

# JavaScript Libraries

Ajax Experience - July 2007

John Resig ([ejohn.org](http://ejohn.org))

Question: How do you want  
to write JavaScript?

# I) Plug-and-Play

- Drop in a “calendar widget” or “tabbed navigation”
- Little, to no, JavaScript experience required.
- Just customize some options and go.
- No flexibility.

## 2) Some Assembly Required

- Write common utilities
  - Click a link, load a page via Ajax
  - Build a dynamic menu
  - Creating interactive forms
- Use pre-made code to distance yourself from browser bugs.
- Flexible, until you hit a browser bug.

# 3) Down-and-Dirty

- Write all JavaScript code from scratch
- Deal, directly, with browser bugs
- Quirksmode is your lifeline
- Excessively flexible, to the point of hinderance.

# What we've just seen...

- Widgets
- Libraries
- Raw JavaScript

# What we've just seen...

- Widgets
- Libraries
- Raw JavaScript

# Why use a library?

- Makes JavaScript bearable
- Gets the job done fast
- Simplifies cross-browser support
- Sort of like C stdlib - no one just codes C by hand

# What kind of libraries exist?

	Open Source	Commercial
Client/ Server	AjaxCFC Qcodo	Atlas Backbase for Struts
Browser Only	Prototype Dojo jQuery Mootools Yahoo UI	Backbase SmartClient

# What kind of libraries exist?

	Open Source	Commercial
Client/ Server	AjaxCFC Qcodo	Atlas Backbase for Struts
Browser Only	Prototype Dojo jQuery Mootools Yahoo UI	Backbase SmartClient

# Open Source Libraries

	Browser Only	Client/Server
Task Specific	Scriptaculous moo.fx Open Rico	AjaxCFC Qcodo
General Purpose	Prototype Dojo jQuery Mootools Yahoo UI	Ruby on Rails CakePHP

# Open Source Libraries

	Browser Only	Client/Server
Task Specific	Scriptaculous moo.fx Open Rico	AjaxCFC Qcodo
General Purpose	Prototype Dojo jQuery Mootools Yahoo UI	Ruby on Rails CakePHP

# What should a library have?

- DOM Traversing/Manipulation
- Event binding
- Ajax Support
- Basic Animations

# What should a library have?

- Some want complex controls
  - Drag-and-drop
  - Auto-complete search
- Others want language enhancements
  - `Object.extend()`
  - `Array.each()`
- DOM, Event, Effects, Ajax is a good middle-ground

# Secondary Features

- Documentation
- Community Support
- File Size
- Popularity

# Secondary Features

- Documentation
- Community Support
- File Size
- Popularity

(Practical)



(Esoteric)

# The Libraries

Dojo

# Dojo: Overview

- Started early 2005 by Alex Russell
- Large development community
- Lots of corporate backing (IBM, AOL)
- Big code base, tons of features

# Dojo: Focus

- Building complete web applications
- A package hierarchy, e.g.:  
`dojo.addClass( ... )`
- Focus has transcended into widgets (Dijit)
- Huge number of features
- Today we're looking at Dojo 0.9

# Dojo: DOM Support

- Good DOM Modification support - handles many cross-browser issues
- Great DOM Traversal with Selectors

```
dojo.forEach( dojo.query("table tr:nth-child(even)"), function(row){  
    dojo.addClass( row, "odd" );  
});
```

# Dojo: Events and Effects

- Events support is complete and cross browser
- Good selection of generic animations

```
dojo.connect(dojo.byId("mylink"), "click", function(){  
    dojo.fadeOut({ node: this, duration: 500 });  
});
```

# Dojo:Ajax

- Supports a wide variety of Ajax communication
- Cross-domain Ajax support
- Built-in back button support

# Dojo: Docs & Community

- Documentation: Rough.  
In the middle of a major documentation overhaul for 0.9.
- Community: Lively  
Active forum with frequent discussions.

# Dojo: File Size & Popularity

- **File Size: Large (100Kb compressed)**  
It's highly recommended that you remove unused modules.
- **Popularity: Substantial**  
Backing from many corporate sponsors (IBM).

Prototype

# Prototype: Overview

- Started early 2005 by Sam Stephenson
- Incredibly popular, tied with Ruby on Rails' popularity
- Development backed by 37 Signals

# Prototype: Focus

- Improving the usability of the JavaScript language
- Big emphasis on adding in 'missing' JavaScript features
- Clean structure, clean objects and 'classes'

# Prototype: DOM Support

- DOM Modification works quite well:  
`Insertion.Bottom("list", "<li>Another item</li>");`
- So does DOM Traversing:  
`$$("table > tr").invoke("addClassName", "row");`

# Prototype: Events & Ajax

- Event binding just works:  

```
Event.observe("button", "click", function(){  
    alert("Thanks for the click!");  
});
```
- So does Ajax:  

```
new Ajax.Request("test.html", {  
    method: "GET",  
    onComplete: function(res){  
        alert( res.responseText );  
    }  
});
```

# Prototype: Details

- Code quality is fantastic, great features
- No included support for animations
  - [moo.fx](#)
  - [Script.aculo.us](#)
- Updated frequently

# Prototype: Docs & Community

- Documentation: Good  
Official API documentation, some tutorials
- Community: Distributed  
Many bloggers love Prototype, little centralized discussion (23 posts/day).

