Bay Area Video Coalition

Introduction to JavaScript

Course Outline/

Instructor: Richard S. Mitchell, urchard@comcast.net
Web site: www.urchard.com/teaching/javascript/

I. Introductions
  A. What does everyone know about HTML, CSS, and JavaScript?
  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. Setting up Aptana
      a. Preferences : Aptana Studio : Themes
      b. Preferences : General : Editors : Text Editors
        i. Displayed tab width: 2–4
        ii. Insert spaces for tabs
    3. Setting the browser default background color
      a. Preferences : Content : Colors…
    4. Using your localhost server
      a. http://127.0.0.1:8888/
      b. http://localhost:8888/
    5. 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 IE7/8
  B. Using XHTML.
III. JavaScript
  A. About JavaScript: a brief history and description.
  B. Interacting with JavaScript
    1. browser popups from JS
      a. **`window.alert(msg)`**
      b. **`window.confirm(msg)`**
      c. **`window.prompt(msg, defaultValue)`**
    2. Firefox tools
      a. Tools : Web Developer
      b. Firebug
  C. Ways to include JavaScript in the HTML document.
    1. Script element in the body. **S–2**
    2. Script element in the head. **S–3**
    3. Scripts in external files. **S–4**
  D. Example 1: hello world
  E. Basic lexical rules **S–5-7**
    1. case sensitive
    2. tokens
      a. keywords—defined by JS
      b. identifiers—defined by the user
      c. reserved words—words which should not be used as identifiers since they may become keywords in the future
    3. whitespace chars: space, tab, newline
      a. separate tokens
      b. ignores more than one
    4. semicolon/new line
  F. Comments **S–8**
      a. C-style

        b. C++-style
        c. jsDoc
        d. commenting out …

G. Numbering systems **S–9**
    a. binary: 8 bits make a byte (4 bits make a nibble)
    b. decimal
    c. hexadecimal

H. Example 2: resources/counting_presentation.pdf

I. Example 3: Decimal / hexadecimal conversion

J. Primitive data **S–10-11**
1. numbers
    a. int
    b. float
    c. hexadecimal
    d. special: NaN and Infinity
2. boolean
3. **`null`**
4. **`undefined`**
5. **Scratchpad demo** `typeof`

K. Data structures **S–12-14**
1. **`Array`**
2. **`Object`**
3. **`Function`**
4. **`Date`**
5. **`String`**
6. **Scratchpad demo:** data structures

L. Variables **S–15-22**
1. declare with **`var`**
2. lexical rules
3. dynamic typing and conversions
4. **`parseInt`** and **`parseFloat`**
5. constants
6. literals
7. evaluation: **`undefined`** vs. undeclared

M. Expressions—any snippet of code that resolves to a value: **S–23-25**
1. literals: number, string, logical (**`true`** | **`false`**)
2. arithmetic
3. objects
4. variables: declared and initialized

N. Operators **S–26-30**
1. unary:
    a. operator—operand
    b. operand—operator
2. binary: operand—operator—operand
3. arithmetic
    a. unary: **`+ - ++ --`**
    b. binary: **`+ - * / %`**
4. bitwise
    a. unary: **`~`**
    b. binary: **`& | ^ << >> >>>`**
5. assignment
6. comparison: **`== != === !== < <= >= >`**
7. string—concatenation: **`+ +=`**
8. logical: **`! && ||`**
9. special
    a. conditional: **`(`**`condition`**`) ?`** `true_value` **`:`** `false_value`
    b. comma (**`,`**)
    c. **`delete`**

        d. `in`
        e. `instanceOf`
        f. `new`
        g. `this`
        h. `typeof`
        i. `void`
    10. operator precedence and associativity
        a. developer.mozilla.org/en/JavaScript/Reference/Operators/Operator_Precedence

O. Example 4: Temperature conversion

P. Controlling program flow **S–31-35**
    1. blocks: groups of statements
    2. conditional statements
        a. `if (...)`
        b. `else`
        c. `else if (...)`
        d. these evaluate `false`: `undefined`, `null`, `0`, `NaN`, empty string (`""`)
    3. looping statements
        a. `while`
        b. `for`

Q. Example 5 Char, word, and newline counting

R. Functions **S–36-38**
    1. defining and invoking
    2. parameters
    3. `return` statement
    4. variables in function

S. JavaScript and the DOM **S–39**
    1. input and output
        a. input-text element
        b. textarea element
    2. Triggering an action
        a. input-button element

T. Example 6: Encode/decode URI

U. Useful String methods **S–40**
    1. `charAt`
    2. `indexOf`
    3. `substr`
    4. `toLowerCase`, `toUpperCase`
    5. *number*`.toString(`*radix*`)`
        a. **Scratchpad demo** `Number(15).toString(2)  // returns 1111`
        b. **Scratchpad demo** `15.toString(16)`—why the error? set a variable and try it …

V. Example 7: Add and strip thousands separators

W. Example 8: Various mouse overs

X. Example 9: popup menu

Y. Working with objects **S–41-44**
    1. Another way to define/access object items: `obj["key"] =` *`some_value`*`;`
    2. Defining a method
        a. define method and assign
        b. assign anonymous function
    3. Using an object's prototype to add properties and methods
    4. `this` keyword

Z. More program flow control **S–45-48**
    1. advanced conditional
        a. `switch`
        b. **Scratchpad demo** converting HTML special chars
    2. loop enhancements
        a. `break`, `continue`, and `label`
        b. **Scratchpad demo** filtering for even numbers
        c. `do` … `while`

        d. **Scratchpad demo** int2bin
        e. Explicate functionality …
    3. object enumeration
        a. **`for`** … **`in`**
        b. **Scratchpad demo** `for ... in`

AA. Regular Expressions **S–49-50**

1. **Example 10: Test email format**
2. explanation

BB. Exceptions

1. **`try`** … **`catch`** … **`finally`**
2. **`throw`**