

Developing for Ajax

Ajax 101

A Look Under the Hood



Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

Learning Ajax

- Learning Ajax, means understanding
 - **Triggers** (event or timer)
 - **Operations** (ajax xhr)
 - **Updating** (dom)

*Triggering
Events &
Timers*

*Using XHR
for Ajax
Operations*

*Updating
the DOM*



Operation. Using XHR

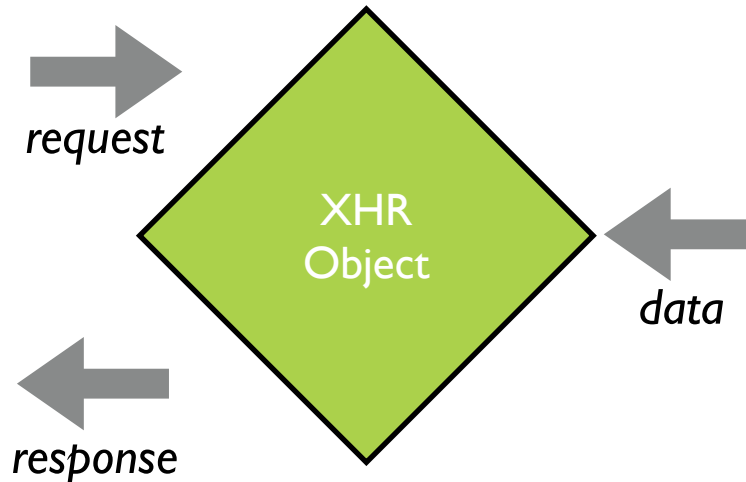
Simple Ajax 'Hello World'

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>
```

/response.xml

```
<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('test.xml')">
Make an ajax request for data
</span>

