Client/Server Programming for Internet Applications

Lecture 6
jQuery (Part I)

TCSS460

Summer 2020

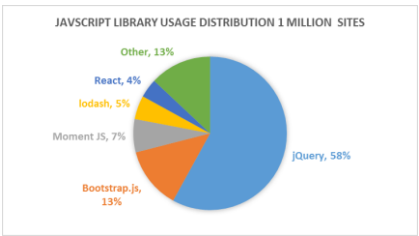
jQuery Foundations
→ most popular JavaScript Library

- a **framework** is a reusable software (environment) which
  - you can use in your own software and
  - provides specific functionality that improves the speed or reliability of the development process
- most web frameworks provide **features** needed by web developers
  - interact with HTTP headers
  - AJAX communication
  - authentication
  - DOM manipulation
  - handling browser differences

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017 2

jQuery Foundations → most popular JavaScript Library (cont'd)

- **jQuery** was introduced in 2005
  - John Resig (founder) was looking into how to better combine CSS selectors with the JavaScript
    - AJAX and animations were added within first year
    - additional modules: **jQuery** UI extension and mobile device support
  - **jQuery** provides many useful shortcuts and succinct ways of doing things



data courtesy of BuiltWith.com

Including jQuery

- importing jQuery for use in your application is as easy as including a link to a file in the `<head>` section of your HTML page
  - there are several different jQuery versions available
    - regular, minified and slim
    - **minified** → used for production sites (removes all comments and additional white space – aim: be as small as possible)
    - **slim** → removes all code pertaining to AJAX and visual effects
  - you can download a version of jQuery and use it in your projects or
  - you can also link to the jQuery file that is hosted on a **content delivery network (CDN)**

```
<script src="http://code.jquery.com/jquery-3.1.0.min.js"></script>
<script type="text/javascript">
    window.jQuery ||
        document.write('<script src="/jquery-3.1.0.min.js"><\/script>');
</script>
```

jQuery Selectors

- jQuery provides a powerful and simple mechanism for **selecting** elements
  - `getElementById()`
  - `querySelector()`
- the power of jQuery resides in the function named **`jQuery()`**
- **`jQuery()`**
  - takes one or two arguments and provides a wide variety of different properties and methods
  - also defines an **alias** for this function named **`$();`**

```
temp = jQuery('body');
temp = $('body');
```

5

jQuery Selectors (cont'd)

ex1

```
<script src="http://code.jquery.com/jquery-3.1.0.min.js"></script>
<p id="here">hello<span> there</span></p>
<ul>
    <li>United States</li>
    <li>France</li>
    <li>Spain</li>
    <li>Thailand</li>
</ul>
<script>
    /* selecting using regular JavaScript */
    var node = document.getElementById("here");
    var link = document.querySelectorAll("ul li");

    /* equivalent selection using jQuery */
    var node = $("#here");
    console.log(node);
    var link = $("ul li");
    console.log(link);
</script>
```

6

jQuery Selectors (cont'd)

- what does `$()` return?
  - returns the jQuery set **object**
    - an array-like structure that contains a set of DOM elements that match the selector
    - `$()` function always returns a set of results, rather than a single object
    - example:
      - `temp = $('body');`
      - `temp = $('body p');`
      - `temp = $('body ul li');`

7

Manipulating after a selection
→ JavaScript versus jQuery

ex2

- example illustrating how JavaScript and jQuery are equivalent programmatically to change the CSS for the markup
- jQuery allows you to do similar things as JavaScript but in a more succinct manner due to the power of the functions defined within jQuery

```
<script src="http://code.jquery.com/jquery-3.1.0.min.js"></script>
<p id="here">hello<span> there</span></p>
<script>
    /* manipulating after a selection -- using regular JavaScript */
    document.getElementById("here").innerHTML = "new <b>content</b>";
    var items = document.querySelectorAll("ul li");
    for (var i = 0; i < items.length; i++) {
    items[i].style.backgroundColor = "yellow";
    }

    /* manipulating after a selection -- using jQuery */
    $("#here").html("new <b>content</b>");
    $("ul li").css("background-color", "yellow");
</script>
```

