CUSTOM FUNCTIONS: NAMING RULES, SYNTAX & SCOPE

Preslee Dayton

Naming (Variable) Rules

- Variable names:
  - are case-sensitive
  - can contain letters, numbers, and underscores
  - can't contain special characters
  - can't begin with a digit or two underscores
  - can't use names that are reserved by PHP

    Ex: $this

Syntax

- Syntax refers to the rules that must be followed when writing PHP code.
  - PHP has a syntax that is similar to Java and JavaScript.

Syntax – General Info.

- A *statement* controls the operations of a program.
- A *comment* helps document what the code does.
- To code a *single-line comment*, write two forward slashes (//) or a pound sign (#) and continue until the end of the line.
- To code a *multiple-line comment*, code /*, followed by the comment, followed by */.
- To make code easier to read, use indentation and extra spaces to align statements.

Syntax Rules

- PHP statements end with a semicolon.
- PHP ignores extra whitespace in statements.
  - Spaces, tabs, and new line characters.

Scope

- In PHP, variables can be declared anywhere in the script. The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes.
  - Local
  - Global
  - Static

Scope - Local & Global Variables

- Local – a variable declared inside a function and can only be accessed inside that function.
- Global – a variable declared outside a function and can only be accessed outside that function.

```
<?php
$x=5; // global scope- variable written outside the function.

function myTest() {
$y=10; // local scope- variable written inside the function.
echo "<p>Test variables inside the function:</p>";
echo "Variable x is: $x";
echo "<br>";
echo "Variable y is: $y";
}

myTest();

echo "<p>Test variables outside the function:</p>";
echo "Variable x is: $x";
echo "<br>";
echo "Variable y is: $y";
?>
```

Scope - Global Variable, cont.

- The global keyword can also be used to access a global variable from within a function.

```
<?php
$x=5;
$y=10;

function myTest() {
  global $x,$y;
  $y=$x+$y;
}

myTest();
echo $y; // outputs 15
?>
```

Scope – Static Variables

- Static – normally, when a function is completed/executed, all of its variables are deleted. When the writer doesn't want the local variable(s) to be deleted, a state scope can be used.

```
<?php

function myTest() {
  static $x=0;
  echo $x;
  $x++;
}

myTest();
myTest();
myTest();

?>
```

References

- http://www.w3schools.com/php/php_variables.asp
- Murach's PHP and MySQL by Joel Murach & Ray Harris