Cleaning your website with Ajax:

Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript

Dion Almaer & Ben Galbraith
Who are we?

- **Dion Almaer**
  - CTO Adigio, Inc
  - TSS Editor
  - JCP Member

- **Ben Galbraith**
  - CTO Amirsys, Inc
  - Utah JUG Leader
  - JCP Member

Ajaxian.com

Dion and Ben's AJaX Mission
Roadmap

- **What is Ajax?**
  - History of rich clients on the web
  - JavaScript Pain
  - XMLHttpRequest (Example)

- Ajax Architecture Pieces
  - Ajax Engine Abstractions
  - Remoting Toolkits
  - UI Toolkits
  - Ajaxian Web Frameworks

- Ajax Usability

- Doing JavaScript Right
  - Actually a nice language!
  - Unit Testing
  - E4X: No more DOM!

- Who is using this now?
  - Google Maps
  - Google Suggest
  - Tadalist
  - Generic use cases

- RSS Reader Demo

- Conclusion / Warnings
JavaScript Pain

- Worked on a dHTML version of WebMail around 1998
  - Was not part of the core business
  - Were able to enforce the browser (IE 4)
    - *Without this, it was impossible*
  - Had a lot of restrictions
  - Some powerful hooks (e.g. data binding)
  - Very tough to use, and frustrating experience

- Learned?
  - Painful to use
  - JavaScript was a quirky language at the time
  - *Should have kept working on that to create GMail!*
This lead to...

The Myth

Compelling rich UI’s can not be delivered with browser technology
Defining Ajax: It’s no soccer team or cleaning supplies...

Ajax 2 – FC Thun 0
Defining Ajax

**Ajax**

- standards-based presentation using XHTML and CSS
- dynamic display and interaction using the Document Object Model
- data interchange and manipulation using XML and XSLT
- asynchronous data retrieval using XMLHttpRequest
- and JavaScript binding everything together

Core “New” Technology
Ajax Client Model

- No more single request/response restrictions
  - Have flexibility to go back to the server when we want!

Server Events Everywhere

- **Web 1.0**
  - GET via links on text and images
  - POST via SUBMIT buttons and type="image"

- **Web 2.0**
  - All of the above plus
  - Blur
  - Key press
  - Hover
  - Sneezes? 😁

“It’s the architecture, Stupid!” 😊
XMLHttpRequest

- **Asynchronous Processing is here**
  - Allows you to kick off an HTTP request in the background
  - Callbacks kick back into the JavaScript code
  - Supported in various browsers

- **Why the big deal?**
  - Changes the request/response paradigm...
XMLHttpRequest: Example

- Asynchronously update fields
XMLHttpRequest:
Working With Browsers

● **Browser Dependent Code**
  
  • The old problems never die 🙁

```javascript
function getHTTPObject() {
    var xmlhttp;
    /*@cc_on
        @if (@_jscript_version >= 5)
            try {
                xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
            } catch (e) {
                xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (E) {
                xmlhttp = false;
            }
        @else
            xmlhttp = false;
        @end @*/
    if (!xmlhttp && typeof XMLHttpRequest !== 'undefined') {
        try {
            xmlhttp = new XMLHttpRequest();
        } catch (e) {
            xmlhttp = false;
        }
    }
    return xmlhttp;
}
```

● **Cross Browser XMLHttpRequest Code:**
  
  • Supports: IE, Opera (Gecko built in!)
XMLHttpRequest: Make the XMLHttpRequest

- Handle the response from the server
  - Setup the HTTP request URL
  - Configure the callback for the response
  - Send the async message

```javascript
function updateCityState() {
  if (!isWorking) {
    var zipValue = document.getElementById("zip").value;
    xmlhttp.open("GET", url + escape(zipValue), true);
    isWorking = true;
    xmlhttp.onreadystatechange = handleHttpResponse;
    xmlhttp.send(null);
  }
}
```

<input size="5" name="zip" id="zip" onblur="updateCityState();" type="text">
XMLHttpRequest: Handling HTTP Response

