Ajax 101

Where we get down and dirty with Ajax operations, JavaScript events and the DOM

Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

Anatomy of a Pattern

- Ajax design patterns contain three steps
  - **Trigger** (event or timer)
  - **Operation** (Ajax, remote scripting)
  - **Update** (presentation)

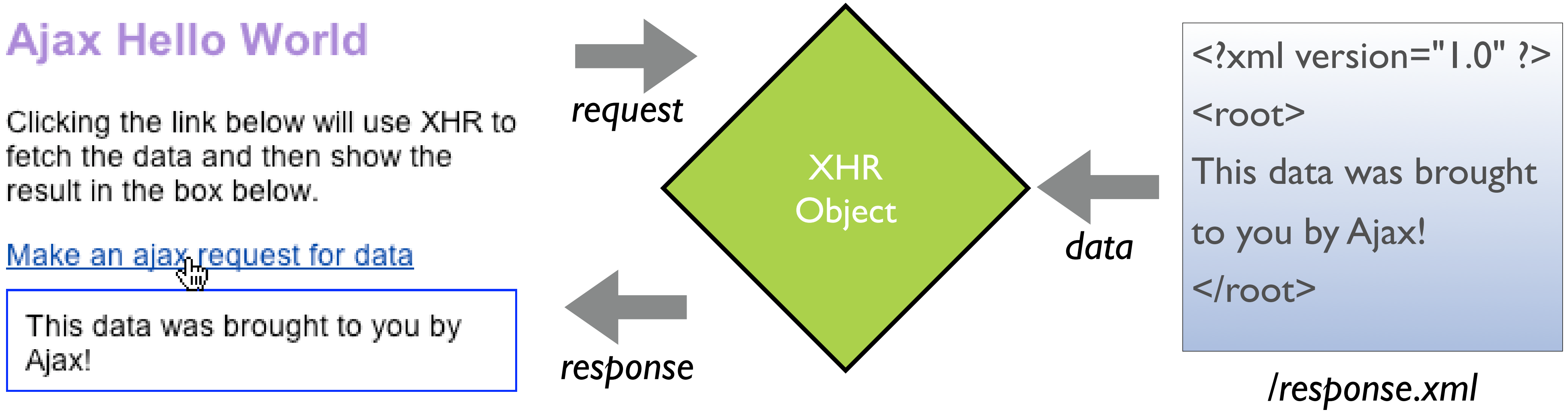
Trigger
Operation
Update

Operation. Using XHR

Operation. Using XHR

- The five operations are not built into XHR
- The simple send/response mechanism can be used to implement lookup, persist, validate, invoke and message
- To create these operations, one must understand how to use XHR
- A simple HelloWorld example will illustrate using XHR

Simple Ajax 'Hello World'



Ajax Hello World

Clicking the link below will use XHR to fetch the data and then show the result in the box below.

Make an ajax request for data

This data was brought to you by Ajax!

```
<?xml version="1.0" ?>
<root>
This data was brought
to you by Ajax!
</root>
```

- Clicking the link makes an XHR request
- Response is inserted into the area outlined in blue

Ajax How To

1. Create a request object
2. Write a callback
3. Make the request
4. Parse the response

1. Create a Request Object

```
if browser is mozilla or safari or opera then
   create a new XMLHttpRequest

otherwise it is IE then
   create a new ActiveXObject
otherwise
   error - browser does not support XMLHttpRequest
```

- IE5+ implements XHR as an ActiveX object
- Mozilla 1.0+, Safari 1.2+, Opera 8+, IE7 provide an XMLHttpRequest object in their API
- All XHR objects have the same methods & properties

XHR Methods

Method	Description
open(“method”, “url”, [, asynchFlag [, “username” [, “password”]]])	Sets up the request object for sending a request
send(content)	Sends the request to the server. Can be null.
abort()	Stops the request.
getAllResponseHeaders()	Returns all response headers for the HTTP request as key/value pairs.
getReponseHeader(“header”)	Returns the string value of the specified header.
setRequestHeader(“header”, “value”)	Sets the specified header to the supplied value.

Source: Foundations of Ajax - APress

XHR Properties

Property	Description
onreadystatechange	The event handler that fires at every state change.
readystate	The state of the request: 0=uninitialized, 1=loading, 2=loaded, 3=interactive, 4=complete
responseText	The response from the server as a text string
responseXML	The response from the server as an XML document
status	The HTTP status code from the server for the request object: 200: Ok; 201: Created; 400: bad request, 403: Forbidden; 500: internal sever error
statusText	The text version of the HTTP status code

2. Write a Callback

```
function handleAjaxResponse
begin
   do something with the data that is returned from XHR
end
```

- JavaScript function is invoked when the readystate changes on the XHR object

3. Make the Request

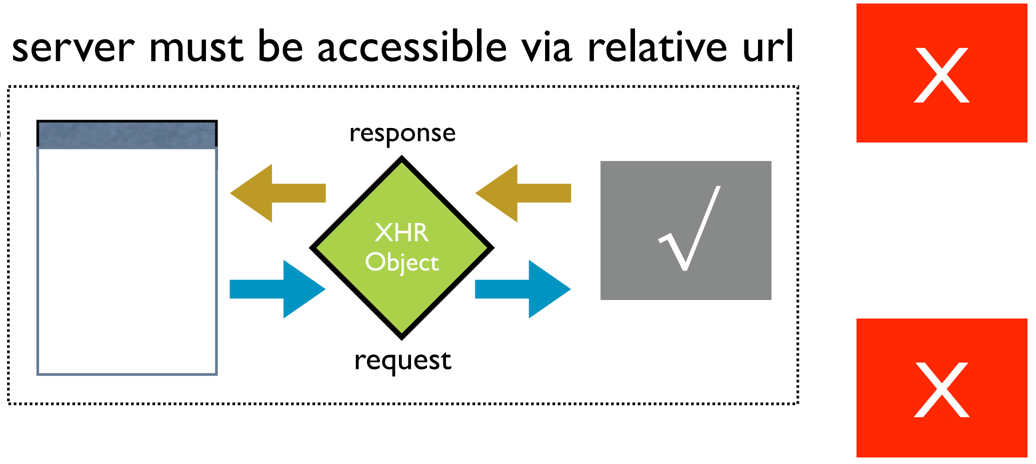
Ajax
XHR

```
set onreadystatechange to callback function - handleAjaxResponse

open a request on the xhr object

send the request through the xhr object
```

- The JavaScript function *getResponse* will be invoked when the readystate property changes on the XHR object
- Same site rule
- ‘GET’ or ‘POST’
- Asynchronous flag



3. Make the Request - Choosing Between GET and POST

- Use **GET** for
  - For retrieve
  - REST services
  - When passing parameters
  - Idempotent URLs
  - Small amount of data
- Use **POST** for
  - Modification
  - Large amounts of data passed to server
  - Non-idempotent URLs

3. Make the Request: Handling the Request on the Server Side

- Its just a normal HTTPRequest
  - Normal mechanism for getting request parameters
- Raw POST (xhr.send(someData))
  - Java/JSP: request.getInputStream() - read as raw post
  - Rails: @request.raw_post
  - PHP: $data = file_get_contents('php://input')

4. Parse the Response

```
function handleAjaxResponse
begin
   if response is valid then
       get responseXML
       get rootNode
       get helloArea on the page
       stuff the rootNode value into the helloArea DIV
   endif
end
```

- **readystate** values

  0 - Uninitialized

  1 - Loading

  2 - Loaded

  3 - Interactive

  4 - Completed

4. Parse the Response: Formatting an XML Response on the Server

- XML version must be first line

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- Set up response header's Content-type

```
“Content-Type”, “text/xml”
```

- Use well-formed XML

4. Parse the Response: Handling an XML Response on the Client

Ajax
XHR

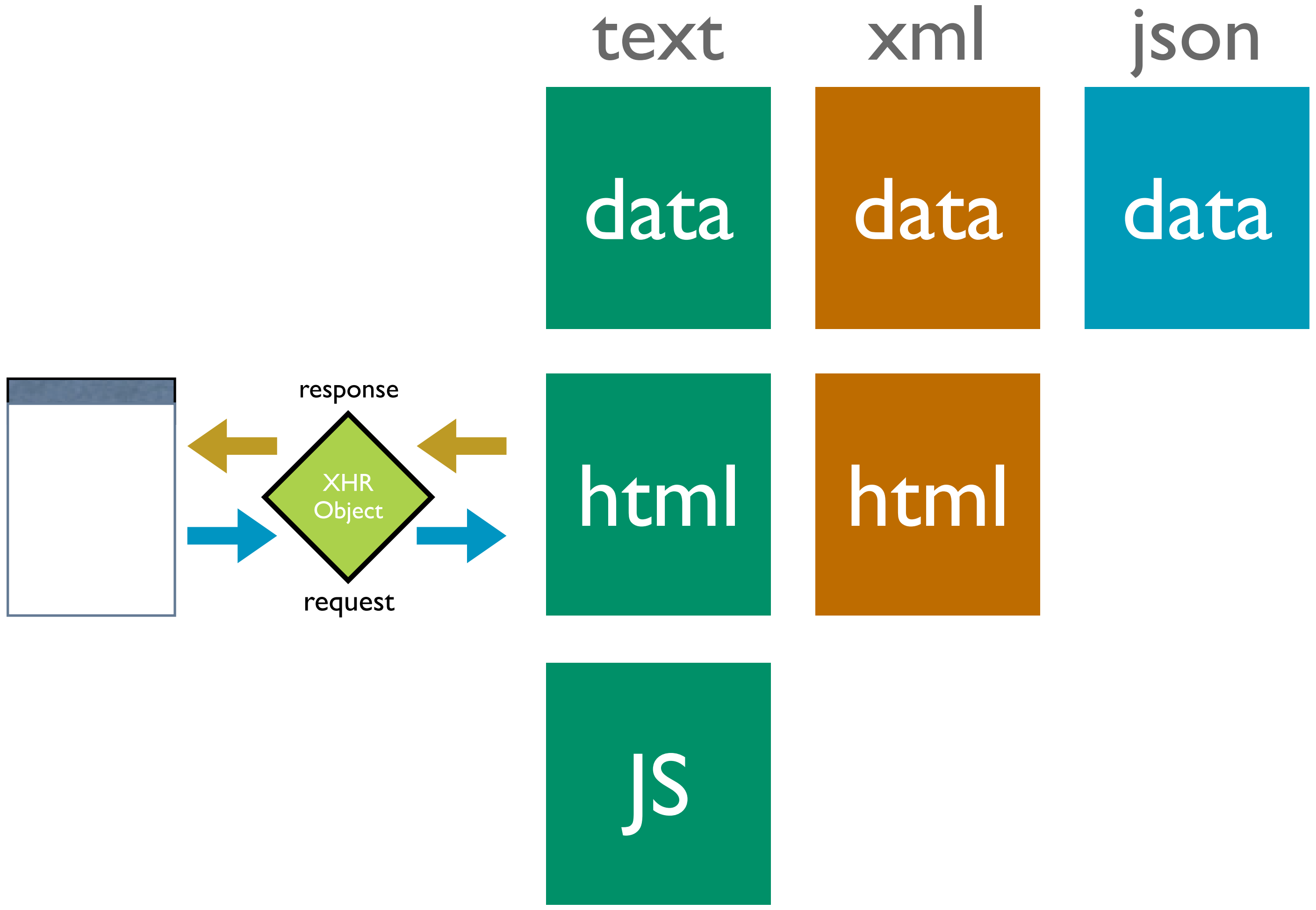
- Use XHR property **responseXML** to get the resp
  as an XML DOM (XmlDocument)
- Use standard JavaScript DOM methods
  - Mozilla, Safari, Opera & IE support a common set of methods and properties
  - Watch out for IE only stuff (e.g., **children** property)
  - Watch out for whitespace issues
- Other options
  - Use XML library for JS (e.g., XML for <script>)

4. Parse the Response: Xml Document API

Property/Method	Description
documentElement	Returns the root element of the document
firstChild	Is the first element within another element (the first child of the current node)
lastChild	Is the last element within another element (the last child of the current node)
nextSibling	Is the next element in the same nested level as the current one
previousSibling	Is the previous element in the same nested level as the current one
nodeValue	Is the value of a document element
getElementsByTagName	Used to place all elements into an object

