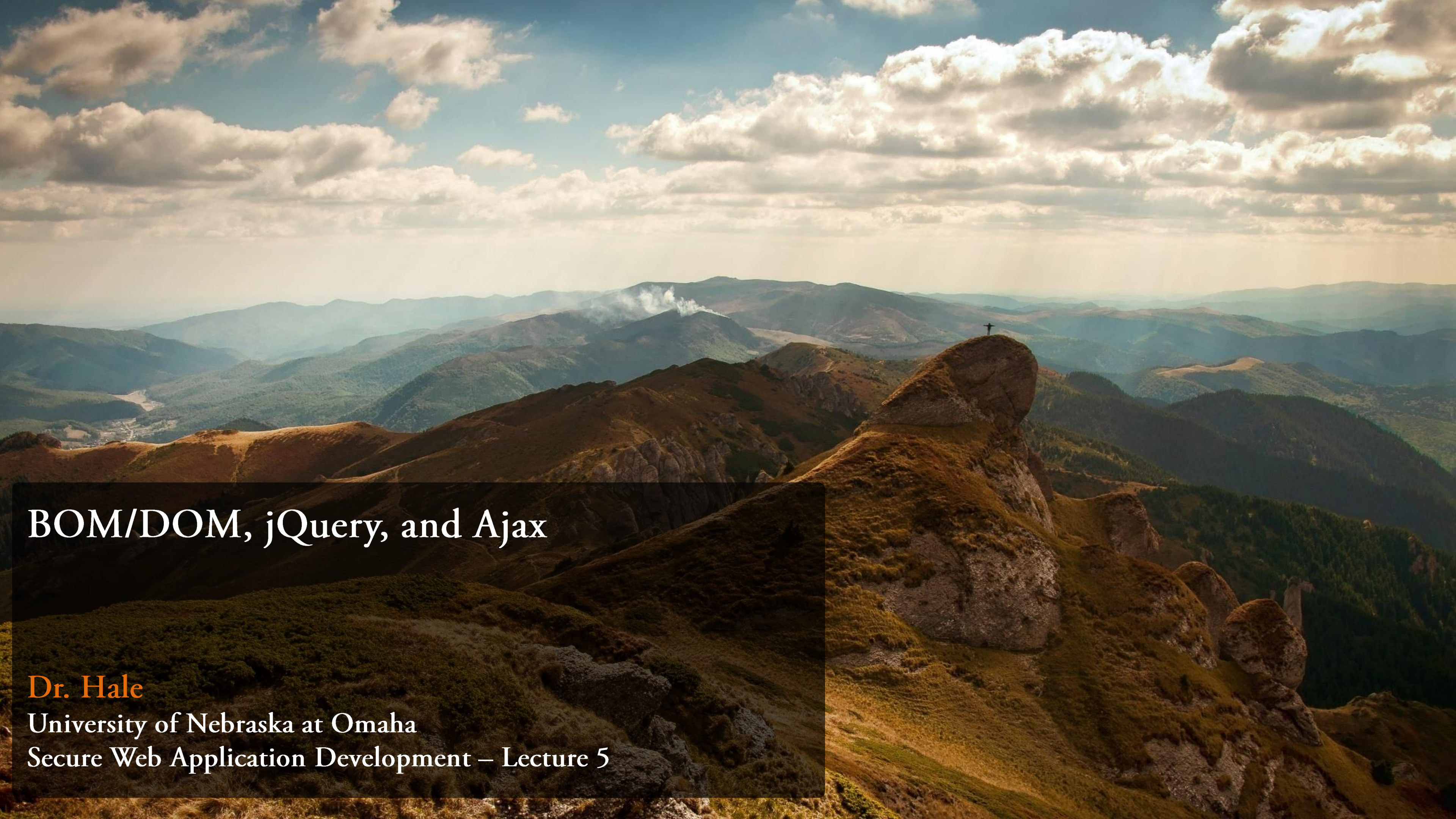


BOM/DOM, jQuery, and Ajax

Dr. Hale
University of Nebraska at Omaha
Secure Web Application Development – Lecture 5

Today's topics:

BOM/DOM Overview

- In more detail
- How is it used for javascripting

jQuery

- Basic syntax overview
- jQuery selectors
- Functions and effects
- Animation
- Event handling