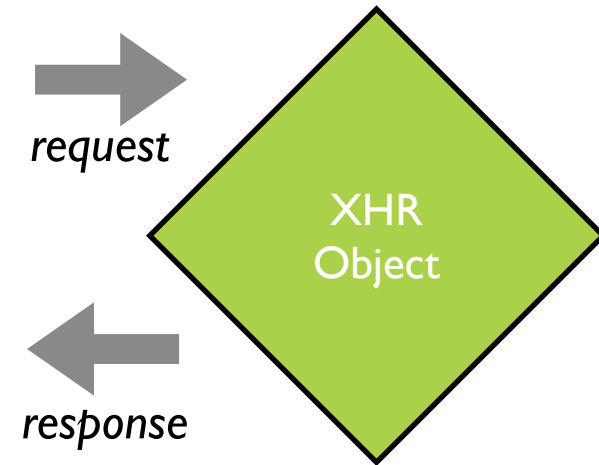
<div id="helloArea"></div>
```

/index.html

- 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. Make the request
3. Handle the response



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, 5.5, 6 implements XHR as an ActiveX object (Msxm12.XMLHTTP/Microsoft.XMLHTTP)
- Mozilla 1.0+, Safari 1.2+, Opera 8+, IE7 provide XMLHttpRequest object natively
- All XHR objects have same methods & properties

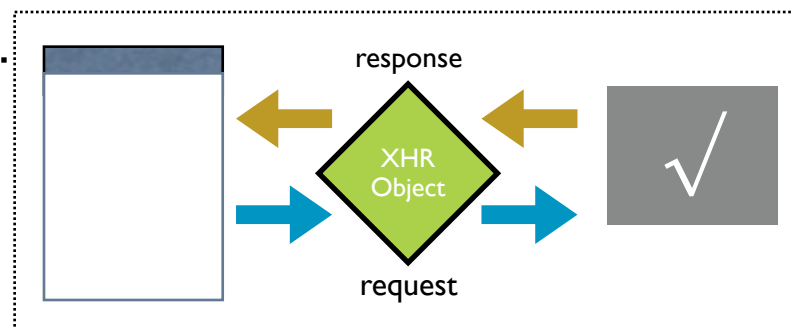
set `onreadystatechange` to callback function - `handleAjaxResponse`

open a request on the `xhr` object

send the request through the `xhr` object

- The JavaScript function *handleAjaxResponse* will be invoked when the `readystatechange` property changes on the XHR object
- Same site rule
- 'GET' or 'POST'
- Asynchronous flag

server must be accessible via relative url



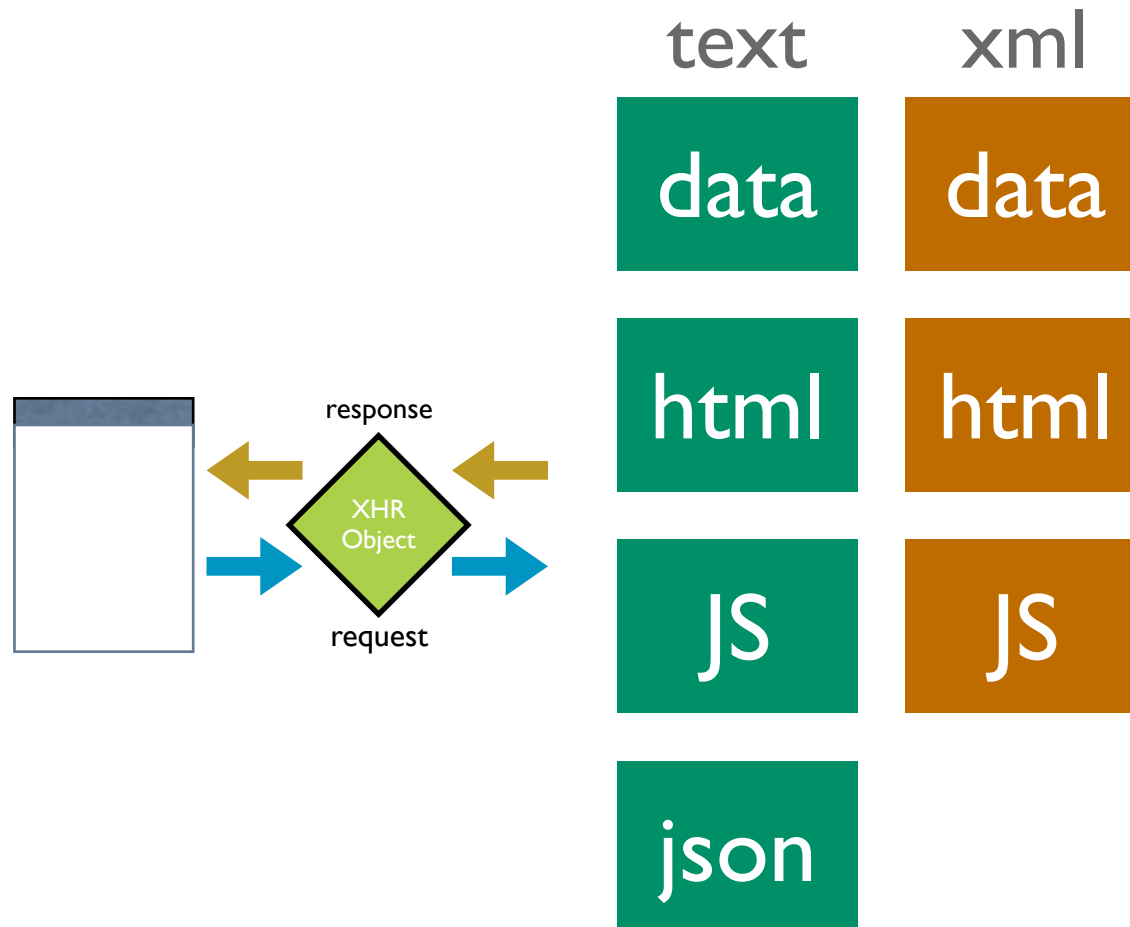
```
function handleAjaxResponse
begin