new **content**

8

jQuery Basic Selectors

- recall: four basic selectors
  - **universal** selector, **class** selectors, **id** selectors, & **elements** selectors
- implementation of selectors in jQuery mirrors CSS specification
  - `$("*")` Universal selector matches all elements (and is slow).
  - `$("tag")` Element selector matches all elements with the given element name
  - `$(".class")` Class selector matches all elements with the given CSS class.
  - `$("#id")` —Id selector matches all elements with a given HTML id attribute.
- to select the single element with id="grab" you would write

```
var singleElement = $("#grab");
```

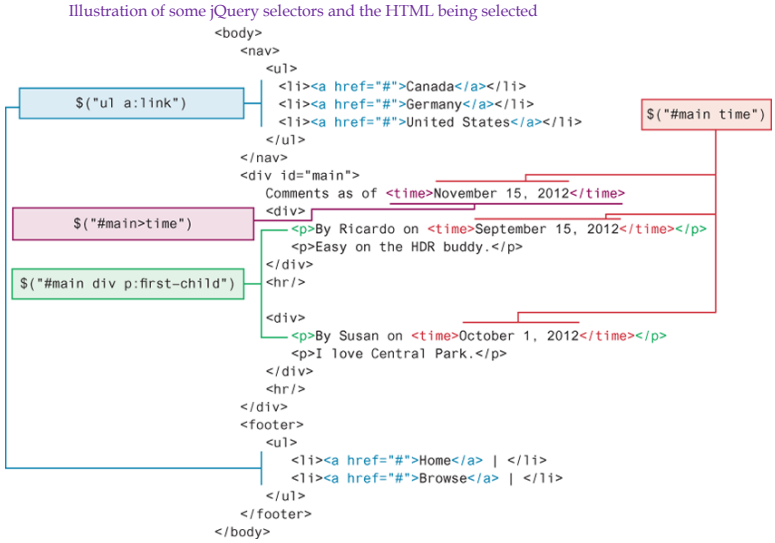
- to get a set of all the <a> elements the selector would be

```
var allAs = $("a");
```

9

jQuery Basic Selectors (cont'd)

Illustration of some jQuery selectors and the HTML being selected



Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

10

Advanced Selectors

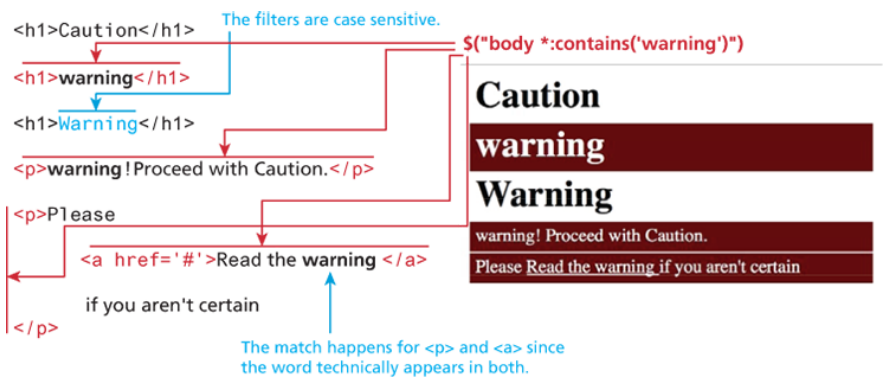
https://api.jquery.com/category/selectors/attribute-selectors/

```
// attribute selector
// allows you to retrieve all <img> elements with an src attribute beginning with /artist/
var artistImages = $("img[src^='/artist/']");

// Pseudo-Element Selector
// allows you to append to any selector using the colon and one of :link,
// :visited, :focus, :hover, :active, :checked, :first-child, :first-line, and :first-letter.
var visitedLinks = $("a:visited");

// Contextual Selector
// allows you to specify elements with certain relationships to one another in your CSS
var para = $("div p");

// jQuery Filters
// allows you to select elements that have a particular child using :has(),
// have no children using :empty, or match a particular piece of text with :contains().
var allWarningText = $("body *:contains('warning')");
```