- **Handle the response from the server**
  - State checking
  - Getting response text
  - Handling XML DOM returned

```javascript
function handleHttpResponse() {
    if (xhr.readyState == 4) {
        isWorking = false;
        if (xhr.responseText.indexOf('invalid') == -1) {
            var xmlDoc = xhr.responseXML;
            var city = xmlDoc.getElementsByTagName('city').item(0).firstChild.data;
            var state = xmlDoc.getElementsByTagName('state').item(0).firstChild.data;
            document.getElementById('city').value = city;
            document.getElementById('state').value = state;
        }
    }
}
```
## XMLHttpRequest: Methods

### Methods available on XMLHttpRequest

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>open(&quot;method&quot;, &quot;URL&quot;[, asyncFlag[, &quot;userName&quot;[, &quot;password&quot;]]])</code></td>
<td>Setup a request (note async)</td>
</tr>
<tr>
<td><code>send(content)</code></td>
<td>Send the request (content = post data)</td>
</tr>
<tr>
<td><code>abort()</code></td>
<td>Stop any request</td>
</tr>
<tr>
<td><code>getAllResponseHeaders()</code></td>
<td>Return a Map of headers</td>
</tr>
<tr>
<td><code>getResponseHeader(&quot;header&quot;)</code></td>
<td>Return the header value</td>
</tr>
<tr>
<td><code>setRequestHeader(&quot;label&quot;, &quot;value&quot;)</code></td>
<td>Set headers</td>
</tr>
</tbody>
</table>
XMLHttpRequest: Properties

### Properties set on XMLHttpRequest

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onreadystatechange</td>
<td>Setup the callback event handler</td>
</tr>
<tr>
<td>readyState</td>
<td>Object status integer:</td>
</tr>
<tr>
<td></td>
<td>0 = uninitialized</td>
</tr>
<tr>
<td></td>
<td>1 = loading</td>
</tr>
<tr>
<td></td>
<td>2 = loaded</td>
</tr>
<tr>
<td></td>
<td>3 = interactive</td>
</tr>
<tr>
<td></td>
<td>4 = complete</td>
</tr>
<tr>
<td>responseText</td>
<td>Text value of response</td>
</tr>
<tr>
<td>responseXML</td>
<td>XML DOM of response</td>
</tr>
<tr>
<td>status</td>
<td>Status code (numeric)</td>
</tr>
<tr>
<td>statusText</td>
<td>Status text</td>
</tr>
</tbody>
</table>
 Ajax Return Approaches

**Returning Data**

- Your own format / XML
- Could have the server in web service mode, returning clean XML
- The JavaScript client will then use responseXML, which gives an XML node back which you can walk with DOM and E4X
- Or you could return back your own strings

**Returning HTML**

- Use “partial” content on the server side
- JavaScript calls the controller to pass back a partial piece of HTML
- Inject the HTML via `innerHTML`

**Returning JavaScript**

- `eval()`
- Have the server responsible for returning JavaScript code
  - The JavaScript client will `eval(http.responseText)` the response

*Ruby on Rails*
Is Flash Ajax?

- A common question/debate surrounds Flash, and its role

- Is Flash Ajax itself?
  - It does allow you to follow a similar architecture
  - ActionScript is even *almost* JavaScript

- Our View:
  - Flash is a fantastic tool in the toolbox for developers
  - Now, you can bridge JavaScript from a HTML page, with ActionScript in a Flash component
  - Some components are impossible, or very hard to implement with HTML/CSS/JS, and we can fall back to Flash
Apple readies AirPort 4.0 software for AirPort Express launch

Apple Computer, Inc. is rapidly completing the development of AirPort 4.0, a major upgrade to its wi... Read more...

An Inside look at Safari 1.3 Developer Preview

Exclusive - Apple Computer seeded developers this week with Safari 1.3 Developer Preview 1, which is... Read more...

Mac OS X Server "Tiger" features Xgrid, iChat and Blog server

Following the first public demonstration of Mac OS X 10.4 Tiger yesterday, Apple quietly previewed t... Read more...