    if response is valid then
        get responseXML
        get rootNode
        get helloArea on the page
        stuff the rootNode value into the helloArea DIV
    endif
end
```

- **readystate** values

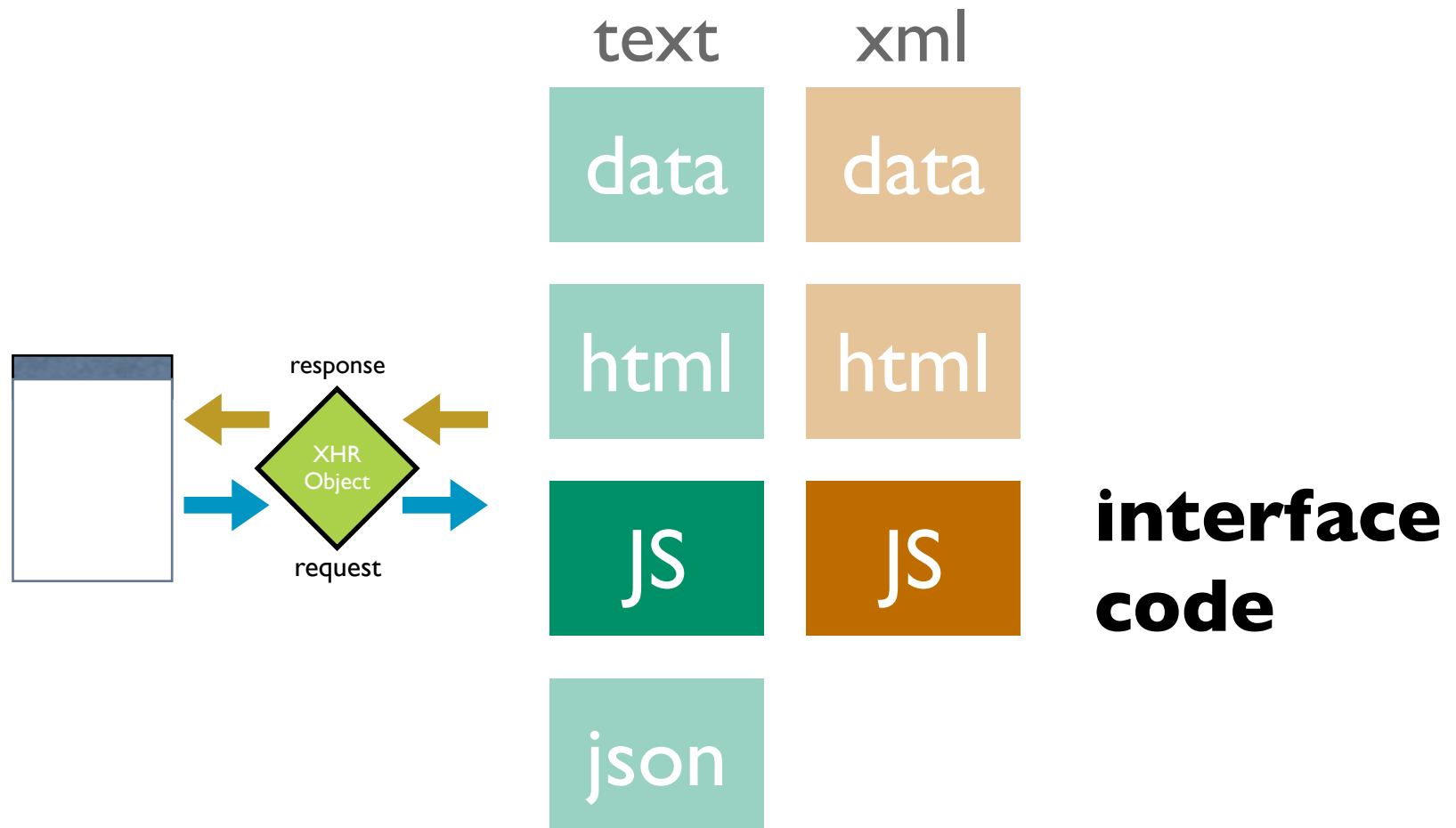
- 0 - Uninitialized
- 1 - Loading
- 2 - Loaded
- 3 - Interactive
- 4 - Completed

- Use XHR property **responseXML** to get the response 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>)

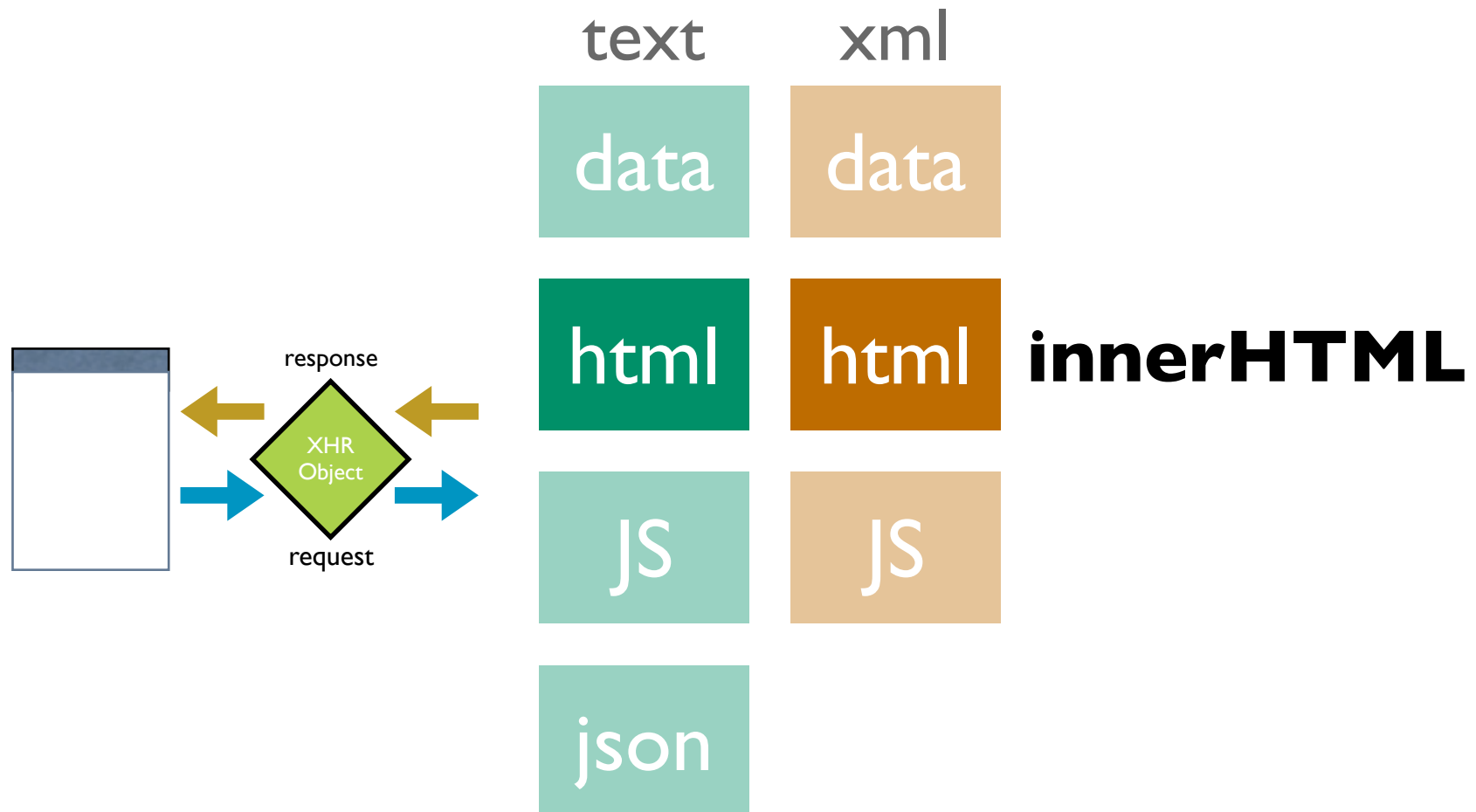
- Two types: `responseText`, `responseXML`
- Many forms: data, html, JS, JSON



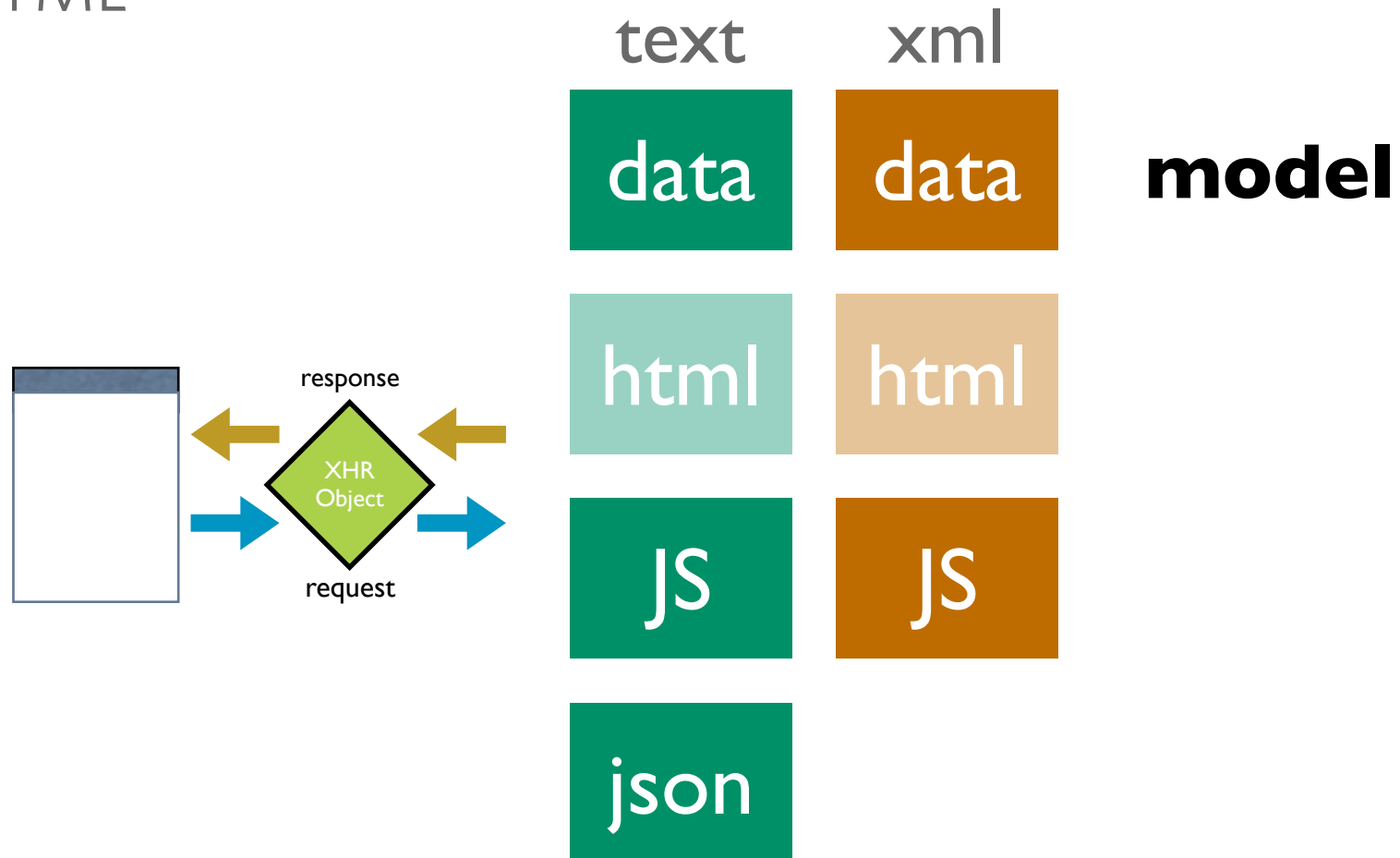
- JavaScript generated on server-side
- JavaScript is eval'ed on client side to generate GUI
- Dangerous!



- HTML generated on server side
- Stuffed into innerHTML of DOM element



- Model data generated on server side
 - w3c DOM elements created from model data
 - Substitute model data into HTML strings and insert via innerHTML



