CIS 4004: Web Based Information Technology Fall 2013

Using jQuery – Part 1

Instructor :   Dr. Mark Llewellyn
                    markl@cs.ucf.edu
                    HEC 236, 407-823-2790
            http://www.cs.ucf.edu/courses/cis4004/fall2013

Department of Electrical Engineering and Computer Science
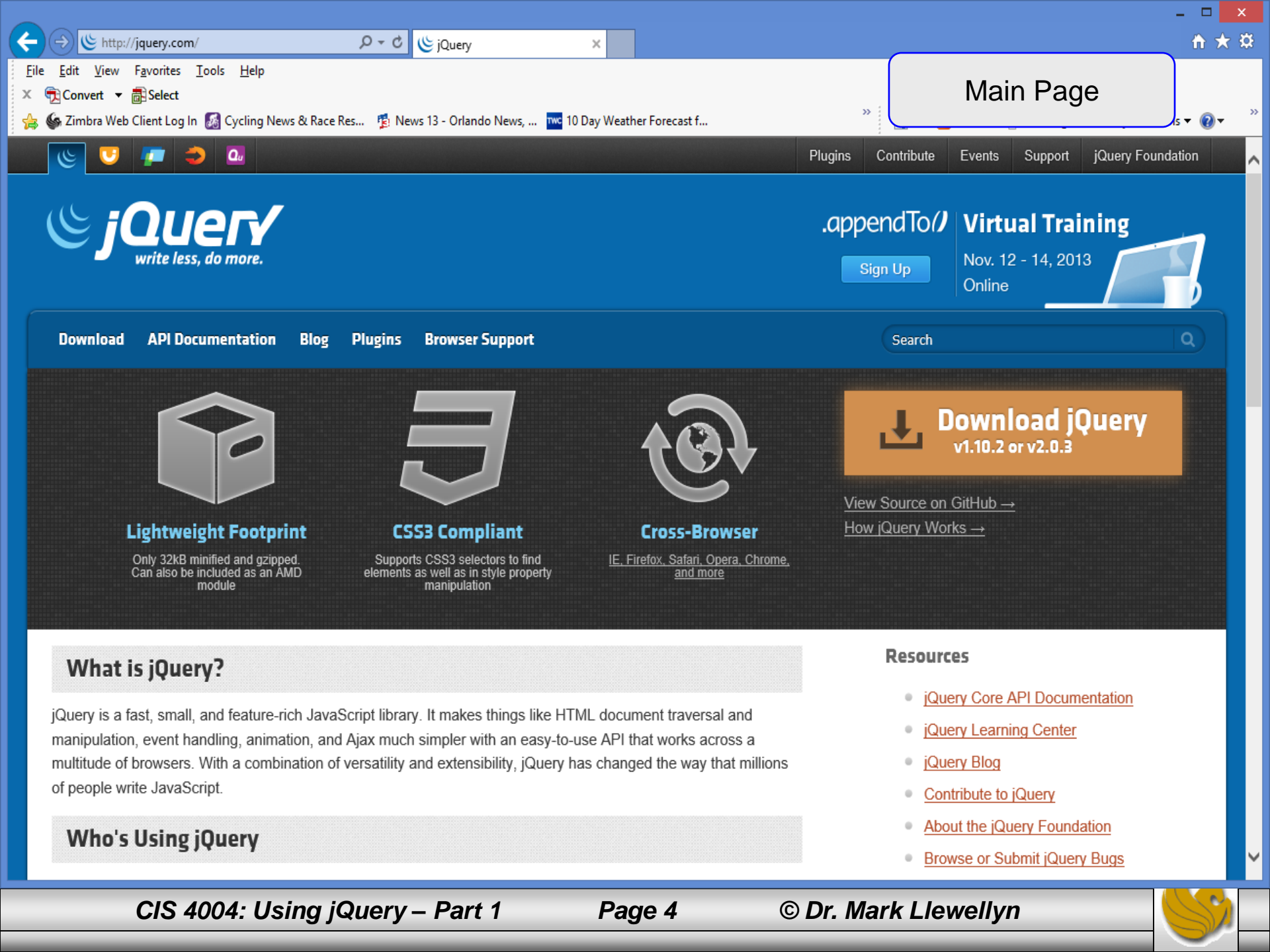University of Central Florida

Using jQuery

- The first step in learning to use jQuery is to download the jQuery library from the jQuery web site.

- The jQuery library is a JavaScript file that can be accessed in either of two ways:

  - Download the jQuery.js and host it locally on your machine.

  - Use a hosted version from a CDN (content delivery network).

- I recommend downloading a copy of jQuery to your computer for development and testing, which does not require an Internet connection.

Downloading jQuery

- Go to www.jquery.com.

- Click the download link located in the main navigation bar at the top of the page, which takes you to a page that offers many different ways to access the jQuery library.

- I'll use both techniques (downloaded and CDN) in the various examples that appear in this set of notes.

- The following pages illustrate some of the pages you'll see at the jQuery site.

- Note that the current version of jQuery is 1.10.2, and this version is supported by both Google and Microsoft CDNs.

http://jquery.com/   jQuery

File   Edit   View   Favorites   Tools   Help

Convert   Select

Zimbra Web Client Log In   Cycling News & Race Res...   News 13 - Orlando News, ...   10 Day Weather Forecast f...

Plugins   Contribute   Events   Support   jQuery Foundation

jQuery
write less, do more.

.appendTo() | **Virtual Training**

Sign Up   Nov. 12 - 14, 2013
Online

Download   **API Documentation**   **Blog**   **Plugins**   **Browser Support**

Search

Lightweight Footprint
Only 32kB minified and gzipped.
Can also be included as an AMD
module

CSS3 Compliant
Supports CSS3 selectors to find
elements as well as in style property
manipulation

Cross-Browser
IE, Firefox, Safari, Opera, Chrome,
and more

Download jQuery
v1.10.2 or v2.0.3

View Source on GitHub →
How jQuery Works →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Who's Using jQuery

Resources

- jQuery Core API Documentation
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery
- About the jQuery Foundation
- Browse or Submit jQuery Bugs

File   Edit   View   Favorites   Tools   Help

Convert   Select

Zimbra Web Client Log In   Cycling News & Race Res...   News 13 - Orlando News, ...

Page   Safety   Tools

Download Page

Plugins   Contribute   Events   Support   jQuery Foundation

jQuery
write less, do more.

