

Computer Science & Engineering 120
Learning to Code

Manipulating Data II – jQuery

Christopher M. Bourke
cbourke@cse.unl.edu

Part I: Introduction to jQuery

Topic Overview

- ▶ Introduction to jQuery
- ▶ Using jQuery

Overview

- ▶ jQuery is a JavaScript *library* that simplifies common tasks
- ▶ Introduced in 2006
- ▶ The most popular JavaScript library used today
- ▶ Provides easy ways of selecting *sets* of DOM elements
- ▶ Provides functions that can be *applied* to such results
- ▶ Companion libraries jQuery UI and jQuery Mobile provide user interface elements

Including jQuery

- ▶ jQuery can be downloaded and used as a local library (<http://jquery.com/>)
- ▶ Easiest to *hot link* to a Content Delivery Network
- ▶ Include the library using a URL/ `<script>` tag in the `<head>`

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
```

Using jQuery I

Selectors

- ▶ jQuery allows you to *select* an element or a *set* of DOM elements
- ▶ Use the dollar sign/parentheses and provide a *selector*: `$("#p")`
- ▶ Just as with CSS, we can specify multiple selectors, combinators
- ▶ Can select by ID or by class(es)

Using jQuery II

Selectors

```
1 //selects all the paragraph elements:
2 $("p");
3
4 //select all paragraphs and divs:
5 $("p, div");
6
7 //select all anchors within a paragraph:
8 $("p a");
9
10 //select by id using #
11 $("#lead");
12
13 //select by class
14 $(".strong");
```

Using jQuery I

Functions

- ▶ jQuery selectors result in a set (possibly empty, possibly of size 1) that can then be manipulated
- ▶ jQuery provides many *functions* that can be applied to the selector result
- ▶ We illustrate usage with the `css()` function
- ▶ Allows you to alter styling properties
- ▶ Takes two arguments: property and value (both strings).
- ▶ Syntax: selector + dot + function + arguments

Using jQuery II

Functions

```
1 $("p").css("font-size", "14pt");
2
3 $("p").css("color", "red");
4
5 $("p").css("background-color", "rgb(225, 225, 225)");
6
7 $("#lead").css("background-color", "rgb(100, 100, 100)");
```

Chaining

- ▶ jQuery is designed with a “fluent” design pattern
- ▶ Essentially, this allows you to *chain* function calls into one executable statement

```
1 $("p").css("font-size", "14pt")
2     .css("color", "red")
3     .css("background-color", "rgb(225, 225, 225)");
4
5 //whitespace does not matter, but
6 //long lines should be avoided
7 $("p").css("font-size", "14pt").css("color", "red").css("background-color", "rgb(225, 225, 225)");
```

Part II: Common jQuery Functions

Topic Overview

- ▶ `css()`
- ▶ `addClass()`, `removeClass()`
- ▶ `text()`
- ▶ `html()`
- ▶ `attr()`
- ▶ `val()`
- ▶ `ready()`

CSS

- ▶ Recall that providing two arguments (strings) to the `css()` function sets the CSS property to a given value
- ▶ Alternatively, you can provide *one* argument (css property as a string) you can *get* the property's value
- ▶ If the selector has multiple elements, returns the value of the *first* element

```
1 var c = $("#intro").css("color");
2 //now c contains its color as a string
3 console.log(c);
```

addClass and removeClass

- ▶ Style attributes can also be defined by an element's class
- ▶ `addClass()` and `removeClass()` allow you to add/remove a class from an element
- ▶ Can add/remove multiple classes (delimited by a space)
- ▶ Alternatively: `toggleClass()` will add/remove if the element doesn't/does have it already

```
1 $("p").removeClass("strong");
2 $("p").addClass("alert strong");
```

text

- ▶ `text()` allows you to get/set the inner text of an element
- ▶ The element `<p>Greetings, husker fans!</p>` has inner text `Greetings, husker fans!`
- ▶ Multiple items: inner text is concatenated into one string
- ▶ No argument: get the text, 1 argument (string): set the text
- ▶ Old text is removed and lost

```
1 var s = $("#intro").text();
2 console.log(s);
3 $("#intro").text("This is some new text.");
```

html

- ▶ `html()` allows you to get/set the *full* inner HTML of an element
- ▶ HTML tags of descendants are preserved
- ▶ Similar get/set with zero/one argument