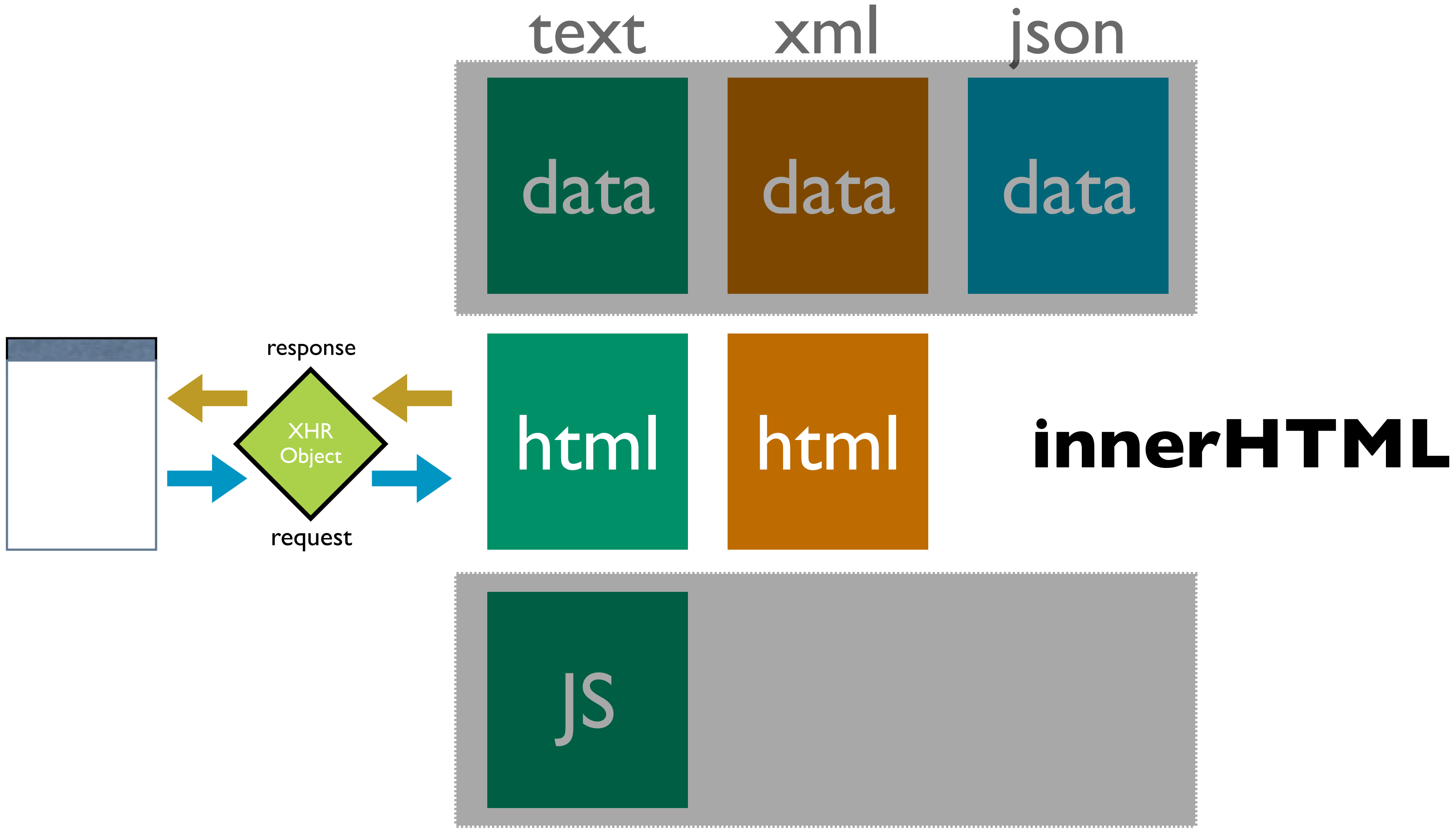
Response Options

- Different types, different formats



Response Options

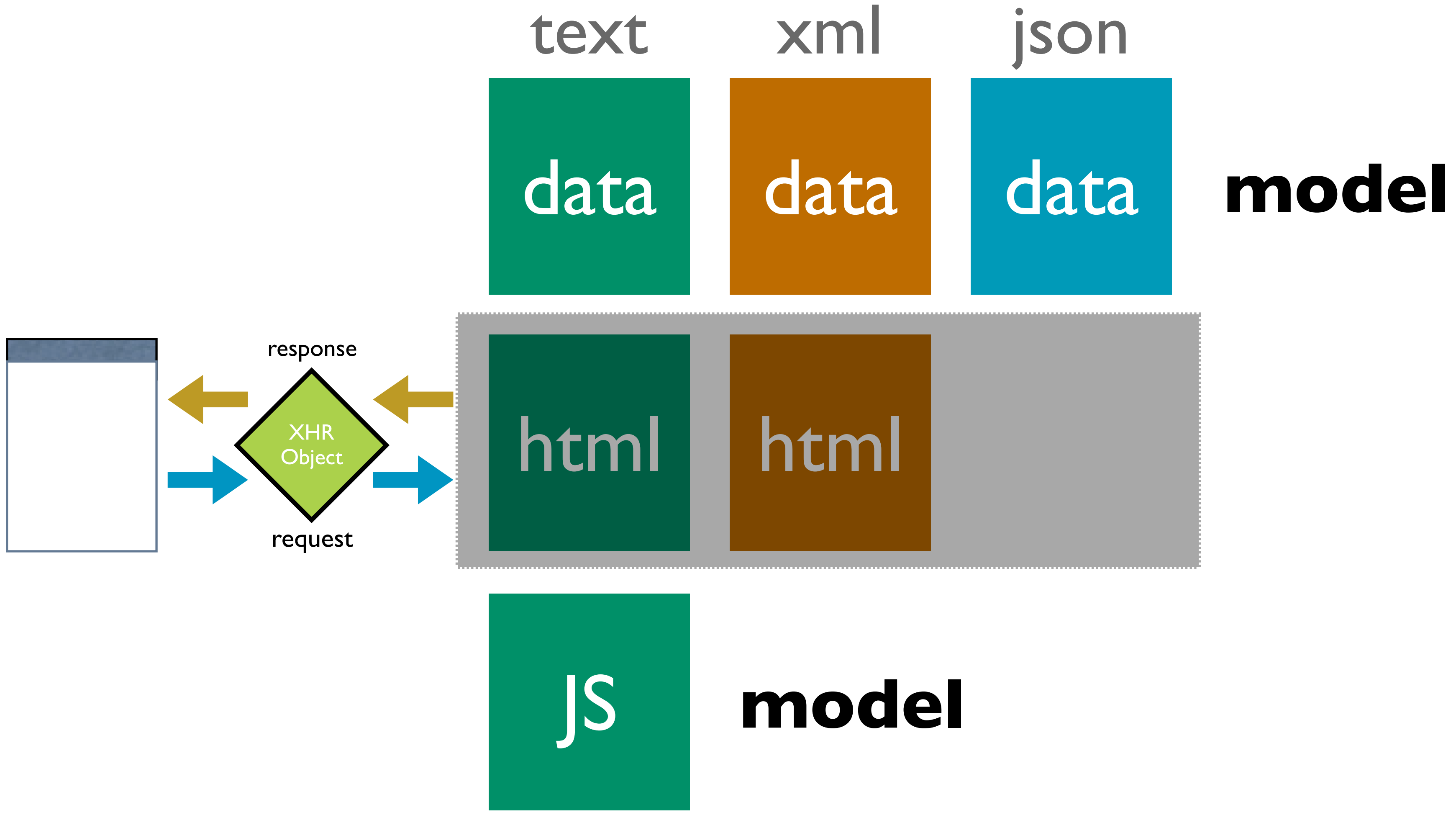
- Snippets of HTML returned, stuffed into innerHTML



Ajax
XHR

Response Options

- Data returned from the server, interface built from it



JSON

- JavaScript supports several string based notations
  - Allows the basic types to be represented as string literals
  - Strings are easy to pass over the wire
- JSON (JavaScript Object Notation - json.org)

JSON Response: Object Literal

Ajax
XHR

{"**name**": "Jack B. Nimble", "at large": true, "grad
"A", "level": 3}

name	Jack B. Nimble
at large	true
grade	A
level	3

JSON Response: Array Literal

```
["Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"]
```

array of 7 named days

```
[
[0, -1, 0],
[1, 0, 0],
[0, 0, 1]
]
```

3x3 array

JSON Response: JSON is JavaScript

- JSON's simple values are the same as used in JavaScript
- No restructuring is requested: JSON's structures are JavaScript!
- JSON's object is the JavaScript object
- JSON's array is the JavaScript array

- Parsing is simple, native

JSON Response: Parsing JSON

- Obtain responseText
- Parse the responseText

```
responseData = eval('(' + responseText + ')');
   OR
responseData = JSON.parse(responseText);
```

JSON Response: Y! JSON API

Ajax
XHR

- http://api.search.yahoo.com/WebSearchService/V1/webSearch appid=YahooDemo&query=finances&start=1&results=1&output=json

```
{
 "ResultSet":
 {
     "totalResultsAvailable":"69200000",
     "totalResultsReturned":"1",
     "firstResultPosition":"1",
     "Result":
     {
        "Title":"Yahoo! Finance",
        "Summary":"manage the market and your money with Yahoo! Finance. Includes
        stock market quotes, business news, mutual funds, online bill pay, banking
        tools, loans, insurance, retirement planning, and tax tips and advice.",
        "Url":"http:\/\/finance.yahoo.com\/",
        "ClickUrl":"http:\/\/finance.yahoo.com\/",
        "ModificationDate":"1137225600",
        "MimeType":"text\/html"
     }
 }
}
```

Ajax Operations

- **Lookup** I can get information when I need it
- **Persist** I can save in real-time
- **Validate** I can prevent errors early
- **Invoke** I can make things happen now
- **Message** I can communicate instantly

Trigger. JavaScript Events

Trigger. JavaScript Events

- Ajax interactions are kicked off by event & timer triggers
- There are issues with event management within the browsers
- You do need to understand these implications to write good Ajax applications

First, the Events

onAbort	onBlur
onChange	onClick
onDblClick	onDragDrop
onError	onFocus
onKeyDown	onKeyPress
onKeyUp	onLoad
onMouseDown	onMouseMove
onMouseOut	onMouseOver
onMouseUp	onMove
onReset	onResize
onSelect	onSubmit
onUnload	

Problem in a Nutshell

- Different event models
- Timing Issues
- Confusing madness around the **this** pointer
- Browser incompatibilities
- Memory leak issues

Different Event Models

- Two Models (actually three!)
  - Classic style
    - <element onclick=func() >
    - element.onclick=func;
  - W3C event model
    - addEventListener, removeEventListener
  - IE event model
    - attachEvent, detachEvent

Timing Issues

- Attempting to attach events before the page is loaded will cause problems
- Do it on the onload

Confusing *this* Madness

- Inline registration

```
<a href="#" onclick="clickHandler(this)">
function clickHandler(anchorDOMElem) {
   // this == window --> window owns the function
   this = anchorDOMElem; // fix the 'this' pointer
}
```

- Event handler with function

```
anchorDOMElem.onclick=clickHandler;
function clickHandler() {
   //this == anchorDOMElem --> anchorDOMElem owns the
function
}
```

- Event handler with Object prototype

```
function AnchorLink(anchorDOMElem) {
   // this == AnchorLink object
   this.anchorDOMElem = anchorDOMElem; // save DOM elem
   this.anchorDOMElem.onclick = this.clickHandler; // event handler
   this.anchorDOMElem.anchorLinkObj = this; // store this on dom elem
}
AnchorLink.prototype.clickHandler = function() {
   // this == anchorDOMElem, not AnchorLink object
   // confusing since this normally refers to AnchorLink
   // grab our normal this
   anchorLinkObj = this.anchorLinkObj;
}
```

- Arbitrary number of event handlers
- Way to remove events
- Defines capture & bubble flow
- No way to get list of handlers
- And the browsers play differently:

Internet Explorer	Mozilla (FF), Safari, Opera [W3C]
addEventListener(), removeEventListener()	attachEvent(), detachEvent
this == window object	this == DOM event object

Memory Leaks

- IE's garbage collection does reference counting
- Attaching events and not removing them when finished will cause memory leaks
- Can use an Observer pattern to cache event handlers and at the end clean them up
- Most Ajax frameworks provide this capability