.appendTo()   **Virtual Training**

Nov. 12 - 14, 2013

Online

Sign Up

Download   **API Documentation**   **Blog**   **Plugins**   **Browser Support**

Search

Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. Starting with jQuery 1.9, we also make available sourcemap files that can be used to debug the compressed file in sourcemap-aware browsers such as Google Chrome. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience.
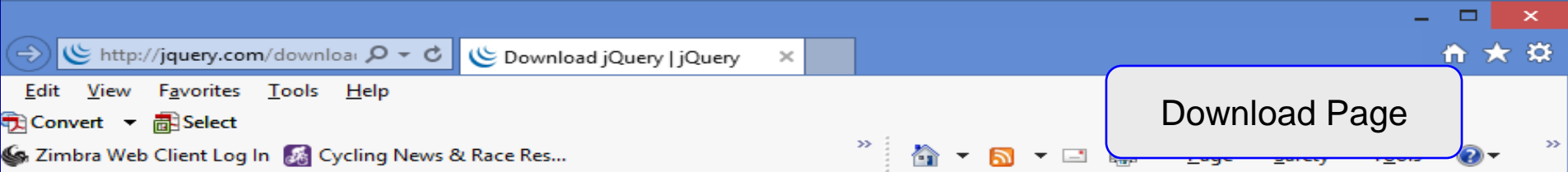
To locally download these files, right-click the link and select "Save as..." from the menu.

jQuery 1.x

The jQuery 1.x line had major changes as of jQuery 1.9.0. We *strongly* recommend that you also use the jQuery Migrate plugin if you are upgrading from pre-1.9 versions of jQuery or need to use plugins that haven't yet been updated. Read the jQuery 1.9 Upgrade Guide and the jQuery 1.9 release blog post for more information.

Download the compressed, production jQuery 1.10.2

Download Page

CDNs can offer a performance benefit by hosting jQuery on servers spread across the globe. This also offers an advantage that if the visitor to your webpage has already downloaded a copy of jQuery from the same CDN, it won't have to be re-downloaded.

jQuery's CDN provided by MaxCDN

To use the jQuery CDN, just reference the file directly from `http://code.jquery.com` in the script tag:

```
1   <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
2   <script src="http://code.jquery.com/jquery-migrate-1.2.1.min.js"></sc
```

Starting with jQuery 1.9, sourcemap files are available on the jQuery CDN as well. If compressed files are included directly from the CDN as shown above, sourcemap-aware browsers such as Google Chrome will use them when you enable sourcemap support.

To see all available files and versions, visit http://code.jquery.com

Other CDNs

The following CDNs also host compressed and uncompressed versions of jQuery releases. Starting with jQuery 1.9 they may also host sourcemap files; check the site's documentation.

Note that there may be delays between a jQuery release and its availability there. Please be patient, they receive the files at the same time the blog post is made public. Beta and release candidates are not hosted by these CDNs.

- Google CDN
- Microsoft CDN
- CDNJS CDN

jQuery

snippet: `<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>`

site: http://jquery.com/

versions: 2.0.3, 2.0.2, 2.0.1, 2.0.0, 1.10.2, 1.10.1, 1.10.0, 1.9.1, 1.9.0, 1.8.3,...

note: 1.2.5 and 1.2.4 are not hosted due to their short and unstable lives in the wild.

jQuery UI

snippet:

`<script src="//ajax.googleapis.com/ajax/libs/jqueryui/1.10.3/jquery-ui.min.js"></script>`

site: http://jqueryui.com/

versions: 1.10.3, 1.10.2, 1.10.1, 1.10.0, 1.9.2, 1.9.1, 1.9.0, 1.8.24, 1.8.23, 1.8...

note: This library depends on jQuery. You must also load jQuery before loading this module. Version 1.8.3 is not hosted due to its short life, and the alias `1.8.3` actually loads 1.8.4.

MooTools

snippet:

`<script src="//ajax.googleapis.com/ajax/libs/mootools/1.4.5/mootools-yui-compressed.js"></sc`

site: http://mootools.net/

versions: 1.4.5, 1.4.4, 1.4.3, 1.4.2, 1.4.1, 1.4.0, 1.3.2, 1.3.1, 1.3.0, 1.2.5, 1.2....

Prototype

snippet:

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

hello.php | welcome.php | php.ini | index.jsp | MySQLServlet.java | jquery-1.10.2.js

> The first part of the actual jQuery library

```
 1  /*!
 2   * jQuery JavaScript Library v1.10.2
 3   * http://jquery.com/
 4   *
 5   * Includes Sizzle.js
 6   * http://sizzlejs.com/
 7   *
 8   * Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors
 9   * Released under the MIT license
10   * http://jquery.org/license
11   *
12   * Date: 2013-07-03T13:48Z
13   */
14  (function( window, undefined ) {
15
16  // Can't do this because several apps including ASP.NET trace
17  // the stack via arguments.caller.callee and Firefox dies if
18  // you try to trace through "use strict" call chains. (#13335)
19  // Support: Firefox 18+
20  //"use strict";
21  var
22      // The deferred used on DOM ready
23      readyList,
24
25      // A central reference to the root jQuery(document)
26      rootjQuery,
27
28      // Support: IE<10
29      // For `typeof xmlNode.method` instead of `xmlNode.method !== undefined`
30      core_strundefined = typeof undefined,
31
32      // Use the correct document accordingly with window argument (sandbox)
33      location = window.location,
34      document = window.document,
```

JavaScript file   length : 273199   lines : 9790   Ln : 1   Col : 1   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

Downloading jQuery

- The non-compressed version of jQuery 1.10.2 is about 267KB, so it does not have a very large footprint on your system. The compressed (minified version) is only about 91KB.

- In production environments it is sometimes desirable to use the CDN solution as they offer the benefit of a large, high-speed network which can serve up the jQuery library from many different locations. The CDNs have proven to be quite reliable and fast-loading solutions that will free up the bandwidth from your own site.

Understanding The jQuery Wrapper

- Before you get started programming with jQuery, you need to understand what the jQuery wrapper is and how it applies to the DOM.

- A wrapper, in most programming languages, is something that wraps something else to extend the functionality, most often an object.

- The jQuery wrapper attaches itself to the DOM by using selectors and allows you to extend the DOM.

- jQuery doesn't actually offer any new methods; it just takes methods that already exist in native JavaScript and makes them much easier to interact with.

Understanding The jQuery Wrapper

- The power of the jQuery wrapper is being able to extend the DOM with much less code than native JavaScript.

- The following code is an example of the jQuery selector statement:

```
$.(selector)
```

- As with JavaScript, jQuery has many event methods to choose from, but one very important one is called the `document.ready()` event handler method, which executes only after the DOM is fully loaded. Since the power of jQuery (like JavaScript) comes from manipulating the DOM, you want to make sure its ready before you do anything with it.

Understanding The jQuery Wrapper

- The `document.ready()` event handler allows you to put all of your JavaScript jQuery code within this event to make sure the code is executed when the DOM is ready.

- This event is similar to the JavaScript `onload` event, except the `document.ready()` event handler only fires after the DOM has loaded.

- To explain the jQuery wrapper, I will walk you through how to set up the `document.ready()` statement.

Understanding The jQuery Wrapper

- The first step involves setting up a selector that is preceded by the dollar sign ($), which is the alias for accessing the jQuery itself.

- You pass the selector between the two parentheses; in this case, I'm passing the document selector for the DOM.