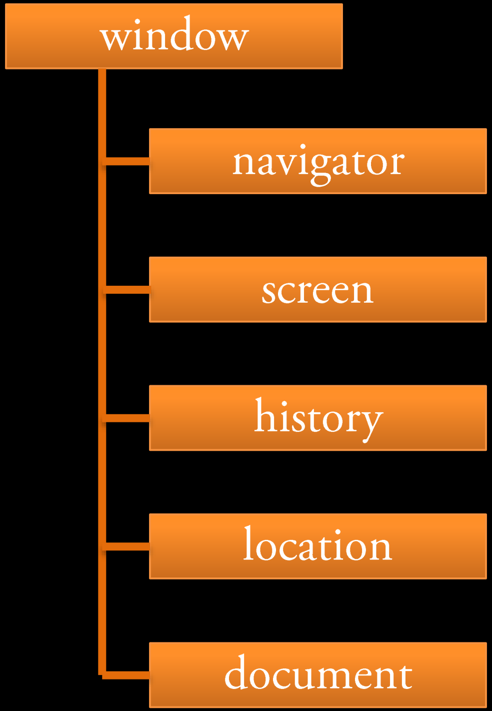
Ajax

- Callbacks and Ajax
- Promises and Ember integration

BOM/DOM

Browser
Object
Model

BOM Overview



Top level object representing the browser

Information about browser and OS

Client display capabilities

Recently visited sites information

Current URL Information

Everything that constitutes the current web page (the DOM)

B
O
M

O
v
e
r
v
i
e
w

a little ‘history’

- Until recently different browsers implemented the BOM differently
    - to the bane of every web developer ever
    - increasing standardization these days has led a ubiquitous BOM (finally)
- The BOM is the browsers API

BOM Overview

Navigator

Property	Description
appCodeName	Returns the code name of the browser
appName	Returns the name of the browser
appVersion	Returns the version information of the browser
cookieEnabled	Determines whether cookies are enabled in the browser
language	Returns the language of the browser
onLine	Determines whether the browser is online
platform	Returns for which platform the browser is compiled
product	Returns the engine name of the browser
userAgent	Returns the user-agent header sent by the browser to the server

Note: Methods are linked to w3c definitions if viewing online

BOM Overview

Screen

Property	Description
availHeight	Returns the height of the screen (excluding the Windows Taskbar)
availWidth	Returns the width of the screen (excluding the Windows Taskbar)
colorDepth	Returns the bit depth of the color palette for displaying images
height	Returns the total height of the screen
pixelDepth	Returns the color resolution (in bits per pixel) of the screen
width	Returns the total width of the screen

Note: Methods are linked to w3c definitions if viewing online

BOM Overview

History

Property	Description
length	Returns the number of URLs in the history list

Method	Description
back()	Loads the previous URL in the history list
forward()	Loads the next URL in the history list
go()	Loads a specific URL from the history list

Note: Methods are linked to w3c definitions if viewing online

BOM Overview

Location

Property	Description
hash	Sets or returns the anchor part (#) of a URL
host	Sets or returns the hostname and port number of a URL
hostname	Sets or returns the hostname of a URL
href	Sets or returns the entire URL
origin	Returns the protocol, hostname and port number of a URL
pathname	Sets or returns the path name of a URL
port	Sets or returns the port number of a URL
protocol	Sets or returns the protocol of a URL
search	Sets or returns the querystring part of a URL

Note: Methods are linked to w3c definitions if viewing online

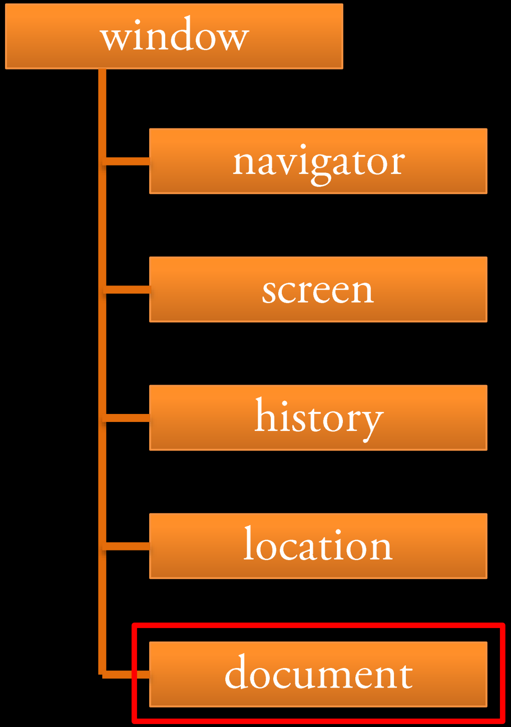
BOM Overview Location

Method	Description
assign()	Loads a new document
reload()	Reloads the current document
replace()	Replaces the current document with a new one