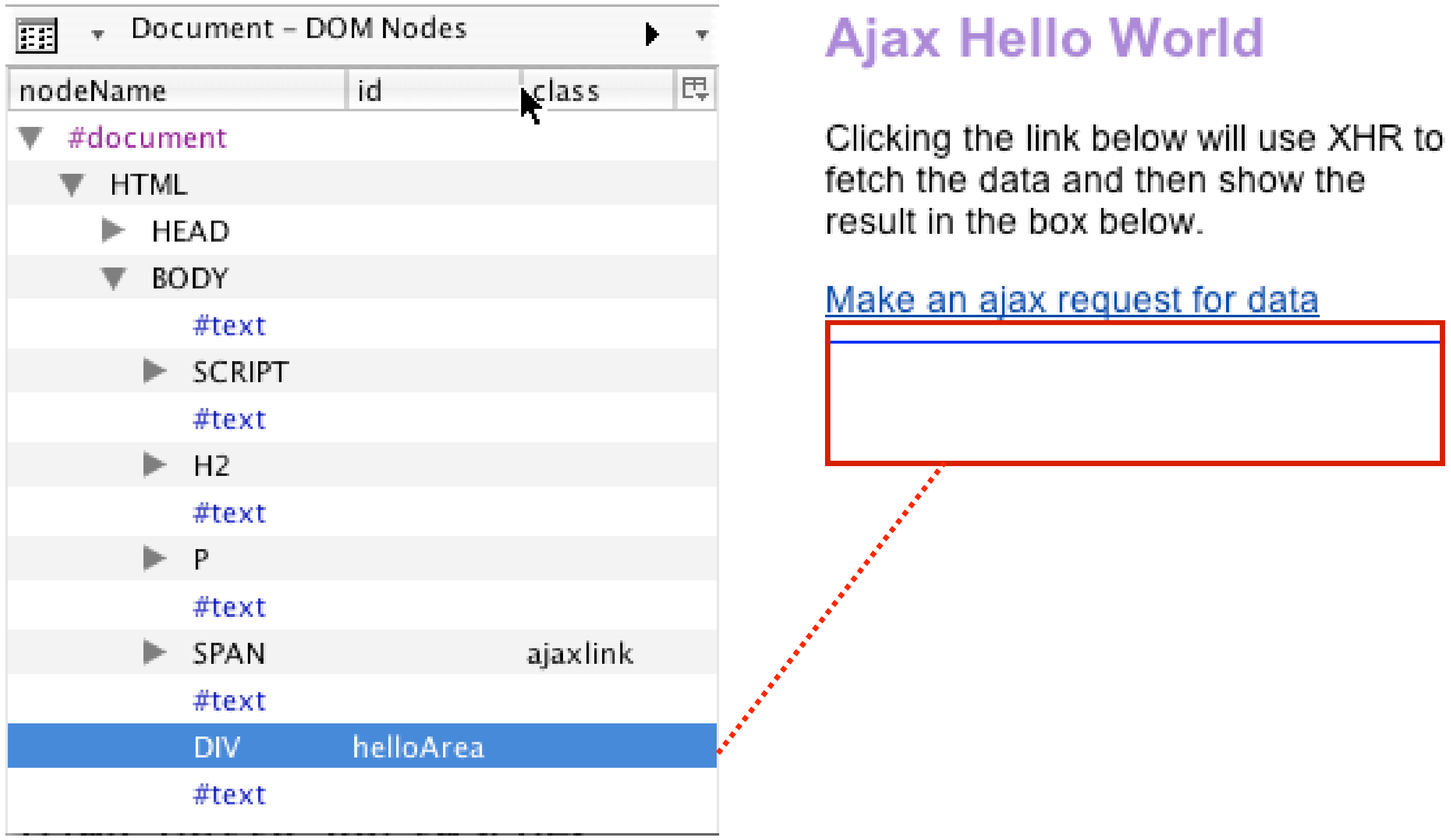
Update. The DOM

Update. The DOM

- Browsers represent the user interface as a set of objects (or elements)
- Since a page has a hierarchy of containment, each of these elements are related as parent-child or siblings
- This takes the shape of a tree
- The tree of elements is called the *document object model* or DOM
- Any change to the DOM structure or a DOM element is reflected immediately on the web page

DOM Example

- Represented as a tree of nodes



```
Document - DOM Nodes
nodeName                 id          class
▼ #document
  ▼ HTML
    ▶ HEAD
    ▼ BODY
        #text
      ▶ SCRIPT
        #text
      ▶ H2
        #text
      ▶ P
        #text
      ▶ SPAN                         ajaxlink
        #text
        DIV              helloArea
        #text
```

Ajax Hello World

Clicking the link below will use XHR to fetch the data and then show the result in the box below.

Make an ajax request for data

Using the DOM

- JavaScript is the DOM manipulator
- Use it to
  - Find DOM elements
  - Add new elements
  - Remove elements
  - Modify element attributes

Finding DOM Elements

- document.getElementById
  - Prototype library shortcut: $(“idName”) or $(id)
- parentNode
- childNodes

DOM Manipulation

- Creating new interface elements
  - innerHTML, createElement(), createTextNode(), appendChild()
- Changing element styles
  - Visual attributes
  - Geometry

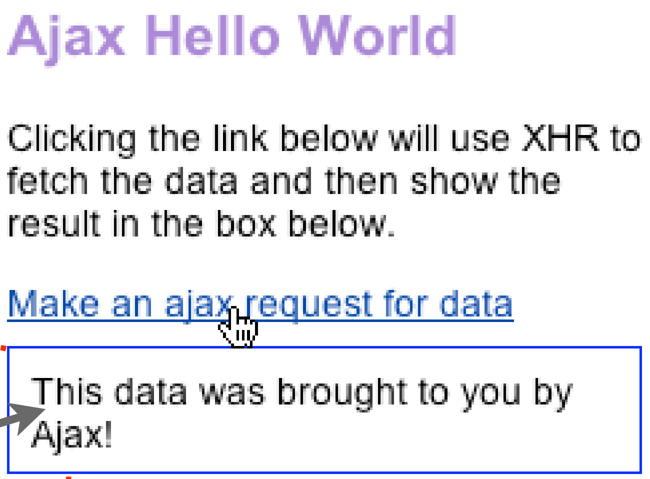
Example of DOM Manipulation (Hello World)

DHTML
DOM

```
<h2>Ajax Hello World</h2>
<p>Clicking the link below will use XHR to fetch the data and
then show the result in the box below.</p>

<span class="ajaxlink" onclick="makeRequest('response.jsp')">
Make an ajax request for data
</span>

<div id="helloArea"></div>
```



Ajax Hello World

Clicking the link below will use XHR to fetch the data and then show the result in the box below.

Make an ajax request for data

This data was brought to you by Ajax!

```
var helloArea = document.getElementById("helloArea");
helloArea.innerHTML=rootNode.firstChild.data;
```

Keeping it Clean

- Separate presentation style from content with CSS
  - Supports degradability
  - Supports accessibility
  - Simplifies maintenance
- This is called good semantic markup

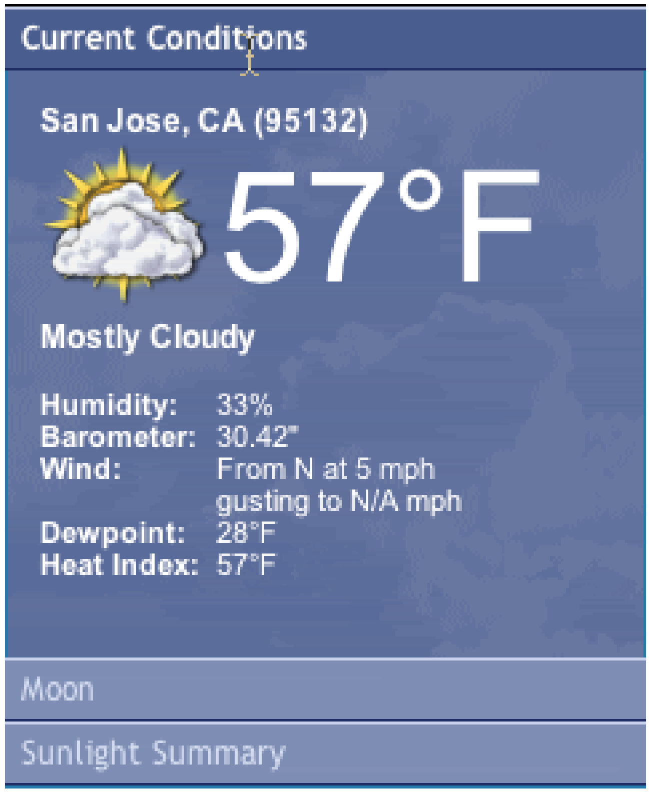
Semantic Markup Example

DHTML
DOM

```
<div id="weather">
 <div id="current">
   <div id="currentHeader" class="accordionTabTitleBar">
     Current Conditions
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="ccInfo">
     </div>
   </div>
 </div>

 <div id="moon">
   <div id="moonHeader" class="accordionTabTitleBar">
     Moon
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="moonInfo">
     </div>
   </div>
 </div>

 <div id="sun">
   <div id="sunHeader"  class="accordionTabTitleBar">
     Sunlight Summary
   </div>
     <div class="weatherTabContentBox">
     <div class="weatherPanel" id="sunInfo">
     </div>
   </div>
 </div>
</div>
```


Current Conditions

San Jose, CA (95132)

57°F

Mostly Cloudy

Humidity: 33%
Barometer: 30.42"
Wind: From N at 5 mph gusting to N/A mph
Dewpoint: 28°F
Heat Index: 57°F

Moon

Sunlight Summary

```
new Rico.Accordion( 'weather', {panelHeight:220, expandedBg:'#39497b'}
```

