INLS 672
Web Development 2

PHP Functions

UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

Functions: what and why

A function is a set of statements that performs a specific task.

- Often <u>return a value</u>, but not required
- May be <u>passed parameters</u> to perform a task with specific values
- Built-in or user-defined

Simpler code

- Easier to understand a program when it is broken down into functions
- Easier to read than one long sequence of statements

Code reuse

- Reduces the duplication of code and makes programs smaller
- If a task is performed multiple times at different places in your program, it can be written once and called when needed

Easier to debug and test

- By testing each function, it is easier to isolate and fix errors
- Easier to maintain: only have to fix it once

Built-in PHP Functions

- Already available, included with the PHP interpreter
- Over 5000 built-in functions
- PHP Manual: Function Reference
- Some commonly used built-in functions
  - Array
  - Calendar, Date and Time
  - Math
  - String

Some Useful String Functions

`empty($str)` returns true if empty

`strlen($str)` returns length of a string

`substr($str, $i, $len)` returns a substring starting at index, $i

`strpos($str1, $str2)` searches $str1 for occurrence of $str2

`str_replace($str1, $new, $str2)` replaces $str1 with $new in $str2

`str_split($str)` converts a string to an array

`sprintf` returns a formatted string

String comparison

`strcmp($str1, $str2)` case-sensitive string comparison

`strcasecmp($str1, $str2)` case-insensitive string comparison

References

- PHP Manual: String Functions
- w3schools: PHP String Functions

Converting Strings to Numbers

Using type casting

```
$value = (int) '78.5';     // result is 78
$value = (int) 'abc';      // result is 0

$value = (float) '55.3';   // result is 55.3
$value = (float) 'xyz';    // result is 0
```

Using functions

`intval($var)` returns an integer value for the `$var` parameter

```
$value = intval('46.4');       // result is 46
```

`floatval($var)` returns a floating point value for the `$var` parameter

```
$value = floatval('9.5');       // result is 9.5
```

Converting numbers to strings: use sprintf function

PHP Manual: String conversion to numbers

User-defined PHP Functions

Syntax of function definition

```
function function_name ([param1, param2, … ])  {
    // your function code goes here
    [return value;]    }
```

Example of a user-defined function, **`coin_toss()`**, that calls a built-in PHP function, **`mt_rand()`**

`coin_toss()` is a function with no parameters that returns a value

```
function coin_toss ()  {
   $toss = mt_rand(0,1); // generates a random # b/w 0 and 1
   if ($toss == 0)
      $coin = 'heads';
   else
      $coin = 'tails';
   return $coin; }
```

How to call this function: **`echo coin_toss();`**

Defining and Calling Functions

Function with 1 parameter, returns no value

```
function display_error ($error)  {
   echo 'An error occurred: ' . $error; }
```

How to call: `display_error('Value out of range');`

Function with 3 parameters, returns a value

```
function avg_of_3 ($x, $y, $z)  {
   $avg = ($x + $y + $z) / 3;
   return $avg; }
```

How to call: `$average = avg_of_3(10, 20, 30);`

Variable Scope

- Scope refers to the visibility, or accessibility, of a variable
- Determines what code can access a variable you have defined
- Local variables are defined inside of a function, and are only available to the code in the function
- Global variables are defined outside of a function
  - Unlike other languages, they are available only to code that runs at the 'global level'
  - By default, they are not accessible within a function
  - To make a global variable accessible within a function, you have to use the `global` statement

Variable Scope Examples

A variable defined outside of a function has <u>global scope</u>

```
$a = 10;                    //$a has global scope
function show_a() {
   echo $a;   }             //$a has a null value here
show_a();                   //displays nothing
```

To use this variable within a function, you must use **global**

```
$b = 10;                    //$b has global scope
function show_b() {
   global $b;               //$b refers to the global variable
   echo $b;   }
show_b();                   //displays 10
```

A variable defined inside a function has <u>local scope</u>

```
function show_c() {
   $c = 10;                 //$c has local scope in show_c
   echo $c;   }             //displays 10
show_c();
echo $c;                    //$c has a null value
```

Temperature Conversion with Functions

Use a function to convert and display result

temperature_conversion.php

```
...
<body>
<?php
include 'temp_conversion_library.php';
$degrees = '';
$conversion_type = 'ctf';
$message = '';
...
```

temperature_conversion_library.php

```
<?php
function convert_temperature($temp_value, $conv_type) {
   $result = sprintf("%0.2f", $temp_value * 9/5 + 32);
   $message = "$temp_value&deg; Celsius is $result&deg; Fahrenheit";
   return $message;
}
?>
```

PHP include statement: includes and evaluates a file