jQuery

An Overview

Plan for Today

- Quick review: what is jQuery
- How to use jQuery
- Selectors and filters
- Manipulating page content
- Chaining
- Event handling
- Effects and animation

What is jQuery

- Free open source JavaScript library (i.e., it is just a JavaScript file)
- Simplifies common front-end web development tasks
- Takes care of cross-browser compatibility issues
- With jQuery you can:
  - select DOM elements and
    - modify them
    - animate them
    - attach event handlers to them
  - create new HTML elements and add them to the DOM
  - use AJAX to update parts of your page with external content
  - ...and all this in a much simpler way than in raw JavaScript

How to use jQuery

1. Add a reference to the jQuery file into your webpage
    - Download the file (http://jquery.com/download)
        - You have the local file, so you don't depend on a connection
    - ...or Link to a CDN (content delivery network)

      https://code.jquery.com/

      https://developers.google.com/speed/libraries/#jquery
        - The file will be cached by most browsers >> your page will load faster
    - ...or do both!

    - Use the full version for development (optionally)
    - Use the minified version for production (to gain speed)

2. Add a reference to the JavaScript file with your code

Adding references to jQuery

- To a CDN:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

- To both a CDN and a local file:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/jquery-3.1.1.js"></script>')</script>
```

jQuery code: where to place it

- A web page cannot be manipulated safely until the DOM has been loaded

- jQuery detects if the page is ready:

```
$(document).ready(function() {
      /* this code will only run once the page is ready */
});
```

- Instead of using the code above, you may use a shortcut:

```
$(function() {
      /* this code will only run once the page is ready */
});
```

"Unpacking" $(document).ready(...

- Consider this code:
  `$(document).ready(function() { /* your code */ });`
  - `$(document)` is a jQuery object
  - `.ready()` is a method called on the jQuery object
  - `function() { /* your code */ }` is the argument passed to that method
    - it is an ***anonymous function***:

      a function declaration without a name;

      the function body is whatever you put within { }

      It is executed just like any other JavaScript code we've used this semester

- The shortcut collapses the object and the method call into this: $():
  `$(function() { /* your code */ });`

Writing jQuery code: Overall approach

1. **Select** an HTML element: `$("p")`
   - returns zero or more elements that match the CSS selector "p" wrapped in a jQuery object
2. optionally, assign to a variable: `var $p = $("p")`
   - by convention, variable name starts with '$')

3. **Do something** with your selection:
   - animate it: `$("p").fadeIn("slow")`
   - change it: `$("p").html("new text")`
   - add an event handler to it:

```
$("p").on("click", function() {
    /* anything you like -- more jQuery */
});
```

Selectors

`$("p")`	select all paragraphs
`$(".foo")`	select all elements that have class "foo"
`$("#bar")`	select the element with id="bar"
`$("p.foo")`	select all paragraphs that have class "foo"
`$("*")`	select all elements on the page
	... and so on. You may use (almost) any CSS selectors

Filters

jQuery provides filters that enable us to make more specific selections:

:not(selector)	all elements except those that match selector
:first	first paragraph
:last	same as above but last
:even	all even elements in the set
:odd	all odd element in the set
:eq(index)	elements with a specific index number
:gt(index)	same as above but greater than
:lt(index)	same as above but less than

Filters

:header	all <h1> - <h6> elements
:animated	all selected elements that are currently being animated
:contains('text')	all selected elements containing 'text'
:hidden	all selected elements that are hidden
:visible	all selected elements that are visible
[attribute]	all selected elements that have the specified attribute
[attribute='value']	same as above, but also must have a specific value

...there are more! http://api.jquery.com/category/selectors/

Manipulating page content

- .html() method gets the content of the selection (including markup).
- It only retrieves content from the first element in the matched set.
- .text() method gets the text content only of the selection (no tags)

- New content is added by passing an argument to the method:
  - .text("new text")
  - .html("<b>new html</b>")
  - It updates all of the elements in the matched set. This is known as **implicit iteration**.

Manipulating page content

- .remove() method removes the selected set
- .replaceWith() method replaces the selected set
- More ways of adding content:
  - .before("new html")
  - .after("new html")
  - .prepend("new html")
  - .append("new html")
- Get and set attributes:
  - attr()
  - removeAttr()
  - addClass()
  - removeClass()
  - toggleClass()

Manipulating page content

- .css() gives access to manipulate the selections CSS directly:

```
.css({
  ‘font-weight': 'bold',
  ‘color’: 'red',
});
```

- Pay attention to the slightly different syntax:
  - {'property' : 'value', 'property' : 'value', 'property' : 'value'}

Chaining

If you need to call multiple methods on your selection, instead of doing this:

```
    $("selector").method1();
    $("selector").method2();
    $("selector").method3();
```

you can (and should) do this:

```
    $("selector").method1()
        .method2()
        .method3();
```

For example: `$(p).text("new text").hide().fadeIn(500);`

Event handling

- .on()   method that handles events
- $(selector).on("event name", function);
- In most cases, we use anonymous functions:

```
$("button").on("click", function() {
        /* do something (with anything else on the page) */
});
```

- Or you could pass the event object (like we did in the Travelling Hobbit)

```
$("button").on("click", function(e) {
        /* do something using event object e */
});
```

Event handling

Events you could use:
focus, blur, change, input, keydown, keyup, keypress, click, dblclick, mouseup, mousedown, mouseover, mousemove, mouseout, hover, submit, select, ready, load, unload, error, resize, scroll

Effects & animation

.show()	display selected
.hide()	hide selected
.toggle()	display or hide: based on current state
.fadeIn()	fade in
.fadeOut()	fade out (make transparent)
.fadeTo()	change opacity to a specific value
.fadeToggle()	same as above, based on current state

Effects & animation

.slideUp()	display selected using sliding motion
.slideDown()	hide selected using sliding motion
.slideToggle()	same as above, based on current state
.delay()	delay execution
.stop()	stop animation if running
.animate()	execute custom animation (animate numeric css properties)

Resources

- Lynda tutorial *(you must be logged in to lynda.uni.edu)*
- https://learn.jquery.com/
- http://api.jquery.com/