- The alias and the selector make up the jQuery wrapper.

```
$(document)
```

- The ready event gets attached after the selector statement and is interchangeable with other events.

```
.ready()
```

Understanding The jQuery Wrapper

- The function is the event handler routine, which is applied after the DOM is loaded and ready and does not include graphics.

- The function is placed within the parentheses of the ready event because you are passing the function you want to be run to the ready event handler.

```
.ready(function() {

    //jQuery DOM code goes here

    alert("The DOM is fully loaded and ready.");

} ) ;
```

- This is example is illustrated on the next two pages, once using a local copy of the jQuery library and once using a CDN.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

example 1 - local.html

```
 1  <!DOCTYPE html>
 2  <html lang="en">
 3    <head>
 4      <meta charset="utf-8">
 5      <title>jQuery - Example 1</title>
 6      <script src="jquery-1.10.2.min.js"></script>
 7      <script>
 8          $(document).ready(function() {
 9              alert("The DOM is fully loaded and ready.");
10          } );
11      </script>
12  <body>
13      <div class="container">
14      </div>
15  </body>
16  </html>
17
```

Using a local copy of the jQuery library

length : 321   lines : 17 | Ln : 17   Col : 1   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

example 1 - local.html   example 1 - cdn.html

Using a CDN to access the jQuery library

```html
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="utf-8">
5       <title>jQuery - Example 1</title>
6       <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
7       <script>
8           $(document).ready(function() {
9               alert("The DOM is fully loaded and ready.");
10          } );
11      </script>
12  <body>
13      <div class="container">
14      </div>
15  </body>
16  </html>
17
```

Hyper Text Markup Language file   length : 365   lines : 17   Ln : 1   Col : 1   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS
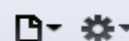
Understanding The jQuery Wrapper

- Most jQuery-specific code needs to be set up within a `document.ready()` event handler.

- However, native JavaScript such as variables, arrays, and so on can be set up outside of the document ready event handler because they don't need to wait for the DOM to be ready and are hidden from the DOM as they are specifics within the actual script.

- The code on the following page illustrates this concept in that the script relies on the DOM being loaded before new content can be added. There are three variables being set in this script – two of them are defined outside of the `document.ready()` event and one is defined inside the `document.ready()` event since it requires access to the for loop set up inside the function.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

| hello.php ✕ | welcome.php ✕ | php.ini ✕ | Undo x.jsp ✕ | MySQLServlet.java ✕ | example 1.html ✕ | example 2.html ✕ |

```html
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <title>jQuery - Example 2</title>
6      <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
7      <script>
8          var numShows=3;
9          var numTickets=15;
10         $(document).ready(function() {
11             for(i=0; i<numTickets; i++) {
12                 var numTotal = i + 1;
13                 $('.container').append("<p>There are " + numTotal + " tickets available</p>");
14             }//end for loop
15         } );
16     </script>
17  <body>
18      <div class="container">
19      </div>
20  </body>
21  </html>
22
```

Variables defined outside the function → (points to lines 8-9)

Variable defined inside the function → (points to line 12)

Hyper Text Markup Language file   length : 517   lines : 22   Ln : 6   Col : 68   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

There are 1 tickets available

There are 2 tickets available

There are 3 tickets available

There are 4 tickets available

There are 5 tickets available

There are 6 tickets available

There are 7 tickets available

There are 8 tickets available

There are 9 tickets available

There are 10 tickets available

There are 11 tickets available

There are 12 tickets available

There are 13 tickets available

There are 14 tickets available

There are 15 tickets available

Using Selectors, Filters, and CSS: The Core of jQuery

- Selectors are the core building blocks of jQuery. Everything you do with the DOM in jQuery incorporates the use of a selector because you need to choose which elements in the DOM you are selecting and manipulating.

- Filters give you more flexibility in selecting elements based on a characteristic in the DOM, where CSS selectors can't help. Filters are often used in conjunction with selectors to give you great depth of control when selecting specific elements based on, say, their position in a group of elements, or their visibility, or a characteristic of a form input such as checked or disabled.

- These are all things that you've had some experience with using native JavaScript. Remember that jQuery doesn't give you any additional functionality, it just makes it easier to do. The jQuery moto is "write less, do more".

Using Selectors, Filters, and CSS: The Core of jQuery

- Selectors, an essential feature of the jQuery library, are powered by the jQuery Sizzle selector engine. Sizzle can be used with other languages, but its real power is best used with all the other jQuery methods. The jQuery Sizzle selector engine is JavaScript code written to handle selectors in jQuery.

- The jQuery selector is a string expression that classifies a single or set of DOM elements referred to as the matched set and is ready to be worked with in jQuery.

- The selector is always declared immediately after the jQuery alisas ($).

- After DOM elements are selected and methods have been applied, the matched set becomes a jQuery object.

Using Selectors, Filters, and CSS: The Core of jQuery

- Working with selectors in jQuery is simple because most of the selectors are those that you've worked with using CSS. There are a few special selectors specific to jQuery, but we'll hold off on those for the time being.

- The selector is a way for you to navigate the DOM, and, in its most basic form, allows you to select an element and the syntax is identical to CSS selector syntax, whether it's an ID, class, tag or, an attribute.

- When you use a selector, the statement you create automatically loops through all of the nodes in the DOM looking for elements you have specified in your selector. The result of this loop is also known as the matched set.

Using Selectors, Filters, and CSS: The Core of jQuery

- The JavaScript parts that make up a selector are:

  - The jQuery alias ($ is the convention, but jQuery also works).

  - The DOM elements, which you are selecting, wrapped in quotes within the two parentheses.

  - Anything after the selector is the jQuery method, which you are applying. The jQuery method can do anything from adding CSS to animating elements on the page.

The jQuery Alias	The Selector	The jQuery method or action
`$ or jQuery`	`('div')`	`.css('border', '1px solid black');`

The anatomy of a jQuery statement

Selecting Page Elements By Using CSS Selectors

- JavaScript has native functions that can select elements by ID and tag (we've used both before). The downside of these functions is that you have to use a different function for each of the three types of elements.

- When you use selectors in jQuery, one selector can handle multiple types of elements. This leads to cleaner code and generally less of it.

- As we did with JavaScript and DOM manipulation, you'll probably want to do most of your development using jQuery utilizing the developer tools of your browser. For many of the subsequent examples in this set of notes, the developer tools will be turned on in order to highlight the changes in the DOM that have occurred via the jQuery manipulation of the DOM.

Selecting Elements Using The Wildcard (*) Selector

- If you would like to select all the elements in your DOM or within other elements, use the wildcard (*) selector. The wildcard is wrapped in quotes between the parentheses directly after the alias.