Ajax & Web Services

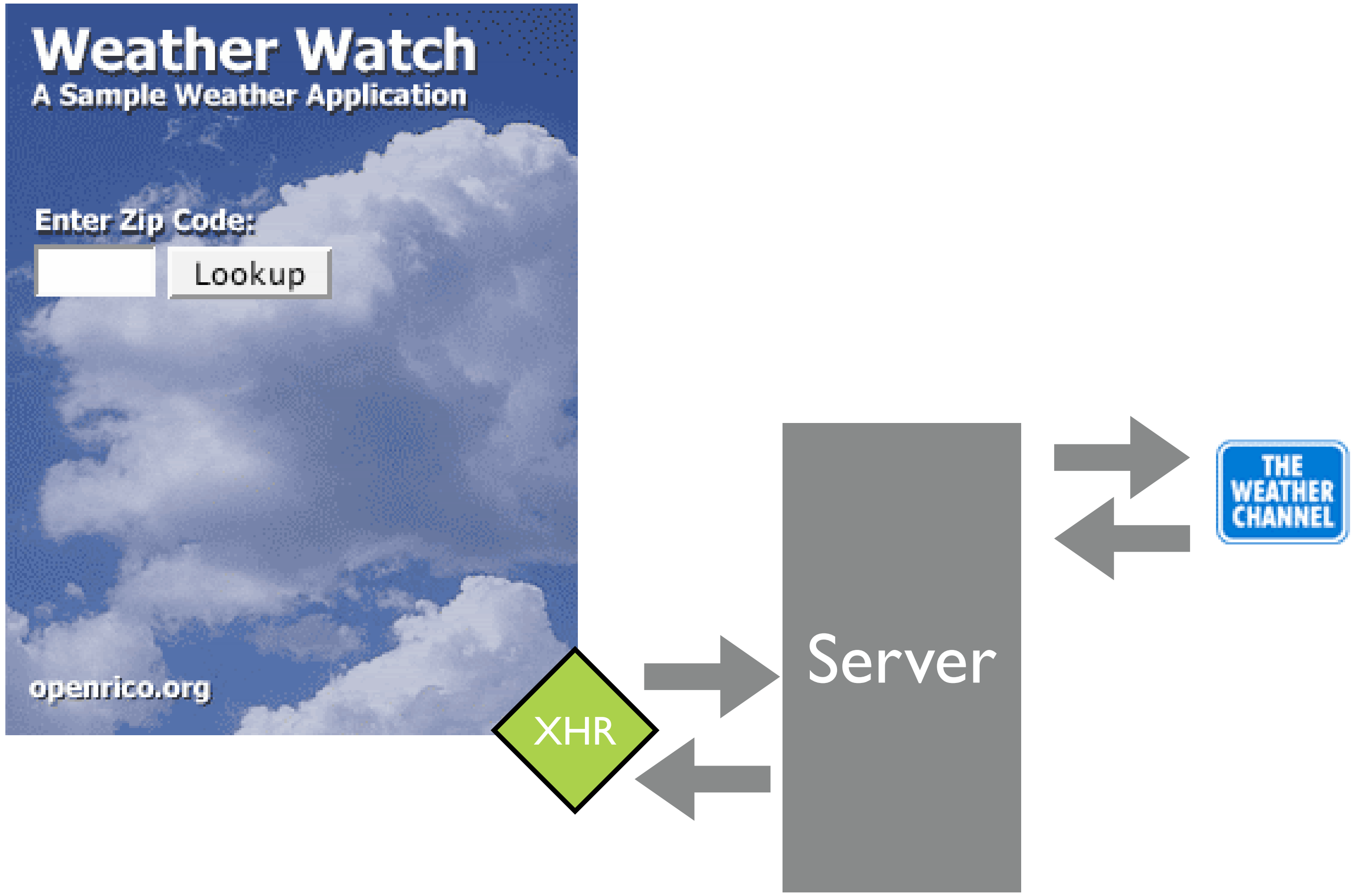
XML and JSON Examples

Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

Building an Ajax Weather Widget

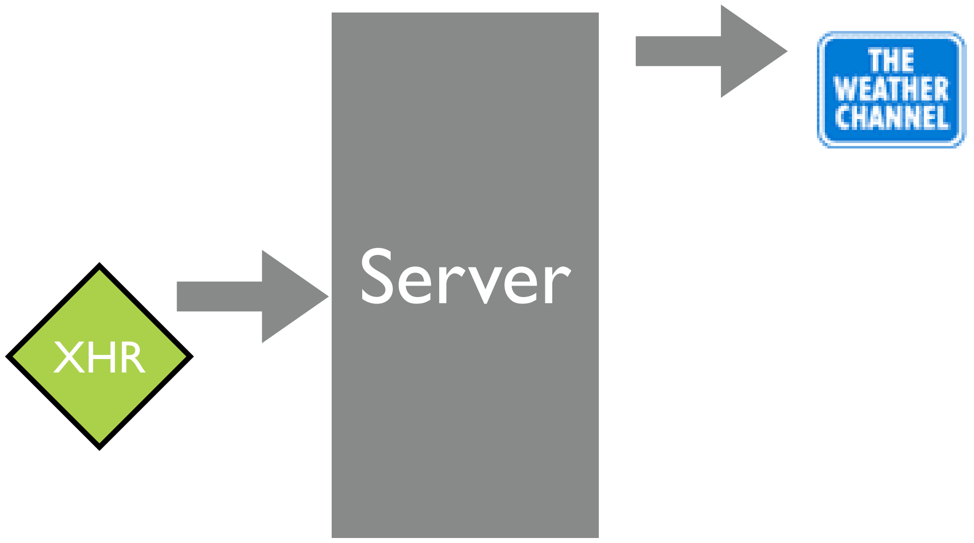
Using an XML Service



REST Service - weather.com

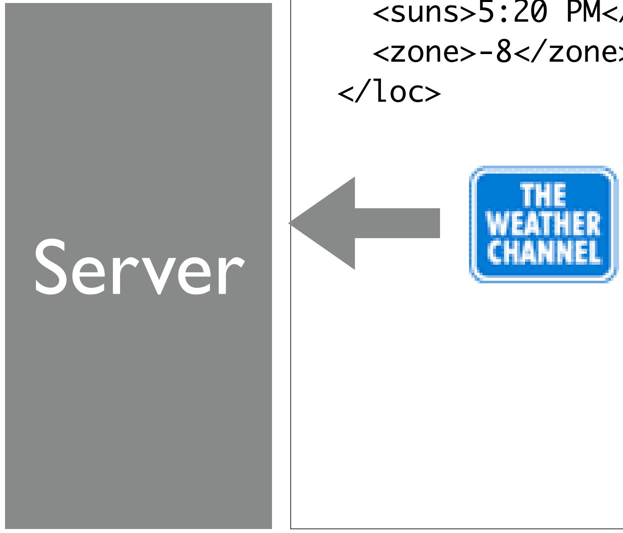
weather.com exposes its weather service via an URL that includes a partner key and license key

```
http://xoap.weather.com/weather/local/95132?
cc=*&link=xoap&prod=xoap&par=PARTNER_KEY&key=LICENSE_KEY
```



XML Response from weather.com

```
<?xml version="1.0" encoding="ISO
-8859-1"?>
<weather ver="2.0">
  <head>
    <locale>en_US</locale>
    <form>MEDIUM</form>
    <ut>F</ut>
    <ud>mi</ud>
    <us>mph</us>
    <up>in</up>
    <ur>in</ur>
  </head>
  <loc id="95132">
    <dnam>San Jose, CA (95132)</
dnam>
    <tm>8:37 PM</tm>
    <lat>37.4</lat>
    <lon>-121.86</lon>
    <sunr>7:18 AM</sunr>
    <suns>5:20 PM</suns>
    <zone>-8</zone>
  </loc>
```



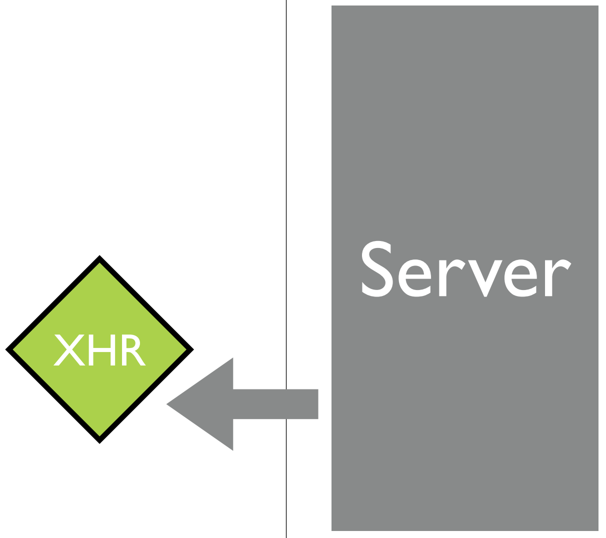
```
  <cc>
    <lsup>1/21/06 7:53 PM PST</lsup>
    <obst>San Jose, CA</obst>
    <tmp>49</tmp>
    <flik>46</flik>
    <t>Mostly Cloudy</t>
    <icon>27</icon>
    <bar>
      <r>30.27</r>
      <d>rising</d>
    </bar>
    <wind>
      <s>7</s>
      <gust>N/A</gust>
      <d>350</d>
      <t>N</t>
    </wind>
    <hmid>80</hmid>
    <vis>10.0</vis>
    <uv>
      <i>0</i>
      <t>Low</t>
    </uv>
    <dewp>43</dewp>
    <moon>
      <icon>21</icon>
      <t>Waning Gibbous</t>
    </moon>
  </cc>
</weather>
```

weather.com responds with an XML response that describes the weather's current conditions

Rico/Prototype Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ajax-response>
  <response type="element" id="ccInfo">
  <div class="weatherTitle">San Jose, CA (95132)</div>
  <div>
  <span><image id="ccImg" src="/images/weather/27.png"></image></span>
  <span class="weatherTemp">49&#176;F</span>
  </div>
  <div class="weatherDescr">Mostly Cloudy</div>
  <div>
  <table cellspacing="0" cellpadding="0" border="0">
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Humidity:</td>
    <td>80&#37;</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Barometer:</td>
    <td>30.27&quot;</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Wind:</td>
    <td>From N at 7 mph</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle"></td>
    <td>gusting to N/A mph</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Dewpoint:</td>
    <td>43&#176;F</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Heat Index:</td>
    <td>46&#176;F</td></tr>
  </table>
  </div>
  </response>
</ajax-response>
```

My server code translates the weather.com XML into a Rico/Prototype ajax-response. Notice the response is an HTML code snippet. The response is of type element, and mapped to id="ccInfo"

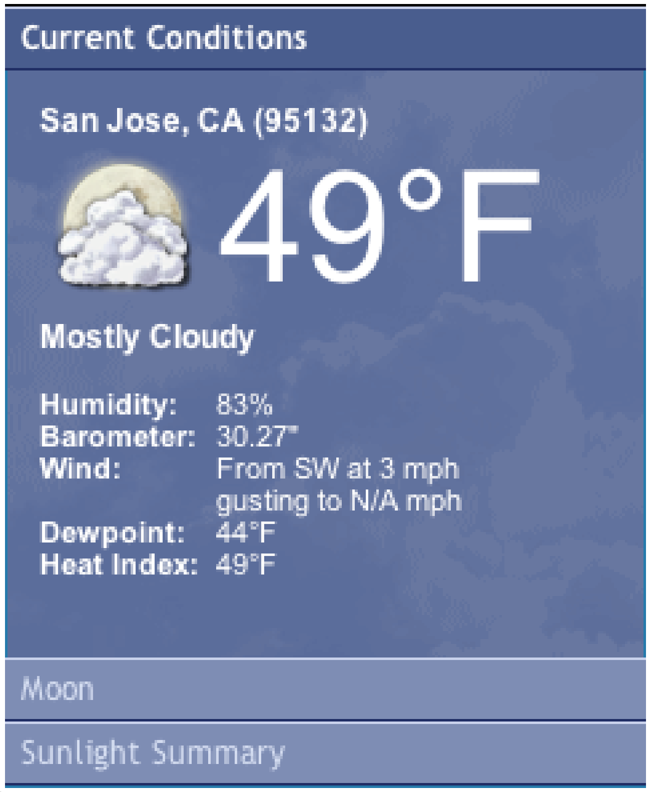


Simple DIV Structure Defines the Accordion

```
<div id="weather">
 <div id="current">
   <div id="currentHeader" class="accordionTabTitleBar">
     Current Conditions
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="ccInfo">
     </div>
   </div>
 </div>

 <div id="moon">
   <div id="moonHeader" class="accordionTabTitleBar">
     Moon
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="moonInfo">
     </div>
   </div>
 </div>

 <div id="sun">
   <div id="sunHeader"  class="accordionTabTitleBar">
     Sunlight Summary
   </div>
     <div class="weatherTabContentBox">
     <div class="weatherPanel" id="sunInfo">
     </div>
   </div>
 </div>
</div>
```



San Jose, CA (95132)

49°F

Mostly Cloudy

Humidity:	83%
Barometer:	30.27"
Wind:	From SW at 3 mph gusting to N/A mph
Dewpoint:	44°F
Heat Index:	49°F

