

Bay Area Video Coalition

Introduction to Programming

Syllabus

4 November 2011

Time: 9:00 a.m. to 5:00 p.m.

Instructor: Richard S. Mitchell, rsm@urchar.com

Web site: www.urchar.com/teaching/intro_programming

Course description

This course is an introduction to programming. The main emphasis will be Web development, namely JavaScript and PHP.

Recommended text and resources

Javascript

<http://www.w3schools.com/js/default.asp>

<https://developer.mozilla.org/en/JavaScript/Guide>

<https://developer.mozilla.org/en/JavaScript/Reference>

PHP

<http://www.w3schools.com/php/default.asp>

<http://www.php.net/>

Course Objectives

This one-day class will introduce students to concepts common to most programming languages:

- data types and variables
- functions
- program-flow control
- server-side vs. client-side scripting

Programming conventions and best practices will be emphasized from the very beginning.

Course Outline

- I. Introductions
 - A. What does everyone know about programming?
 - B. What does everyone hope to get from this class?
 - C. My background ...
- II. Getting started
 - A. Setting up an environment for development
 1. Text editors and IDEs
 2. browser developer tools
 - a. Mozilla: Firebug, DOM Inspector, Web Developer
 - b. Safari: from the Safari Preferences, click 'Advanced', then select 'Show Develop menu in menu bar'.
 - c. Internet Explorer: IE Developer Toolbar for IE7
 3. setting the browser default background color
 4. using MAMP (<http://localhost:8888/>)
 - B. Our sandbox file: index.php
- III. Programming basics
 - A. What is programming?
 1. Programming is controlling a computer and remembering how you did it.
 2. In the case of Web development:
 - a. control the server
 - b. communicate with other services, e.g. a database
 - c. control the client, usually a browser
 - d. control the page elements
 - e. respond to user actions
 - f. remember stuff
 - g. control access

- B. What is a programming language?
 - C. First step: “hello world”
 - 1. in JavaScript
 - a. `document.write()`
 - 2. in PHP
 - a. `print()` and `echo`
- IV. Adding scripts to HTML
- A. JavaScript
 - 1. body
 - 2. head
 - 3. external
 - B. PHP
 - 1. escaping
 - a. `<?php echo $str ?>`
 - b. `<?=$str?>` is the same
- V. A second step: temperature conversion
- A. JavaScript
 - B. PHP
- VI. Our little program explained
- A. Comments
 - 1. single line
 - 2. multi-line
 - B. Variables
 - 1. declaring
 - 2. assigning value
 - 3. changing value
 - 4. using
 - 5. data types
 - C. Statements and blocks
 - 1. A statement is a command to do something.
 - 2. Curly braces group statements.
 - D. Controlling the flow
 - 1. while loop
 - E. Operators
 - 1. assignment
 - 2. conditions (testing)
 - 3. arithmetic
 - 4. precedence
 - 5. concatenation
 - F. Output
 - 1. JavaScript
 - a. `document.write` and `.writeln`
 - b. using the DOM for display
 - 2. PHP
 - a. `print` (and `println`)
 - b. `echo`
 - c. `printf` functions: formatted output
 - 3. the difference:
 - a. JS: displayed by the browser
 - b. PHP: in the response from the server
- VII. Third step: more than one way ...
- A. Using the for control statement
 - 1. JavaScript
 - 2. PHP
 - B. Explained
 - 1. initialization—done once before loop is entered
 - 2. condition—tested before each loop
 - 3. increment—performed after loop and then tested
 - 4. skip the variable (`$celsius`): expressions return a value