Apple seeds Safari 1.3 DP1, Java 1.5 DP to developers

Apple Computer seeded developers this week with Safari 1.3 Developer Preview 1, which will run on lo... Read more...

Apple introduces displays, Mac OS X 10.4 Tiger

The attendees began taking their seats, the lights dimmed, and the Beatles played in the background... Read more...

Apple likely to replay WWDC keynote with delayed webcast [Upd...

Apple Computer, Inc. will most likely provide a delayed webcast of its World Wide Developers Confer... Read more...

Screenshots of Mac OS X 10.4 Tiger?

Within 24 hours Apple Computer is expected to unveil the majority of its plans for the next major re... Read more...

Hon Hai to manufacture Apple's AirPort Express

Hon Hai Industrial Co., Ltd., Taiwan's leading manufacturer of Information technology products, has ... Read more...

Run script "setDateFilter('1',110250801,null)"
Roadmap

- What is Ajax?
- **Ajax Architecture Pieces**
  - Ajax Engine Abstractions
  - Remoting Toolkits
  - UI Toolkits
  - Ajaxian Web Frameworks
- Ajax Usability
- Doing JavaScript Right
- RSS Reader Demo
- Conclusion / Warnings
Ajax Architecture

Ajaxian Web Frameworks
Rails, Tapestry, WebWork, ASP.NET, ...

UI Toolkit
Dojo, SmartClient, Backbase, Scriptaculous...

Remoting Toolkit
DWR, JSON-RPC, dojo.io.bind(), Prototype

JavaScript Utilities & Tools

XMLHttpRequest
iframe
...
Dojo Toolkit

- The Dojo Toolkit has a nice Ajax abstraction
  - Handles the back button
  - Abstraction above XMLHttpRequest / iframes / etc
  - Error handling

```javascript
function divChange() {
    var divToChange = document.getElementById("divToChange");
    var lastContent;

    dojo.io.bind({
        url: "/samples/dojoSampleText.txt",
        load: function(type, data, evt) {
            divToChange.innerHTML = data;
        },
        error: function(type, error) { alert(error) },
        backButton: function() {
            lastContent = divToChange.firstChild.nodeValue;
            divToChange.innerHTML = 'before change.';
        },
        forwardButton: function() {
            divToChange.innerHTML = lastContent;
        },
        changeURL: true,
        mimetype: "text/plain"
    });
}
/* and there’s more optional attributes:
transport: XMLHttpRequest
method: <http method>
content: { key1: 'value1', ... }
sync: true || false
useCache: true || false
formNode: document.getElementById('yourForm') */
```
Prototype

- Prototype offers simple remoting too
  - Tight integration with Rails, but works with anything
  - Has some effects
  - Nice additions to your JavaScript
    - e.g. `$('foo')`

```javascript
function getZipData(zipCode) {
    new Ajax.Updater("rewrite", url, {
        asynchronous: true,
        method: "get",
        parameters: "zip=" + zipCode + ",type=html",
        onSuccess: function(request) {
            new Effect.Highlight('rewrite');
        },
        onFailure: function(request) {
            assignError(request.responseText);
        }
    });
}

/*
    onLoading: function(request) {Element.show(fieldname + 'Progress');},
    onComplete: function(request) {Element.hide(fieldname + 'Progress');}
*/
}

function assignError(error) {
    $('zipError').innerHTML = "Error: " + error;
}
```
DWR: Direct Web Remoting

- DWR is a way of calling Java code on the server directly from Javascript in the browser.
- Speaks to a web app
- Helpful abstractions for the Java and JavaScript side

**Config: dwr.xml**

```xml
<dwr>
  <allow>
    <create creator="new" javascript="JDate">
      <param name="class" value="java.util.Date"/>
    </create>
    <create creator="new" javascript="Demo">
      <param name="class" value="your.java.Bean"/>
    </create>
  </allow>
</dwr>
```

**HTML**

Credit Card #:
```html
<input id="cardnum" type="text" onchange="checkCardNum()"/>
<span id="cardnum-error" class="error"></span>
```

**JavaScript**

```javascript
function checkCardNum() {
  CreditCardValidator.isValid(replyCardNum, DWRUtil.getValue("cardnum"));
}
```

