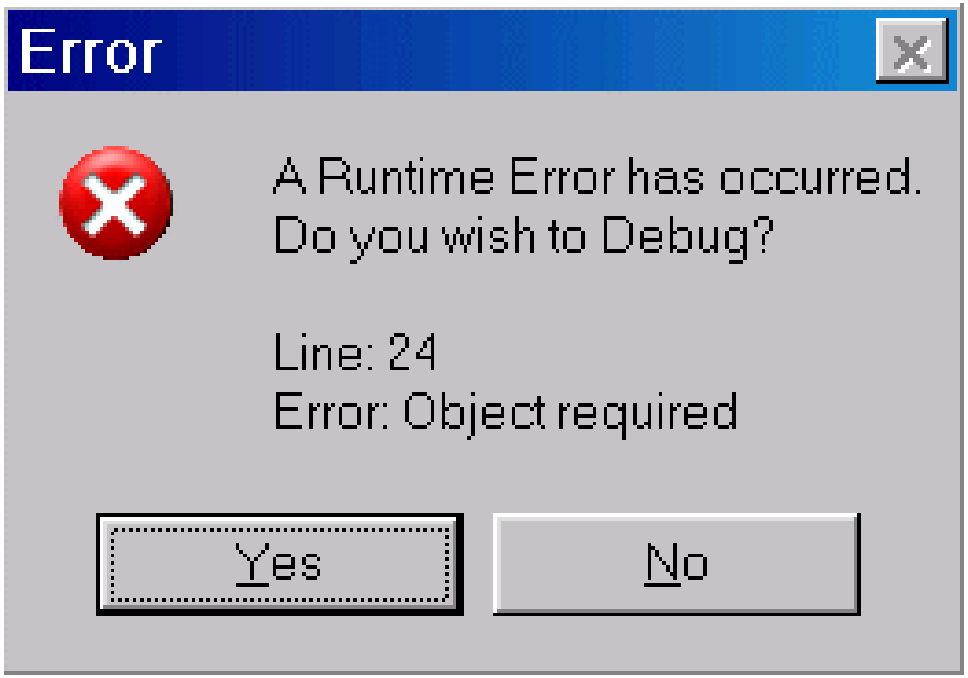
JavaScript Doesn't Have to Hurt:

Techniques, Tools and Frameworks to Ease JavaScript Development

Matt Hughes
Chariot Solutions

Is JavaScript painful?

- It certainly has been
  - Severe browser incompatibilities
  - Anemic tooling support
- Abstract it away?
  - Google Widget Toolkit
  - JavaServer Faces
- Make it irrelevant?
  - Adobe Flex
- Does it Matter?
  - Even if JS died today, we'd still have to maintain 100, 000 LOC



Overview

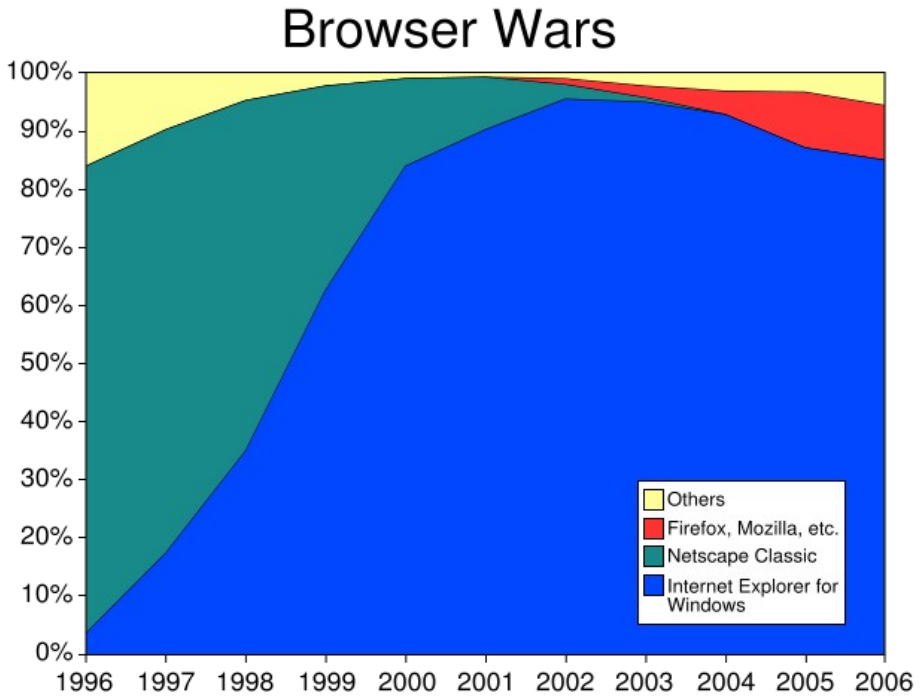
- There and Back Again
- Modern JavaScript Development
- IDEs
- Debuggers
- Other tools
- Sample project (FormValidator)