Note: Methods are linked to w3c definitions if viewing online

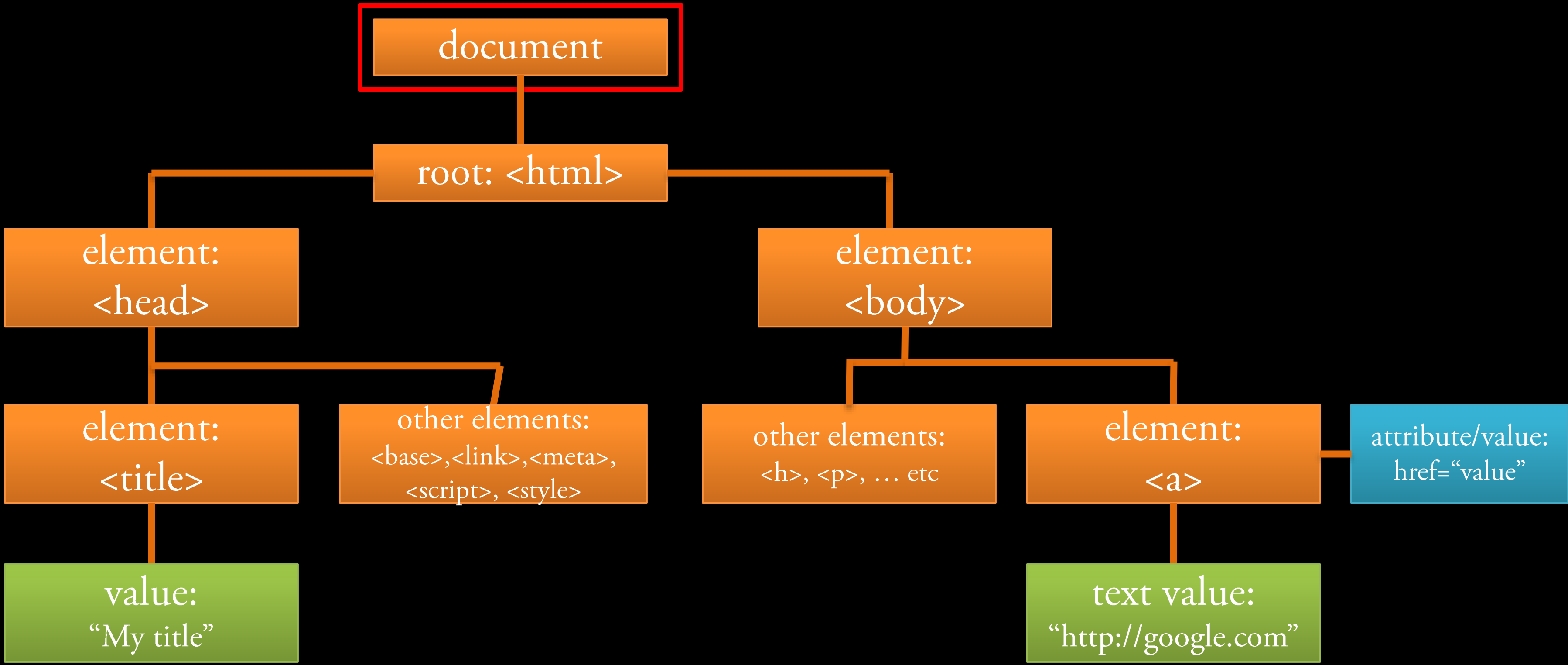
Document
Object
Model

DOM Overview



D
O
M

O
v
e
r
v
i
e
w


document
root: <html>
element: <head>
element: <body>
element: <title>
other elements: <base>,<link>,<meta>, <script>, <style>
other elements: <h>, <p>, … etc
element: <a>
attribute/value: href="value"
value: "My title"
text value: "http://google.com"

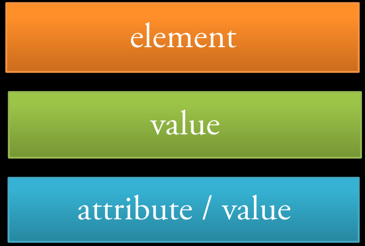
IT’S A TREE!