- 5. a possible upgrade: constants
- VIII. Turning it into a function
 - A. JavaScript
 - B. PHP
 - 1. optional args
- IX. Using if-else
 - A. JavaScript
 - B. PHP
- X. Strings
 - A. Javascript object
 - 1. instantiating: constructor or literals
 - 2. properties
 - a. length
 - 3. methods
 - a. `string.charAt(n)`
 - b. `string.indexOf(string)`
 - c. `string.substr()`
 - d. `string.toLowerCase()`
 - B. PHP
 - 1. string functions
 - a. `substr(string, start, length)`
 - b. `strpos(string, search)`
 - c. `strtolower(string)`
- XI. Arrays
 - A. Javascript object
 - 1. instantiating
 - 2. properties
 - a. length
 - 3. methods
 - a. `array.pop() / array.push(value)`
 - b. `array.reverse()`
 - c. `array.slice()`
 - B. PHP
 - 1. array functions
 - a. `count()`
 - b. `array_pop() / array_push()`
 - c. `array_reverse()`
 - d. `array_slice`
 - e. `array_merge()`
- XII. Dates and Times
 - A. Javascript object
 - 1. instantiating
 - 2. methods
 - a. `Date()`—Date object with current date-time
 - b. `Date(dateString)`—Date object
 - c. `date.getDate()`—day of the month
 - d. `date.getMonth()`—month index from zero
 - e. `date.getFullYear()`—four-digit year
 - B. PHP
 - 1. date-time functions
 - a. `time()`—current system date-time
 - b. `date(format, time)`—format a date-time
 - c. `strtotime(string)`—convert string date to time
- XIII. Debugging
 - A. Javascript—alert, confirm, prompt
 - B. PHP—firephp

Example V.A. JavaScript temperature conversion

```
/*
 * print a table of Fahrenheit to Celsius conversions
 */
var fahr, celsius;
var lower = 0; // lower limit of table
var upper = 300;
var step = 20;

fahr = lower;

while (fahr <= upper) {
  celsius = 5 * (fahr - 32) / 9;
  document.write(fahr + " : " + Math.round(celsius) + "<br />");
  getEl("js_output").innerHTML += fahr + " : " + Math.round(celsius) + "<br />";
  fahr = fahr + step;
}
```

Example V.B. PHP temperature conversion

```
/*
 * print a table of Fahrenheit to Celsius conversions
 */
$lower = 0; // lower limit of table
$upper = 300;
$step = 20;
$fahr = $lower;

while ($fahr <= $upper) {
  $celsius = 5 * ($fahr - 32) / 9;
  println(sprintf('%s : %s<br />', $fahr, round($celsius)));
  $fahr += $step;
}
```

Example VII.A. for loop in JavaScript

```
var fahr, celsius;
var lower = 0;
var upper = 300;
var step = 20;

for (fahr = lower; fahr <= upper; fahr += step) {
  celsius = 5 * (fahr - 32) / 9;
  document.write(fahr + " : " + Math.round(celsius) + "<br />");
}
```

Example VII.B. for loop in PHP

```
$lower = 0;
$upper = 300;
$step = 20;

for ($fahr = 0; $fahr <= 300; $fahr += 20) {
    println(sprintf('    %s : %s<br />', $fahr, round(5 * ($fahr - 32) / 9)));
}
```

Example VII.B. for loop in PHP with constants

```
define('LOWER', 0);
define('UPPER', 300);
define('STEP', 20);

for ($fahr = LOWER; $fahr <= UPPER; $fahr += STEP) {
    println(sprintf('    %s : %s<br />', $fahr, round(5 * ($fahr - 32) / 9)));
}
```

Example JS function—no args

```
function fahrToCel()
{
    var fahr;
    var lower = 0;
    var upper = 300;
    var step = 20;

    for (fahr = lower; fahr <= upper; fahr += step) {
        document.write(fahr + " : " + Math.round(5 * (fahr - 32) / 9) + "<br />");
    }
}
```

Example PHP function—no args

```
function fahr_to_cel()
{
    define('LOWER', 0);
    define('UPPER', 300);
    define('STEP', 20);

    for ($fahr = LOWER; $fahr <= UPPER; $fahr += STEP) {
        println(sprintf('    %s : %s<br />', $fahr, round(5 * ($fahr - 32) / 9)));
    }
}
```

Example JS function—with arguments

```
function fahrToCel(lower, upper, step)
{
    // tested separately
    if (!step) {
        return;
    }

    if (!upper) {
        return;
    }

    if (!lower && lower != 0) {
        return;
    }

    // tested using OR
    // if (!step || !upper || !lower && lower != 0) {
    //     return;
    // }

    for (var fahr = lower; fahr <= upper; fahr += step) {
        document.write(fahr + " : " + Math.round(5 * (fahr - 32) / 9) + "<br />");
    }
}
```