The Landscape has Changed

- Who: Web Designers
  => Developers
- How: Copy-n-Paste snippets
  => Object-Oriented code,
  Comprehensive Frameworks
- Tools: Dreamweaver/Notepad
  => Full-blown IDE
- Purpose: Small widgets
  => Desktop functionality

What caused the shift?

- In no particular order...
  - Mozilla's Rise From the Ashes
  - Push for Web Standards
  - Browsers Got Better
  - AJAX/Web 2.0 Craze
  - Developers Took Second Look at JavaScript
  - Frameworks Made Things *Just Work*

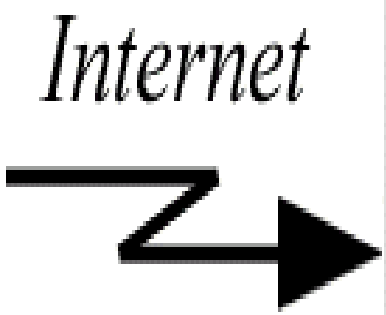


Wikipedia

Reintroduction to JavaScript

In case you weren't clear...

```
<html>
<script language="JavaScript">
function pushStack(theForm) {
   theForm.stack.value = theForm.display.value
   theForm.display.value = 0
}
</script>
</html>
```



Netscape's JavaScript Guide

Reintroduction to JavaScript (take two)

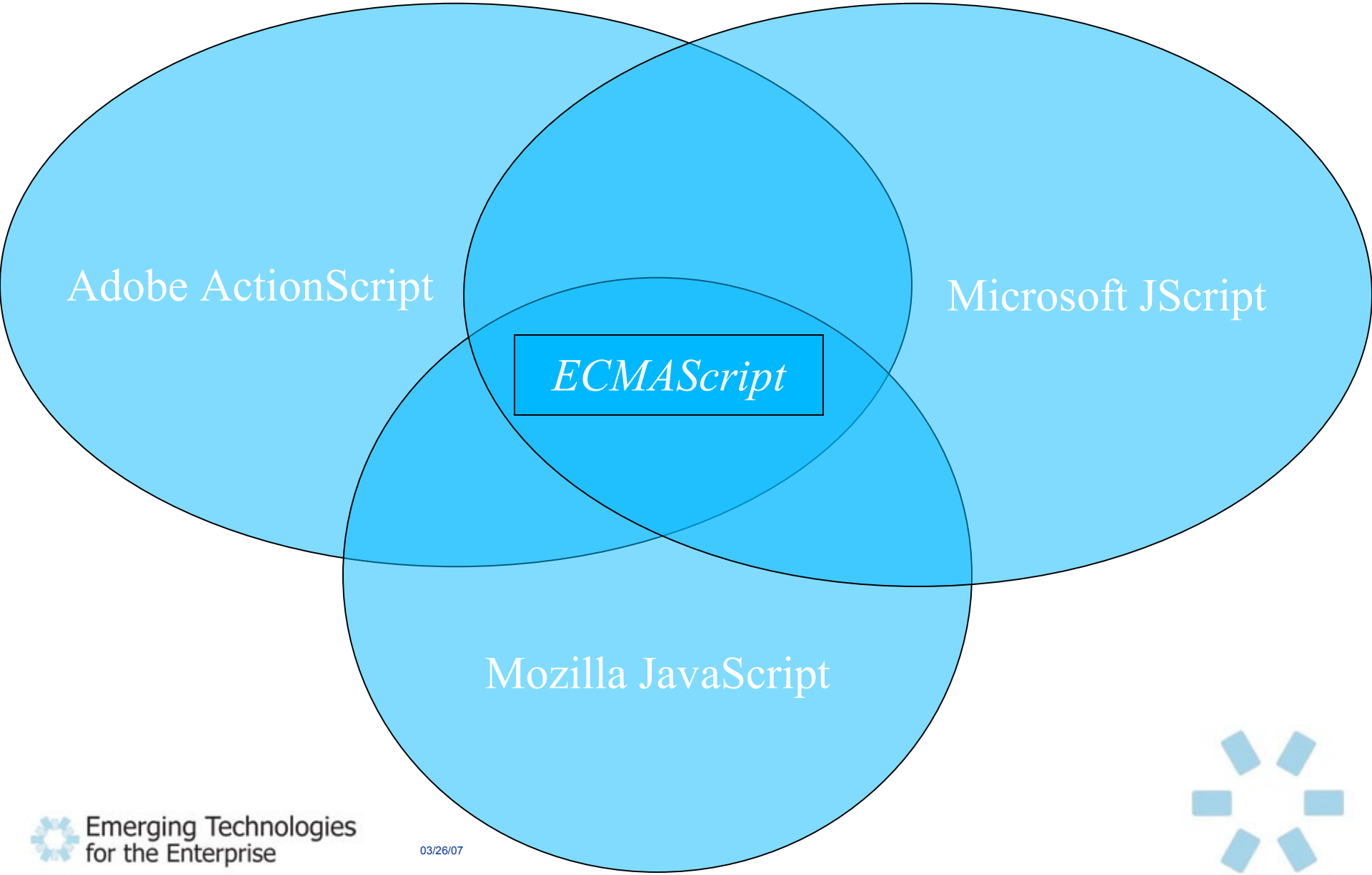
"The World's Most
Misunderstood Programming Language"
-- Douglas Crockford