**Servlet: web.xml**

```xml
<servlet>
  <servlet-name>dwr-invoker</servlet-name>
  <servlet-class>uk.ltd.getahead.dwr.DWRServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>dwr-invoker</servlet-name>
  <url-pattern>/dwr/*</url-pattern>
</servlet-mapping>
```
JSON: JavaScript Object Notation

- JSON (JavaScript Object Notation) is a lightweight data-interchange format.
- Bindings for Java, JavaScript, Ruby, Perl, C, ...
- No parsing required on the JavaScript side
- Built on:
  - Object, Array, String, Number, Boolean, null

```json
[
  {
    "name": "Dion Almaer",
    "age": 12,
    "pets": ["sandy", "fluffy"]
  },
  {
    "name": "Ben Galbraith",
    "age": 11,
  }
]
```

The X in Ajax: (Not just XML)

HTML
JavaScript / JSON
SVG?
JSON-RPC

- Think XML-RPC but with JSON instead of XML 😊
- Java ORB available

```
function onLoad() {
    jsonrpc = new JSONRpcClient("/jsonrpc/JSON-RPC");
}

function clickHello() {
    var whoNode = document.getElementById("who");
    var result = jsonrpc.hello.sayHello(whoNode.value);
    alert("The server replied: " + result);
}
```
UI Toolkit

- JavaScript Widgets have been around for a long time:
  - ActiveWidgets (http://www.activewidgets.com/grid/)
  - Backbase (http://www.backbase.com)
  - SmartClient (http://www.smartclient.com)
  - Dojo (again) (http://dojotoolkit.org)
UI Toolkit

- Low level effects have been encapsulated in libraries:
  - Scriptaculous:
    - [http://script.aculo.us/visual-effects](http://script.aculo.us/visual-effects)
    - Builds on Prototype effects
  - Drag and Drop encapsulated for us

```javascript
new Effect.Fade('id_of_element');
new Effect.Fade('id_of_element',
    { transition: Effect.transitions.wobble });
```

```text
this is the first list
- Item 1 from first list.
- Item 2 from first list.
- Item 3 from first list.

and now the second list
- [DRAG HERE] Item 1 from second list.
- [DRAG HERE] Item 2 from second list.
- [DRAG HERE] Item 3 from second list.
```
Ajax Frameworks: Another Look

**Javascript: Application Frameworks**
AJFORM, BackBase, Bindows, Dojo, Isomorphic, OpenRico, qooxdoo, ThyAPI, Tibet

**Javascript: Infrastructural Frameworks**
AjaxCaller, Flash JavaScript Integration Kit, Google AJAXSLT, HTMLHttpRequest, Interactive Website Framework, LibXMLHttpRequest, MAJAX, RSLite, Sack, Sarissa, XHConn

**Server-Side: Multi-Language**
Cross-Platform Asynchronous INterface Toolkit, SAJAX, Javascript Object Notation, Javascript Remote Scripting, Bitkraft for ASP.NET

**Server-Side: Java**
WebORB for Java, Echo 2, WidgetServer, Direct Web Remoting, SWATO, AJAX JSP Tag Library, AJAX JSF Framework, ThinkCAP JX, Struts-Layout, Tapestry Tacos, Wicket, WebWork