D
O
M

O
v
e
r
v
i
e
w

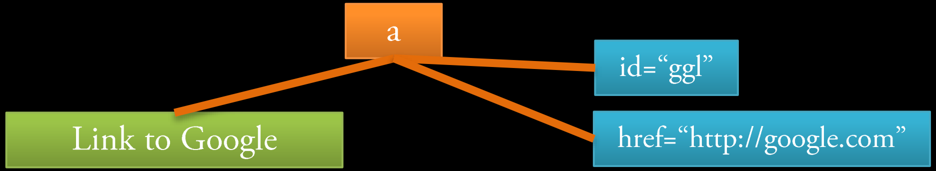
elements

- may have attributes
  - id, name, href, {{inert your own}}
- may have values
  - text, images, javascript, etc
- are composable
  - can be nested in different ways hierarchically

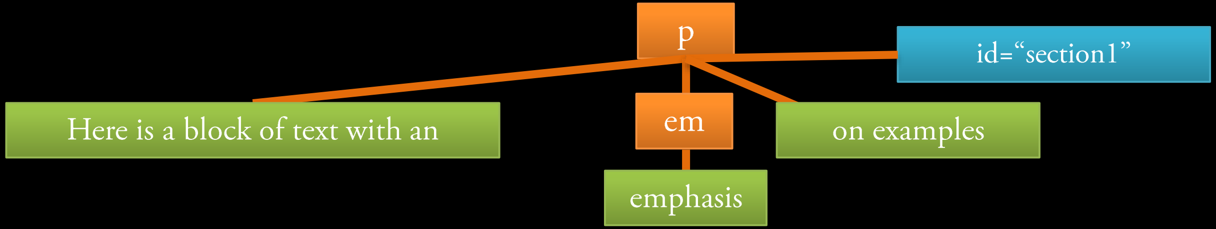


Example:

<a id="ggl" href="http://google.com">Link to Google</a>



<p id="section1" style="some CSS;">Here is a block of text with an <em> emphasis </em> on examples. </p>



DOM Overview

nodes/parents/siblings

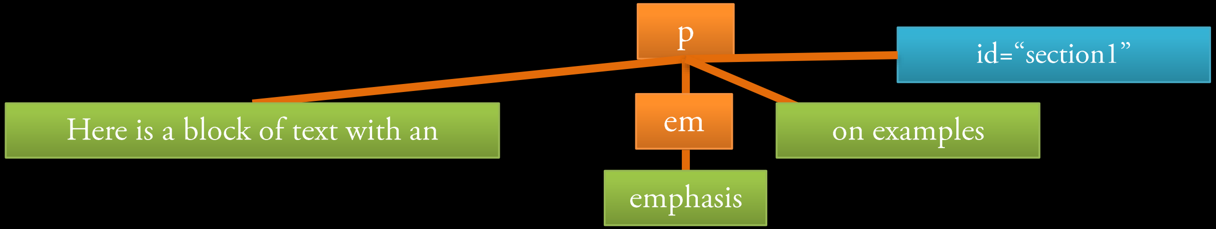
- each item can be seen as a 'node'
- Using this hierarchy, javascript can iterate over nodes, parents, and siblings
- e.g., em.parentNode (referencing the em below) would return the p element
- p.nodeValue would return the combined text and em blocks

🟧 element

🟩 value

🟦 attribute / value

<p id="section1" style="some CSS;">Here is a block of text with an <em> emphasis </em> on examples. </p>



