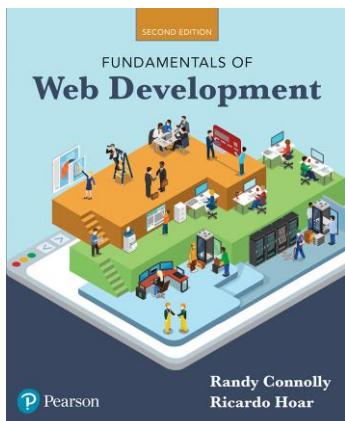


CSE 686 Internet Programming

Week 12: Introduction to JavaScript, Part 3: jQuery

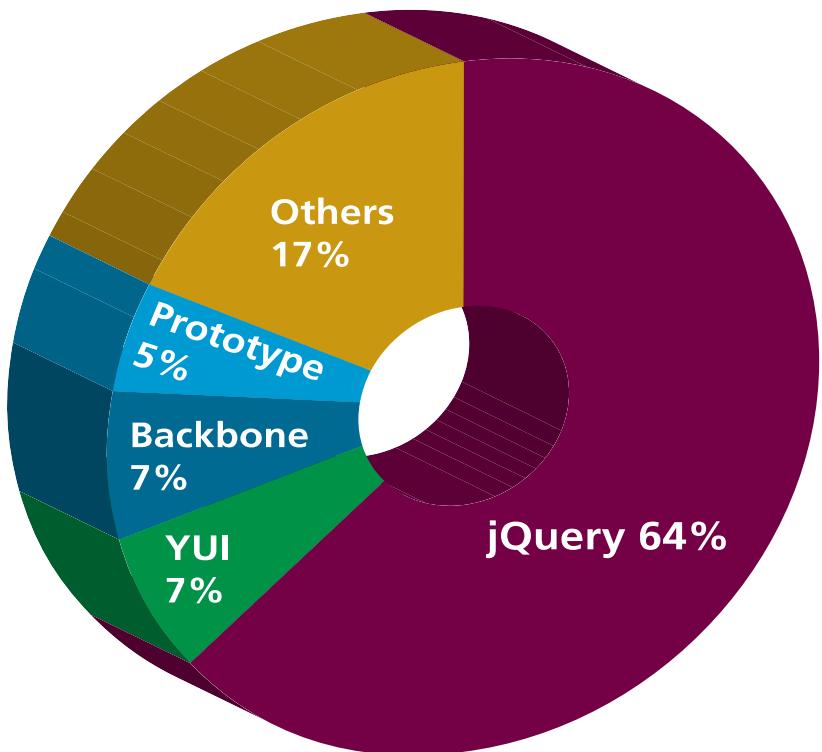
Edmund Yu, PhD
Associate Teaching Professor
esyu@syr.edu

April 4, 2018



A Popular Framework

Top 10,000 Sites



Top Million Sites

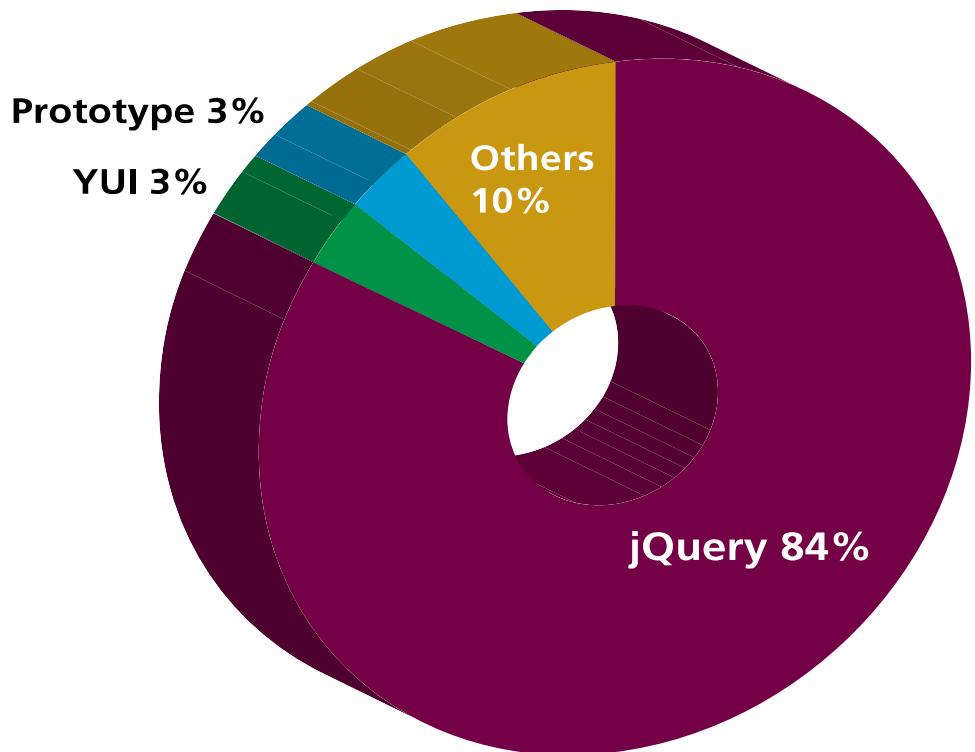


Figure 10.1 Comparison of the most popular JavaScript frameworks (data courtesy of BuiltWith.com)