**Server-Side: .Net**
MonoRail, WebORB for .NET, Ajax.NET, ComfortASP.NET, AjaxAspects, Microsoft Atlas

**Server-Side: PHP**
AjaxAC, JPSpan, XAJAX, PEAR::HTML::Ajax, CPAINT
JavaScript Utilities

- There are many tools that you need to make life barable in JavaScript land:
  - **JavaScript Munging**
    - Jazmin / JavaScript Minifier (http://www.crockford.com/javascript/jsmin.html)
    - Obfuscator (http://www.shanenq.bobos.ca/#jso_description)
    - ShrinkSafe (http://alex.dojotoolkit.org/shrinksafe)
  - **Debugging**
    - Ajax Debugging Tool (http://blog.monestuff.com/archives/000252.html)
    - IE Leak Detector (http://blog.monestuff.com/archives/000252.html)
    - DOM Inspector (built into Mozilla, need to grab IE plugins)
    - TrimBreakpoint (http://trimpath.com/project/wiki/TrimBreakpoint)
  - **XML Munging**
    - Google AJAXSLT (http://goog-ajaxslt.sourceforge.net)
    - JSXML (http://sourceforge.net/projects/jsxml)
    - E4X
  - **Misc**
    - JSDoc (http://jsdoc.sourceforge.net/)
    - JsLint (http://www.crockford.com/javascript/lint.html)
  - **Unit Testing**
    - JsUnit (http://sourceforge.net/projects/jsunit)
    - Selenium (http://selenium.thoughtworks.com)
JavaScript Utilities: Dojo, please come back

- This is another area where full frameworks like Dojo come in nice:
  - **dojo.io**
    - The remoting package, that makes Ajax requests easy.
  - **dojo.event**
    - Browser compatible event system.
  - **dojo.lang**
    - Support for mixins, and object extension.
      - *Prototype does a great job here too*
  - **dojo.graphics**
    - Support for nifty HTML effects (e.g. fadeIn/Out, slideTo/By, explode/implode, etc).
      - *Prototype and Scriptaculous do a great job here too*
  - **dojo.dnd**
    - Drag and Drop support.
  - **dojo.animation**
    - Create animation effects.
  - **dojo.hostenv**
    - Support for JavaScript packages.
  - **widgets**
    - Great widgets such as the rich text editor!
JsUnit:
Testing in the client

- The Test First/Driven folks will be happy to know that they can write a few thousand lines of unit test code for their JavaScript too

```javascript
<script language="javascript" src="jsUnitCore.js"></script>
<script language="javascript" src="myJsScripts.js"></script>

<script language="javascript">
  function testWithValidArgs() {
    assertEquals("2 times 3 plus 5 is 11", 11, multiplyAndAddFive(2, 3));
    assertEquals("Should work with negative numbers", -15, multiplyAndAddFive(-4, 5));
  }
  function testWithInvalidArgs() {
    assertNull("A null argument should result in null", multiplyAndAddFive(2, null));
    assertNull("A string argument should result in null", multiplyAndAddFive(2, "a string"));
  }
  function testStrictReturnType() {
    assertEquals("Should return a number, not a string", "11", multiplyAndAddFive(2, 3));
  }
  function testWithUndefinedValue() {
    assertNull("An undefined argument should result in null", multiplyAndAddFive(2, JSUNIT_UNDEFINED_VALUE));
  }
</script>
```

Selenium:
- New opensource project
- Test runner in your browser
Roadmap

- What is Ajax?
- Who is using this now?
- Ajax Approaches

**Ajax Usability**
- Graceful failback
- Better UI

- JavaScript Right
- RSS Reader Demo
- Conclusion

http://www.1976design.com/blog/
Graceful Failback?

- There is a lot of discussion on handling graceful failback
  - As always, it is a balance!
  - Are you using Ajax to do a few subtle things?
  - Are you building a crazy rich framework?

Some helpful techniques

- Sam Pullara has an interesting solution to try to help out
  - XPath + Ajax
- CSS Selectors
  - http://ripcord.co.nz/behaviour/
- HTTP 204

BAD

```html
<li>
  <a onclick="this.parentNode.removeChild(this)" href="#">Click me to delete me</a>
</li>
```

BEETTER

```javascript
var myrules = {
  '#example li': function(el) {
    el.onclick = function() {
      this.parentNode.removeChild(this);
    }
  }
};
Behaviour.register(myrules);
```
Better Usability: Yellow Fade Technique

- This is an example of how you can make better usability

37 Signals pattern:
Denoting Server Access

- Web users are trained to expect server access at certain points.
- To help usability we need to give feedback when a piece of the page is hitting the server.
Roadmap

- What is Ajax?
- Ajax Architecture Pieces
- Ajax Usability
- Doing JavaScript Right
  - The language
  - Treating it right
- Who is using this now?
- RSS Reader Demo
- Conclusion / Warnings
JavaScript isn’t as bad as you think