```
1 var s = $("#myList").html();
2 console.log(s);
3 $("#myList").text("<li>A</li><li>B</li>");
```

attr

- ▶ `attr()` allows you to get/set the attribute value of an element
- ▶ Get: one argument (attribute name)
- ▶ Set: two arguments (name, value)

```
1 var url = $("#homeLink").attr("href");
2 $("#homeLink").attr("href", "http://cse.unl.edu");
```

val

- ▶ Entered/selected values of form elements are not part of the HTML
- ▶ `val()` allows you to get/set values of form elements

```
1 var fName = $("#firstName").val();
2 if(fName === "") {
3     $("#firstName").val("Please Enter valid value");
4 }
```

ready

- ▶ We may want some code to execute when a page is loaded (initialize an interface element or get some data)
- ▶ However, we want to wait to ensure that the page is entirely loaded
- ▶ Especially important if our code relies on other script file(s)
- ▶ We can place code in the `ready()` function which will only be executed after the page is downloaded, loaded, and *ready*

```
1 $(document).ready( function() {  
2   //code to be executed after the document is ready...  
3 });
```

Part III: Manipulating DOM Elements

Topic Overview

- ▶ Manipulation of DOM elements
- ▶ Basic animations
- ▶ jQuery UI

Adding & Removing DOM Elements I

- ▶ jQuery allows you to add/remove DOM elements
- ▶ To create a DOM element, you can simply format a valid HTML string
- ▶ Various functions add/remove from different parts of the element.

Adding & Removing DOM Elements II

append()

- ▶ Adds to the inner part of the element at the end
- ▶ `$("#lead").append(" And more...");`
- ▶ Changes
`<p id="lead">All that!</p>`
to
`<p id="lead">All that! And more...</p>`

Adding & Removing DOM Elements III

prepend()

- ▶ Adds to the inner part of the element at the *beginning*
- ▶ `$("#lead").prepend("Hey! ");`
- ▶ Changes
`<p id="lead">All That!</p>`
to
`<p id="lead">Hey! All that!</p>`

Adding & Removing DOM Elements IV

after()

- ▶ Adds content to the DOM *after* the matched element(s)
- ▶ `$("#lead").after("<p>And a bag of chips.</p>");`
- ▶ Changes
`<p id="lead">All That!</p>`
to
`<p id="lead">All that!</p><p>And a bag of chips.</p>`

Adding & Removing DOM Elements V

before()

- ▶ Adds content to the DOM *before* the matched element(s)
- ▶ `$("#lead").before("<p>All These!</p>");`
- ▶ Changes
`<p id="lead">All That!</p>`
to
`<p>All These!</p><p id="lead">All that!</p>`

Adding & Removing DOM Elements VI

empty()

- ▶ Empties the content of the element (its inner text and all descendants)
- ▶ `$("#lead").empty();`
- ▶ Changes
`<p id="lead">All That!</p>`
to
`<p id="lead"></p>`

Adding & Removing DOM Elements VII

remove()

- ▶ Removes the element (and its inner text and all descendants) entirely
- ▶ Take care, all will be lost
- ▶ `$("#lead").remove();`
- ▶ Changes
`<p id="lead">All That!</p>`
to

Effects

- ▶ You can `hide()` and `show()` elements to make them invisible/visible *without* removing them from the DOM
- ▶ Alternatively, you can show/hide with some effects: `fadeIn()`, `fadeOut()`, `fadeToggle()`
- ▶ Or: `slideUp()`, `slideDown()`, `slideToggle()`
- ▶ You can provide a "speed": `"slow"` `"fast"`, `{duration:1500}` (milliseconds)

```
1 $("#lead").fadeOut();
2 $("#lead").fadeIn("slow");
3 $("#lead").fadeToggle({duration:3000});
4
5 //chaining:
6 $("#lead").fadeOut("fast").fadeIn("fast");
```

jQuery UI

- ▶ jQuery UI (<http://jqueryui.com/>) is an add-on library built on jQuery
- ▶ Adds many more animation effects
- ▶ Comes with customizable User Interface "widgets"
- ▶ Need to include (using the same CDN) script and CSS
- ▶ Initialize elements in the `ready()` function

```
1 <link rel="stylesheet"
2 href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/themes/smoothness/jquery-ui.css">
3 <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js"></script>
4
5 <script>
6 $(document).ready( function() {
7     $("#leadParagraph").tooltip();
8 });
9 </script>
```