Including jQuery

- ❖ Use a Content Delivery Network (CDN)

```
<script src="http://code.jquery.com/jquery-3.1.0.min.js">  
</script>
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>Chapter 10</title>
6
7     <script src="http://code.jquery.com/jquery-3.1.0.min.js"></script>
8     <script type="text/javascript">
9         window.jQuery ||
10        document.write('<script src="/jquery-3.1.0.min.js"></script>');
11    </script>
12
13
14 </head>
15 <body>
16
17     nothing yet
18
19 </body>
20 </html>
```

jQuery Selector

- ❖ Remember getElementByID(), etc...
- ❖ The power of jQuery resides in the function named **jQuery()**. There's also an alias for this function named **\$()**. (See next slides)

jQuery Selector

```
/* selecting using regular JavaScript */  
var node = document.getElementById("here");  
var link = document.querySelectorAll("ul li");
```

```
/* equivalent selection using jQuery */
```

```
var node = $("#here");  
var link = $("ul li");
```



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Chapter 10</title>
6      <script src="http://code.jquery.com/jquery-3.2.1.min.js" ></script>
7
8  </head>
9  <body>
10
11 <p id="here">hello <span>there</span></p>
12 <ul>
13     <li>France</li>
14     <li>Spain</li>
15     <li>Thailand</li>
16 </ul>
17
18 <script>
19 /* selecting using regular JavaScript */
20 var node = document.getElementById("here");
21 var link = document.querySelectorAll("ul li");
22
23 /* equivalent selection using jQuery */
24 node = $("#here");
25 link = $("ul li");
26 </script>
27
28
29 </body>
30 </html>
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>Chapter 10</title>
6     <script src="http://code.jquery.com/jquery-3.2.1.min.js" ></script>
7 </head>
8 <body>
9 <p id="here">hello</p>
10 <ul>
11 <li>France</li>
12 <li>Spain</li>
13 <li>Thailand</li>
14 </ul>
15
16 <script>
17 /* selecting using regular JavaScript      */
18 document.getElementById("here").innerHTML = "new content";
19
20 var items = document.querySelectorAll("ul li");
21
22 for (var i=0; i<items.length; i++) {
23     items[i].style.backgroundColor = "yellow";
24 }
25
26 /* equivalent selection using jQuery */
27 $("#here").html("new content");
28 $("ul li").css("background-color", "yellow");
29 </script>
30 </body>
31 </html>
```

jQuery Tutorial

Definition and Usage

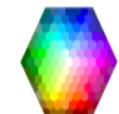
[ORDER NOW](#)

The `html()` method sets or returns the content (`innerHTML`) of the selected elements.

When this method is used to **return** content, it returns the content of the FIRST matched element.

When this method is used to **set** content, it overwrites the content of ALL matched elements.

Tip: To set or return only the text content of the selected elements, use the `text()` method.

COLOR PICKER

Syntax

HOW TO

Return content:

```
$(selector).html()
```

Tabs**Dropdowns****Accordions****Convert Weights****Animated Buttons****Side Navigation****Top Navigation****Modal Boxes****Progress Bars****Parallax****Login Form****HTML Includes****Google Maps****Range Sliders****Tooltips****Slideshow****Filter List****Sort List**

Set content:

```
$(selector).html(content)
```

Set content using a function:

```
$(selector).html(function(index,currentcontent))
```

Parameter

Description

<code>content</code>	Required. Specifies the new content for the selected elements (can contain HTML tags)
<code>function(index,currentcontent)</code>	Optional. Specifies a function that returns the new content for the selected elements

SHARE

[jQuery Get Started](#)
[jQuery Syntax](#)
[jQuery Selectors](#)
[jQuery Events](#)[jQuery Effects](#)[jQuery Hide/Show](#)
[jQuery Fade](#)
[jQuery Slide](#)
[jQuery Animate](#)
[jQuery stop\(\)](#)
[jQuery Callback](#)
[jQuery Chaining](#)[jQuery HTML](#)[jQuery Get](#)
[jQuery Set](#)
[jQuery Add](#)
[jQuery Remove](#)
[jQuery CSS Classes](#)
[jQuery css\(\)](#) jQuery
[jQuery Dimensions](#)[jQuery Traversing](#)[jQuery Traversing](#)
[jQuery Ancestors](#)
[jQuery Descendants](#)
[jQuery Siblings](#)
[jQuery Filtering](#)

jQuery css() Method

The `css()` method sets or returns one or more style properties for the selected elements.

Return a CSS Property

To return the value of a specified CSS property, use the following syntax:

