

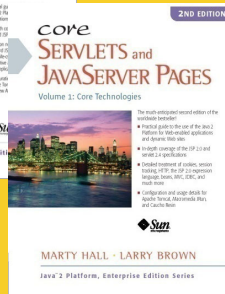
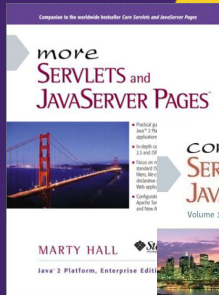


JavaScript: A Crash Course

Part I: Basics and Core Language Syntax

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/ajax.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Ajax & GWT training, see training courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
 - Java 6, servlets/JSP (intermediate and advanced), Struts, JSF 1.x, JSF 2.0, Ajax, GWT 2.0 (with GXT), custom mix of topics
 - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, Google Closure) or survey several
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, Web Services, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- Overview
- JavaScript references
- Embedding in browser
- HTML versions
- Basic syntax
- Arrays
- Strings and regular expressions

4

© 2010 Marty Hall



Intro

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Books

- **JavaScript the Definitive Guide**
 - By David Flanagan, O'Reilly. The only really complete reference on the JavaScript language. Thorough and well-written.
 - Makes the global variable blunder when covering Ajax.
- **JavaScript: The Good Parts**
 - By Douglas Crockford (of JSON and YUI fame), O'Reilly
 - Outstanding advanced guide to best practices in core JavaScript, especially functions, objects, and regular expressions. *Very* short.
 - No coverage of Ajax or DOM scripting. “The *Effective Java* of JS”.
- **Pro JavaScript Techniques**
 - By John Resig (of jQuery fame), APress
 - Excellent guide to best practices; not a thorough reference
 - Makes the global variable blunder when first covering Ajax.
- **DOM Scripting**
 - By Jeremy Keith, FriendsOf Press
 - Focuses on manipulating DOM and CSS
 - Makes the global variable blunder when briefly covering Ajax.

6

Online References

- **JavaScript tutorial (language syntax)**
 - <http://www.w3schools.com/js/>
 - http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Guide
- **JavaScript API references (builtin objects)**
 - <http://www.w3schools.com/jsref/>
 - <http://www.devguru.com/technologies/ecmascript/QuickRef/>
 - <http://www.devguru.com/technologies/JavaScript/>
 - <http://www.javascriptkit.com/jsref/>
 - http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Reference
- **HTML DOM reference (with JavaScript Examples)**
 - http://www.w3schools.com/html/dom/dom_reference.asp
- **Official ECMAScript specification**
 - <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

7

Firebug

- **Install Firebug in Firefox**
 - <http://getfirebug.com/>
- **Use Firebug command line for interactive testing**
 - <http://getfirebug.com/commandline>
- **Can use Firebug Lite in IE, Opera, Chrome**
 - Not great, but better than nothing
 - But Chrome and IE 8 now have Firebug Wannabe environments that are reasonably good
 - <http://getfirebug.com/firebuglite>
 - See especially “bookmarklet” link
- **For more details on Firebug usage**
 - See section on Ajax development and debugging tools

8

© 2010 Marty Hall



Embedding JavaScript in HTML

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Loading Scripts

- **script with src**

- `<script src="my-script.js" type="text/javascript"></script>`

- Purpose

- To define functions, objects, and variables.
- Functions will later be triggered by buttons, other user events, inline script tags with body content, etc.