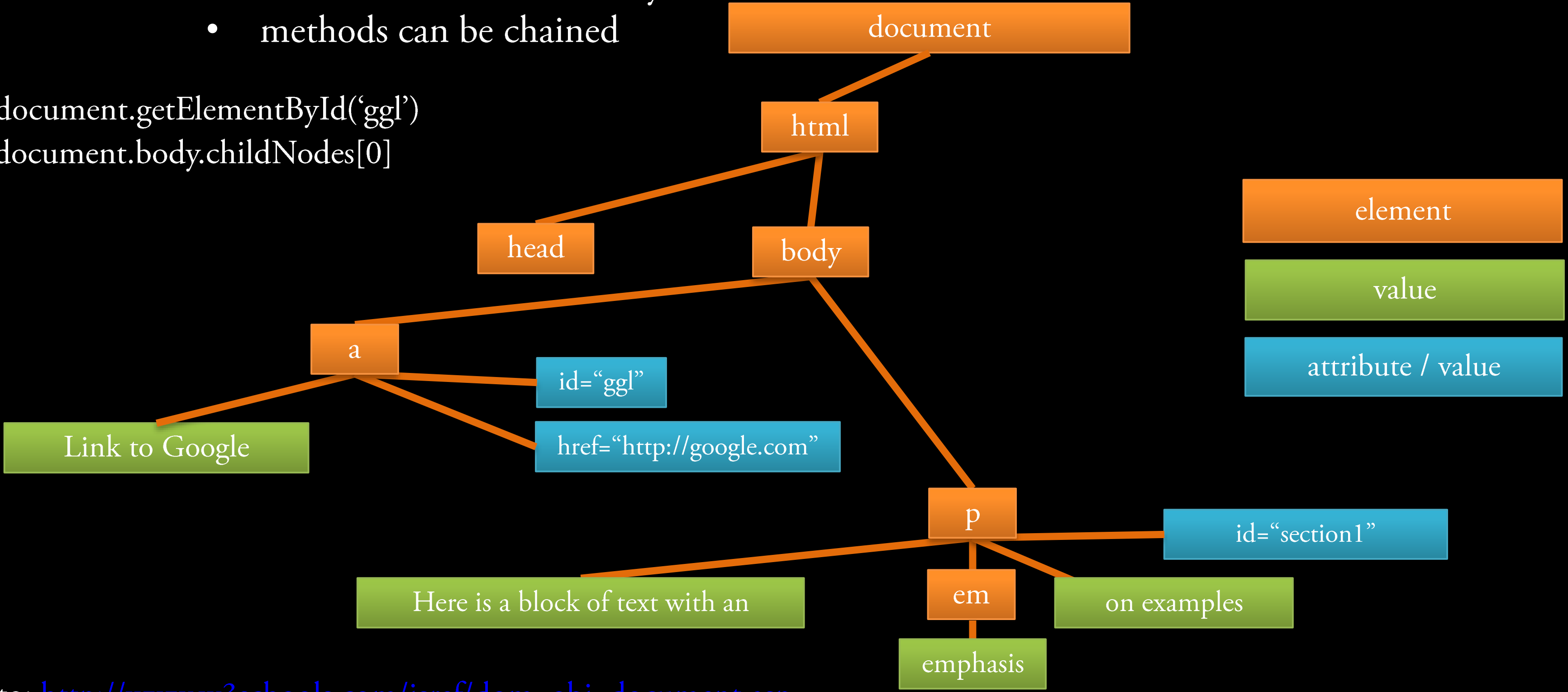
D
O
M

O
v
e
r
v
i
e
w

nodes (con't)

- can be referenced by type, id, etc
- can be traced hierarchically
- methods can be chained

```
document.getElementById('ggl')
document.body.childNodes[0]
```



refer to: http://www.w3schools.com/jsref/dom_obj_document.asp
for methods/properties of the document object

You’ve been using the DOM in ember all along.

Learn How to Use 'Magic'

jQuery

Legendary Sorcery

1 : **Use selector** *(Get an element from the DOM with ease)*

1 : **Apply a function***(Apply some client-side scripts to a DOM element)*

Wield the power of modernity

jQuery
write less, do more.
AJAX
Asynchronous Javascript And XML
M. Hale
™ & © 1993-2012 Wizards of the Coast, inc. 220774/220817

Overview

- jQuery is a library of javascript (JS) functions
- wraps basic DOM capabilities to
    - access DOM
    - modify DOM
    - animate elements
    - handle events
    - manage in-page server-side connections
- provides browser-independence for JS

Overview (con't)

- jQuery is open source and free
- massive community resources
- two versions
  - jquery-version-number.min.js (minified) – use for production
  - jquery-version-number.js (development) – use for coding
- easily included in your project (in head or body, body is best for speed)
  - <script src="../path-to-js-directory/jquery-version-number.js"></script>

http://jquery.com/

Philosophy

Facilitate interactions between javascript and HTML

(basically) Find DOM element –> do something to it

Syntax

$(selector).action()

- The $ symbol is the jQuery namespace (analogous to a package in java)
- the (selector) is the query that locates DOM element(s)
- An action() is a composable sequence of functions to apply to the selected elements.

Examples

- $("p").html("some text") – overwrite all <p> elements in the document to state 'some text'
- $("#content").fadeOut() – hide the element with id "content" using an animated fade effect
- $("#content").fadeTo(1000,.5) – change opacity of element with id content to 50% over a 1000ms time interval
- $(".row").hide() – hide all elements with a class attribute named "row"
- $(".row").width("600px") – change the 'width' property of all elements with a class named 'row'

NOTE: DEMO EET on jQuery page

Unpacking the Syntax

Basic Javscript

```
var c = document.getElementById('content');
c.innerHTML = 'test';
```

jQuery

```
$("#content").html('test');
```

NOTE: jQuery may implement multiple versions to handle different browser types

Whats up with the $ (bills)

- jQuery(args) is a selector function
- $ is a variable name in the global window space that references jQuery

i.e.

```
var $ = jQuery;
```

Hence

```
$("p").html("some text")
```

is equivalent to

```
jQuery("p").html("some text")
```

More on $(selector)

Selector can be:

- a CSS selector expression (all of examples shown so far)
- a string of HTML e.g., $(:contains(‘some html’))
- a javascript object e.g., $(myVar)

Most developers basically will only use the first type

NOTE: DEMO EET on jQuery page

http://api.jquery.com/category/selectors/

More on $(selector)