```
css("propertyname");
```

The following example will return the background-color value of the FIRST matched element:

Example

```
$(“p”).css(“background-color”);
```

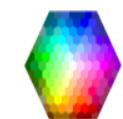
[Try it Yourself »](#)

Set a CSS Property

To set a specified CSS property, use the following syntax:

```
css("propertyname", "value");
```

The following example will set the background-color value for ALL matched elements:

[WordPress.com](#)[COLOR PICKER](#)[HOW TO](#)[Tabs](#)[Dropdowns](#)[Accordions](#)[Convert Weights](#)[Animated Buttons](#)[Side Navigation](#)[Top Navigation](#)[Modal Boxes](#)

[jQuery Ancestors](#)
[jQuery Descendants](#)
[jQuery Siblings](#)
[jQuery Filtering](#)[jQuery AJAX](#)
[jQuery AJAX Intro](#)
[jQuery Load](#)
[jQuery Get/Post](#)[jQuery Misc](#)
[jQuery noConflict\(\)](#)
[jQuery Filters](#)[jQuery Examples](#)
[jQuery Examples](#)
[jQuery Quiz](#)
[jQuery Certificate](#)[jQuery References](#)
[jQuery Selectors](#)
[jQuery Events](#)
[jQuery Effects](#)
[jQuery HTML/CSS](#)
[jQuery Traversing](#)
[jQuery AJAX](#)
[jQuery Misc](#)
[jQuery Properties](#)

jQuery HTML / CSS Methods

The following table lists all the methods used to manipulate the HTML and CSS.

The methods below work for both HTML and XML documents. Exception: the `html()` method.

Method	Description
<u>addClass()</u>	Adds one or more class names to selected elements
<u>after()</u>	Inserts content after selected elements
<u>append()</u>	Inserts content at the end of selected elements
<u>appendTo()</u>	Inserts HTML elements at the end of selected elements
<u>attr()</u>	Sets or returns attributes/values of selected elements
<u>before()</u>	Inserts content before selected elements
<u>clone()</u>	Makes a copy of selected elements
<u>css()</u>	Sets or returns one or more style properties for selected elements
<u>detach()</u>	Removes selected elements (keeps data and events)
<u>empty()</u>	Removes all child nodes and content from selected elements
<u>hasClass()</u>	Checks if any of the selected elements have a specified class name
<u>height()</u>	Sets or returns the height of selected elements
<u>html()</u>	Sets or returns the content of selected elements
<u>innerHeight()</u>	Returns the height of an element (includes padding, but not border)



END-OF-MONTH SALE

UP TO 70% OFF

+ \$1 BOOKS YOUR RESORT

ENDS AT MIDNIGHT APRIL 1ST!

42 hrs

57 min

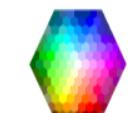
48 sec

✓ ALL INCLUSIVE

- All Meals
- All Drinks
- Activities
- Entertainment

DEALS FROM \$84

COLOR PICKER



HOW TO

- Tabs
- Dropdowns
- Accordions
- Convert Weights
- Animated Buttons
- Side Navigation
- Top Navigation

Basic Selectors

`$('*')`—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.

Some Examples