- **script with body content**

- `<script type="text/javascript">JavaScript code</script>`

- Purpose

- To directly invoke code that will run as page loads
 - E.g., to output HTML content built by JavaScript
- Don't use this approach for defining functions or for doing things that could be done in external files.
 - Slower (no browser caching) and less reusable

10

Example (phish.js)

```
function getMessage() {
    var amount = Math.round(Math.random() * 100000);
    var message =
        "You won $" + amount + "!\n" +
        "To collect your winnings, send your credit card\n" +
        "and bank details to oil-minister@phisher.com.";
    return(message);
}

function showWinnings1() {
    alert(getMessage());
}

function showWinnings2() {
    document.write("<h1><blink>" + getMessage() +
        "</blink></h1>");
}
```

"alert" pops up dialog box

"document.write" inserts text into page at current location

11

Example (loading-scripts.html)

```
<!DOCTYPE ...><html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Loading Scripts</title>
...
<script src="./scripts/phish.js"
      type="text/javascript"></script>
</head>
<body>
...
  <input type="button" value="How Much Did You Win?"
        onclick='showWinnings1 ()' />
...
  <script type="text/javascript">showWinnings2 ()</script>
...
</body></html>
```

Loads script from previous page

Calls showWinnings1 when user presses button. Puts result in dialog box.

Calls showWinnings2 when page is loaded in browser. Puts result at this location in page.

12

Example (Results)



13

Loading Scripts: Special Cases

- **Internet Explorer bug**
 - Scripts with src fail to load if you use `<script.../>`.
 - You must use `<script src="..." ...></script>`
- **XHTML: Scripts with body content**
 - It is an error if the body of the script contains special XML characters such as `&` or `<`
 - E.g. `<script...>if (a<b) { this(); } else { that(); }</script>`
 - So, use CDATA section unless body content is simple and clearly has no special characters
 - `<script type="text/javascript"><![CDATA[
JavaScript Code
]]></script>`

14

© 2010 Marty Hall



HTML Versions and JavaScript

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **XHTML**
 - Most common version used with Ajax apps or Dynamic HTML apps (JavaScript apps that manipulate the DOM)
 - Follows XML syntax, lowercase tags
- **HTML 5 (sort of)**
 - Growing in popularity for Ajax or DHTML apps.
 - Version used now is basically XHTML but with a simpler DOCTYPE and <html> start tag
 - Doesn't matter if browser really supports HTML 5
- **HTML 4**
 - Very common in non-JavaScript apps
 - Not recommended for Ajax apps

16

XHTML

- **Summary**
 - Follows XML syntax. Lowercase tags, end tags required, quotes around attribute values.
- **Basic structure**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>...</title></head>
<body> ... </body></html>
```
- **Pros**
 - Code corresponds very directly to internal (DOM) representation by the browser
- **Cons**
 - DOCTYPE and <html> start tag are long and tedious

17

Pseudo-HTML 5

- **Summary**
 - Follows XML syntax. XHTML (transitional) syntax but with simpler DOCTYPE and <html> start tag.
- **Basic structure**

```
<!DOCTYPE html>
<html>
  <head><title>...</title></head>
  <body> ... </body></html>
```
- **Pros**
 - Code corresponds very directly to internal (DOM) representation by the browser
- **Cons**
 - Not strictly compliant with spec. May get warnings from formal validators, especially with non-CSS formatting.

18

HTML 4

- **Summary**
 - Does not follow XML syntax. Tags not case sensitive. End tags and quotes on attribute values sometimes optional.
- **Basic structure**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD><TITLE>...</TITLE></HEAD>
  <BODY> ... </BODY></HTML>
```
- **Pros**
 - Simple code. Widely used in non-Ajax apps.
- **Cons**
 - Source code and internal browser representation can be substantially different, requiring mental translation when thinking of how to manipulate DOM from JavaScript.

19



Basic JavaScript Syntax

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Variables

- **Introduce with “var”**
 - For global variables (!) and local variables.
 - No “var” for function arguments
- **You do not declare types**
 - Some people say JavaScript is “untyped” language, but technically it is “dynamically typed” language
 - JavaScript is *very* liberal about converting types
- **There are only two scopes**
 - Global scope
 - Be very careful with this when using Ajax.
 - Can cause race conditions.
 - Function (lexical) scope
 - There is *not* block scope as in Java

Operators and Statements

- **Almost same set of operators as Java**
 - + (addition and String concatenation), -, *, /
 - &&, ||, ++, --, etc
 - The == comparison is more akin to Java's "equals"
 - The === operator (less used) is like Java's ==
- **Statements**
 - Semicolons are technically optional
 - But highly recommended
 - Consider
 - return x
 - return
x
 - They are not identical! The second one returns, then evaluates x. You should act as though semicolons are required as in Java.
- **Comments**
 - Same as in Java (/* ... */ and // ...)

