Introduction to PHP (Hypertxet PreProcessor, HyperText Processor)
Part 1 of 2

References

- Chapter 8 Introduction to Server-Side Development with PHP of the Text Book Entitled Fundamentals of Web Development, by Randy Connolly and Ricardo Hoar
- PHP 5.6.14 Released Oct. 1, 2015
- PHP 5.6.1 Released Oct. 2, 2014, http://php.net/
- PPH Documentation, http://php.net/urlhowto.php
- PHP Language Reference, http://php.net/manual/en/langref.php
    - Basic Syntax
    - Data Types
    - Variables
    - Constants
    - Expressions
    - Operators
    - Control Structure
    - Include Files
    - Classes and Objects
    - Namespaces
    - Exceptions
    - Predefined Variables

Basic Syntax
PHP Tags

<? …. ?> shorthand version of <?php ….?>
<script language = “php”>
….
</script>

PHP Comments: three types of Comments

/* */ Multiple-line comment, Block comment
// End-of-line comment, One line comment at the end of the statement
Single-line comment, at the beginning of the line

PHP Data Types

- Boolean // true, false
- Integer // whole numbers
- Float // Decimal numbers
- String // Letters, characters
- Array // A collection of data of various data types

- Object // Instances of classes

Variables

- Case sensitive
- Loosely types
- Always start with the $ symbol before the variable name
- Examples
    - $count = 42;
    - $id = 1234;
    - $artist1 =”Picasso”;
    - $artist2 =”Raphael”;
    - $php5 = ‘version’;
    - $os = “Microsoft Windows Prof 7”;
    - $this_int = 50; // Standard decimal notation
    - $that_int = 062; // Octal number
    - $my_int = 0x32; //Hexadecimal

Variable Manipulation

```
<?php
        $sum = 10 + 2;              //12
        $sum = $sum + 5;            //17
        $sum = %sum % 2;            //1
        $answer = 5;
        $answer += 2;
        $answer *= 2;
        $answer ++;
        $answer --;
?>
```

String Escape Sequences (Escape Characters) in PHP

Escape String	Meaning
\n	New line
\t	Horizontal tab
\\	Back slash character
\$	$ character
\’	Single quote
\”	Double quote
\###	ASCII character (octal)
\x##	ASCII character (hexadecimal)

Constants

- Typically defined near the top of PHP file via the **define()** function
- Examples

```
<?php
# uppercase for constants – programming convention
Define("DATABASE_LOCAL", "localhost");
Define("DATABASE_NAME", "ArtStore");
Define("DATABASE_USER", "Fred");
Define("DATABASE_PASSWD", "F5^7%ad");
….
…
$db = new mysqli(DATABASE_LOCAL, DATABASE_NAME, DATABASE_USER,
DATABASE_NAME);
?>
```

Control Structures

- if .. else
- switch .. case
- while, do while
- for

Include Files

- include "somefile.php"
- include_once "somefile.php"
- require "somefile.php"
- require_once "somefile.php"

Reference to Include File PHP Scripts Examples

```
include "files.php";                    // http://php.net/manual/en/function.include.php
include_once "file.php";                // http://php.net/manual/en/function.include-once.php
require ('library.inc');                // http://php.net/manual/en/function.require.php
```

Functions

- PHP Built-in
  - echo() //Output to HTML
  - define() // Define constants
  - printf() // Formatted output
- User Defined
  - Syntax: function, return
  - Calling a function
  - Parameters
    - Passing by values
    - Passing by reference
  - Variable scopes

```
/*
*  This function returns nicely formatted System Time string using the current
*  System time.
*/
function getTime(){
    return date("H:1:s");
}

/* This function outputs the footer menu
 *
 */
function outputFooterMenu() {
    echo '<dic id="footer">';
    echo '<a href=#>Home </a> | <a href=#Product</a>  | ';
    echo '<a href=#>About us </a> | <a href=#Contact us</a>';
    echo '<div>';
}
```

Example

```
<? php
        $answer = 10;
        echo(++$answer)." ";
        echo "$answer<BR>";
        $answer += 10;
        echo ($answer++)." ";
        echo $answer; ?
        unset($answer);         //destroy the variable
?>
<?php
        %thankyou_string = "Thank you ";
        %thankyou_string = thankyou_string . "for your comments!";
        // String concatenation operator period .
?>
```

Predefined Variables

- $Superglobals // Bulit-in variables that are always available in all scopes
- $GLOBALS // All variables available in global scope
- $_SERVER // Server and execution environment info
- $_GET // HTTP GET variables
- $_POST // HTTP POST variables
- $_FILES // HTTP file upload variables
- $_REQUEST // HTTP request variables
- $_SESSION // Session variables

- $_ENV // Environment variables
- $_COOKIE // HTTP cookies
- $php_errormsg
- $HTTP_RAW_POST_DATA // Raw POST data
- $http_response_header // HTTP response headers
- $argc // The number of arguments passed to script
- $argv // Array arguments passed to script

PHP Functions Reference

- **echo()**
- **unset()** **// Destroy variables**
- **func_get_args()** **// get function arguments**
- **func_num_args()** **// get function argument count**

String Manipulation Functions

- **strcmp()** **// string comparison, case sensitive**
- **strcasecmp()** **// string comparison, non case sensitive**
- **substr()** **// the sub string**
- **strlen()** **// The number of chars in the string**
- **strops()** **// The character position**
- **chop()** **// Remove all white spaces from its ends**
- **trim()** **// remove all white spaces from both ends**
- **strtolower()** **// covert to lower case characters**
- **strtoupper()** **// convert to upper case characters**

Arithmetic Functions

- **floor(), ceil(), round(), srand(), rand(), abs(), min(), max()**

Ouput Functions

- **print()**
- **printf()** **// formatted output**

PHP Functions

- func_get_args() // http://us3.php.net/manual/en/function.func-get-args.php
- func_num_args()

```
<!DOCTYPE html>
<!—funct_arguments.php - A trivial example to illustrate a php document -->
<html lang = "en">
  <head>
    <title> funct_arguments.php </title>
<meta charset = "utf-8" />
  </head>
  <body>
```

```
<?php
    function func_dynamic() {
    echo “ITC 250/CPET 499 Web Systems: “.func_num_args(). “Number of arguments.<br>”;
    $args = func_get_args();
    for($i = 0; $i < count($args); $i++{
       echo “Passed arguments: {args[$i]} <br>;
        }
     }
    func_dynamic(5, 4, 3, 2, 1 );
?>
 </body>
 </html>
```