```
<script>
onloads.push( accord );
function accord() {
  new Rico.Accordion( 'weather',
       {panelHeight:220, expandedBg:'#39497b'} );
}
</script>
```

The Ajax Code

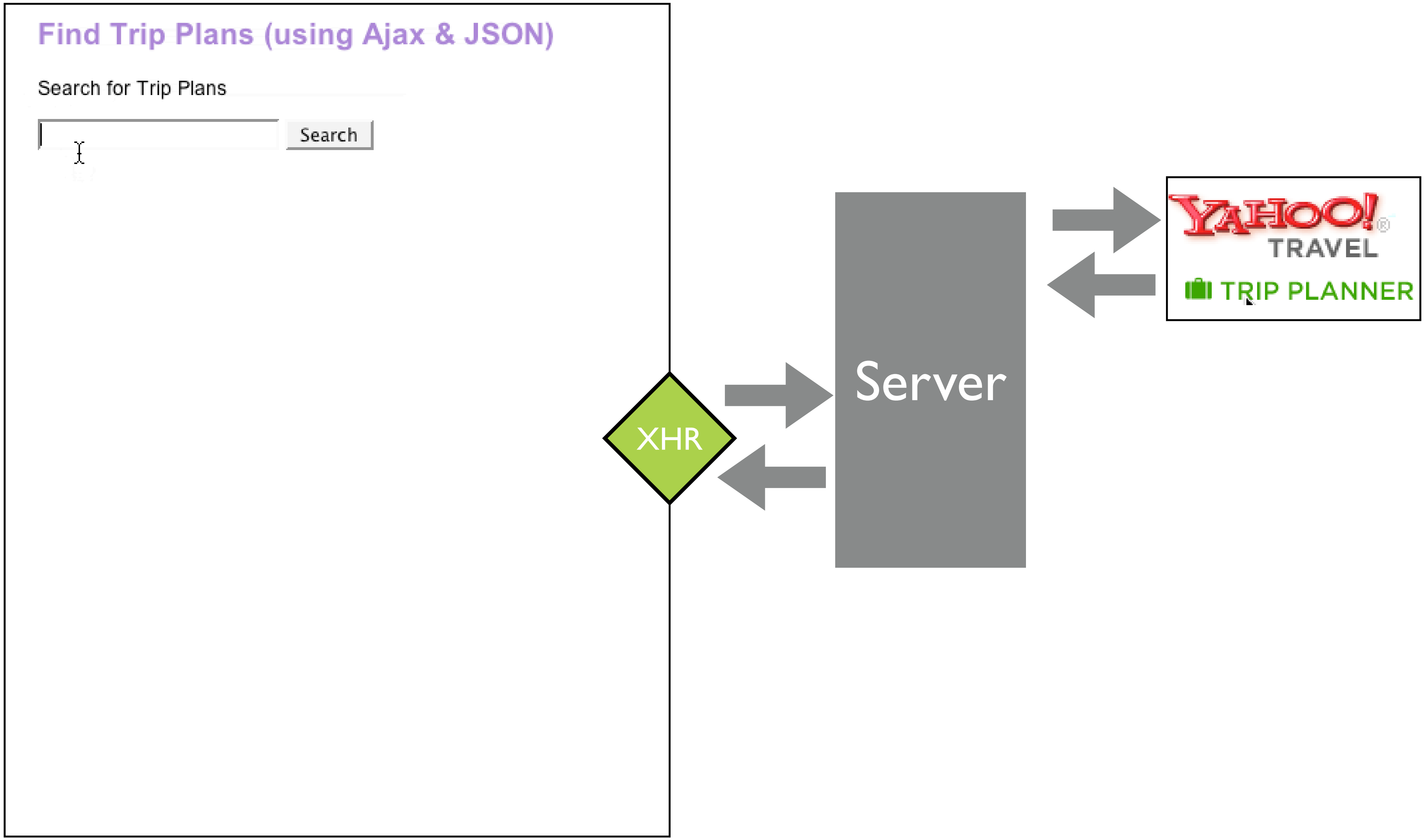
```
function registerAjaxStuff() {
   ajaxEngine.registerRequest( 'getWeatherInfo', 'ajax_weather_info' );
   ajaxEngine.registerAjaxElement( 'ccInfo' );
   ajaxEngine.registerAjaxElement( 'moonInfo' );
   ajaxEngine.registerAjaxElement( 'sunInfo' );
   ajaxEngine.registerAjaxElement( 'sponsoredLinks' );
   $('zip').onkeydown = handleEnterKey.bindAsEventListener($('zip'));
}
function getWeatherInfo() {
   $('checkanother').style.visibility='visible';
   new Rico.Effect.Position( $('zipinput'), 200, null, 100, 10);

   new Rico.Effect.FadeTo( 'frontdoor', 0, 100, 10,
         {complete:function() {$('frontdoor').style.display = 'none';}}
         );

   ajaxEngine.sendRequest( 'getWeatherInfo', "zip=" + $('zip').value);
}
function resetWeather() {
   $('zipinput').style.left = '12px';
   $('checkanother').style.visibility='hidden';

   $('frontdoor').style.display = ''
   $('zip').focus();
   new Rico.Effect.FadeTo( 'frontdoor', .99999, 100, 10, {complete:emptyContents});
}
```

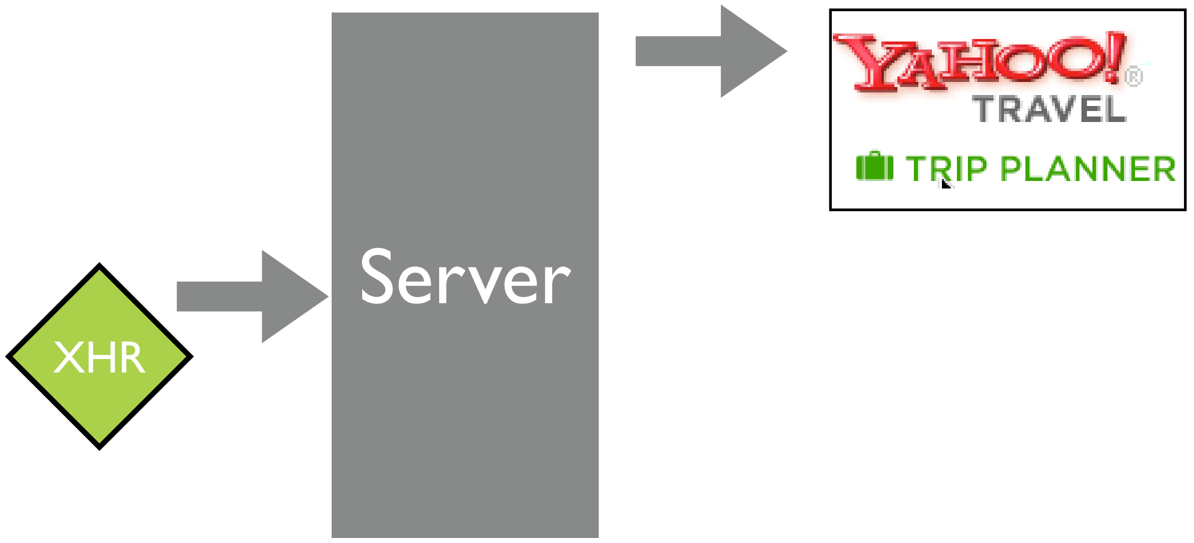
JSON Trip Finder



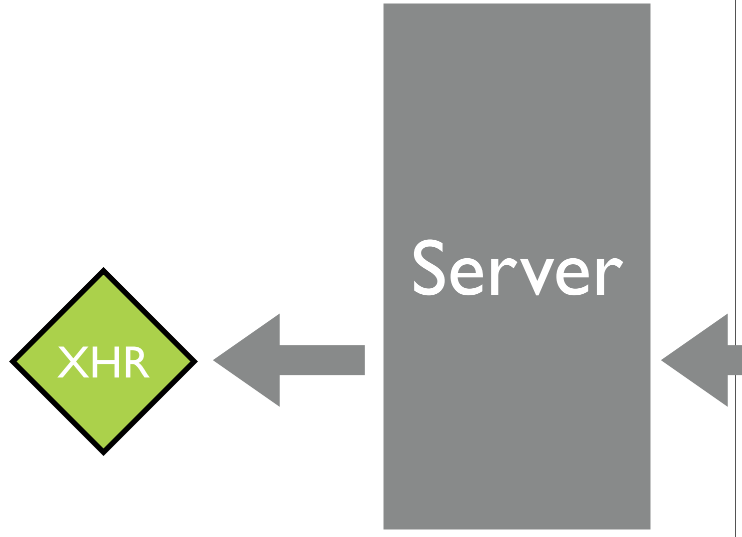
REST Service Request - Yahoo! Trip Planner - JSON

Yahoo! Trip Planner exposes its service via an URL that requires a free partner key. Notice output=json.

```
http://api.travel.yahoo.com/TripService/V1/tripSearch?
appid=PARTNER_KEY&query=alaska&output=json
```



JSON Response



```
{"ResultSet":{"firstResultPosition":
1,"totalResultsAvailable":"16","totalResultsReturned":
10,"Result":[

{"id":"351102","YahooID":"jk_95661","Title":"Cruise to Alaska
from Vancouver","Summary":"Things to do: Whistler Blackcomb -
Hiking,... Hotel: Hyatt Regency Vancouver, Howa
Jho...","Destinations":"Vancouver, Ancho
Vancouver, Ketchikan,
Se...","CreateDate":"1130437928","D
{"Url":"http:\/\/us.i1.yimg.com\/u
lp\/cd\/
100x100_cde24409e413d4d6da27953f6a
"Width":"66"},"Geocode":{"Latitude"
-123.120499","precision":"not availa
travel.yahoo.com\/trip\/?pid=351102

{"id":"400333","YahooID":"brdway_grl","Title":"Alaska","Summa
ry":"Things to do: Moose's Tooth Pub and Pizzer...
Restaurant: Orso, Simon's & Seafort's
Salo...","Destinations":"Anchorage","CreateDate":"1134676973"
,"Duration":"10","Image":{"Url":"http:\/\/us.i1.yimg.com\/
us.yimg.com\/i\/travel\/tg\/poi\/ca\/
100x1                      31d3093b94afa37e.jpg","Height":"75","
Widt                       {"Latitude":"61.190361","Longitude":"
-149                       :"not available"},"Url":"http:\/\/
trave                      ?pid=400333&action=view"},
...
]}}
```

Y! service responds with a JSON text string, valid JavaScript object notation. This is passed directly back as the XHR result

The Ajax/JSON Code