22

Conditionals and Simple Loops

- **if/else**
 - Almost identical to Java except test can be converted to true/false instead of strict true/false
 - "false": false, null, undefined, "" (empty string), 0, NaN
 - "true": anything else (including the string "false")
- **Basic for loop**
 - Identical to Java except for variable declarations
 - for(**var** i=0; i<someVal; i++) { doLoopBody(); }
- **while loop**
 - Same as Java except test can be converted to boolean
 - while(someTest) { doLoopBody(); }
- **do/while loop**
 - Same as Java except test can be converted to boolean

23

Other Conditionals and Loops

- **switch**

- Differs from Java in two ways
 - The “case” can be an expression
 - Values need not be ints (compared with ===)

- **for/in loop**

- On surface, looks similar to Java for/each loop, but
 - For arrays, values are array indexes, not array values
 - Use this loop for objects (to see property names), not arrays!
Fails with Prototype or other extended arrays
 - For objects, values are the property names
- `var person = { firstName: "Brendan", lastName: "Eich" };
for(var property in person) {
 doSomethingWith(person[property]);
}`

24

The Math Class

- **Almost identical to Java**

- Like Java, static methods (Math.cos, Math.random, etc.)
 - As we will see in next lecture, these are not *really* static methods, but syntax is similar to static methods in Java.
- Like Java, logs are base e, trig functions are in radians

- **Functions**

- Math.abs, Math.acos, Math.asin, Math.atan, Math.atan2, Math.ceil, Math.cos, Math.exp, Math.floor, Math.log, Math.max, Math.min, Math.pow, Math.random, Math.round, Math.sin, Math.sqrt, Math.tan

- **Constants**

- Math.E, Math.LN10, Math.LN2, Math.LOG10E, Math.PI, Math.SQRT1_2, Math.SQRT2

25



Arrays

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Array Basics

- **One-step array allocation**
 - `var primes = [2, 3, 5, 7, 11, 13];`
 - `var names = ["Joe", "Jane", "John", "Juan"];`
 - No trailing comma after last element (see later slide)
- **Two-step array allocation**
 - `var names = new Array(4);`
`names[0] = "Joe";`
...
`names[3] = "Juan";`
- **Indexed at 0 as in Java**
 - `for(var i=0; i<names.length; i++) {`
`doSomethingWith(names[i]);`
`}`

Looping Down Arrays in JavaScript

- **Java-style for loop**

- Roughly same as in Java. Don't forget the "var"!

```
for(var i=0; i<someArray.length; i++) {  
    var value = someArray[i];  
    doSomethingWith(value);  
}
```

- **JavaScript-specific for loop**

- Relies on fact that a nonexistent array index results in a value of undefined (not an exception) and that undefined means "false" in a test.

```
for(var i=0, value; value=someArray[i]; i++) {  
    doSomethingWith(value);  
}
```

- **for-in loop**

- *Not* recommended for looping down normal arrays.
 - Returns indexes, not values
 - Array-like objects can have extra properties

28

More on Arrays

- **Arrays can be sparse**

- ```
var names = new Array();
names[0] = "Joe";
names[100000] = "Juan";
```

- **Arrays can be resized**

- Regardless of how arrays is created, you can do:
  - ```
myArray.length = someNewLength;
```
 - ```
myArray[anyNumber] = someNewValue;
```
  - ```
myArray.push(someNewValue)
```

 - These are legal regardless of which way myArray was made

- **Arrays have methods**

- **push**, **pop**, join, reverse, sort, concat, slice, splice, etc.
 - See API reference

