jQuery and AJAX for Responsive Web Sites

Lecture 2
Selectors, DOM,
Events

October 2012   41

Objectives

- Loose ends
- jQuery selectors
- HTML and its tree representation
- Events and the DOM tree
- jQuery event handlers intro

October 2012   42

Loose Ends

"+100 ms –> -1 % sales"
Greg Linden, 'Make Data Useful'

"foo" === 'foo'

//foo.com/jquery.min.js //

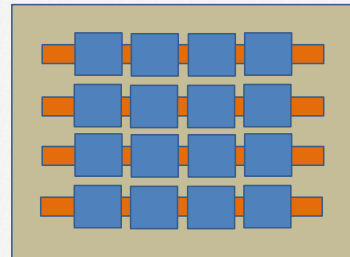moodle.extn.washington.edu

October 2012   43

Homework #1

October 2012   44

SurveyMonkey

October 2012   45

Office Hours

Monday and Thursday, 6pm to 7:30pm

Uptown Espresso
4th and Wall St.

Skype Niek will also be available: "niekjsanders"

October 2012   46

Selectors and Chains

Photo by Phillipe Put, CC-ND

October 2012   47

$

< Dollar Demo >

October 2012   48

The jQuery Dollar

- *$* just a variable name
- *$(foo)* is same as *jQuery(foo)*
- More next time

October 2012   49

---

jQuery.noConflict()

Categories: Core | Miscellaneous > Setup Methods

jQuery.noConflict([removeAll])   **Returns: _Object_**

Description: Relinquish jQuery's control of the $ variable.

jQuery.noConflict([removeAll])   version added: 1.0

removeAll A Boolean indicating whether to remove all jQuery variables from the global scope (including jQuery itself).

Many JavaScript libraries use $ as a function or variable name, just as jQuery does. In jQuery's case, $ is just an alias for jQuery, so all functionality is available without using $. If we need to use another JavaScript library alongside jQuery, we can return control of $ back to the other library with a call to $.noConflict():

```
<script type="text/javascript" src="other_lib.js"></script>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $.noConflict();
  // Code that uses other library's $ can follow here.
</script>
```

This technique is especially effective in conjunction with the .ready() method's ability to alias

October 2012   50

---

jQuery Dollar

$('li')

**$('
')**

$(function(e) { … })

October 2012   51

$('.puzzleBlock')

⬇

[<span>, <span>, <span>]

"Wrapped Set"

October 2012   52

2

$

< Wrapped Set Demo >

Wrapped Sets

- Look like simple arrays
- But they're not
- $(nativeArray)
- jqWrapSet.toArray()

Wrapped Set Wrap Up

```
// match DOM elements, output a wrapped set
// ---> [<strong>, <strong>, <strong>]
$('p strong');

// create new HTML elements, output a wrapped set
// ---> [<li>]
$('<li></li>')


// these two statements are the same… outputs wrapped set with doc obj
// ---> [#document]
$( function(e) { ... } );

jQuery(document).ready( function(e) { … } );
```

$

< Set Manipulation Docs >

(wrapped set) → Manipulation Function → (wrapped set)

Example: addClass()

jQuery Style

```
function jQueryManipulator( wrappedSet ) {

    for ( var i = 0; i < wrappedSet.length; ++i ) {
        doStuffTo( wrappedSet[i] );
    }

    return wrappedSet;
}

// method 1
var foo = jQueryManipulator( bar );
var baz = jQueryManipulator( foo );

// method 2
var baz = jQueryManipulator( jQueryManipulator( bar ) );
```
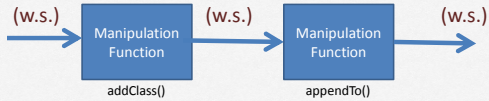
addClass()   appendTo()

59

Verbose Method

```
// select all the text input form fields
var foo = $('input[type=text]');

// add a CSS class to each item in wrapped set
foo = foo.addClass( 'puzzleBlock' );

// append wrapped items to html body element
foo = foo.appendTo( 'body' );
```

60

---

jQuery-esque

```
$('input[type=text]').addClass( 'puzzleBlock' )
                     .appendTo( 'body' )
```

61

Chaining Odds and Ends

- Naming convertion for wrapped sets
  - var foo$ = $('li');
- Lookup speeds
  - $('#meow')
  - $('ul > li')
  - $('a[href*=".png"]')
- Avoiding reflows
  - Next slide!

62

DOM Reflows

```
var newList$ = $('<ul></ul>');

for ( var i=0; i < 100; ++i ) {
    $('<li></li>').text(i).appendTo( newList$ );
}

newList$.appendTo( 'body' );
```

63

!