```
<body>
  <nav>
    <ul>
      $("ul a:link")           |<li><a href="#">Canada</a></li>
      <li><a href="#">Germany</a></li>
      <li><a href="#">United States</a></li>
    </ul>
  </nav>
  <div id="main">
    Comments as of <time>November 15, 2012</time>
    $("#main>time")          |<div>           |<time>November 15, 2012</time>
    <p>By Ricardo on <time>September 15, 2012</time></p>
    <p>Easy on the HDR buddy.</p>
    $("#main div p:first-child") |<hr/>
    <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
      <hr/>
    </div>
    <footer>
      <ul>
        <li><a href="#">Home</a> | </li>
        <li><a href="#">Browse</a> | </li>
      </ul>
    </footer>
  </body>
```

jQuery Selectors					
jQuery Ancestors		jQuery Descendants			
jQuery Siblings		jQuery Filtering			
jQuery Ancestors	jQuery Descendants	jQuery Siblings	jQuery Filtering	jQuery AJAX	jQuery Examples
jQuery Descendants	jQuery Siblings	jQuery Filtering	jQuery AJAX	jQuery Examples	jQuery References
jQuery Siblings	jQuery Filtering	jQuery AJAX	jQuery Examples	jQuery References	jQuery Selectors
jQuery Filtering	jQuery AJAX	jQuery Examples	jQuery References	jQuery Selectors	jQuery Events
jQuery AJAX	jQuery Examples	jQuery References	jQuery Selectors	jQuery Events	jQuery Effects
jQuery Examples	jQuery References	jQuery Selectors	jQuery Events	jQuery Effects	jQuery HTML/CSS
jQuery References	jQuery Selectors	jQuery Events	jQuery Effects	jQuery HTML/CSS	jQuery Traversing
jQuery Selectors	jQuery Events	jQuery Effects	jQuery HTML/CSS	jQuery Traversing	jQuery AJAX
jQuery Events	jQuery Effects	jQuery HTML/CSS	jQuery Traversing	jQuery AJAX	jQuery Misc
jQuery Effects	jQuery HTML/CSS	jQuery Traversing	jQuery AJAX	jQuery Misc	jQuery Properties
jQuery HTML/CSS	jQuery Traversing	jQuery AJAX	jQuery Misc	jQuery Properties	
jQuery Traversing	jQuery AJAX	jQuery Misc	jQuery Properties		
jQuery AJAX	jQuery Misc	jQuery Properties			
jQuery Misc	jQuery Properties				
jQuery Properties					
jQuery Selectors					
Use our jQuery Selector Tester to demonstrate the different selectors.					
Selector	Example	Selects			
*	<code>\$('*')</code>	All elements			
#id	<code>\$("#lastname")</code>	The element with id="lastname"			
.class	<code>\$(".intro")</code>	All elements with class="intro"			
.class,.class	<code>\$(".intro,.demo")</code>	All elements with the class "intro" or "demo"			
element	<code> \$("p")</code>	All <p> elements			
e1,e2,e3	<code> \$("h1,div,p")</code>	All <h1>, <div> and <p> elements			
:first	<code> \$("p:first")</code>	The first <p> element			
:last	<code> \$("p:last")</code>	The last <p> element			
:even	<code> \$("tr:even")</code>	All even <tr> elements			
:odd	<code> \$("tr:odd")</code>	All odd <tr> elements			
:first-child	<code> \$("p:first-child")</code>	All <p> elements that are the first child of their parent			
:first-of-type	<code> \$("p:first-of-type")</code>	All <p> elements that are the first <p> element of their parent			
:last-child	<code> \$("p:last-child")</code>	All <p> elements that are the last child of their parent			

Attribute Selectors

- ❖ An attribute selector provides a way to select elements by either the presence of an element attribute or by the value of an attribute.
- ❖ A list of sample CSS attribute selectors was given in Chapter 4 (Table 4.4), but to jog your memory with an example, consider a selector to grab all **img** elements with an **src** attribute beginning with **/artist/**

```
var artistImages = $("img[src ^='/artist/']");
```

- ❖ Recall that you can select by attribute with square brackets ([attribute]), specify a value with an equals sign ([attribute=value]) and search for a particular value in the beginning, end, or anywhere inside a string with ^, \$, and * symbols, respectively ([attribute^=value], [attribute\$=value], [attribute*=value]). (next slide)

Attribute Selectors

Selector	Matches
[]	A specific attribute.
[=]	A specific attribute with a specific value.
[~=]	A specific attribute whose value matches at least one of the words in a space delimited list of words.
[^=]	A specific attribute whose value begins with a specified value.
[*=]	A specific attribute whose value contains a substring.
[\${=]}	A specific attribute whose value ends with a specified value.

Selector	Matches	Example
[]	A specific attribute.	[title] Matches any element with a title attribute
[=]	A specific attribute with a specific value.	a[title="posts from this country"] Matches any <code><a></code> element whose title attribute is exactly “posts from this country”
[~=]	A specific attribute whose value matches at least one of the words in a space-delimited list of words.	[title~="Countries"] Matches any <code>title</code> attribute that contains the word “Countries”
[^=]	A specific attribute whose value begins with a specified value.	a[href^="mailto"] Matches any <code><a></code> element whose <code>href</code> attribute begins with “mailto”
[*=]	A specific attribute whose value contains a substring.	img[src*="flag"] Matches any <code></code> element whose <code>src</code> attribute contains somewhere within it the text “flag”
[\${=]}	A specific attribute whose value ends with a specified value.	a[href\$=".pdf"] Matches any <code><a></code> element whose <code>href</code> attribute ends with the text “.pdf”

Pseudo-Element Selectors

- ❖ Pseudo-elements are special elements, which are special cases of regular ones.
- ❖ As you may recall from Chapter 4, these pseudo-element selectors allow 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. (See next slides)
- ❖ These selectors can be used in combination with the aforementioned selectors, or alone.
- ❖ Selecting all links that have been visited, for example, would be specified with:

