Introduction to PHP

CSE/ISE 102: Intro to Web Design (Section 02)
Stony Brook University

What is PHP?

- PHP = "PHP: Hypertext Preprocessor"
- An HTML-embedded scripting language
  - syntax is similar to C, Java, and Perl
  - the server processes the PHP code and translates it into HTML before serving the Web page

PHP Script Formatting

- PHP commands are enclosed in the following delimiters: <?php and ?>
- Commands are followed by semicolons
- Variable names start with $
- We can insert many PHP commands into a single HTML file
- Comments are similar to JavaScript

Simple Scripts

- <?php echo "Hello, world!"; ?>
- <?php echo $_SERVER['HTTP_USER_AGENT']; ?>

Printing and Strings

- Use 'echo' to print something to standard output
  - use . to concatenate values
- Strings can be single- or double-quoted
  - single-quoted strings don't expand variables or \n sequences

Special Global Variables

- $_SERVER — hash of server variables
- $_GET — returns hash of GET variables
- $_POST — returns hash of POST variables

- Access values by variable name
  - e.g., $_GET["name"]

Server Variables

- SERVER_ADDR
- SERVER_NAME
- SERVER_PROTOCOL
- REQUEST_METHOD
- REQUEST_TIME
- QUERY_STRING
- REMOTE_HOST
- REMOTE_ADDR
- and more...

Variable Scope

- Scope rules are slightly different in PHP
- PHP variables only have a single scope
- Global variables are not automatically usable inside functions
  - they must be declared 'global' inside the function, or use the $GLOBALS array

A More Complex Example

```
<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') != FALSE)
{
    echo 'You are using Internet Explorer<br />';
}
?>
```

If Statements

- Similar to that of C, Java, and JavaScript:

```
if (expression)
    statement
```

- Use curly braces to surround blocks
- We can also add elseif and else clauses
  - Note the extra 'e' in elseif

Mixing Modes

- We can mix PHP and HTML modes:

```
<?php if ( condition ) { ?>
<h3>The if condition was true</h3>
<?php
} else { ?>
<h3>The if condition was false</h3>
<?php } ?>
```

Useful String Functions

- strpos() searches a string for another string
  - if found, returns the numeric position relative to the start of the source string
  - otherwise, returns FALSE
- e.g., $pos = strpos($mystring, $findme);

More String Functions

- strlen()
- strtoupper()/strtolower()
- ucfirst() — make first character uppercase
- ucwords() — capitalize first letter of each word

More String Manipulation

- str_replace (search, replace, originalString)
  - returns new string
- substr_replace(original, replacement, start, length)
  - replaces *length* characters, beginning at *start*, with *replace*

explode() and implode()

- explode() breaks a string into an array of individual words
  - explode(delimiter, original_string)
- implode() combines an array of strings
  - implode(delimiter, array_of_parts)

PHP Functions

- Define a function with the "function" keyword
  - you can also specify function parameters

```
function my_function ($a, $b)
{
    echo $a;
}
```

Fancier Functions

- We can specify default parameter values in the function header:

  function foo ($a = 123, $b = 456) { ... }

- Functions can also return values

PHP and Forms

- Use your PHP document as your form action
- In your PHP file, use $_GET or $_POST to retrieve the form data
  - use the `htmlspecialchars()` function to escape any special characters
  - $a = htmlspecialchars($_GET['name']);

```
<html>
 <head>
 <title>My Form</title>
 </head>
 <body>
 <form action="fav_words.php" method=post>
 My name is:
 <br> <input type="text" name="YourName">
 My favorite word is:
 <br /><input type="text" name="FavoriteWord">
 <input type="submit" name="submit" value="That's
Right!">
 </form>
 </body>
</html>
```

```
<html>
<head>
<title>Great!</title>
</head>
<?php
// Capture the values posted to this php program from the text fields
// which were named 'YourName' and 'FavoriteWord' respectively
$YourName = $_REQUEST['YourName'] ;
$FavoriteWord = $_REQUEST['FavoriteWord'] ;
?>
<body bgcolor="#FFFFFF" text="#000000">
<p>
Hi <?php print $YourName; ?>
<p>
You like the word <b> <?php print $FavoriteWord; ?>!?!</b>
<p>That's my favorite word too!
</body>
</html>
```

Arrays

- To create an array, declare the variable and assign a value to a specific position:
  - `$emp_array[0] = "Bob";`
  - `$emp_array[1] = "Sally";`
  - `$emp_array[2] = "Joe";`
- Use square brackets to refer to an element
- Use `count()` to get number of values

Associative Arrays

- Like arrays, but they use keys rather than numbers to identify values:
  - $salaries["Bob"] = 40000;
  - $salaries["Sally"] = 65000;
  - $salaries["Joe"] = 50000;

Loops

- We can also use while and for loops in PHP
- These work just like in JavaScript and Perl
  - while (condition) { ... }
  - for (setup ; test ; update) { ... }

Loops and Associative Arrays

- Use a foreach loop to process an associative array
- Syntax: $something as $key => $value

```
foreach($myArray as $name => $age)
{
    echo "Name: $name, Age: $age <br>";
}
```

What's the Date?

- Use the date() function to get a timestamp
  - by default, date() gives you the current date
- The argument specifies the format
  - e.g., date("m/d/y")
    - "m" = month, "d" = day, "y" = year

Date Components

- r: full date, time, and timezone
- a/A: am or pm
- g/G: hour (1-12 or 0-23) without leading zeros
- h/H: hour with leading zeros
- i: minutes with leading zeros
- s: seconds with leading zeros
- d/j: day of month (#)
- D/l: day of week (abbrev)
- m/n: month number
- M/F: month as text
- Y/y: 4- or 2-digit year