```
// var helloArea = YAHOO.util.Dom.get("hello-area");  
var helloArea = document.getElementById("hello-area");  
var para = document.createElement("p");  
var paraText = document.createTextNode("Hello ");  
var em = document.createElement("em");  
var emText = document.createTextNode("World");  
helloArea.appendChild(para);  
para.appendChild(paraText);  
para.appendChild(em);  
em.appendChild(emText);
```

```
<p>A message will be inserted below:</p>  
<div id="hello-area" style="height:20px;width:220px;  
margin-top:8px;padding:4px;border:1px solid blue"></div>
```

A message will be inserted below:

A message will be inserted below:

Hello *World*

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)

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

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


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

array of 7 named days

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

3x3 array

- 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

- Obtain responseText
- Parse the responseText

```
responseData = eval('(' + responseText + ')');
```

OR

```
responseData = JSON.parseJSON(responseText);
```

- <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"
    }
  }
}
```

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

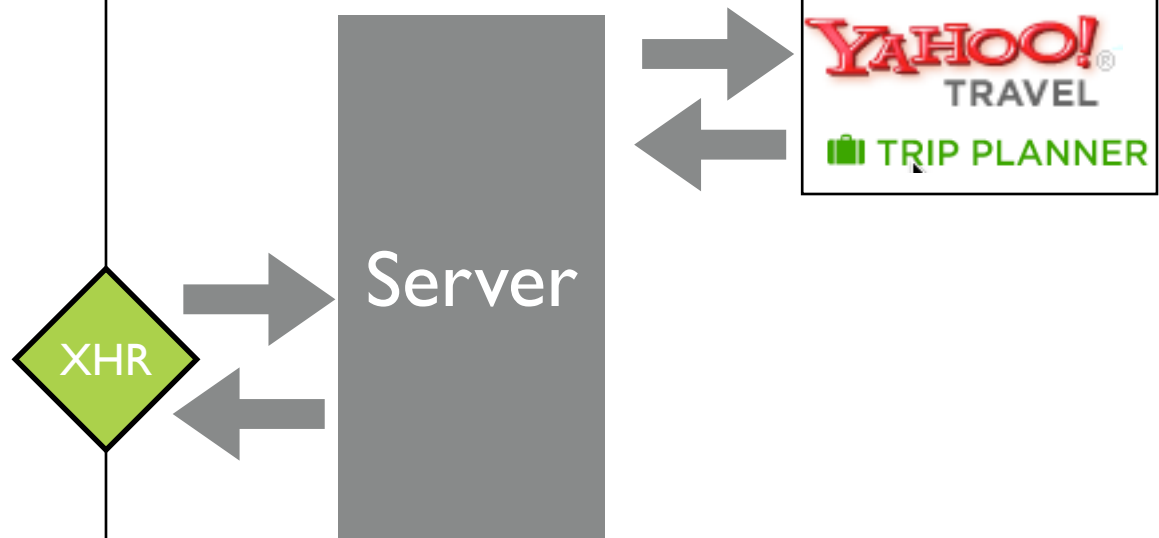
```
{
  "ResultSet":
  {
    "totalResultsAvailable": "69200000",
    "totalResultsReturned": "1",
    "firstResultPosition": "1",
    "Result":
    {
      "Title": "Yahoo! Finance",
      "Summary": "manage the market and your money with Yahoo! Finance. Includes stock market quotes, business news, 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"
    }
  }
}
```

```
responseData = JSON.parseJSON(responseText);
var resultsAvail = responseData.ResultSet.totalResultsAvailable;
var resultsReturned = responseData.ResultSet.totalResultsReturned;
var firstResultTitle = responseData.ResultSet.Result.Title;
var firstResultSummary = responseData.ResultSet.Result.Summary;
```