Object-oriented

- Functions are first class citizens
- Closures
- Duck typing
- Exceptions
- Open classes
  - String.prototype.trim = function() { ...}

Give it another look.

“...sounds like a skin disease” - Brendan Eich



Not Your Father's JavaScript

- Prototypical Object-Orientation
- Unobtrusive JavaScript
- Namespaces
- Unit Tests

These aren't brand-new techniques but they are becoming more *important* as JavaScript code bases grow and RIA functionality becomes more integral to your application.

JavaScript Does Little By Itself

JavaScript
+
DOM (Document Object Model)
+
BOM (Browser Object Model)

Frameworks – What's Available

YahooUI

Frameworks - What They Do and Why You Want One

Abstract the DOM Away

“Fill in the Potholes”
(Prototype, Dojo, YahooUI, Mochikit)

... and AJAX and all that eye candy
(Script.aculo.us and some of the above)

The Events Pothole

```
// Register the event
if (widget.addEventListener) {
    widget.addEventListener('click', handleClick, false);
} else if (widget.attachEvent) {
    widget.attachEvent('onclick', handleClick);
} else {
    widget.onclick = handleClick;
}
// Handle the event
function handleClick(event) {
    var event = (event | window.event);
    // work with event
    if (event.preventDefault) {
      preventDefault();
    } else {
      event.returnValue = false;
    }
}
```


And how Prototype fills it

```
// Register the event
Event.observe(widget, 'click', handleEvent);

// Handle the event
function handleClick(event) {
    // work with event
    Event.stop(event);
}
```

If you find yourself thinking, “Well I could write that function”...

Stop! That is just one pothole of many.

Basic Features

Structure View	Some view that shows all symbols available in current scope
Code Formatting	Correctly format code
Find Usages	Find usages of symbol in project. This goes beyond mere search as it should only match instances of that declaration, not other string matches.
Templates	Has code templates for commonly used constructs
Live Documentation	Display documentation of functions as you use them
Go to Declaration	Can you jump to a function/variable declaration? Can you jump to a function even if it is declared in another file?

Advanced Features

Code Completion	Supports code completion
Refactoring	Supports code refactoring (Extract method, extract functions to external script)
Error Checking	As-you-type error checking
Code Analysis	Checks similar to Checkstyle/Lint
Understands OO	Code completion works with user-defined objects, etc
Built-in Debugger	Has a built-in debugger
External Libraries	Supports integration with external JS libraries (Prototype, etc.)

IDE Shootout – The Grades

- A
  - Intellij IDEA ($499)
- B
  - Aptana (Free / OS)
- C
  - Active State Komodo 4.0 ($295)
  - Spket (Free / OS)
- F
  - Netbeans 5.5 (Free / OS)
  - Eclipse 3.2 with WST (Free / OS)
  - Microsoft Web Developer Express 2005 (Free / Closed)

Debuggers

- Mozilla
  - Venkman
  - Firebug
  - Active State Komodo 4.0
- Microsoft
  - Script Debugger
  - Script Editor (comes with Office 2003 *thanks, a lot*)
  - Visual Web Developer 2005 Express Edition
- Safari
  - Drosera

Firebug

The Holy Grail of Web Development

- DTrace for your web application
- Runs in Firefox
- Current page is the context
- Debugger
- Code Profiler
- Logging
- Interactive Console links ***everything***