Selectors are tiered, chainable, composable, and can use DOM properties

(Tiered) $(“#content h2”)

//all <h2> elements under any element with id ‘content’

(Chain) $(“h2.block:contains(‘What is jQuery’)”)

//all <h2> elements with class block that contains the string ‘What is jQuery

(Composed) $(“h2.block:contains(‘What is jQuery’), section.project-tiles”)

//the (chain) dom elements plus all <section> elements of class “project-tiles”

(DOM OPs) $(“#corporate-members li:last-child”)

//the last <li> element under an element with id ‘corporate-members’ http://api.jquery.com/category/selectors/

$(selector) Summary

Syntax	Selector description
$("*")	All elements
$("this")	Current HTML element (useful for iteration or buttons)
$("element")	Element Selector: All elements of type 'element'
$(".class")	Class Selector: All elements with class 'class'
$("[attribute]")	Attribute Selector: All elements with attribute 'attribute'
$(":dom-prop")	DOM methods: All elements matching the dom property 'dom-prop'
$("e1, e2")	Composed selector: all elements of type e1 or e2
$(myVar)	Object Selector: Javascript object named myVar
$(:contains('string'))	String Selector: All elements that contain the 'string'

.action() Simple Effects and Animation

jQuery comes equipped with a good variety of out of the box capabilities

- Display: hide() / show() / toggle() / slideUp() / slideDown() / slideToggle()
- Opacity: fadeIn / fadeOut() / fadeTo() / fadeToggle()
- Custom: animate() / stop() / delay() / finish() / clearQueue() / dequeue() / myFunction()

.action() Display hide/show

jQuery makes hiding and showing elements easy

- Can be used for menuing, content queuing, rendering improvement, data loading, etc
- Hide / show – no frills basics

.hide([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

.show([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

NOTE: DEMO EET

.action() Display slideUp/slideDown

jQuery makes hiding and showing elements easy

- Can be used for menuing, content queuing, rendering improvement, data loading, etc
- slideUp / slideDown – accordion-like hide/show functionality

.slideDown([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

.slideUp([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

jQuery Magic

.action() Display slideUp/slideDown

jQuery makes hiding and showing elements easy

- Can be used for menuing, content queuing, rendering improvement, data loading, etc
- slideUp / slideDown – accordion-like hide/show functionality

.slideDown([duration] [, complete])

duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

.slideUp([duration] [, complete])

duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

j
Q
u
e
r
y

M
a
g
i
c

.action() Opacity fadeIn/fadeOut/fadeTo

jQuery makes hiding and showing elements easy

- Can be used for menuing, content queuing, rendering improvement, data loading, etc
- fadeIn/fadeOut – adjust transparency entirely to 0 or 1
- fadeTo – adjust transparency to a specific value

.fadeIn([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

.fadeOut([duration] [, complete])
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

.fadeTo(duration, opacity [, complete])
duration Type: String or Number
A string or number determining how long the animation will run.
opacity Type: Number
A number between 0 and 1 denoting the target opacity.
complete Type: Function()
A function to call once the animation is complete.

j
Q
u
e
r
y

M
a
g
i
c

.action() Animate

Custom animations using CSS

.animate(properties [, duration] [, easing] [, complete])

properties
Type: PlainObject
An object of CSS properties and values that the animation will move toward.
duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
easing (default: swing)
Type: String
A string indicating which easing function to use for the transition.
complete
Type: Function()
A function to call once the animation is complete

NOTE: can only be applied to positioned elements (fixed, absolute or relative)

.action() Chaining

You can chain multiple actions

```
$("h2").hide().html("some text").fadeIn(1000)
```

.action() Eventing

jQuery provides an event system that allows for handling and triggering

- Can ‘bind’ event handlers to DOM elements
- Handlers can be any function
- Can ‘trigger’ events on elements programmatically (within browser scripting constraints)

http://api.jquery.com/category/events/

.action() Eventing

Binding event handlers

```
$().ready(function(){
        $("h2").click(function(){
              alert('test');
        });
});
```

→

```
window.onload = init;
function init() {
     var es = document.getElementsByTagName('h2')
     for (var i ==0; i <= es.length; i++) {
          es[i].onclick = function(){alert('test')}
     }

}
```

http://api.jquery.com/category/events/

.action() Eventing

Binding custom events and triggering the event handlers programmatically

```
$().ready(function(){
      $("h2").click(function(){
           alert('test');
      });
});
```

$("h2:first").trigger("click")

```
$().ready(function(){
      $('element').bind('event-type',function(){
           alert('test bind');
      });
});
```

