a hidden form field and then look for that field in the post array

```
<input type="hidden" name="submitcheck" value="submit">
```

```
if (isset($_POST['submitcheck']))
```

Assignments

- Assignment 1 is Posted
- Due **Sunday, September 20, 2015 at 11:59pm Chicago Time**
- We will discuss in class on September 23 so no submissions will be accepted after 6:00pm on September 23.