JSON Trip Finder

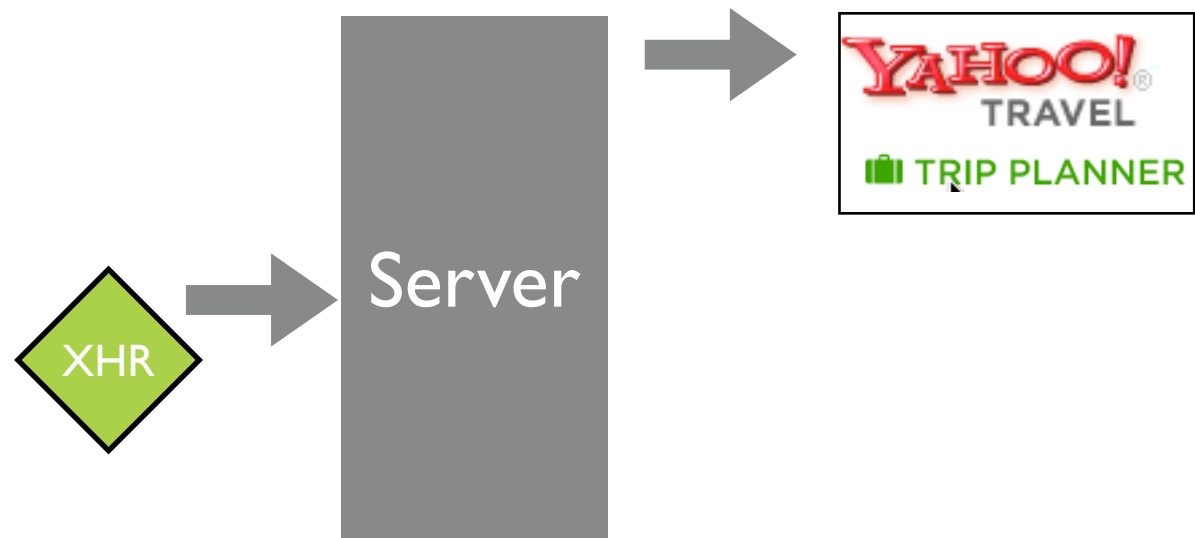
Find Trip Plans (using Ajax & JSON)

Search for Trip Plans



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`



Each response is represented as an object in an array (list) of responses

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

```
{
  "ResultSet": {
    "firstResultPosition": 1,
    "totalResultsAvailable": "16",
    "totalResultsReturned": 10,
    "Result": [
      {
        "id": "351102",
        "YahooID": "j",
        "Title": "from Vancouver",
        "Summary": "Hiking,... Hotel: Hyatt R Jho...",
        "Destinations": "Vancouver, Ketchikan, Se...",
        "CreateDate": "113043",
        "Image": {
          "Url": "http://us.i1.yimg.com/.../100x100_cde24409e413d4d6da27953f6ab0bbae.jpg",
          "Height": "100",
          "Width": "66"
        },
        "Geocode": {
          "Latitude": "49.284801",
          "Longitude": "-123.120499",
          "precision": "not available"
        },
        "Url": "http://travel.yahoo.com/trip/?pid=351102&action=view"
      },
      {
        "id": "400333",
        "YahooID": "brdway_grl",
        "Title": "Alaska",
        "Summary": "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/.../100x100_cde24409e413d4d6da27953f6ab0bbae.jpg",
          "Height": "75",
          "Width": "66"
        },
        "Geocode": {
          "Latitude": "61.190361",
          "Longitude": "not available"
        },
        "Url": "http://travel.yahoo.com/trip/?pid=400333&action=view"
      }
    ]
  }
}
```

Server

XHR

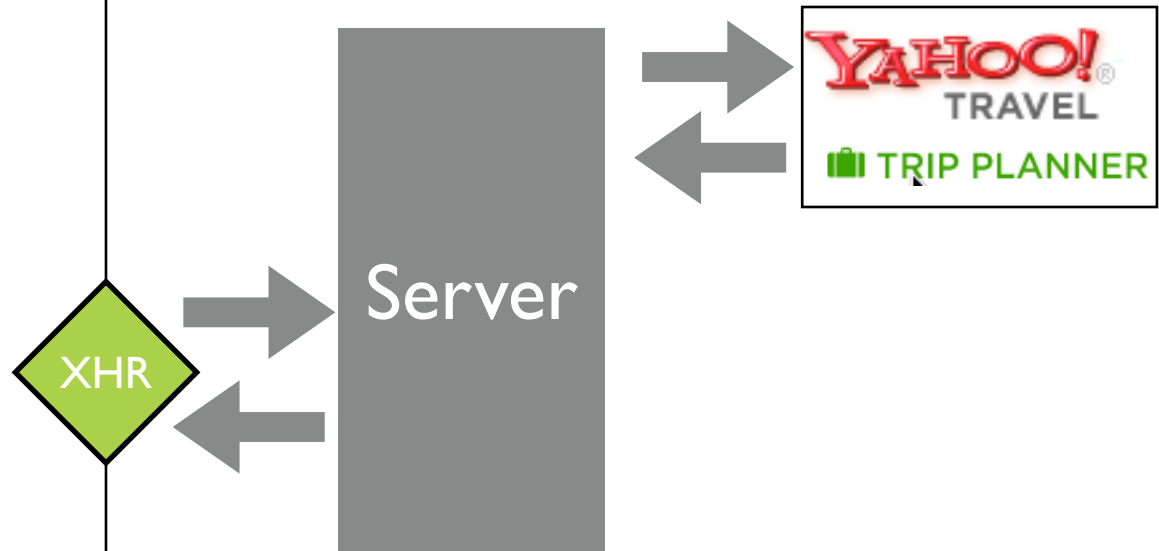



```
<script>
function showResponse() {
    if (xhr.readyState == 4) && (xhr.status == 200) {
        var helloArea = document.getElementById("helloArea");
        var theTrip = eval( '(' + xhr.responseText + ')' );
        var tripPlanHTML = "";
        for(var i=0; i<theTrip.ResultSet.totalResultsReturned; i++) {
            var result = theTrip.ResultSet.Result[i];
            tripPlanHTML = tripPlanHTML +
                '<div style="padding:4px;border:1px solid gray;width:' +
                result.Image.Width +
                ';"></img></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.');
```