- JavaScript got a bad rap
  - Buggy implementations
  - Copy ‘n paste reuse
  - Web hacking versus “programming”

- JavaScript is OO
  - Prototype-based OO versus class based
  - Everything is an object
  - Even functions

- Spend the time to grok it

```javascript
Number.prototype.toColorPart = function() {
  var digits = this.toString(16);
  if (this < 16) return '0' + digits;
  return digits;
}
```
Treating JavaScript as a first class citizen

- JavaScript doesn’t get treated like our other programming assets on projects

- We need to change this:
  - Testing matters
  - Having releases of JavaScript matters

- I even created a Maven plugin that lets me define JavaScript dependencies

```xml
<dependency>
  <groupId>ajaxian</groupId>
  <artifactId>ajaxian-permissions</artifactId>
  <version>0.1</version>
  <type>js</type>
</dependency>
```
ECMAScript for XML (E4X)

- **Sick of DOM?**

```javascript
xmlDocument.getElementsByTagName('city').item(0).firstChild.data;
```

- **ECMAScript for XML == first class citizenship for XML**

- **Create a DOM**

```javascript
var order = <order>
    <customer>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
    </customer>
    <item>
        <description>Big Screen Television</description>
        <price>1299.99</price>
        <quantity>1</quantity>
    </item>
</order>
```

- **Walk the XML tree a la XPath**

```javascript
// Construct the full customer name
var name = order.customer.firstname + " " + order.customer.lastname;

// Calculate the total price
var total = order.item.price * order.item.quantity;
```
JavaScript Templates

- Now the browser is doing a lot of work... how about using templates?

```html
Hello ${customer.firstName|default:"Mr."} ${customer.lastName}<br/>
Your recent orders are:
<ul>
  {for order in customer.getRecentOrders()}
    <li>${order.orderNum} - ${order.status|capitalize}:
        ${order.grandTotal}
        {if order.requestedShipDate< new Date().getTime() && order.status!= "shipped"}
            Please contact customer service.
        {/if}
    </li>
  {forelse}
    <li>You have no orders yet. </li>
  {/for}
</ul>
```
AOP?

- Now we know that JavaScript is cool ;)

```javascript
Aspects = new Object();

Aspects.addBefore = function(obj, fname, before) {
    var oldFunc = obj[fname];
    obj[fname] = function() {
        return oldFunc.apply(this, before(arguments, oldFunc, this));
    };
};

Aspects.addAfter = function(obj, fname, after) {
    var oldFunc = obj[fname];
    obj[fname] = function() {
        return after(oldFunc.apply(this, arguments), arguments, oldFunc, this);
    };
};

Aspects.addAround = function(obj, fname, around) {
    var oldFunc = obj[fname];
    obj[fname] = function() {
        return around(arguments, oldFunc, this);
    };
};

Aspects.addBefore(this, "makeGreeting", aopizeAdvice);
alert(makeGreeting("world")); // Hello AOP world!

Aspects.addAfter(this, "makeGreeting", shoutAdvice);
alert(makeGreeting("world")); // HELLO AOP WORLD!

Aspects.addAround(this, "makeGreeting", ciaoAdvice);
alert(makeGreeting("world")); // Bye-bye!
</script>
```
SOAP Access?

```xml
search="Word"
xmllhttp.open("POST", "http://api.google.com/search/beta2",true);
xmllhttp.onreadystatechange=function() {
    if (xmllhttp.readyState==4) {
        alert(xmllhttp.responseText)
    }
}
xmllhttp.setRequestHeader("Man", "POST http://api.google.com/search/beta2 HTTP/1.1")
xmllhttp.setRequestHeader("MessageType", "CALL")
xmllhttp.setRequestHeader("Content-Type", "text/xml")
xmllhttp.send("<?xml version='1.0' encoding='UTF-8'?>"+"\n\n"+"<SOAP-ENV:Envelope">' xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"+
' xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"'+
' xmlns:xsd="http://www.w3.org/1999/XMLSchema">'+'</ns1:doGoogleSearch>'+'</SOAP-ENV:Body>"'+
'xmlns:ns1="urn:GoogleSearch"'+
' SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">'+'</ns1:doGoogleSearch>'+
' </SOAP-ENV:Body></SOAP-ENV:Envelope>")
```