- **Regular objects can be treated like arrays**

- You can use numbers (indexes) as object properties

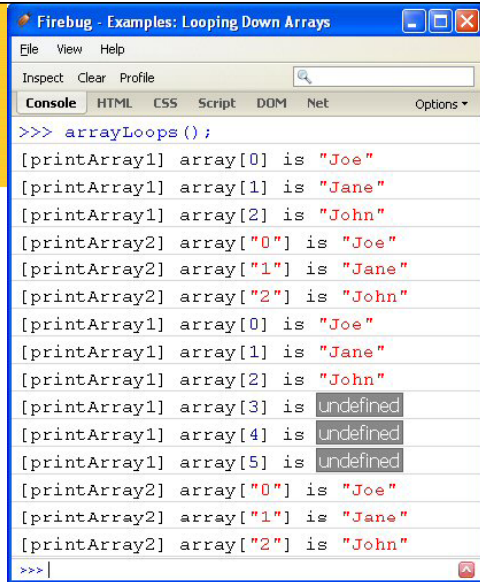
29

Arrays Example

```
function arrayLoops() {
    var names =
        ["Joe", "Jane", "John"];
    printArray1(names);
    printArray2(names);
    names.length = 6;
    printArray1(names);
    printArray2(names);
}
```

```
function printArray1(array) {
    for(var i=0; i<array.length; i++) {
        console.log("[printArray1] array[%o] is %o", i, array[i]);
    }
}
```

```
function printArray2(array) {
    for(var i in array) {
        console.log("[printArray2] array[%o] is %o", i, array[i]);
    }
}
arrayLoops();
```



console.log is a printf-like way to print output in Firebug Console window. For testing/debugging only.

Direct call for interactive testing in Firebug console. (Cut/paste all code into console command line.)

30

© 2010 Marty Hall



Strings and Regular Expressions

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

String Basics

- **You can use double or single quotes**
 - `var names = ["Joe", 'Jane', "John", 'Juan'];`
- **You can access length property**
 - E.g., `"foobar".length` returns 6
- **Numbers can be converted to strings**
 - Automatic conversion during concatenations.
 - `var val = 3 + "abc" + 5; // Result is "3abc5"`
 - Conversion with fixed precision
 - `var n = 123.4567;`
`var val = n.toFixed(2); // Result is 123.46 (not 123.45)`
- **Strings can be compared with `==`**
 - `"foo" == 'foo'` returns true
- **Strings can be converted to numbers**
 - `var i = parseInt("37 blah"); // Result is 37 – ignores blah`
 - `var d = parseFloat("6.02 blah"); // Ignores blah`

32

Core String Methods

- **Simple methods similar to Java**
 - `charAt`, `indexOf`, `lastIndexOf`, `substring`, `toLowerCase`, `toUpperCase`
- **Methods that use regular expressions**
 - `match`, `replace`, `search`, `split`
- **HTML methods**
 - `anchor`, `big`, `bold`, `fixed`, `fontcolor`, `fontsize`, `italics`, `link`, `small`, `strike`, `sub`, `sup`
 - `"test".bold().italics().fontcolor("red")` returns `'<i>test</i>'`
 - These are technically nonstandard methods, but supported in all major browsers
 - But I prefer to construct HTML strings explicitly anyhow