```
var visitedLinks = $("a:visited");
```

Pseudo Selectors

- ❖ A **pseudo-element selector** is a way to select something that does not exist explicitly as an element in the HTML document tree but which is still a recognizable selectable object.
- ❖ A **pseudo-class selector** does apply to an HTML element, but targets either a particular state or, in CSS3, a variety of family relationships.

Table 4.5 Common Pseudo-Class and Pseudo-Element Selectors

Selector	Type	Description
<code>a:link</code>	pseudo-class	Selects links that have not been visited.
<code>a:visited</code>	pseudo-class	Selects links that have been visited.
<code>:focus</code>	pseudo-class	Selects elements (such as text boxes or list boxes) that have the input focus.
<code>:hover</code>	pseudo-class	Selects elements that the mouse pointer is currently above.
<code>:active</code>	pseudo-class	Selects an element that is being activated by the user. A typical example is a link that is being clicked.
<code>:checked</code>	pseudo-class	Selects a form element that is currently checked. A typical example might be a radio button or a check box.
<code>:first-child</code>	pseudo-class	Selects an element that is the first child of its parent. A common use is to provide different styling to the first element in a list.
<code>:first-letter</code>	pseudo-element	Selects the first letter of an element. Useful for adding drop-caps to a paragraph.
<code>:first-line</code>	pseudo-element	Selects the first line of an element.

Pseudo Selectors

```
<style>
    a:link {
        text-decoration: underline;
        color: blue;
    }
    a:visited {
        text-decoration: underline;
        color: purple;
    }
    a:hover {
        text-decoration: none;
        font-weight: bold;
    }
    a:active {
        background-color: yellow;
    }
</style>
```

C:\Users\Edmund\Desktop\686\FunWebDev2\0134481747_cl-268317\code-listings\chapter04\listing04-08.html - Notepad++

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

listing04-03.html listing04-04.css listing04-05.css listing04-06.html listing04-07.html listing04-08.html