- The example on the next two pages illustrates this concept by adding a visible border to every element on the page.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?

hello.php   welcome.php   php.ini   index.jsp   MySQLServlet.java   example 1.html   example 2.html   example 3.html

```html
1  <!DOCTYPE html>
2  <html>
3    <head>
4    <title>jQuery - Example 3</title>
5    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
6    <script>
7    $(document).ready(function() {
8      $('*').css('border','1px solid #333');
9    });
10   </script>
11   <body>
12     <div class="container">
13         <h1>Hello From jQuery.</h1>
14     </div>
15     <div class="container">
16         <h2>This is inside a second div element in the page.</h2>
17     </div>
18   </body>
19  </html>
20
```

Hyper Text Markup Language file   length : 453   lines : 20   Ln : 5   Col : 66   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

Browser window titled "jQuery - Example 3"

Content area:
Hello From jQuery.

This is inside a second div element in the page.

Developer tools panel:

Elements | **Resources** | **Network** | **Scripts** | **Timeline** | **Profiles** | **Audits** »   Search Elements

```
<!DOCTYPE html>
<html style="border-top-width: 1px; border-right-width: 1px;
border-bottom-width: 1px; border-left-width: 1px; border-top-
style: solid; border-right-style: solid; border-bottom-style:
solid; border-left-style: solid; border-top-color: rgb(51, 51,
51); border-right-color: rgb(51, 51, 51); border-bottom-color:
rgb(51, 51, 51); border-left-color: rgb(51, 51, 51); ">
  <head style="border-top-width: 1px; border-right-width:
  1px; border-bottom-width: 1px; border-left-width: 1px;
  border-top-style: solid; border-right-style: solid; border-
  bottom-style: solid; border-left-style: solid; border-top-
  color: rgb(51, 51, 51); border-right-color: rgb(51, 51, 51);
  border-bottom-color: rgb(51, 51, 51); border-left-color:
  rgb(51, 51, 51); ">…</head>
  <body style="border-top-width: 1px; border-right-width:
  1px; border-bottom-width: 1px; border-left-width: 1px;
  border-top-style: solid; border-right-style: solid; border-
  bottom-style: solid; border-left-style: solid; border-top-
  color: rgb(51, 51, 51); border-right-color: rgb(51, 51, 51);
  border-bottom-color: rgb(51, 51, 51); border-left-color:
  rgb(51, 51, 51); ">
    <div class="container" style="border-top-width: 1px;
```

html > body

Styles panel:
```
▼ Styles
element.style {
  ▼border: 1px solid ■#333;
    border-top-width: 1px;
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-left-width: 1px;
    border-top-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-top-color: ■#333;
    border-right-color: ■#333;
    border-bottom-color: ■#333;
    border-left-color: ■#333;
}
Matched CSS Rules
body {                          user agent stylesheet
  display: block;
  ▶ margin: 8px;
}
```

Selecting Elements By Using HTML5 Tags

- Other CSS selectors work the same way that the wildcard selector works, they just are more specific in the elements that are part of the matched set.

- You can select any element within the DOM using the element selector – you need to pass a tag name to the selector, which is present in the page.

- This selector uses the native JavaScript method `getElementByTagName()`.

- The example on the next two pages illustrates this concept by setting the font-family and font-size properties of the `<h1>` tags .

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

`hello.php` `welcome.php` `php.ini` `index.jsp` `MySQLServlet.java` `example 1.html` `example 2.html` `example 3.html` `example 4.html`

```
 1   <!DOCTYPE html>
 2   <html>
 3     <head>
 4       <title>jQuery  - Example 4</title>
 5       <meta charset="utf-8">
 6       <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
 7       <script>
 8           $(document).ready(function() {
 9             $('h1').css('font-family','arial,verdana');
10             $('h1').css('font-size','64pt');
11           });
12       </script>
13     </head>
14   <body>
15     <div class="container">
16         <h1>Hello From jQuery.</h1>
17     <div class="container">
18         <h2>This is inside a second div element in the page.</h2>
19     </div>
20     </div>
21   </body>
22   </html>
```

Hyper Text Markup Language file   length : 540   lines : 23   Ln : 6   Col : 70   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

File   Edit   View   History   Bookmarks   Develop   Window   Help

file:///C:/COURSES/CIS%204004%20-%20Web%20Based%20Information%20

Google

WebCourses   Apple   Yahoo!   Google Maps   YouTube   Wikipedia   News (137) ▾   Popular ▾

jQuery - Example 4

Hello From jQuery.

This is inside a second div element in the page.

Elements   Resources   Network   Scripts   Timeline   Profiles   Audits   »   Search Elements

```
<!DOCTYPE html>
▼<html>
  ► <head>…</head>
  ▼<body style>
    ▼<div class="container">
        <h1 style="font-family: arial, verdana; font-size: 64pt;
        ">Hello From jQuery.</h1>
      ▼<div class="container">
          <h2>This is inside a second div element in the page.</h2>
        </div>
      </div>
  </body>
</html>
```

▼ Computed Style   ☐ Show inherited

  ► display: block;
    height: 184px;
  ► margin-bottom: 8px;
  ► margin-left: 8px;
  ► margin-right: 8px;
  ► margin-top: 8px;
    width: 840px;

▼ Styles   ⚙▾

element.style {

}

Matched CSS Rules

body {   user agent stylesheet
    display: block;
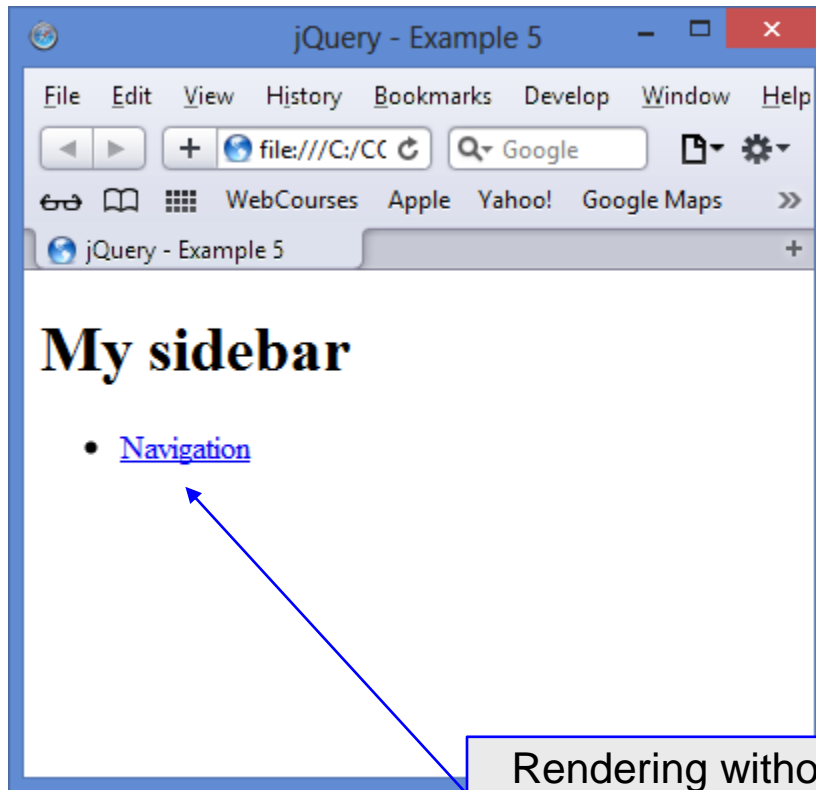
html   body   div.container

Selecting Elements By Using The ID Selector

- The jQuery selector for selecting IDs is the ID ('#') selector.