JSON Trip Finder

Find Trip Plans (using Ajax & JSON)

Search for Trip Plans





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
onDbClick	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

- Element Attributes
 - `<element onclick="func()" >`
- Element Properties (function reference)
 - `element.onclick=func;`
- Event Binding

Internet Explorer	Mozilla (FF), Safari, Opera [W3C]
<code>attachEvent()</code> <code>detachEvent()</code>	<code>addEventListener()</code> <code>removeEventListener()</code>

▪ Element Attributes

```
<a href="#" onclick="clickHandler(this)">
function clickHandler(anchorDOMElem) {
    // this == window --> window owns the function
    // anchorDOMElem was set to anchor DOM element
    this = anchorDOMElem; // fix this pointer
}
```

▪ Element Properties

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

▪ Event Binding

```
function AnchorLink(anchorDOMElem) {
    this.anchorDOMElem = anchorDOMElem;
    this.anchorDOMElem.onclick = this.clickHandler;
    this.anchorDOMElem.anchorLinkObj = this;
}
AnchorLink.prototype.clickHandler = function() {
    // this == anchorDOMElem, not AnchorLink object
    // confusing since this normally refers to AnchorLink
    // grab our normal this
    anchorLinkObj = this.anchorLinkObj;
    anchorLinkObj.doIt();
}
AnchorLink.prototype.doIt = function() {
    // this == AnchorLink Object
}
```


- Memory Leaks
 - IE's garbage collection does simple reference counting
 - Attaching events and not removing them when finished will cause memory leaks
 - Can use an Observer style pattern to cache event handlers and at the end clean them up
- Timing
 - Attempting to attach events before the page is loaded will cause problems
 - Do it on the onload

- Standardizes event binding

- Attach an event

```
var helloAreaDiv = document.getElementById("hello-area");  
function handleClick(e) { alert("click"); }  
YAHOO.util.Event.addListener(helloAreaDiv, "click", handleClick);
```

- Attach multiple events

```
var ids = ["el1", "el2", "el3"]; // array can contain object references, element ids, or both  
function handleClick(e) { alert(this.id); }  
YAHOO.util.Event.addListener(ids, "click", handleClick);
```

- Handles automatic scope correction

- By default “this” refers to the DOM element that the event was attached to
 - Can be overridden

- First parameter is the event object

- Pass arbitrary objects to event handler

- Automatic Listener Cleanup
- Custom Events
 - For custom objects or overriding events on widgets
- onAvailable event
 - Define a function that will execute as soon as an element is detected in the DOM (during load)

```
<script type="text/javascript">
```

```
function TestObj(id) { YAHOO.util.Event.onAvailable(id, this.handleOnAvailable, this); }
```

```
TestObj.prototype.handleOnAvailable = function(me) { alert(this.id + " is available"); }
```

```
var obj = new TestObj("myelementid"); </script>
```

```
<div id="myelementid">my element</div>
```



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. Specifically the Browser 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

- Think of JavaScript as the DOM manipulator
- Use it to
 - Find DOM elements
 - Add new elements
 - Remove elements
 - Modify element attributes
 - Position elements
 - Style elements

Finding DOM Elements

- `document.getElementById`
 - Prototype library shortcut: `$("#idName")` or `$(id)`
 - Yahoo! library shortcuts:
 - `YAHOO.util.Dom.get("idName")`
 - `getElementByClassName`
- `parentNode`
- `childNodes`
 - Yahoo! library has ways to find ancestors

DOM Manipulation

- Creating new interface elements
 - `innerHTML`, `createElement()`, `createTextNode()`, `appendChild()`
- Changing element styles
 - Visual attributes
 - Geometry
- Y!UI positioning, region detection
- Y!UI classname & style normalization

```
<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;
```

```
...
<script type="text/javascript">
YAHOO.example.DDApp = function() {
    var dd;
    return {
        init: function() {
            var dd = new YAHOO.util.DD("dragDiv");
        }
    }
} ();
YAHOO.util.Event.addListener(window, "load",
YAHOO.example.DDApp.init);
</script>

<style type="text/css">
#dragDiv {
    background:url(images/macosexlogo.gif) 0 0 no-repeat;
    height:100px;
    width:70px;
}
</style>
</head>

<body>
<h2>Drag and Drop</h2>
<p>The Mac logo is draggable</p>

<div id="dragDiv"></div>

</body>
...
```