● REST fits in nicely 😊 ● SOAP is possible :/
JavaScript Limitations: Signing Code

- Limitation: “Can’t request other resources on the web”
  - WRONG: You can, as long as you sign your code
  - Mozilla is anal about this security
  - IE? Not so much 😊

- Limitation: “But I want to read from the file system!”
  - WRONG: You can, as long as you sign your code

- Limitation: “But I want to write to the file system!”
  - WRONG: You can, as long as you sign your code
JavaScript Limitations: Ugly Security Code

- Need to get permission in every function
  - Mozilla reverts your priv when it comes out of the function

```javascript
try {
    if (netscape.security.PrivilegeManager.enablePrivilege) {
        netscape.security.PrivilegeManager.enablePrivilege('UniversalBrowserRead');
    }
} catch (ex) { // eat it
}

function applyPrivilege(privilege, functionCallback) {
    try {
        if (netscape.security.PrivilegeManager.enablePrivilege) {
            netscape.security.PrivilegeManager.enablePrivilege(privilege);
        }
    } catch (ex) { // eat it
    }
    functionCallback();
}
```
Roadmap

• What is Ajax?
• Ajax Architecture Pieces
• Ajax Usability
• Doing JavaScript Right
• **Who is using this now?**
  • Google Maps
  • Google Suggest
  • Tadalist
  • General Use Cases
• RSS Reader Demo
• Warnings
Google Maps

- Map Pane
- Push Pins
- Shadows
- Scrolling
Google Maps: Placing items

- Images absolutely placed
- 8 bit alpha channel PNGs
- Shadows
  - Two images
- Scrolling
  - 128x128 panels
  - Infinite scroller
Google Maps: Talking to Server

- Google Maps keeps talking back to get more panes
  - Fixed URL for caching
    
    http://mt.google.com/mt?v=.1&x={x tile index}&y={y tile index}=2&zoom={zoom level}

- Google Maps also talks back to do searches via an iframe
  - Why an iframe? Back button saver
    
    http://maps.google.com/maps?q=atlanta&z=13&sll=37.062500%2C-95.677068&sspn=37.062500%2C80.511950&output=js

    ```javascript
    function load() {
      if (window.parent && window.parent._load) {
        window.parent._load({big chunk of XML}, window.document);
      }
    }
    ```
Google Maps: Becomes a platform

- When you get into this world, you are building a platform
- If you are lucky, there will be a <INSERT PROJECT> Hacks book on its way
- Now your users, are no longer passive consumers of your service, but they are actively developing on top of it!

http://www.housingmaps.com
http://chicagocrime.org
http://libgmail.sourceforge.net

- Bookmarklets go crazy!

http://weblog.infoworld.com/udell/2005/02/28.html#a1186
Live Search: Google Suggest

- Aim: Let’s you know what people are searching
- Helps with spelling mistakes
- Gets you to popular content
- Uses XHR to call back to server and populates drop down div

```javascript
eval(http.responseText)
```
TadaList

- Todo List manager
- Dynamically moves between states
- Easy in place adding
- Doesn’t have in place editing
TadaList: Toggle Time

```javascript
function toggle_item(checkbox, item) {
    xml_request(toggle_url + item.id + '?completed=' + (checkbox.checked ? 1 : 0));
    switch_lists(item, checkbox.checked);

    if (movement_controls_on) { toggleMovementControls(); } changed_list = true;

    toggle_reorder_link(checkbox.checked);
    toggle_delete_link(checkbox.checked);

    item.children[1].id = 'fade';
    fadeIn(7);

    if (parent_table_of_a_row(item).id == "completed_items") { checkbox.checked = true;
    } else {
        checkbox.checked = false;
    }
}
```
General Solutions

There are many areas in which Ajax fits in, including:

- Generally: CRUD on the DOM
- Feedback messages
- Admin Features
- Updating Calculations
- Portal
Feedback messages

- Whenever you want to update something on the server and just say “oh, I did it”, why re-render the page that you are on if you don’t need too?
Deleting from the Admin Panel

- When you move something off of your pane (via move/delete) why re-render when you can just delete that element from the DOM?
Recalculate

- Why not re-calculate on the fly?
- Why re-render from the server?