- The ID selector uses the native JavaScript method getElementById().

- The ID selector always includes the # (hash) symbol when referencing the ID in the selector. Without that symbol, the selector will not work correctly.

- The following example, illustrates using the ID selector. In this case we want to hide the `#sidebar div` using CSS.
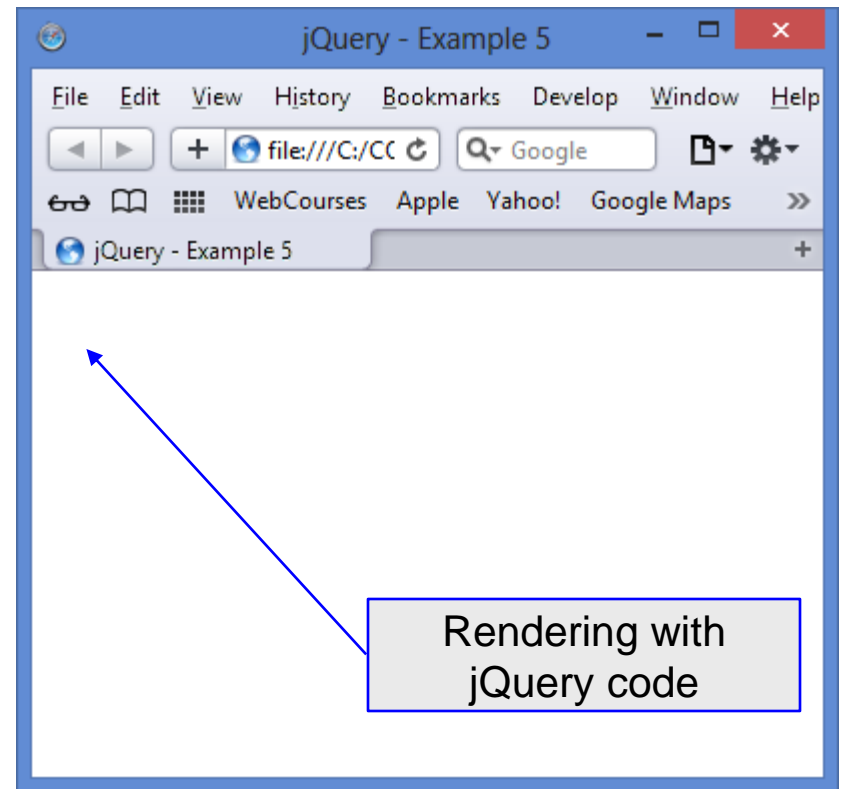
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?   X

php.ini ☒ index.jsp ☒ MySQLServlet.java ☒ example 1.html ☒ example 2.html ☒ example 3.html ☒ example 4.html ☒ example 5.html ☒   ◄ ►

```html
 1  <!DOCTYPE html>
 2  <html lang="en"
 3    <head>
 4      <title> jQuery - Example 5</title>
 5      <meta charset="utf-8">
 6      <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'></script>
 7      <script>
 8          $(document).ready(function() {
 9            $('#sidebar').css('display','none');
10          });
11      </script>
12    </head>
13
14    <body>
15      <div id='sidebar'>
16      <h1>My sidebar</h1>
17      <ul>
18        <li><a href='/nav'>Navigation</a></li>
19      </ul>
20      </div>
21    </body>
22  </html>
```

Hyper Text Markup Language file   length : 464   lines : 22   Ln : 6   Col : 70   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

Rendering without jQuery code

Rendering with jQuery code

Selecting Elements By Class

- Similar to selecting by ID, you can also select elements in your page by class (.class).

- This selector uses the native JavaScript method `getElementByClassName().`

- The class selector selects all elements of a given class in the DOM.

- Using jQuery allows you to accomplish the same thing as with native JavaScript, but do so using less code.

- The following example, illustrates using the class selector. In this case we want to apply a border, padding, and a width to a set of images.

Selecting Elements By Class

- The CSS method allows multiple CSS properties to be passed in using an object literal (a comma separated list composed of name-value pairs), which will keep this statement very clean and concise.

- In this example, I used a class selector to select all instances of the (.telephone) class on the page. Three sets of CSS properties are passed to the CSS method, therefore they need to be included in brackets.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

Tabs: index.jsp   MySQLServlet.java   example 1.html   example 2.html   example 3.html   example 4.html   example 5.html   example 6.html

```html
1    <!DOCTYPE html>
2    <html>
3      <head>
4        <title> jQuery - Example 6</title>
5        <meta charset="utf-8">
6        <script src="jquery-1.10.2.min.js" type="text/javascript"></script>
7        <script>
8            $(document).ready(function() {
9              $('.telephone').css({'border':'1px solid #ccc','width':'250px', 'padding':'10px'});
10           });
11       </script>
12     </head>
13     <body>
14       <div id='container'>
15         <h1>Phone Boxes Galore!</h1>
16         <div class='telephone'><img src='images/phonebooths.jpg'></div>
17         <div class='telephone'><img src='images/phonebooths.jpg'></div>
18         <div class='telephone'><img src='images/phonebooths.jpg'></div>
19         <div class='telephone'><img src='images/phonebooths.jpg'></div>
20         <div class='telephone'><img src='images/phonebooth
21       </div>
22     </body>
23   </html>
```

Note the brackets enclosing the multiple CSS styles

Hyper Text Markup Language file   length : 758   lines : 24   Ln : 6   Col : 33   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

Rendering without jQuery code

Rendering with jQuery code

Phone Boxes Galore!

Selecting One or Many Elements with Multiple Classes

- Sometimes you might have applied multiple classes to the same element and you might want to select only element with those classes applied.

- The class selector accepts multiple classes.

- The example on the next page illustrates this concept. In this markup there are six elements with multiple classes applied to them. I want to hide the two elements using CSS that have the `book` and `inactive` classes applied to them.