```
1 |     <!DOCTYPE html>
2 |     <html>
3 |         <head>
4 |             <title>Share Your Travels</title>
5 |             <style>
6 |                 a:link {
7 |                     text-decoration: underline;
8 |                     color: blue;
9 |                 }
10 |                a:visited {
11 |                    text-decoration: underline;
12 |                    color: purple;
13 |                    font-weight: bold;
14 |                }
15 |                a:hover {
16 |                    text-decoration: none;
17 |                    font-weight: bold;
18 |                }
19 |                a:active {
20 |                    background-color: yellow;
21 |                }
22 |            </style>
23 |        </head>
24 |        <body>
25 |            <p>Links are an important part of any web page. To learn more about
26 |            links visit the <a href="#">W3C</a> website.</p>
27 |            <nav>
28 |                <ul>
29 |                    <li><a href="#">Canada</a></li>
30 |                    <li><a href="#">Germany</a></li>
31 |                    <li><a href="#">United States</a></li>
32 |                </ul>
33 |            </nav>
34 |        </body>
```



File Edit View Favorites Tools Help

Links are an important part of any web page. To learn more about links visit the [W3C](#) website.

- [Canada](#)
- [Germany](#)
- [United States](#)

Contextual Selectors

- ❖ Another powerful CSS selector included in jQuery is the contextual selectors, introduced in Chapter 4. (See next slides)
- ❖ These selectors allowed you to specify elements with certain relationships to one another in your CSS. These relationships included descendant (space), child (>), adjacent sibling (+), and general sibling (~).
- ❖ To select all elements inside of elements you would write
`var para = $("div p");`

Contextual Selectors

- ❖ A contextual selector (in CSS3 also called combinators) allows you to select elements based on their ancestors, descendants, or siblings.
- ❖ It selects elements based on their context or their relation to other elements in the document tree.
- ❖ While some of these contextual selectors are used relatively infrequently, almost all web authors find themselves using **descendant** selectors.
- ❖ A descendant selector matches all elements that are contained within another element. The character used to indicate descendant selection is the **space** character.

Descendant Selectors

context selected element

div p { ... }

Selects a `<p>` element
somewhere
within a `<div>` element

`#main div p:first-child { ... }`

#main div p

Selects the first `<p>` element
somewhere within a `<div>` element
that is somewhere within an element
with an `id="main"`

Contextual Selectors

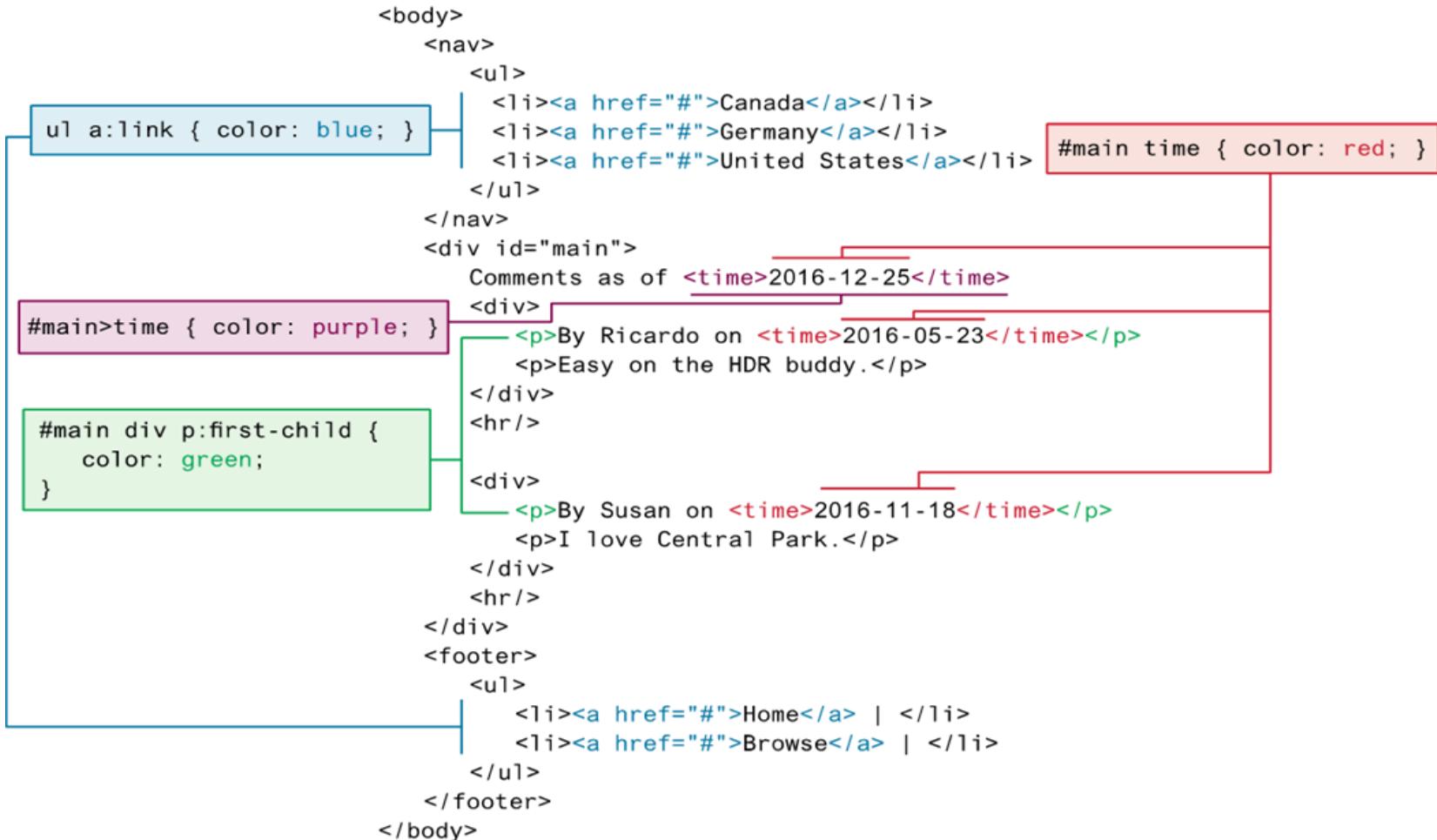




figure04-08.html