$(“element").trigger(“event-type")

http://api.jquery.com/category/events/

.action() Eventing

Some built in events

Event	Description
.error()	Browser event that handles errors (can be HTTP, such as 404 for imgs)
.resize()	Browser event triggered on resize
.scroll()	Browser event triggered on scrolling
.load(), .ready(), .unload()	Document loading methods that signal when the DOM is loaded, ready, or unloaded
.blur(), .change(), .focus(), .focusin(), .select(), .submit()	Form event handlers that handle bluring, changing, focusing, selecting, or submitting form elements
.focusout(), keydown(), keypress(), keyup()	Keyboard event handlers that handle moving out of an element, an event when the key is first down, the press event, and the event when the key is first up
.click(), .dblclick(), .focusout(), .hover(), .mousedown(), .mouseenter(), .mouseleave(), .mousemove(), .mouseout(), .mouseover(), .mouseup(), .toggle()	Mouse event handlers for dealing with different events related to using a mouse or touchpad. Note there are additional built in handlers for jquery-mobile (e.g., swipeleft(), swiperight()).

.action() Eventing is messy

As you can imagine this gets messy and begins to look like spaghetti (keep in mind)

```
$().ready(function(){
        $("h2").click(function(){
              alert('test');
        });
        $(“:button”).click( function() { // handle ALL button clicks
              if( $(this).is(“#btn1”) { // $(this) is current element
                       // code for handling btn1 goes here…
              }
              if( $(this).is(“#btn2”) {
                       // code for handling btn2 goes here…
              }
        });
        …
        $(“some other element”).bind(“some event”, function(){
             //a bunch of other handlers go here
        }
});
```

.action() Eventing is messy

To add to the mess you can also have handlers directly on elements!

```
<a href="https://someurl.com" onmouseover="somefunction()">Some link text</a>
```

Now you have event handlers potentially everywhere and no clear idea of structuring or modularity except for some comment blocks.

Callbacks

jQuery allows for actions to have a callback function

- A 'callback' gets executed after the action or event completes
- Ex. Return to the hide action, add a callback

.hide([duration] [, complete])

duration (default: 400)
Type: Number or String
A string or number determining how long the animation will run.
complete
Type: Function()
A function to call once the animation is complete.

```
$("h2").hide(function(){
	alert('hid: '+ this.innerHTML)
});
```

http://api.jquery.com/jquery.ajax/

Ajax

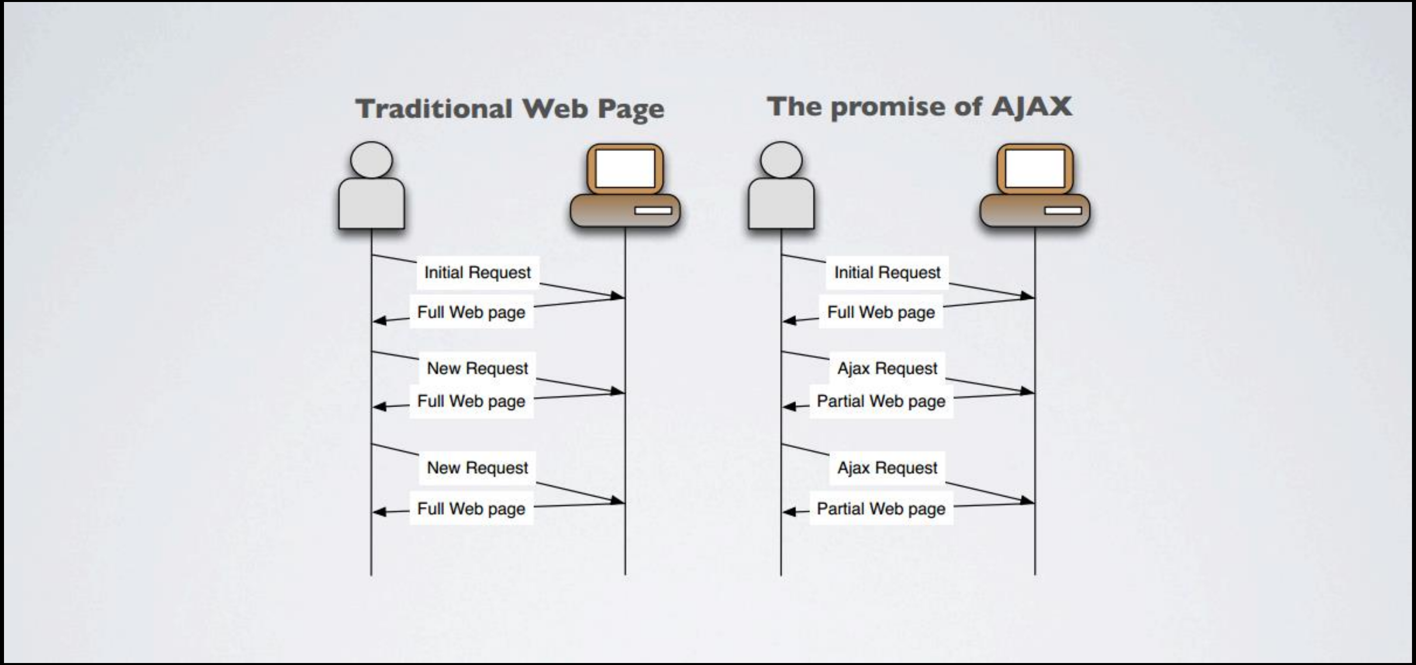
jQuery provides a method called .ajax()