- Notice how the selector includes two different classes.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

MySQLServlet.java   example 1.html   example 2.html   example 3.html   example 4.html   example 5.html   example 6.html   example 7.html

```html
 1  <!DOCTYPE html>
 2  <html lang="en">
 3    <head>
 4       <title>jQuery - Example 7</title>
 5       <meta charset="utf-8">
 6       <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'></script>
 7       <script>
 8           $(document).ready(function() {
 9             $('.book.inactive').css('display','none');
10           });
11       </script>
12    </head>
13  <body>
14      <div class='book inactive'>
15          <p>Travel Guide to NYC</p>
16      </div>
17      <div class='book active'>
18          <p>Travel Guide to San Francisco</p>
19      </div>
20      <div class='book inactive'>
21          <p>Travel Guide to Seattle</p>
22      </div>
23      <div class='book active'>
24          <p>Travel Guide to Miami</p>
25      </div>
26      <div class='book active'>
27          <p>Travel Guide to Palo Alto</p>
28      </div>
29  </body>
30  </html>
```

Hyper Text Markup Language file   length : 695   lines : 31   Ln : 6   Col : 70   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

File Edit View History Bookmarks Develop Window Help

◄ ► + 🌐 file:///C:/COURSES/CIS%204004%20-%20Web%20Based ↻ 🔍 Google

∞ 📖 ⊞ WebCourses Apple Yahoo! Google Maps YouTube Wikipedia News (137) ▼ Popular ▼

🌐 jQuery - Example 7   +

Travel Guide to San Francisco

Travel Guide to Miami

Travel Guide to Palo Alto

📑 Elements 📄 Resources ⊙ Network 📋 Scripts 🚩 Timeline » 🔍 Search Elements   ⊗

```
<!DOCTYPE html>
▼<html lang="en">
  ►<head>…</head>
  ▼<body style>
    ►<div class="book inactive" style=
      "display: none; ">…</div>
    ►<div class="book active">…</div>
    ►<div class="book inactive" style=
      "display: none; ">…</div>
    ►<div class="book active">…</div>
    ►<div class="book active">…</div>
  </body>
</html>
```

▶ Computed Style   ☐ Show inherited
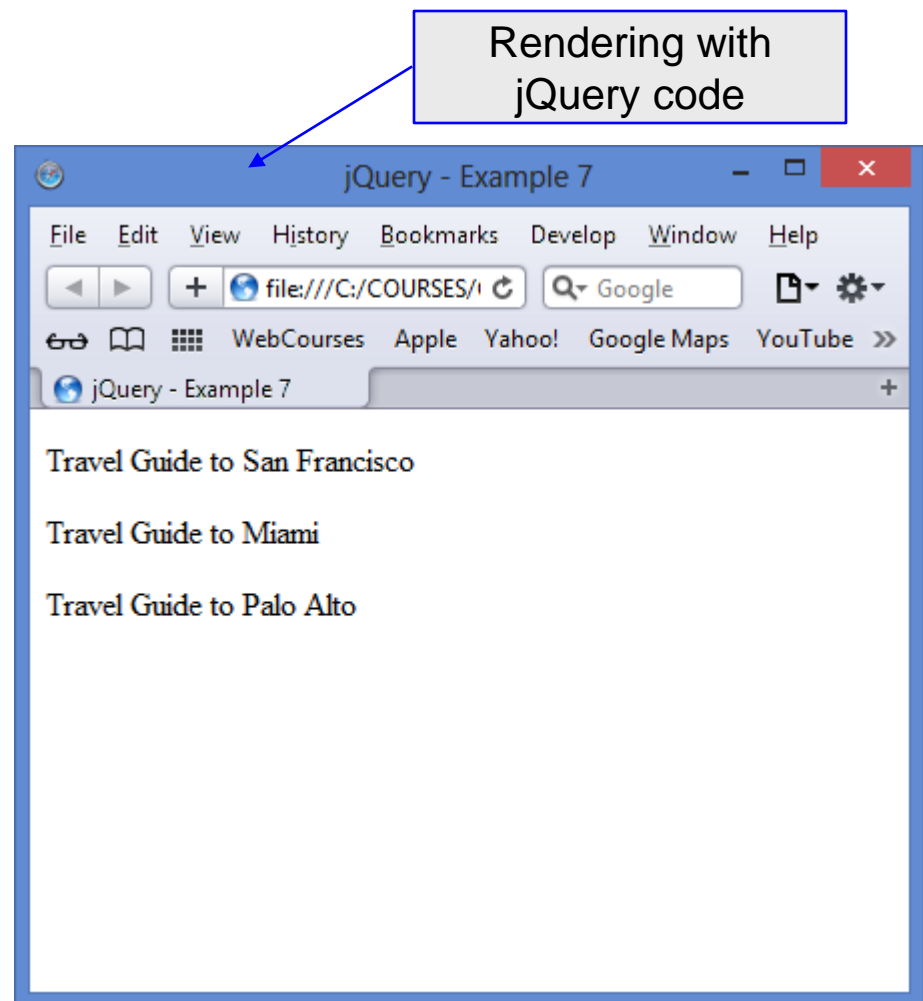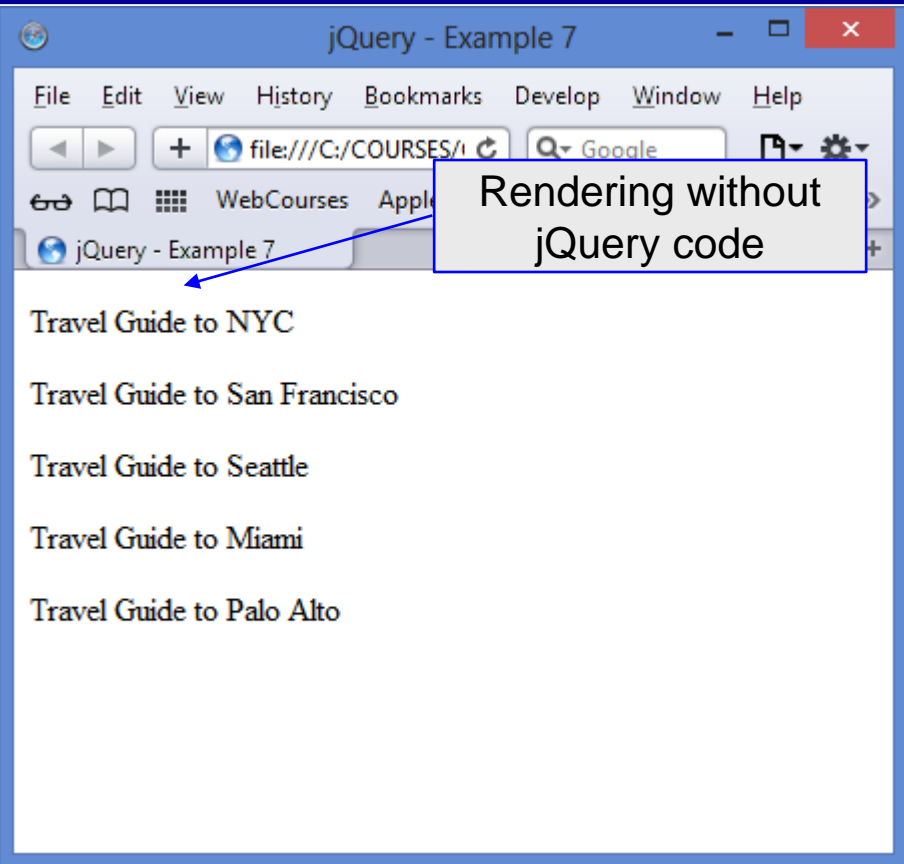▼ Styles   ⚙▾
```
element.style {

}
```
Matched CSS Rules
```
body {                        user agent stylesheet
    display: block;
  ► margin: 8px;
}
```
▶ Metrics
▶ Properties
▶ Event Listeners   ⚙▾

▭ ≽ 🔍 html ⟩ body

Rendering without jQuery code

Travel Guide to NYC

Travel Guide to San Francisco

Travel Guide to Seattle

Travel Guide to Miami

Travel Guide to Palo Alto

Rendering with jQuery code

Travel Guide to San Francisco

Travel Guide to Miami

Travel Guide to Palo Alto

Selecting Page Elements Using Parent-Child Selectors

- Parent child selectors are a useful way to select elements within your page, when `tag, CSS`, and `ID` elements cannot be used.