```
1 <html>
2 <head>
3 <style>
4     ul a:link { color: blue; }
5
6     #main time { color: red; }
7
8     #main div p:first-child {
9         color: green;
10    }
11
12    #main>time { color: orange; }
13
14 </style>
15
16 </head>
17 <body>
18     <nav>
19         <ul>
20             <li><a href="#">Canada</a></li>
21             <li><a href="#">Germany</a></li>
22             <li><a href="#">United States</a></li>
23         </ul>
24     </nav>
25     <div id="main">
26         Comments as of <time>2016-12-25</time>
27         <div>
28             <p>By Ricardo on <time>2016-05-23</time></p>
29             <p>Easy on the HDR buddy.</p>
30         </div>
31         <hr/>
32
33         <div>
34             <p>By Susan on <time>2016-11-18</time></p>
35             <p>I love Central Park.</p>
36         </div>
37         <hr/>
38     </div>
39     <footer>
40         <ul>
41             <li><a href="#">Home</a> | </li>
```



File Edit View Favorites Tools Help

- [Canada](#)
- [Germany](#)
- [United States](#)

Comments as of 2016-12-25

By Ricardo on 2016-05-23

Easy on the HDR buddy.

By Susan on 2016-11-18

I love Central Park.

- [Home](#) |
- [Browse](#) |

Contextual Selectors

Selector	Matches	Example
Descendant	A specified element that is contained somewhere within another specified element.	div p Selects a <code><p></code> element that is contained somewhere within a <code><div></code> element.
Child	A specified element that is a direct child of the specified element.	div>h2 Selects an <code><h2></code> element that is a child of a <code><div></code> element.
Adjacent Sibling	A specified element that is the next sibling (i.e., comes directly after) of the specified element	h3+p Selects the first <code><p></code> after any <code><h3></code> .
General Sibling	A specified element that shares the same parent as the specified element.	h3~p Selects all the <code><p></code> elements that share the same parent as the <code><h3></code> .

jQuery Filters

- ❖ Filters are special jQuery selectors that work with the other CSS selectors.
- ❖ They start with the colon (:) character and some take parameters much like the **nth-child()** selector in CSS.
- ❖ The jQuery documentation divides them into three categories:
 1. Basic filters
 2. Child filters
 3. Content filters

jQuery Traversing
jQuery Ancestors
jQuery Descendants
jQuery Siblings
jQuery Filtering

jQuery AJAX

jQuery AJAX Intro
jQuery Load
jQuery Get/Post

jQuery Misc

jQuery noConflict()
jQuery Filters

jQuery Examples

jQuery Examples
jQuery Quiz
jQuery Certificate

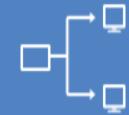
jQuery References

jQuery Selectors
jQuery Events
jQuery Effects
jQuery HTML/CSS
jQuery Traversing
jQuery AJAX
jQuery Misc
jQuery Properties

jQuery Selectors

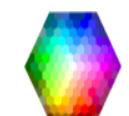
Use our [jQuery Selector Tester](#) to demonstrate the different selectors.

Selector	Example	Selects
*	<code>\$("*")</code>	All elements
#id	<code>\$("#lastname")</code>	The element with id="lastname"
.class	<code>\$(".intro")</code>	All elements with class="intro"
.class,.class	<code>\$(".intro,.demo")</code>	All elements with the class "intro" or "demo"
element	<code> \$("p")</code>	All <p> elements
e1,e2,e3	<code> \$("h1,div,p")</code>	All <h1>, <div> and <p> elements
:first	<code> \$("p:first")</code>	The first <p> element
:last	<code> \$("p:last")</code>	The last <p> element
:even	<code> \$("tr:even")</code>	All even <tr> elements
:odd	<code> \$("tr:odd")</code>	All odd <tr> elements
:first-child	<code> \$("p:first-child")</code>	All <p> elements that are the first child of their parent
:first-of-type	<code> \$("p:first-of-type")</code>	All <p> elements that are the first <p> element of their parent
:last-child	<code> \$("p:last-child")</code>	All <p> elements that are the last child of their parent



Start Now

COLOR PICKER



HOW TO

Tabs
Dropdowns
Accordions
Convert Weights
Animated Buttons
Side Navigation
Top Navigation
Modal Boxes

jQuery Filters

- ❖ Listing 10.4 illustrates a simple but effective use of jQuery filters for programmatically styling rows in a table.

C:\Users\Chris\Desktop\686\FunWebDev\0134481747_ci-268317\code-listings\chapter10\listing10-04.html - Notepad++

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

figure09-14.js functions.js chapter08-project01.js data.js functions.js chapter08-project01.html chapter09-project01.js Chapter09-project01.html example.html example.js figure10-16.html extended-example1.html listing10-16.js listing10-14.html listing10-03.html listing10-04.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Chapter 10</title>
6     <script src="http://code.jquery.com/jquery-3.2.1.min.js" ></script>
7
8   </head>
9   <body>
10
11   <table>
12     <tr><td>Row 0</td></tr>
13     <tr><td>Row 1</td></tr>
14     <tr><td>Row 2</td></tr>
15     <tr><td>Row 3</td></tr>
16     <tr><td>Row 4</td></tr>
17     <tr><td>Row 5</td></tr>
18     <tr><td>Row 6</td></tr>
19   </table>
20
21   <script>
22     /* changes the background color of the even rows */
23     $("table tr:even").css("background-color", "#CFD8DC");
24     /* changes the text color for rows 4 through N */
25     $("table tr:gt(3)").css("color", "#DD2C00");
26   </script>
27
28   </body>
29 </html>
```


jQuery Tutorial

[jQuery HOME](#)

[jQuery Intro](#)

[jQuery Get Started](#)

[jQuery Syntax](#)

[jQuery Selectors](#)

[jQuery Events](#)

jQuery Effects

[jQuery Hide/Show](#)

[jQuery Fade](#)

[jQuery Slide](#)

[jQuery Animate](#)

[jQuery stop\(\)](#)