Drag and Drop

The Mac logo is draggable



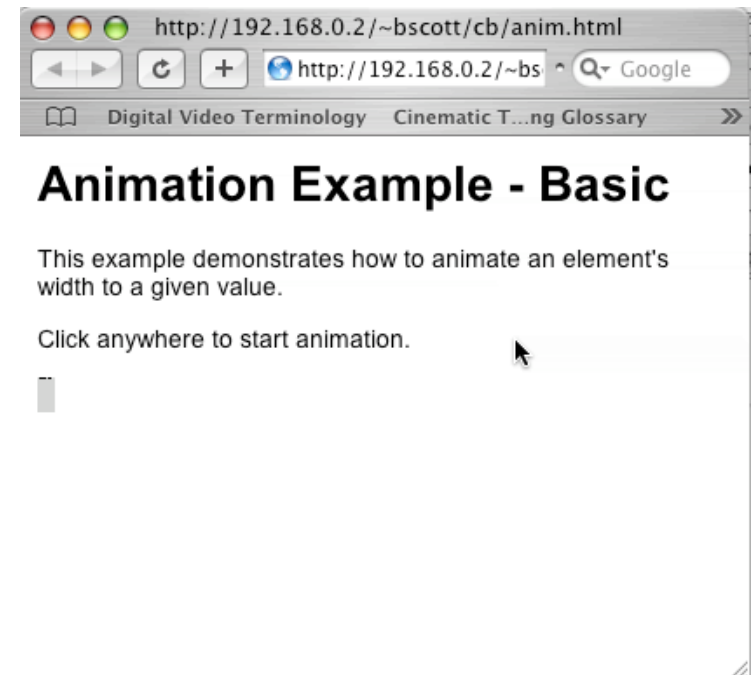
```
<html>
<head>
<style>
#anim { background:#ccc;width:10px;height:20px;font-size:10%;}
</style>
<script type="text/javascript">
YAHOO.namespace('example.anim');
YAHOO.example.anim.init = function() {
    var myAnim = new YAHOO.util.Anim('anim', {
        width: {to: 200},
        height: {to: 150},
        fontSize: {from: 20, to: 120, unit: '%'},
        opacity: {from: 0.25, to: 1.0 },
        1,
        YAHOO.util.Easing.easeOut);

    var animate = function(e) {
        myAnim.animate();
        return false;
    }
    YAHOO.util.Event.addListener(document, 'click',
                                animate);
}
YAHOO.util.Event.addListener(window, 'load',
                                YAHOO.example.anim.init);

</script>
</head>
<body>
<h1>Animation Example - Basic</h1>
<p>This example demonstrates how to animate
an element's width to a given value.</p>
<p>Click anywhere to start animation.</p>

<div id="anim">Lorem ipsum dolor</div>

</body>
</html>
```



Developing for Ajax

Advanced Topics

Problems and Challenges with Building Ajax Applications



Bill W. Scott, Y! Ajax Evangelist

bscott@yahoo-inc.com

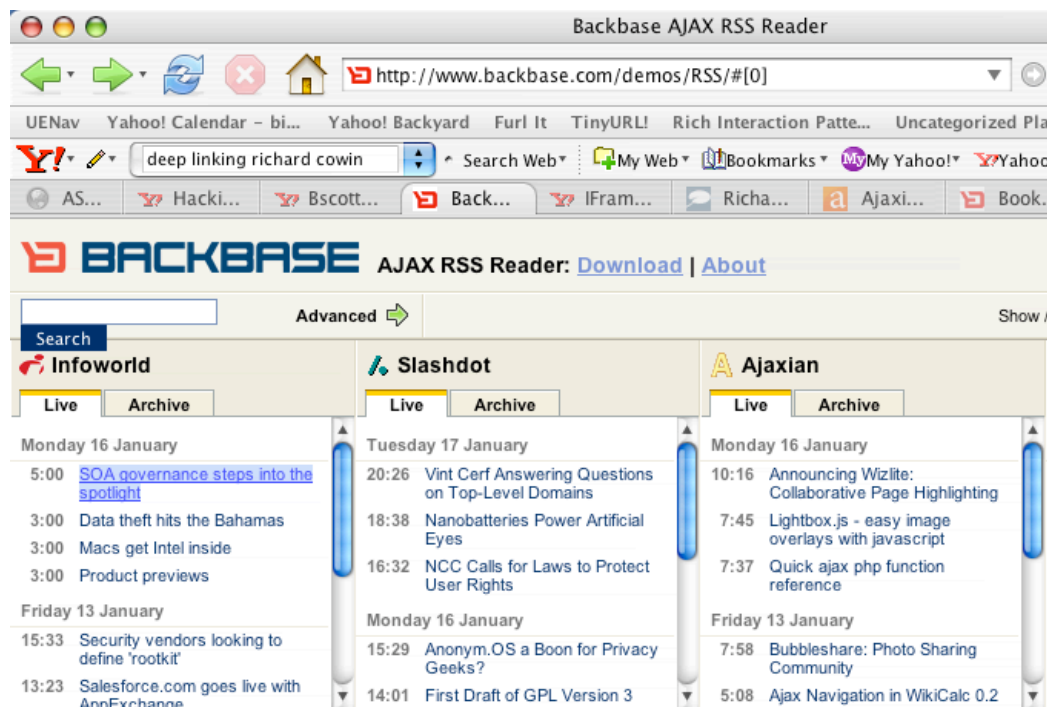
How Ajax Changes Things

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, obfuscation, 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?