< Homework Code >

64

HTML its tree representation.

Photo by joiseyshowaa, CC-SA

October 2012 · 65

A Tree View of HTML

```
<html>
  <head>
    <title>Who doesn't like trees?</title>
  </head>
  <body>
    <h1>Trees!</h1>

    <div>
      <p>Seattle has lots of them.</p>

      <ul>
        <li>They're green!</li>
        <li>They're leafy!</li>
        <li>They smell like tree.</li>
      </ul>
    </div>
  </body>
</html>
```
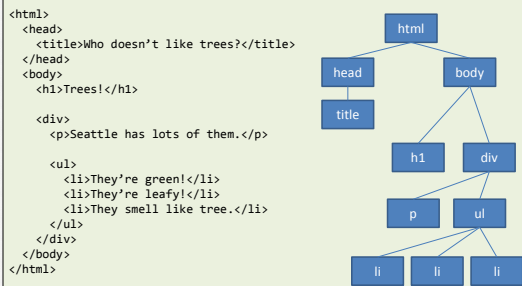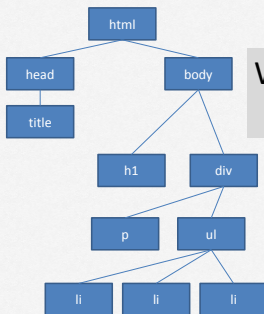
October 2012 · 66

A Tree View of HTML

```
<html>
  <head>
    <title>Who doesn't like trees?</title>
  </head>
  <body>
    <h1>Trees!</h1>

    <div>
      <p>Seattle has lots of them.</p>

      <ul>
        <li>They're green!</li>
        <li>They're leafy!</li>
        <li>They smell like tree.</li>
      </ul>
    </div>
  </body>
</html>
```


October 2012 · 67

HTML its tree representation.

Photo by joiseyshowaa, CC-SA

October 2012 · 68

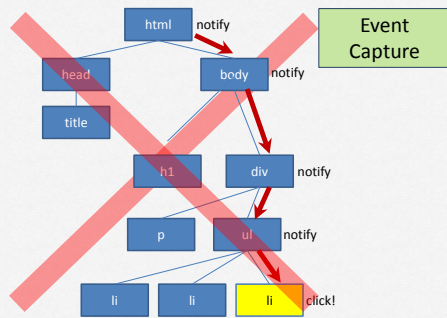## A Tree View of HTML

Why should we care?

October 2012 · 69

A Tree View of HTML
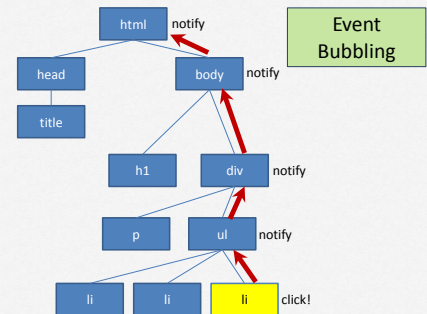
Event Bubbling

October 2012 · 70

5

A Tree View of HTML

71

< DOM View Demo >

72

Events to drive behavior

1. Event fires
2. React
   - Talk to the server (AJAX),
   - manipulate the DOM
   - Apply CSS styling
3. Browser state has changed
4. Repeat

73

Events Bestiary

Blur	Key Down/Up/Press	*DOCUMENT READY*
Change	Mouse Down	Resize
Click	Mouse Enter	Scroll
Double Click	Mouse Leave	Select
ERROR	Mouse Move	Submit
Focus	Mouse Out	Toggle
Focus In/Out	Mouse Over	*UNLOAD*
Hover	Mouse Up	

74

Events Bestiary … more?

AJAX Start

AJAX Send

AJAX Complete

AJAX Stop

AJAX Success

AJAX Error

75

< Event Binding Demo >

76

Events Bestiary ... more!

jQueryUI Drag

CREATE, START, DRAG, STOP

jQueryUI Selectable

CREATE, SELECTED, SELECTING, START, STOP, UNSELECTED

.... STILL MORE!

Window resize
Painting
Mouse
Keyboard
Closes

qt.digia.com

"Mysterious"

- jQuery document ready handler
  - Short hand
  - Long hand
- window.onload

```
$( function(e) {

    // manipulate dom

    // bind handlers

    // foo bar baz!
} );
```

Listening for events.....

on(), off(), one()

- bind, unbind
- live, die
- delegate, undelegate