- Ajax maps to an XMLHttpRequest action in javascript
- The browser interprets an XMLHttpRequest (aka XHR) according to its implementation
    - jQuery wraps these implementations – so that calling is cross-platform
- As of jQuery 1.5 all .ajax() calls implement the Promise interface giving them the methods and behaviors of a Deferred object in jQuery
- This is good because you can chain callback functions using .then(), .done(), .fail(), and .always()

http://api.jquery.com/jquery.ajax/

Ajax (con't)

Core idea



http://api.jquery.com/jquery.ajax/

A
j
a
x

Ajax (con't)

Specifics

- Ajax can return different content types including files, images, html, text/javascript, and JSON/XML/YAML
- Can set the http parameters associated with the request
- Can make GET/POST/PUT/DELETE/etc requests with ajax
- Can capture returned data in a callback function
- Can be used to interact with a REST API and then load data into the DOM

http://api.jquery.com/jquery.ajax/

A
j
a
x

Ajax (con't)

Specifics

- Ajax can return different content types including files, images, html, text/javascript, and JSON/XML/YAML
- Can set the http parameters associated with the request
- Can make GET/POST/PUT/DELETE/etc requests with ajax
- Can capture returned data in a callback function
- Can be used to interact with a REST API and then load data into the DOM
- Is subject to all the same constraints as other http requests
- Uses a cookie by default on each request
- Is one of the most import areas to focus on for clientside security

http://api.jquery.com/jquery.ajax/

A
j
a
x

Ajax (con't)

Syntax

jQuery.ajax(url [, settings])

url

Type: String

A string containing the URL to which the request is sent.

settings

Type: PlainObject

A set of key/value pairs that configure the Ajax request. All settings are optional. A default can be set for any option with $.ajaxSetup().

See jQuery.ajax(settings) below for a complete list of all settings.

```
$.ajax(
            {
               url:"some-url",
               type: "GET/PUT/POST/etc",
               data: myData,
               other-settings
            }
).done(function(data){
            //callback function returns response as 'data"
});
```

http://api.jquery.com/jquery.ajax/

Ajax (con't)

Example

```
$.ajax(
        {url:"http://jquery.com/browser-support/"}
).done(function(data){
        $("#content").append(data)
});
```

Promises

Definition:
Promises are, in a nut shell, better versions of asynchronous callbacks that exist in one of three states: pending, fulfilled, or rejected

Promises

Ember uses a promise library called rsvp.js that lets you create your own promises. Ember data uses promises by default as well.

```
var promise = new RSVP.Promise(function (fulfill, reject){
   //do stuff
});

//invoke the promise
promise.then(function(foo){
       //it worked
}, function(bar){
       //it failed
});
```

Read more: http://www.toptal.com/javascript/javascript-promises

Ember search posts example

```
return this.store.find(‘forumpost').then(function(items){
          //data loaded
          var filteredItems = items.filterProperty('url',params.searchquery);
          if(filteredItems.length>=1){
                    return filteredItems;
          }
          else{
                    t.transitionTo('page404');
          }
}, function(errors){
          //dataload failed
          console.log(errors);
});
```

Read more: http://www.toptal.com/javascript/javascript-promises

Ember search posts example: no ember data

```
return Ember.$.getJSON(‘/api/forumposts’).then(function(items){
          //data loaded
          var filteredItems = items.filterProperty('url',params.searchquery);
          if(filteredItems.length>=1){
                    return filteredItems;
          }
          else{
                    t.transitionTo('page404');
          }
}, function(errors){
          //dataload failed
          console.log(errors);
});
```

Read more: http://www.toptal.com/javascript/javascript-promises

Questions?

Matt Hale, PhD
University of Nebraska at Omaha
Interdisciplinary Informatics
mlhale@unomaha.edu
Twitter: @mlhale_