- The parent-child selector can be very useful when working with nested elements such as navigation menus.

- The parent-child selector allows you to select direct descendants of a given parent element using the > operator. For example, the selector `$('body > p')` will select all of the paragraph elements found within the body as long as they are a child of the body element.

- The example on the next page extends the previous example in that now I want to select the paragraphs within the elements that are members of the `inactive` class and append "Sorry, this book is sold out." to the end of the paragraph tag in red text. I use chaining to add multiple methods to one statement.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?

example 1.html   example 2.html   example 3.html   example 4.html   example 5.html   example 6.html   example 7.html   **example 8.html**

```html
 1   <!DOCTYPE html>
 2   <html lang="en">
 3   <head>
 4       <title>jQuery - Example 8</title>
 5       <meta charset="utf-8">
 6       <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'></script>
 7       <script>
 8         $(document).ready(function() {
 9           $('.book.inactive > p').css('display','none');
10           $('.book.inactive > p').css('display','block').append('    Sorry this book is sold out!').css('color','red');
11         });
12       </script>
13   </head>
14     <body>
15       <div class='book inactive'>
16           <p>Travel Guide to NYC</p>
17       </div>
18       <div class='book active'>
19           <p>Travel Guide to San Francisco</p>
20       </div>
21       <div class='book inactive'>
22           <p>Travel Guide to Seattle</p>
23       </div>
24       <div class='book active'>
25           <p>Travel Guide to Miami</p>
26       </div>
27       <div class='book active'>
28           <p>Travel Guide to Palo Alto</p>
29       </div>
30
31     </body>
32   </html>
```

Hyper Text Markup Language file   length : 858   lines : 33   Ln : 6   Col : 68   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

File   Edit   View   History   Bookmarks   Develop   Window   Help

◄   ►   +   🌐 file:///C:/COURSES/CIS%204004%20-%20Web%20Base   ⟳ | 🔍▾ Google

👓   📖   ⊞   WebCourses   Apple   Yahoo!   Google Maps   YouTube   Wikipedia   News (137) ▾   Popular ▾

🌐 jQuery - Example 8   +

Travel Guide to NYC   Sorry this book is sold out!

Travel Guide to San Francisco

Travel Guide to Seattle   Sorry this book is sold out!

Travel Guide to Miami

Travel Guide to Palo Alto

<📄| Elements   🖼 Resources   🌐 Network   🔧 Scripts   »   🔍 Search Elements   ⊗

```
<!DOCTYPE html>
▼<html lang="en">
  ►<head>…</head>
  ▼<body style>
    ▼<div class="book inactive">
      ►<p style="display: block;
        color: red; ">…</p>
      </div>
    ▼<div class="book active">
        <p>Travel Guide to San
        Francisco</p>
      </div>
    ►<div class="book
      inactive">…</div>
    ►<div class="book active">…</div>
```

▶ Computed Style   ☐ Show inherited ⌃
▼ Styles   ⚙▾

element.style {

}

Matched CSS Rules

body {   user agent stylesheet
    display: block;
  ► margin: 8px;
}

▶ Metrics
▶ Properties
▶ Event Listeners   ⚙▾

⊡ ≥≡ 🔍   html › body

Selecting Page Elements Using Descendent Selectors

- Parent-child selectors only work if the child is directly related to the parent, like `<li>` tags are related between two `<ul>` tags. When you run into a situation where the element you want to target is two or three or more levels away from the parent, then you need to use descendent selectors.

- The difference between parent-child selectors and descendent selectors is in the > operator. If this operator is not included, then the selector matches not only direct child elements but also any/all descendent elements.

- The example on the following page illustrates a descendent selector. Here we are targeting `<li>` tags inside the `<ul class= sidebar-nav>` tags and applying a border to them.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

example 1.html ☒ example 2.html ☒ example 3.html ☒ example 4.html ☒ example 5.html ☒ example 6.html ☒ example 7.html ☒ example 8.html ☒

```html
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4      <title>jQuery - Example 9</title>
 5      <meta charset="utf-8">
 6      <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'></script>
 7      <script>
 8        $(document).ready(function() {
 9          $('ul li').css('border','3px dashed blue');
10        });
11      </script>
12  </head>
13    <body>
14      <ul class='sidebar-nav'>
15      <li>Link 1</li>
16      <li>Link 2</li>
17      <li><ul>
18          <li>Sub Link 1</li>
19          <li>Sub Link 2</li>
20      </li>
21      </ul>
22    </body>
23  </html>
24
```

Note the missing > operator which makes this descendent selector.

Hyper Text Markup Language file length : 494   lines : 24   Ln : 6   Col : 68   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

File   Edit   View   History   Bookmarks   Develop   Window   Help

◄ ►   +   🌐 file:///C:/COURSES/CIS%204004%20-%20Web%20Base   ⟳   🔍 Google

WebCourses   Apple   Yahoo!   Google Maps   YouTube   Wikipedia   News (137) ▼   Popular ▼

🌐 jQuery - Example 9   +

- • Link 1
- • Link 2
- •   ○   Sub Link 1
-          ○   Sub Link 2

◀⊞▶ Elements   🖼 Resources   ⊙ Network   {} Scripts   »   🔍 Search Elements   ⊗

```
<!DOCTYPE html>
▼<html lang="en">
  ►<head>…</head>
  ▼<body style>
    ▼<ul class="sidebar-nav">
      <li style="border-top-width:
      3px; border-right-width: 3px;
      border-bottom-width: 3px;
      border-left-width: 3px; border-
      top-style: dashed; border-right-
      style: dashed; border-bottom-
      style: dashed; border-left-
      style: dashed; border-top-color:
      blue; border-right-color: blue;
      border-bottom-color: blue;
      border-left-color: blue; ">Link
      1</li>
      <li style="border-top-width:
      3px; border-right-width: 3px;
      border-bottom-width: 3px;
      border-left-width: 3px; border-
```

► Computed Style   ☐ Show inherited

▼ Styles   ⚙▾

element.style {

}

Matched CSS Rules

body {   user agent stylesheet
    display: block;
    ► margin: 8px;
}

► Metrics

► Properties

► Event Listeners   ⚙▾

🔲 ≡↓ 🔍   html ▶ body

Selecting Multiple Elements

- You might also run into situations where you need to select multiple types of elements, which could be a combination of classes, IDs, HTML tags, as well as parent-child relationships.