# Prototype: File Size & Popularity

- File Size: Moderate (99KB)  
Unfortunately, not easily compressible with Packer - they recommend GZIP.
- Popularity: Strong  
Apple, NBC, ESPN, CNN, Amazon

jQuery

# jQuery: Overview

- Released January 2006 by John Resig
- Rapid rise in popularity
- Many developers across the globe

# jQuery: Focus

- Improving the interaction between JavaScript and HTML
- Finding elements then performing actions
- Highly-effective, short, code

# jQuery: DOM Support

- DOM Traversing is great - CSS1-3 support and basic XPath:  
`$(“div > p:nth-child(odd)”)`
- DOM Manipulation too:  
`$(“#li”).append(“<li>An item</li>”);`

# jQuery: Events, Effects, and Ajax

- Events are fully functional:  
`$(“div”).click( showItem );`
- And effects:  
`$(“#menu”).slideDown(“slow”);`
- And Ajax!  
`$(“form”).submit(function(){  
 $(“#results”).load(“test.html”);  
});`

# jQuery: Details

- Core features are limited to DOM, Events, Effects, Ajax
- Other features can be added in via plugins

# jQuery: Docs & Community

- Documentation: Complete  
Fully documented, complete API.
- Community: Vibrant  
108+ Mailing List Posts/Day

# jQuery: File Size & Popularity

- File Size: Small  
About 20KB with everything.
- Popularity: Strong  
Users: IBM, Technorati, MSNBC, Digg  
Projects: Drupal, Wordpress

# Mootools

# Mootools: Overview

- Released Sept 2006 by Valerio Proietti
- Created as a fork of Prototype, Base, and moo.fx

# Mootools: Focus

- Designed to fulfill common activities (All the way from low-level iteration to high-level Drag-and-Drop)
- Classical structure (like Prototype)

# Mootools: DOM Support

- Good DOM Selector support:  
`$$("body")`
- Plenty of helper functions for DOM modification:  
`$('#test').appendText("foo");`  
`$('#test').removeClass("foo");`

# Mootools: Events and Effects

- Has traditional event model:

```
$(‘test’).addEvent(‘keydown’, function(){..});
```

- Plenty of visual effects too:

```
var marginChange = new Fx.Style(‘myElement’, ‘margin-top’,{duration:500});  
marginChange.start(10, 100);
```

# Mootools:Ajax

- Ajax support is fully functional:  
`var myAjax = new Ajax(url, {method: 'get'}).request();`

# Mootools: Details

- Lots of plug-and-play features
  - Drag-and-Drop
  - Accordion
  - Scrollbar

# Mootools: Docs & Community

- Documentation: Good  
Good API coverage, some tutorials
- Community: Moderate  
Active forum, hostile against “newbies”

# Mootools: File Size & Popularity

- File Size: Small (approx. 25KB, compressed)  
Has a full compilation system, you can pick-and-choose components.
- Popularity: Rising  
Ubuntu, C|Net

Yahoo! UI

# YUI: Overview

- Released Feb 2006 by Yahoo!
- Maintained and financed internally
- Attempt to standardize internal JavaScript

# YUI: Focus

- Exposing, and solving, common methodologies
- Looking for common idioms (Drag-and-Drop, Calendar, Auto-Complete)

# YUI: DOM Support

- DOM Traversal is minimal, no selectors:  
`YAHOO.util.Dom.get("list")`
- DOM Modification is also minimal, modification only (no DOM creation):  
`YAHOO.util.Dom.addClass("list","hover")`

# YUI: Events and Effects

- Full Event support:

```
YAHOO.util.Event.addListener(  
    "list", "click", function(){  
        alert("List Clicked");  
    });
```

- Animations are also customizable:

```
new YAHOO.util.Anim("list",  
    { width: { from: 10, to: 100 } }, 1,  
    YAHOO.util.Easing.easeOut  
);
```

# YUI:Ajax

- **Ajax connections are simple to setup:**

```
YAHOO.util.Connect.asyncRequest(
    'GET', "test.html", function(){
        alert("File loaded");
    }, null
);
```

# YUI: Details

- Namespaces can become quite verbose. It is recommended that you copy namespaces to other variables.  
`var dom = YAHOO.util.Dom;`
- Closed development (Yahoo employees only, no subversion repository)

# YUI: Docs & Community

- **Documentation: Excellent**  
Everything is thoroughly documented. Lots of explanations, plenty of demos.
- **Community: Moderate**  
Mailing List hovering around 36 posts/day.

# YUI: File Size & Popularity

- **File Size: 40KB (Compressed)**  
Code is quite modular, you can remove unused code easily.
- **Popularity: Possibly Waning**  
Many good bloggers use to talk about it - have moved on.

# Feature Summary

	DOM	Events	Effects	Ajax
Dojo	X	X	X	X
Prototype	X	X	-	X
jQuery	X	X	X	X
Mootools	X	X	X	X
Yahoo UI	/	X	X	X

# More Information

- Dojo: <http://dojotoolkit.org/>
- Prototype: <http://prototypejs.org/>
- jQuery: <http://jquery.com/>
- Mootools: <http://mootools.net/>
- Yahoo! UI: <http://developer.yahoo.com/yui/>

# Ajax Experience Presentations

- Presentations on:
  - Dojo (3)
  - Prototype (3) & Scriptaculous (1)
  - jQuery (3)
- Go to one that sounds interesting and enjoy!