Example PHP function—with arguments

```
function fahr_to_cel($lower = 0, $upper = 300, $step = 20)
{
    for ($fahr = $lower; $fahr <= $upper; $fahr += $step) {
        println(sprintf('      %s : %s<br />', $fahr, round(5 * ($fahr - 32) / 9)));
    }
}
```

IX.A. JavaScript—using if-else

```
var s = "Et harum quidem rerum facilis est et expedita distinctio.";
var IN = true;
var OUT = false;

/**
 * Counts the number of newlines, words, and characters.
 * @param {String} s
 * @type void
 */
function wc(s)
{
  var c, nl, nw, slen = s.length, state;

  state = OUT;
  nl = nw = 0;

  for (var i = 0; i < slen; i++) {
    c = s.charAt(i);

    if (c == "\n") {
      nl++;
    }

    if (c == " " || c == "\n" || c == "\t") {
      state = OUT;
    }
    else if (state == OUT) {
      state = IN;
      nw++;
    }
  }

  document.write("newlines: " + nl + "<br />words: " + nw + "<br />chars: " +
    slen);
}
```

IX.B. PHP—using if-else

```
define('IN', TRUE);
define('OUT', FALSE);

$s = "Et harum quidem rerum facilis est et expedita distinctio.\n";

function wc($s)
{
    $slen = strlen($s);
    $state = OUT;
    $nl = $nw = 0;

    for ($i = 0; $i < $slen; $i++) {
        $c = substr($s, $i, 1);

        if ($c == "\n") {
            $nl++;
        }

        if ($c == ' ' || $c == "\n" || $c == "\t") {
            $state = OUT;
        }
        elseif ($state == OUT) {
            $state = IN;
            $nw++;
        }
    }

    println(sprintf('newlines: %s<br />words: %s<br />chars: %s', $nl, $nw, $slen));
}
```

X.A. Javascript strings

```
document.write("This is a test.".length);
document.write("<br />");
document.write("This is a test.".charAt(6));
document.write("<br />");
document.write("This is a test.".indexOf(" is"));
document.write("<br />");
document.write("This is a test.".toUpperCase());
```

X.B. PHP strings

```
$fmat = '%s<br />';
$s = "This is a test."
println(strlen($s));
println(sprintf($fmat, substr($s, 0, 4)));
println(sprintf($fmat, strpos($s, ' is')));
println(sprintf($fmat, strtoupper($s)));
```

X.A. Javascript arrays

```
var colors = new Array("red", "green", "blue"); // new with constructor
var colors2 = [ "red", "green", "blue" ]; // literal

function print(value)
{
    document.write(value);
    document.write("<br />");
}

print("colors: " + colors);
print("length: " + colors.length);

colors.push("yellow");
print("new colors: " + colors);

colors2.reverse();
print("reversed: " + colors2);

colors2.splice(1, 1);
print("remove blue: " + colors2);
```

X.B. PHP arrays

```
$colors = array("red", "green", "blue"); // with function

function output($s, $a)
{
    println("<pre>$s:");
    print_r($a);
    println('</pre>');
}

output('colors', $colors);
output('length', count($colors));

array_push($colors, 'yellow');
output('new colors', $colors);

$rev_colors = array_reverse($colors);
output('reverse', $rev_colors);

array_splice($rev_colors, 1, 1);
output('remove blue', $rev_colors);
```

XII.A. Javascript date-time

```
var date = new Date("6/1/2010 08:00");
// array literal
var months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];

document.writeln(date.getDate() + " " +
  months[date.getMonth()] + " " +
  date.getFullYear() + "<br />");
document.writeln(date.toLocaleString());
```

XII.B. PHP date-time

```
define('LONG_DATE', 'j M Y');

println(date(LONG_DATE, strtotime('6/1/2010')) . '<br />');
println(date(LONG_DATE, time()));
```