- Using the jQuery selector, you can add multiple elements just by creating a comma separated list.

- In the following example, I need to select five individual classes and two individual IDs using a comma-separated list. I then want to apply a gray background color to all of them using the CSS method, and then, with the last ID, apply the background color to the paragraph tag found within the `#footer` element.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

example 4.html   example 5.html   example 6.html   example 7.html   example 8.html   example 9.html   example 10.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <title> jQuery - Example 10</title>
   <meta charset="utf-8">
   <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'></script>
   <script>
            $(document).ready(function() {
         $('.book-one, .book-two, .book-three, .book-four, .book-five, #header, #footer p').css
         ('background-color','#ccc');

      });
   </script>
</head>
  <body>
    <div id='header'><h1>Book Club</h1></div>
    <div class='book-one'>
        <p>Travel Guide to NYC</p>
    </div>
    <div class='book-two'>
        <p>Travel Guide to San Francisco</p>
    </div>
    <div class='book-three'>
        <p>Travel Guide to Seattle</p>
    </div>
    <div class='book-four'>
        <p>Travel Guide to Miami</p>
    </div>
    <div class='book-five'>
        <p>Travel Guide to Palo Alto</p>
    </div>
    <div id='footer'><p>Copyright 2013</p></div>
  </body>
</html>
```

Hyper Text Markup Language file   length : 898   lines : 35   Ln : 32   Col : 39   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

jQuery – Example 10

jQuery - Example 10

File   Edit   View   History   Bookmarks   Develop   Window   Help

file:///C:/COURSES/CIS%204004%20-%20Web%20Base

Google

WebCourses   Apple   Yahoo!   Google Maps   YouTube   Wikipedia   News (137)   Popular

jQuery - Example 10

Book Club

Travel Guide to NYC

Travel Guide to San Francisco

Travel Guide to Seattle

Travel Guide to Miami

Travel Guide to Palo Alto

Copyright 2013

Rendering with jQuery code

Elements   Resources   Network   Scripts   »   Search Elements

```
<!DOCTYPE html>
▼<html lang="en">
  ►<head>…</head>
  ▼<body style>
    ►<div id="header" style=
      "background-color: rgb(204, 204,
      204); ">…</div>
    ►<div class="book-one" style=
      "background-color: rgb(204, 204,
      204); ">…</div>
    ►<div class="book-two" style=
      "background-color: rgb(204, 204,
      204); ">…</div>
    ►<div class="book-three" style=
      "background-color: rgb(204, 204,
      204); ">…</div>
    ►<div class="book-four" style=
```

html > body

► Computed Style   ☐ Show inherited
▼ Styles
element.style {

}
Matched CSS Rules
body {   user agent stylesheet
    display: block;
    ► margin: 8px;
}
► Metrics
► Properties
► Event Listeners

Filtering DOM Elements Using jQuery Selector Filters

- Filtering allows you to refine the elements that you are targeting with your selectors.

- Filters are very handy when you're trying to target just one, or a few, elements within the DOM. If you have a static HTML document, it is easy to adjust the the HTML, but in cases where the DOM changes with every page request or load, you need to use a dynamic front-end language such as JavaScript to add formatting on the fly.

- A filter is defined by a colon that precedes the actual filter, such as `:filter`.

- There are a number of filters defined in jQuery. The table on the next page lists some of the more popular ones.

Filtering DOM Elements Using jQuery Selector Filters

Filter Name	Filter Function	CSS3 Selector
:even and :odd	finds even and odd based on index	
:header	finds elements that are <h1>…<h6> tags	
:first-child (:last-child and :only-child)	finds first, last, or only child	yes
:nth-child	finds the nth child – n must be specified	yes
:parent	finds the parent of an element	
:visible	finds an element that is visible	
:hidden	finds an element that is hidden	
:contains('this is text')	finds an element which contains the specified text	

Creating A Striped Table Using :even And :odd Filters

- Recall your JavaScript work in Project 4 where you created striped tables using JavaScript to color alternate rows of a table.

- While you were still learning JavaScript and DOM manipulation at the time, this was a fairly difficult task to achieve with native JavaScript.

- As a culminating example for how jQuery can make your coding much easier, I'll show you how to achieve the same effect that you accomplished in Project 4 using filters in jQuery.

- In this case we need the `:even` and `:odd` filters listed in the table on the previous page. I changed the colors a bit from the ones in used in Project 4, but notice how simple the code is to accomplish this task compared to the native JavaScript you wrote for that project.

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

example 4.html   example 5.html   example 6.html   example 7.html   example 8.html   example 9.html   example 10.html   example 11.html

```html
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4    <title> jQuery - Example 11 - Striped Table</title>
 5    <meta charset="utf-8">
 6    <script src="jquery-1.10.2.js" type="text/javascript"> </script>
 7    <script>
 8    $(document).ready(function() {
 9      $('body').css('background','#ccc');
10      $('tr:even').css({'background':'black','color':'yellow'});
11      $('tr:odd').css({'background':'yellow','color':'black'});
12    });
13    </script>
14  </head>
15    <body>
16     <table class="stripe_table">
17  <caption>More JavaScript Books for Your Reading Pleasure</caption>
18    <thead>
19    <tr>
20      <th scope="col">Title</th>
21      <th scope="col">Author</th>
22      <th scope="col">Publisher</th>
23      <th scope="col">Comment</th>
24    </tr>
25    </thead>
26    <tbody>
27    <tr>
28      <td>PPK on JavaScript</td>
29      <td>Peter-Paul Koch</td>
30      <td>New Riders</td>
31      <td>Europe's foremost JS expert gives his personal insights.</td>
32    </tr>
33    <tr>
34      <td>Beginning JavaScript with DOM Scripting and Ajax</td>
35      <td>Chris Heilmann</td>
```

Hyper Text Markup Language file   length : 3552   lines : 196   Ln : 6   Col : 29   Sel : 0 | 0   UNIX   ANSI as UTF-8   INS

More JavaScript Books for Your Reading Pleasure

Title	Author	Publisher	Comment
PPK on JavaScript	Peter-Paul Koch	New Riders	Europe's foremost JS expert gives his personal insights.
Beginning JavaScript with DOM Scripting and Ajax	Chris Heilmann	Apress	Yahoo's JS guru ensures you're not a beginner for long.
Pro JavaScript	John Resig	Apress	Serious JS with the man who brought you jQuery.
Bulletproof Ajax	Jeremy Keith	New Riders	Approachable and readable—a little masterpiece.
Ajax in Action	David Crane & Eric Pascarello	Manning	Modern coding techniques for professional Web programmers.
Programming The World Wide Web	Robert Sebesta	Pearson	A long standing classic - 3 JavaScript chapters, two of which deal with the DOM.