```
<script>
function showResponse() {
    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            var jsontext = xhr.responseText;
            var helloArea = document.getElementById("helloArea");
            var theTrip = eval( '(' + jsontext + ')' );
            var tripPlanHTML = "";
            for(var i=0; i<theTrip.ResultSet.totalResultsReturned; i++) {
                var result = theTrip.ResultSet.Result[i];
                tripPlanHTML = tripPlanHTML + '<div style="padding:4px;
border:1px solid gray;width:'+result.Image.Width+';"><img
src="'+result.Image.Url+'" width="'+result.Image.Width+'"
height="'+result.Image.Height+'"></img></div>'+
'<div ><a href="'+result.Url+'"><span style="font-weight:bold;font-size:
18px;">'+result.Title+'</span></a></div><div>'+result.Summary+'</div><br/>';
            }
            helloArea.innerHTML=tripPlanHTML;
        } else {
              alert('There was a problem with the request.');
        }
    }
}
</script>
<h2>Find Trip Plans (using Ajax &amp; JSON)</h2>
<p>Search for Trip Plans</p>
<input id="searchCriteria" type="text"> <input value="Search" type="button"
onclick="makeRequest('response_trip_planner_mult_json.jsp',
         document.getElementById('searchCriteria').value)">
<div style="height:300px;width:420px;margin-top:8px;padding:8px;"
id="helloArea"></div>
```

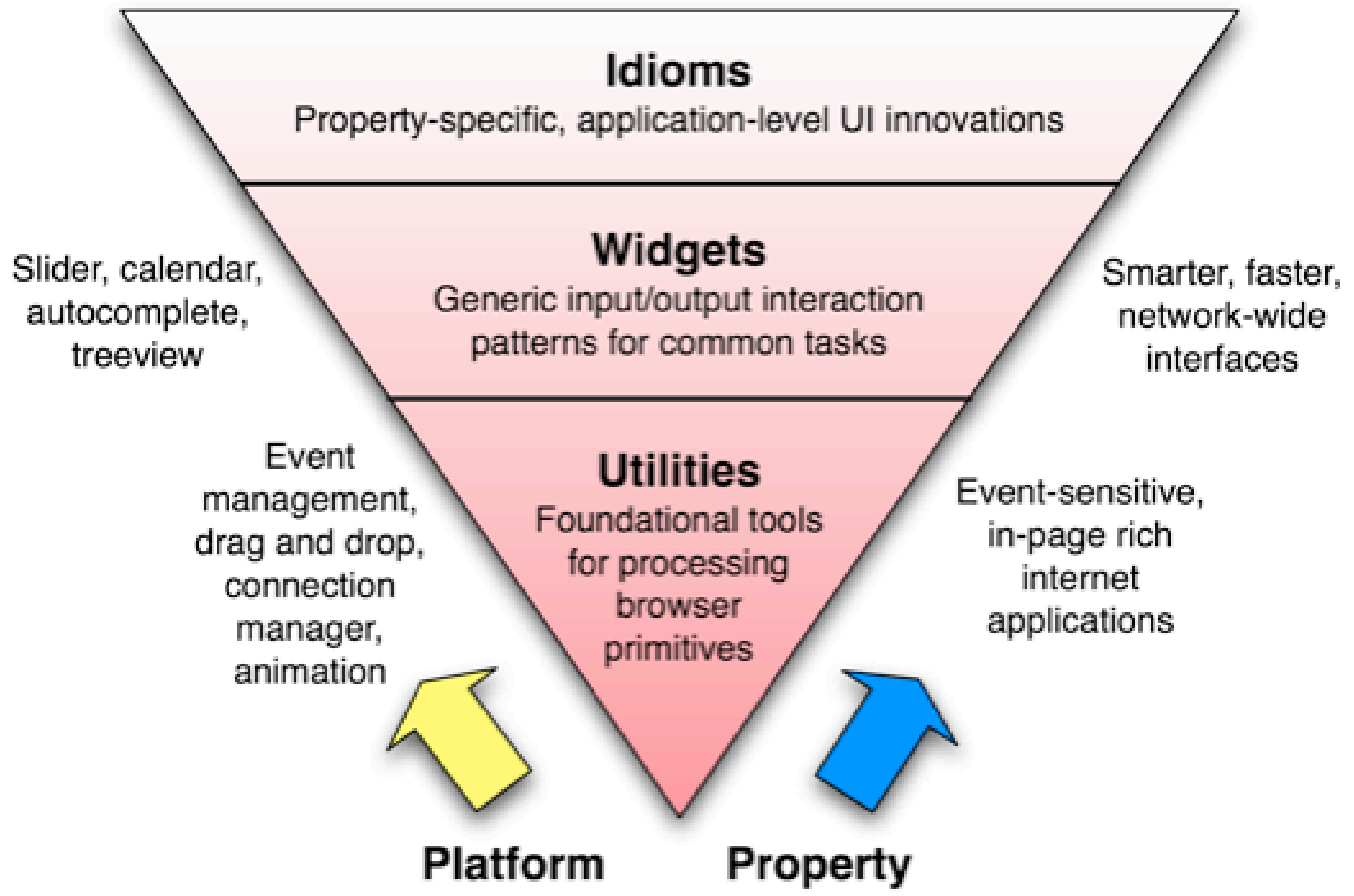
Ajax Toolkits

Designing and Building Ajax Applications with Ajax Toolkits

Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

What a Toolkit Provides


Idioms
Property-specific, application-level UI innovations
Slider, calendar, autocomplete, treeview
Widgets
Generic input/output interaction patterns for common tasks
Smarter, faster, network-wide interfaces
Event management, drag and drop, connection manager, animation
Utilities
Foundational tools for processing browser primitives
Event-sensitive, in-page rich internet applications
Platform
Property

Evaluating Toolkits

- What language is it targeted for?
- Are you looking for XHR abstraction or high level?
- Will you use XML or JSON?
- Is it $$ or open source?
- Does it simplify general JS development?
- Community of support?
- Likelihood it will be around next year?

Ajax Toolkits: Open Source JavaScript Libraries

Prototype	Inspired by Ruby, simplifies DOM manipulation, events, Ajax connectivity. Several frameworks based on it. Notable for the $() function among many other shortcuts. OO extends.
Scriptaculous	Built on Prototype, nice effects, animations, drag and drop, widgets. Nicely documented
Rico	Built on Prototype, nice effects, animations, behaviours (widgets), drag and drop, nice Ajax engine abstraction, style effects. Lots of demos.
Behaviour	Built on Prototype, separates event management from HTML using CSS selectors
Dojo	Widget library, drag and drop, effects, widget framework, event management, animation, bookmarkability, manipulating location history, extensive deployment support
Zimbra	Extensive set of widgets (data list, wizard, button, rich text editor, tree, menus, etc.). MVC, debugging facilities
Microsoft Atlas	In developer release. Support for drag and drop, animation, full XML specification for wiring widgets together, behavior abstraction, will be integrated in full suite of MS dev tools.
MochiKit	Widgets, painless DOM manipulation, task management, JSON-like notation, logging. Active community
Sarissa	Encapsulates XML functionality in browser-independent calls

Ajax Toolkit: Commercial JavaScript Libraries

TIBCO GI	Very extensive framework with full IDE. Dozens of widgets, vector based charting package, support for SOAP. IDE has WYSIWYG GUI layout, step through debugging as well as code completion
Bindows	Been around since 2001, lots of great widgets and library routines. Not always browser independent
JackBe	Focused at corporate environments. Able to emulate a desktop application. Very fast performance even on a dial-up line. Uses a proprietary "screen" format that gets interpreted by a JavaScript runtime engine.
Active Widgets	Rich set of JavaScript widgets. Rich grid widget is a high spot.

Ajax Toolkit: PHP

AjaxAC	Separates events from HTML, subrequest support, event management, easily remotes to PHP methods, start of a widget framework
Cajax	Low level. Simplifies server side programming, limits client side JS required, suggest widget, plug-in for form submission.
CakePHP	OO, modeled after Ruby on Rails, integrated CRUD, fast, flexible templating, nice docs
CPAINT	OO library for Ajax connectivity. Local & remote functions, single or multiple XHR, supports both POST & GET
JPSpan	PHP remoting via JavaScript
XAjax	Low level. PHP remoting via JavaScript. Simplified DOM updates on XHR response.
XOAD	Uses JSON and native PHP serialized objects, security emphasis, server side events, HTML manipulation. Tutorials and documentation.

Ajax Toolkit: Java Frameworks

Ajax JavaServer Faces Framework	Converts JSF applications to Ajax
Ajax JSP Tag Library	Set of tags for Ajax. Autocomplete, callout, select, toggle, update fields
DWR	For remoting Java methods from JavaScript code. Contains a number of widgets.
Echo2	Auto generate HTML & JavaScript from server side. Write in pure Java.
Guise ($$)	Application framework that uses the desktop UI approach to components and events (hides HTML generation and DOM manipulation.) Controls, dialogs, flyovers,
JSP Controls Tag Library	For portlet style JSP components. Supports Ajax and non-Ajax modes
SWATO	JS library based on Prototype. Uses JSON for marshalling. Direct remoting for POJOs. Widgets include auto suggest, template, logger.
AjaxAnywhere	Turns existing JSP/JSF/Struts/Spring components into Ajax Aware components
WebOrb for Java	Ties together Ajax, Flash and Java objects with XML services

Prototype

```
function makeRequest(url) {
    var xhr = new Ajax.Request(url, {method:'get', onComplete: showResponse});
}
function showResponse(xhr) {
    $('helloArea').innerHTML = xhr.responseText;
}
```

- $()
- $F()
- Try.these()
- Ajax.request()
- Ajax.Updater()
- Element.show/hide/toggle/remove
- Object-oriented extensions

Prototype Hello World

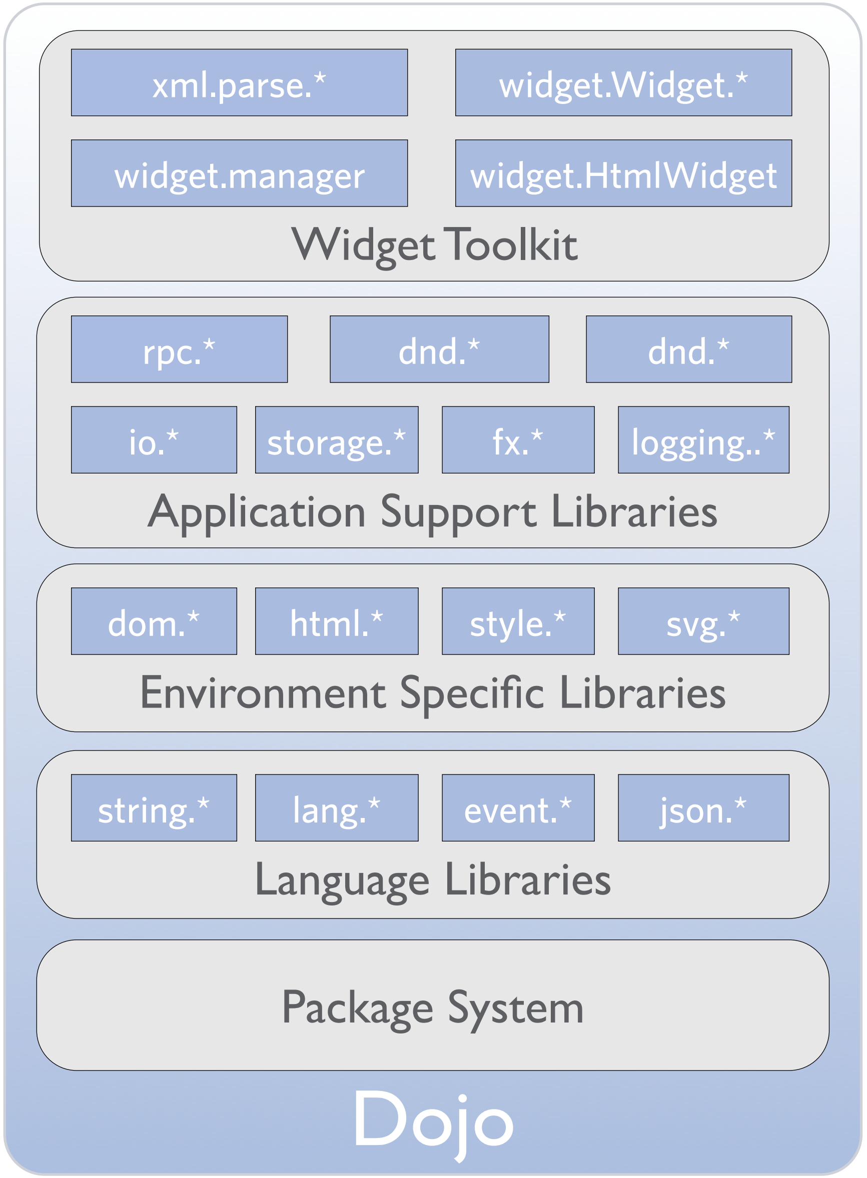
- Makes Ajax request/response simple

```
function makeRequest(url) {
     var xhr = new Ajax.Request(url,
                      { method:'get', onComplete: showResponse });
}