Other Tools

Code Analysis Tool

```
### Warnings
# Enable or disable warnings based on requirements.
# Use "+WarningName" to display or "-WarningName" to suppress.

+no_return_value              # function {0} does not always return a
    value
+redeclared_var               # redeclaration of {0} {1}
+missing_semicolon            # missing semicolon
+useless_assign               # useless assignment
+duplicate_case_in_switch     # duplicate case in switch statements
```

Unit Test Library

```
function testMinConstraint() {
  var minConstraint = new LengthConstraint(4, 10);
  var errMsg = minConstraint.enforceConstraint("aaa");
  assertEquals(errMsg, "Field must be at least 4 characters long.");
}

function testMaxConstraint() {
  var minConstraint = new LengthConstraint(4, 10);
  var errMsg = minConstraint.enforceConstraint("0123456789 ");
  assertEquals(errMsg, "Field must be at most 10 characters long.");
}
```

JavaScript Archive Network

- Online module repository
- Similar to CPAN (Comprehensive Perl Archive Network)
- Dojo-style imports

```
try {
  JSAN.use('Test.Simple');
} catch (e) {
  alert("Requires JSAN");
}
```

Documentation Tool

- Based on JavaDoc
- Uses the same syntax for documentation
- Produces similar output

DEMO

Where is JavaScript Heading?

ECMAScript Fourth Edition

- What does it look like?
  - Formal support for:
    - Classes
    - Private/Public methods
    - Package system
  - *Optional* static typing
    - function trim(value : **String**) : **String** { .. }
- When?
  - Spec finalized mid-2007?
  - Browsers “in 2010”?
- Problems
  - Companies have already “implemented ECMAScript 4”

Where is JavaScript Heading? (2)

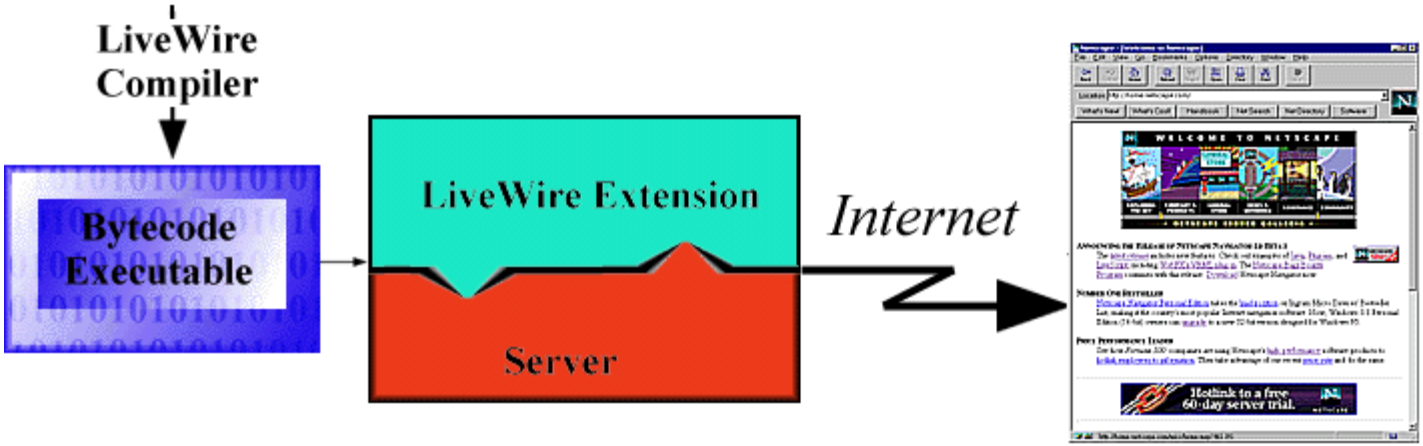
- WHATWG Standards
  - Mozilla, Apple, Opera
  - Web Forms 2.0
  - Web Applications 1.0
  - Web Controls 1.0
- XForms

Where is JavaScript Heading? (3)

- Tool Support Gets Better
- Project Tamarin
  - JIT VM donated by Adobe

Not just the browser anymore

- Adobe Flex and Apollo (ActionScript)
- Scripting in the JVM
- Widgets
  - Google
  - Yahoo
  - OS X



Netscape's JavaScript Guide

- JavaScript on the server
  - LiveWire lives! (renamed to Project Phobos)
- Becoming an important glue language (Greasemonkey, Chickenfoot)