- 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



- 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
- RSH - Really Simple History.
<http://www.onjava.com/pub/a/onjava/2005/10/26/ajax-handling-bookmarks-and-back-button.html>

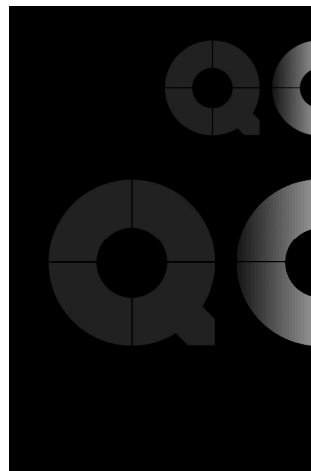
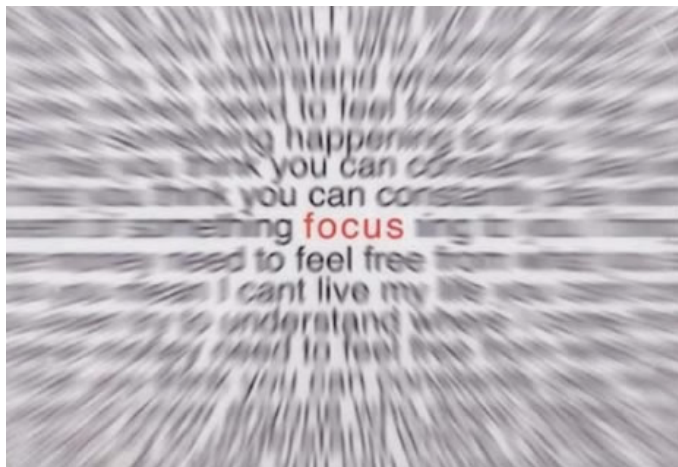
Bookmarking

- 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.

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

 [Print](#)  [Email](#)  [Link to this page](#)

Accessibility



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)

<http://www.mozilla.org/access/>

http://developer.mozilla.org/en/docs/Accessible_DHTML

- 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

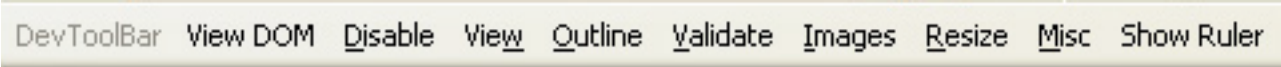
Security: Same Site Rule

- 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
 - XMLHttpRequest (coming)

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

Debugging Tools

- Microsoft Script Editor
- Instant Source (IE Plugin)
- IE Dev Toolbar 
- Venkman (Mozilla)
- Firebug (Mozilla)
- DOM Inspector (Mozilla)
- Web Developer Tools (Mozilla)
- Safari JavaScript Console
- JSLint

DHTML Load - Carousel

An array of images are randomly chosen on load and 'next' events, simulating loading data from within the client space.



Also see the [documentation](#)

Done



- 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>

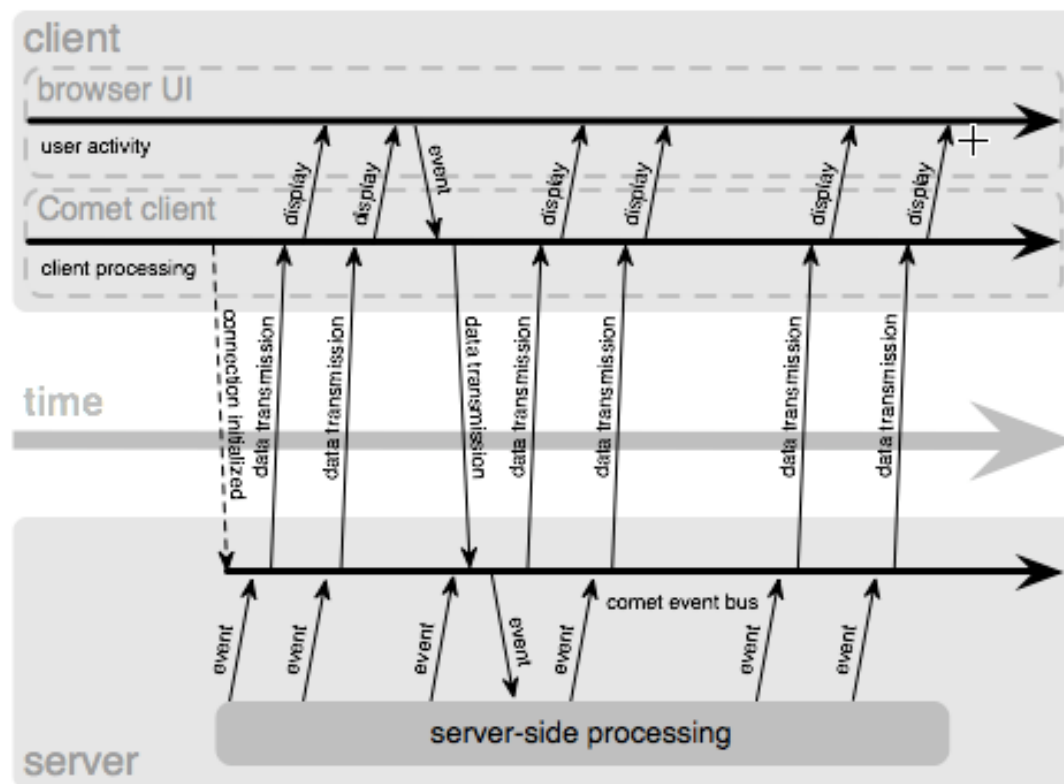
Error Handling

- 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 ($\neq 200$)
 - Roll your own HTTP status error codes
- Minimize server-side errors with intentional validation (error prevention)

Comet

- Can deliver data to client any time
 - Delivered over single, previously opened connection
- Example applications
 - gmail's GTalk integration
 - Jot Live
 - Renkoo
 - Meebo

Comet web application model



What about Flash?

- Adobe Flash/JS Bridge (finance.google.com)
- Adobe Flex/Ajax Bridge (8.5)
- Flash sites (maps.yahoo.com/beta)
- Flex: 2.0 is serious environment (Flash 8.5; free)
- Laszlo: Generates either Flash or DHTML/Ajax
- Communication Bridge
- Storage Mechanism
- Event Extender
- Rich Media Extender (sound, video, streaming)



Questions?