function showResponse(xhr) {
    $('helloArea').innerHTML = xhr.responseText;
}
```

- Rico allows multiple targets in the response

Dojo



- Events: multiple listeners, browser abstraction, memory leak protection
- Aspect-oriented
- I/O: common interface for multiple transports, single/ multi response
- Built-in packaging system
- Widget building w/HTML & CSS fragments, automatic reuse

Microsoft Atlas Features

- Extensible core adds lifetime mgmt, inheritance, multicast event handler, and interfaces
- Base class library: strings, timers, tasks
- UI framework for attaching behaviors to HTML
- Simple connectivity
- Set of rich UI controls (auto complete, popup panels, animation, and drag and drop
- Browser compatibility layer

Microsoft Atlas: Interesting Concepts

- Update panels: mark region for auto update
- Behaviors: encapsulate actions to associate with DHTML events [floating, hover, popup, autocomplete]
- Validators [required, type, range, regex, etc.]
- Data binding: connect controls & manage flow between them [data entered, reflected elsewhere]
- Bindings transformers: support transform event as part of a binding [ToString, invert, etc.]

Microsoft Atlas

- Two models: XML scripting & JS API
- Will be fully supported by Microsoft tools
- Big Q? Tied to ASP.NET?
  - Language on the site is confusing
  - Similar to how XHR got lost!
  - But I have been assured they intend the Atlas Client Scripting Framework to be totally independent of Microsoft technologies (although they want you to use them!)

Advanced Topics

Problems and Challenges with Building Ajax Applications

Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

How Ajax Changes Things

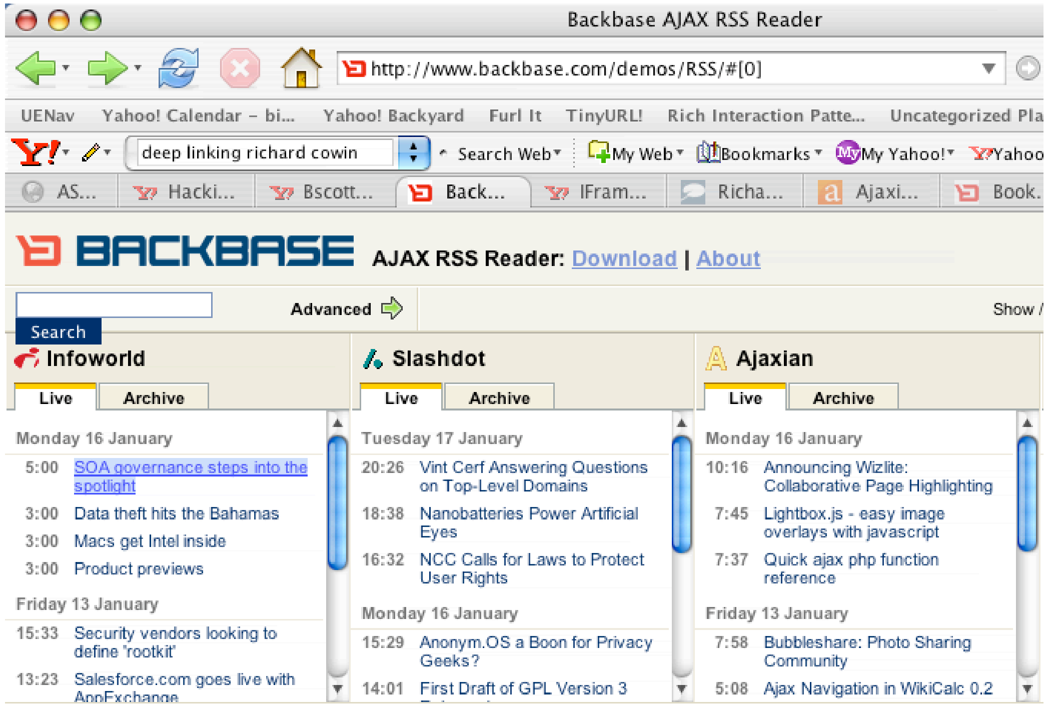
Classic Web	Ajax/Web 2.0	Problems/Challenges
Page to page navigation	Micro content	Back button, SEO, bookmarking, accessibility, security
URL/Link = User's location	Application state	Back button, SEO, bookmarking, accessibility
Browser history	Application history	Back button, bookmarking, accessibility
Back button = Undo	Is unpredictable	Back button, bookmarking, accessibility
Little JavaScript	More JavaScript	Accessibility, degradability, security, memory, performance, debug, obsfucation, error handling
Document model	Application model	Back button, SEO, bookmarking, accessibility
Static content	Dynamic content	SEO, bookmarking, accessibility
Course-grained events	Micro states	Back button
Synchronous	Asynchronous	Error handling
Browser chrome	Application controls	Back button, bookmarking, accessibility
Page Services	Web Services	Security, XML vs. JSON

Back Button

- Problem: Ajax application state changes, URL does not. No browser history of state changes/navigation
- What does the user expect?
  - Often confused with Undo
  - True context view changes should be part of history
    - Navigation tabs; but not content tabs?
    - Steps in a process
    - Tree selection when views change
  - Impact of tabbed browsers?

Back Button Solution: Fragment Identifier

- URL hash, fragment identifier (**http://a.com#loc** does not trigger a page reload
- Fragment identifier (string after '#') can be used to record state changes in the URL



Infoworld	Slashdot	Ajaxian
Live / Archive	Live / Archive	Live / Archive
Monday 16 January	Tuesday 17 January	Monday 16 January
5:00 SOA governance steps into the spotlight	20:26 Vint Cerf Answering Questions on Top-Level Domains	10:16 Announcing Wizlite: Collaborative Page Highlighting
3:00 Data theft hits the Bahamas	18:38 Nanobatteries Power Artificial Eyes	7:45 Lightbox.js - easy image overlays with javascript
3:00 Macs get Intel inside	16:32 NCC Calls for Laws to Protect User Rights	7:37 Quick ajax php function reference
3:00 Product previews	Monday 16 January	Friday 13 January
Friday 13 January	15:29 Anonym.OS a Boon for Privacy Geeks?	7:58 Bubbleshare: Photo Sharing Community
15:33 Security vendors looking to define 'rootkit'	14:01 First Draft of GPL Version 3	5:08 Ajax Navigation in WikiCalc 0.2
13:23 Salesforce.com goes live with AppExchange		

Back Button: Fragment Identifier

- Yahoo! Maps Beta also uses this technique
- Bottom line: tricky to implement
  - Dojo, Backbase provide direct support
  - One approach: http://www.contentwithstyle.co.uk/Articles/38/fixing-the-back-button-and-enabling-bookmarking-for-ajax-apps
    - All links have fragment
    - Clicking link changes URL, not page
    - Timer monitors window.location.href & updates

Back Button: iframe Approach

- Technique: use iframes to control browser history for recording state changes
- Tricky issues abound
  - Fragile across browsers
  - onload issues
  - audible transition (on IE if sound enabled)
- Bottom line: problematic

Search Engine Optimization (Deep Linking)

- All the content may not be statically available for search engine crawlers
  - Won’t find content to index your pages correctly
- Possible solutions
  - **Lightweight Indexing**: leverage existing tags such as meta, title and h1
  - **Extra Links**: extra links are placed on the site.
  - **Secondary Site**: a secondary site is fully accessible to the search engine. (See degraded experience)

source: backbase.com

Bookmarking

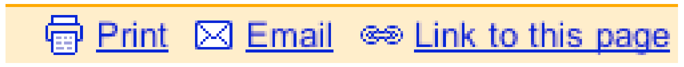
- Since we have broken the history and URL paradigm, bookmarking become problematic
- What does the user expect?
  - Do they expect to bookmark application state? content viewed?
  - Desktop apps often support bookmarking. It is always content based.

Bookmarking Technique

- Allow the user to save a bookmark at an interesting moment in an Ajax application
- Perhaps dynamically generate a link for bookmarking
- The URL generated for the bookmark is sufficient to restore the state

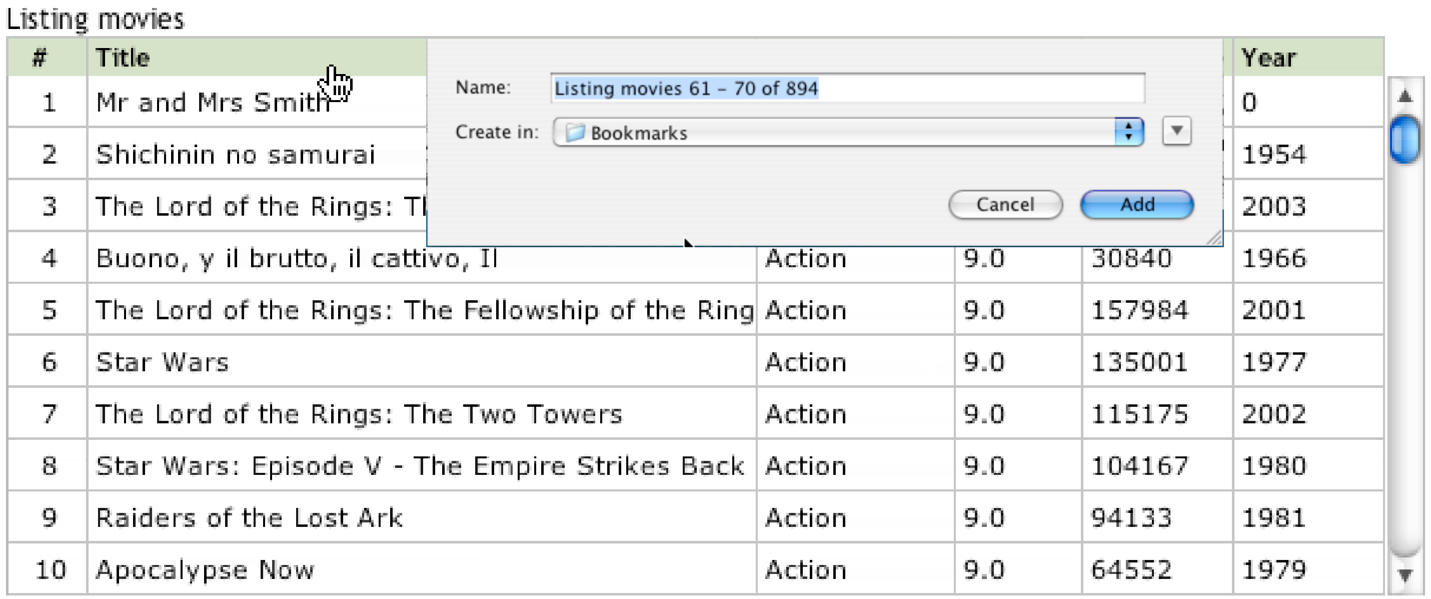
Bookmarking in Google Maps

- Google Maps
  - Link is generated on each new map address
  - Link contains URL parameters to return to the page



Another Bookmarking Example

- OpenRico LiveGrid
  http://richardcowin.typepad.com/blog/2005/07/there_has_been_.html



Listing movies

#	Title				Year
1	Mr and Mrs Smith				0
2	Shichinin no samurai				1954
3	The Lord of the Rings: T				2003
4	Buono, y il brutto, il cattivo, Il	Action	9.0	30840	1966
5	The Lord of the Rings: The Fellowship of the Ring	Action	9.0	157984	2001
6	Star Wars	Action	9.0	135001	1977
7	The Lord of the Rings: The Two Towers	Action	9.0	115175	2002
8	Star Wars: Episode V - The Empire Strikes Back	Action	9.0	104167	1980
9	Raiders of the Lost Ark	Action	9.0	94133	1981
10	Apocalypse Now	Action	9.0	64552	1979

Name: Listing movies 61 – 70 of 894

Create in: Bookmarks

```
http://openrico.org/rico/livegrid.page?
data_grid_index=60&data_grid_sort_col=rating&data_grid_sort_dir=ASC
```

Accessibility

DHTML Provides	Accessibility Expects	Problem
JavaScript enabled markup, new user interface controls	Simple markup	Markup has more meaning than expected. How does the assistive technology understand this is a tree control? How does it understand the state of a control?
Dynamic pages & content	Fairly static pages	How do I announce dynamic content?
Weak support for keyboard navigation	Keyboard navigation	Sight-impaired users will not be using rich interactions; they will use the keyboard (or another device that simulates the keyboard) But how to tab key, arrow key, select items with the keyboard?

- IBM/Mozilla Accessible DHTML API/Spec
  - Direct support for keyboard traversal
    - Supports tab key to a container, arrow keys to navigate inside the container and enter key to select item
    - Setting tabindex=-1 allows focus to be set on a object without it being in the tab order
  - A way to add metadata to HTML markup for assistive technologies to understand:
    - Widget roles, properties, values and events
  - Working with assistive technology vendors to make them aware of this API (funded development)
- Microsoft's plans

Accessibility: Setting tabindex

tabindex value	Purpose	Tab key navigable?
Not set	Accept default behavior	Default behavior Form elements
tabindex = “-1”	For child elements (nodes in a tree)	No. You must set the focus with JavaScript
tabindex = “0”	To tab in HTML code order	Tabbing moves you field to field in the order they are added to the page
tabindex > “0”	To specify an exact order for each field	Value describes the tab order

Accessibility: Example of Markup for Role & Property

- Roles