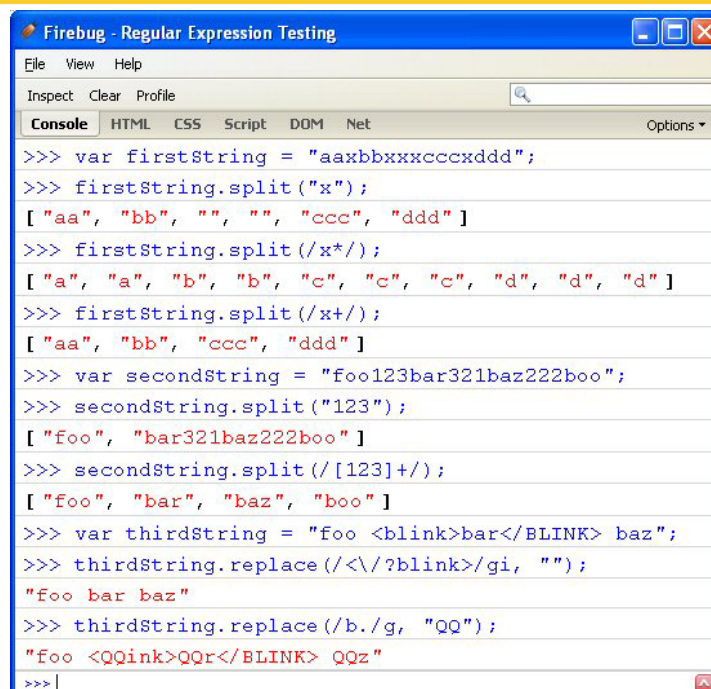
33

Regular Expressions

- **You specify a regexp with /pattern/**
 - *Not* with a String as in Java
- **Most special characters same as in Java/Unix/Perl**
 - ^, \$, . – beginning, end of string, any one char
 - \ – escape what would otherwise be a special character
 - *, +, ? – 0 or more, 1 or more, 0 or 1 occurrences
 - {n}, {n,} – exactly n, n or more occurrences
 - [] – grouping
 - \s, \S – whitespace, non-whitespace
 - \w, \W – word char (letter or number), non-word char
- **Modifiers**
 - /pattern/g – do global matching (find all matches, not just first one)
 - /pattern/i – do case-insensitive matching
 - /pattern/m – do multiline matching

34

Regular Expression: Examples



```
Firebug - Regular Expression Testing
File View Help
Inspect Clear Profile
Console HTML CSS Script DOM Net Options
>>> var firstString = "aaxbbxxxcccddd";
>>> firstString.split("x");
[ "aa", "bb", "", "", "ccc", "ddd" ]
>>> firstString.split(/x*/);
[ "a", "a", "b", "b", "c", "c", "c", "d", "d", "d" ]
>>> firstString.split(/x+/);
[ "aa", "bb", "ccc", "ddd" ]
>>> var secondString = "foo123bar321baz222boo";
>>> secondString.split("123");
[ "foo", "bar321baz222boo" ]
>>> secondString.split(/[123]+/);
[ "foo", "bar", "baz", "boo" ]
>>> var thirdString = "foo <blink>bar</BLINK> baz";
>>> thirdString.replace(/<\/?blink>/gi, "");
"foo bar baz"
>>> thirdString.replace(/b./g, "QQ");
"foo <QQink>QQr</BLINK> QQz"
>>>|
```

35

More Information on Regular Expressions

- **Online API references given earlier (See RegExp class)**
 - http://www.w3schools.com/jsref/jsref_obj_regexp.asp
 - <http://www.devguru.com/technologies/ecmascript/QuickRef/regexp.html>
- **JavaScript Regular Expression Tutorials**
 - http://www.evolt.org/article/Regular_Expressions_in_JavaScript/17/36435/
 - <http://www.javascriptkit.com/javatutors/re.shtml>

36

© 2010 Marty Hall



Wrap-up

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Use Firebug for testing and debugging**
- **Bookmark references**
 - <http://www.w3schools.com/js/>
- **Embedding in browser**
 - `<script src="blah.js" type="test/javascript"></script>`
 - Use XHTML or pseudo-HTML 5 syntax
- **Basic JavaScript syntax**
 - Declare local variables with var. No type declarations.
 - Loops and conditionals similar to Java.
- **JavaScript arrays**
 - Arrays are very different than in Java. Can have extra properties. Can resize them. Can be sparse.
 - But, you usually treat them like normal arrays, except that pop and push are widely used.

38

© 2010 Marty Hall



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.