Note: there may be valid reasons!

amazon.com
Portal / Componentized

- In a portal, really dealing with separate lifecycles for each component.

- Now each component can truly handle its own work without interfering with other pieces.
Roadmap

- What is Ajax?
- Ajax Approaches
- Doing JavaScript Right
- Ajax Usability
- Who is using this now?
- **RSS Reader Demo**
- Conclusion / Warnings
RSS Bling

Ajaxian RSS Bling: Dynamic RSS Reader

/\ Hawkins
The BlueBlog
Bloglines | News
James Gosling on the Java road
James Strachan's Weblog
Martin Fowler's Blik
O'Reilly, Cedric's weblog
Set-Sense

techno.blog("Dion")
blogging about life, the universe, and everything tech

Ansi SQL-AOP 2015 Released

A new ANSI-SQL standard has been released, named SQL-AOP 2015. The aim of this sub-spec is to take SQL to the r and enable AOP support. Now, your DBA can package cross-cutting concerns in "stored aspect libraries" in the database. The features include: Rich pointcut language for SQL Capture before/after around advice on any SQL statement Add DML, DDL, or other code Force caching semantics based on AOP policy advice Logging TX semantics Security fill (including instance based) You can capture rich semantics such as: Before the trigger named "foo" causes a change to make sure that you update the "Load up the cache for tables x, y, and z, when a SELECT captures fields 11, and 12, on t Larrin Ellison was quoted as saying: SQL-AOP is going to revolutionize the world of computing. The database is the net. Finally, my vision of the network computer can come to fruition and I can finally make fun of my good friend Steve Jobs. developer at TheServerSide symposium responded with: "Finally, I can get rid of the logging code that I scatter in my PL scripts. This is a godsend for database computing."

Adobe Open Source

It is great to see more companies come along with open source contributions a la http://opensource.adobe.com. The release revolves around Adam and Eve. Adam is a modeling engine and declarative language for describing constrain relationships on a collection of values, typically the parameters to an application command. When bound to a human int (Hi) Adam provides the logic that controls the Hi behavior. Adam is similar in concept to a spreadsheet or a forms man. Values are set and dependent values are recalculated. Adam provides facilities to resolve interrelated dependencies a track those dependencies, beyond what a spreadsheet provides. Eve consists of a declarative language and layout construct an Hi. The layout engine in Eve takes into account a rich description of UI elements to achieve a high level of maintaining what can be achieved with manual placement. A single Hi description in Eve suffices for multiple OS platforms and languages. This document describes Eve2, the latest version of Eve. Eve2 was developed to work with Adam and to incremental improvements that have been completed since Eve1 was written. It is important to note that Adam and Eve do not
Roadmap

- What is Ajax?
- Who is using this now?
- Ajax Approaches
- Doing JavaScript Right
- Ajax Usability
- RSS Reader Demo

**Conclusions / Warnings**
  - Issues we found
Conclusion

- Don’t use for the technologies sake
- Understand usability testing
  - We have been trained with the SUBMIT button
  - Can’t redefine the way a UI works, has to be natural
- Understand restrictions
  - Implicit JavaScript restrictions
  - Programming constructs that you wish you had
- Still in creative mode
  - Many frameworks coming out there
  - Will take time to work out the better solutions
- Still learning
ajaxian.com & Pragmatic Ajax

- Ajax is changing at a rapid rate
- Ben and I try to keep you up to date at

ajaxian.com

Dion and Ben's Ajax Mission

- Pragmatic Ajax
  - Beta Book released now
  - Give us feedback on what you would like in the book
  - Discount code for OSCON attendees: EuroAjax