```
<span tabindex="0" xhtml2:role="wairole:checkbox"
      property:checked="true"
      onkeydown="return checkBoxEvent(event);"
      onclick="return checkBoxEvent(event);">
  Any checkbox label
</span>
```

- Provides clues to assistive technologies

Degradability

- Degradability - Managing the user experience as you move down in device/browser capability
- At Yahoo! we grade the browser by experience we will provide (A, B, C grade)
  - A-grade experience (majority of our users; greatest visual fidelity and richest interactions
  - B-grade experience (almost A-grade, but either they are on bleeding edge, low adoption or low economic value). Not considered in development or testing
  - C-grade experience (baseline of support; content but little style or interaction). Crawlers and old browsers get this experience.

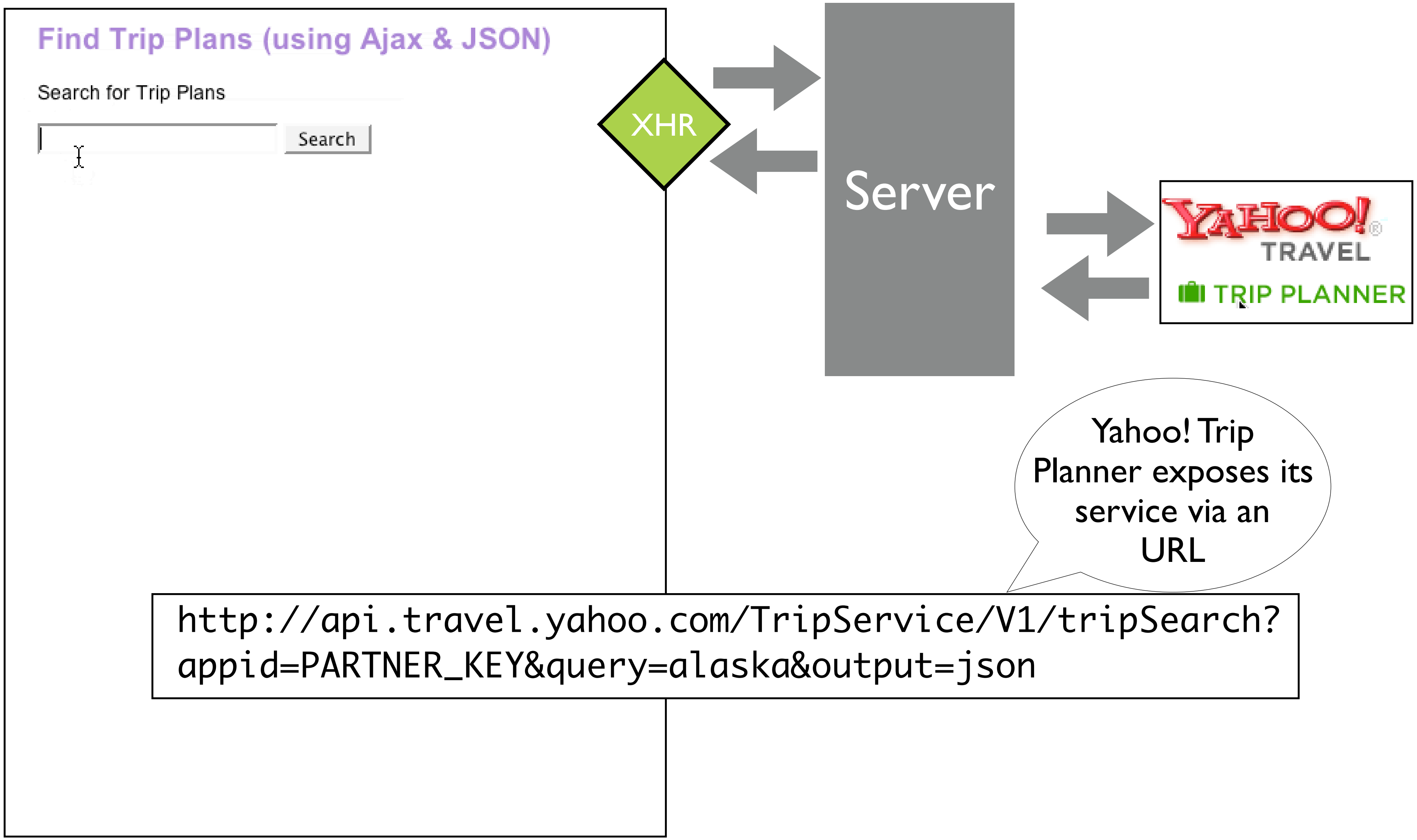
Degradability: Some Solutions

- Pre-emptive nag bar
- Semantic Markup
  - What's good for accessibility is good for degradability
- Design Issues
  - What is the experience for C-grade browsers?
  - What is the experience for non-browsers?

Web Services

- Web 2.0 - Web as a Platform. Lots of web services!
- YAHOO! DEVELOPER NETWORK
  - del.icio.us, Financ, Flick, HotJob, Maps, Merchant Solutions, Music, RSS Feeds, Search, Search Marketing, Shopping, Travel, Traffic, upcoming.org, weather, webjay, widgets
- Google™ Google Web APIs (beta)
  - maps, search, desktop, sitemaps, adwords

Web Services Example


Find Trip Plans (using Ajax & JSON)
Search for Trip Plans
Search
XHR
Server
YAHOO! TRAVEL
TRIP PLANNER
Yahoo! Trip Planner exposes its service via an URL

```
http://api.travel.yahoo.com/TripService/V1/tripSearch?
appid=PARTNER_KEY&query=alaska&output=json
```

Web Services, Some Choices: XML vs JSON

- JSON means
  - Trivial parsing
  - Faster than XML
  - Friendlier for developer
  - Runs everywhere
  - It's JavaScript, simpler programming model
  - Very stable... never will change!
  - Strong community, wide acceptance
- Yahoo! behind it

Security: Same Site Rule

- The domain of the URL request destination must be same as one that serves up page containing script
- Why? XHR requests to a different site would carry cookies. Cookies would spoof authentication.
- Solutions
  - Proxies
  - <script> hack
  - Other remote scripting tricks
  - JSONRequest (coming)

Security: JSONRequest

- I want to access multiple services (from different domains) without setting up a separate server (scalable, simpler to implement)
- Solution: Protocol that POSTs JSON text, gets response, parses the response into JavaScript value
  - No cookies sent
  - No other file formats accepted. Must be valid JS
  - JSON is safe JavaScript (data not functions)
  - Little or no error information provided to a miscreant
  - Accumulates random delays before trying again to frustrate denial of service attacks
  - Can support duplex!

Memory Management

- Hey, Internet Explorer is leaky!
  - Its memory garbage collector is the culprit
    - Uses reference counting; JS uses Mark & Sweep
    - Often closures are blamed
- Common problem that causes memory leaks
  - DOM <--> Application object reference each other
  - Event handlers left hanging around
- Tool for IE: Drip

JavaScript Performance

- JavaScript requires common sense
  - Cache repeatedly accessed objects
  - Always use var for local variables
  - Use numbers for number and strings for strings
  - Avoid using eval() for accessing properties
    eval("obj."+propName) --> obj[propName]
  - Look to your loops
- And a few surprises
  - Build DOM trees downward
  - Use array.join for lengthy string concatenations

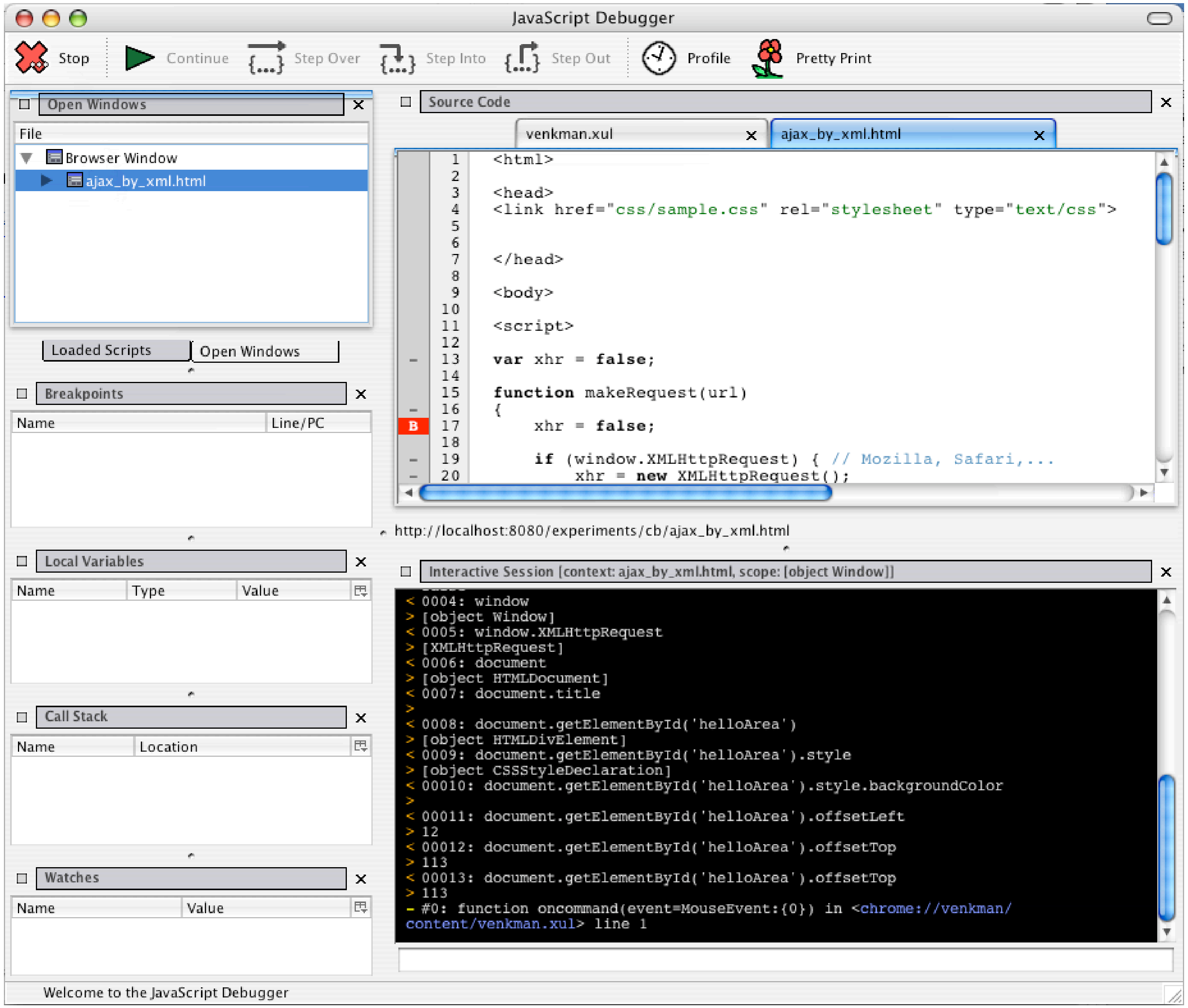
source: Adam Platti

Debugging Tools

- Microsoft Script Editor
- Instant Source (IE Plugin)
- IE Dev Toolbar

- Venkman (Mozilla)
- DOM Inspector (Mozilla)
- Web Developer Tools (Mozilla)
- Safari JavaScript Console
- JSLint

Debugging: Venkman



```
<html>

<head>
<link href="css/sample.css" rel="stylesheet" type="text/css">


</head>

<body>

<script>

var xhr = false;

function makeRequest(url)
{
    xhr = false;

    if (window.XMLHttpRequest) { // Mozilla, Safari,...
        xhr = new XMLHttpRequest();
```

http://localhost:8080/experiments/cb/ajax_by_xml.html

```
< 0004: window
> [object Window]
< 0005: window.XMLHttpRequest
> [XMLHttpRequest]
< 0006: document
> [object HTMLDocument]
< 0007: document.title
>
< 0008: document.getElementById('helloArea')
> [object HTMLDivElement]
< 0009: document.getElementById('helloArea').style
> [object CSSStyleDeclaration]
< 00010: document.getElementById('helloArea').style.backgroundColor
>
< 00011: document.getElementById('helloArea').offsetLeft
> 12
< 00012: document.getElementById('helloArea').offsetTop
> 113
< 00013: document.getElementById('helloArea').offsetTop
> 113
- #0: function oncommand(event=MouseEvent:{0}) in <chrome://venkman/
content/venkman.xul> line 1
```

- Breakpoints
- Local variables
- Watches
- Call Stack
- Profiling
- Source view
- Debug toolbar

Debugging Prevention: JSLint

- Lint are syntax checkers and validators
- JavaScript needs lint
- http://www.crockford.com/jslint/lint.html
- Scans source file & looks for common syntax problems
  - Nice way to catch silly mistakes
  - Try it at: http://jslint.com

Obsfucation or Minification

- JavaScript is easy to see with a Save or a <ctrl>-U
- JavaScript can increase page size & page load
  - Obsfucators mangle the code into unreadable form
  - Minifiers strip white space & comments
- Obsfucators go too far
  - Makes development way too hard
- Minification & compression do just enough
  - JSMin (http://www.crockford.com/javascript/jsmin.html)

Error Handling

- Asynchronous error handling
  - Keep design simple (do you need multiple requests going at once?)
  - Will increase implementation issues
- Normal error handling
  - Check for error codes (!=200)
  - Roll your own HTTP status error codes
- Minimize server-side errors with intentional validation (error prevention)

Error Handling

- Difficult to correct XML, JSON or even HTML over the wire. XML & JSON are slightly harder. Server issue, but how to handle
- Use error handlers provided by Ajax frameworks
  - Prototype try.these
  - JavaScript try/catch
  - DWR has global error handler
  - Dojo error handlers

Questions?