11

Advanced Selectors (cont'd)

An illustration of jQuery's content filter selector




```
$("body *:contains('warning')").css("background-color", "#aa0000");
```

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

12

Common Element Manipulations
→ HTML Attributes and Properties

- We can both set and get an attribute value by using the **attr()** method

```
// link is assigned the href attribute of the first <a> tag
      var link = $("a").attr("href");
// change all links in the page to http://uw.edu
$("a").attr("href","http://uw.edu");
// change the class for all images on the page to fancy
$("img").attr("class","fancy");
```

- the **prop()** method is the preferred way to retrieve and set the value of a property

```
<input class="meh" type="checkbox" checked="checked">
```

```
var theBox = $(".meh");
theBox.prop("checked"); // evaluates to TRUE
```

- jQuery provides an extremely intuitive **css()** method

```
var color = $("#element").css("background-color"); // get the color
$("#element").css("background-color", "red"); // set color to red
```

13

Advanced Selectors (cont'd)

ex3

Sample jQuery selector filters

```
<script src="http://code.jquery.com/jquery-3.1.0.min.js"></script>
<table>
    <tr><td>Row 0</td></tr>
    <tr><td>Row 1</td></tr>
    <tr><td>Row 2</td></tr>
    <tr><td>Row 3</td></tr>
    <tr><td>Row 4</td></tr>
</table>
<script>
    /* changes the background color of the even rows */
    $("table tr:even").css("background-color", "#CFD8DC");
    /* changes the text color for rows 4 through N */
    $("table tr:gt(3)").css("color", "#DD2C00");
</script>
```

Row 0
Row 1
Row 2
Row 3
Row 4

14

Form Selectors

jQuery Form Selectors and their CSS Equivalents When Applicable

Selector	CSS Equivalent	Description
`$(":button")`	`button, input[type='button']`	Selects all buttons.
`$(":checkbox")`	`[type=checkbox]`	Selects all checkboxes.
`$(":checked")`	`No equivalent`	Selects elements that are checked. This includes radio buttons and checkboxes.
`$(":disabled")`	`No equivalent`	Selects form elements that are disabled. These could include `<button>`, `<input>`, `<optgroup>`, `<option>`, `<select>`, and `<textarea>`.
`$(":enabled")`	`No equivalent`	Opposite of `:disabled`. It returns all elements where the disabled attribute=false as well as form elements with no disabled attribute.
`$(":file")`	`[type=file]`	Selects all elements of type `file`.
`$(":focus")`	`No equivalent`	The element with focus.
`$(":image")`	`[type=image]`	Selects all elements of type image.
`$(":input")`	`No equivalent`	Selects all `<input>`, `<textarea>`, `<select>`, and `<button>` elements.
`$(":password")`	`[type=password]`	Selects all `password` fields.
`$(":radio")`	`[type=radio]`	Selects all `radio` elements.
`$(":reset")`	`[type=reset]`	Selects all the `reset` buttons.
`$(":selected")`	`No equivalent`	Selects all the elements that are currently selected of type `<option>`. It does not include checkboxes or radio buttons.
`$(":submit")`	`[type=submit]`	Selects all submit input elements.
`$(":text")`	`No equivalent`	Selects all input elements of type `text`. `$('[type=text]')` is almost the same, except that `$(:text)` includes `<input>` fields with no type specified.

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

15

Module Topics

- jQuery (Part I)
- jQuery (Part II)
- JavaScript Frameworks

TCSS 460 - Summer 2020

16