[jQuery Callback](#)

[jQuery Chaining](#)

jQuery HTML

[jQuery Get](#)

[jQuery Set](#)

[jQuery Add](#)

[jQuery Remove](#)

[jQuery CSS Classes](#)

[jQuery css\(\)](#)

[jQuery Dimensions](#)

GROW BOLDER

LEARN MORE



HOLLYWOOD - ORLANDO

jQuery :gt() Selector

[jQuery Selectors](#)

Example

Select all `<tr>` elements after the 4 first:

```
$("tr:gt(3)")
```

[Try it Yourself »](#)

SQUARESPACE

Make
your
website
today

[Start Now](#)



Definition and Usage

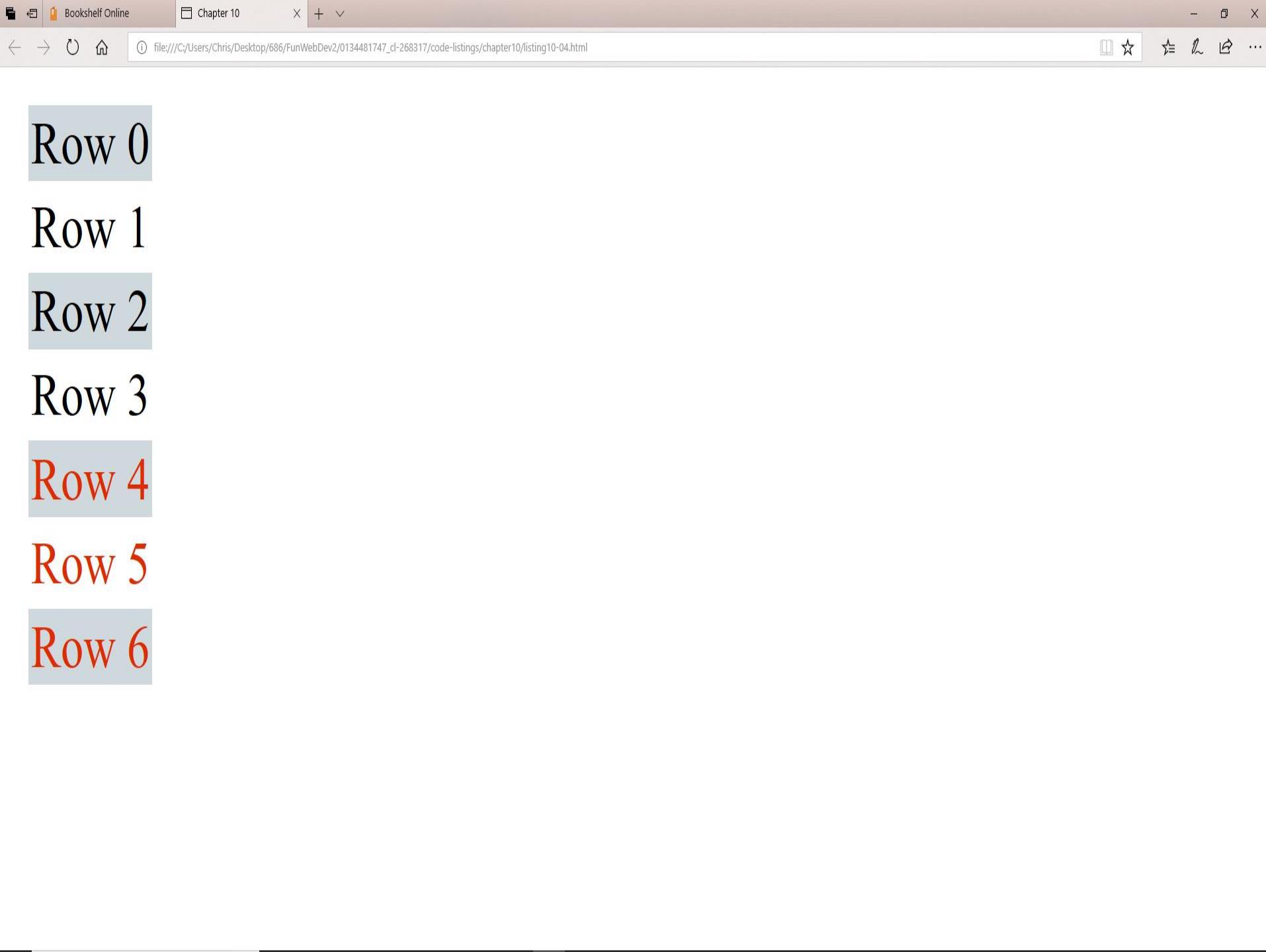
The `:gt()` selector selects elements with an index number higher than a specified number.

The index numbers start at 0.

This is mostly used together with another selector to select the last elements in a group (like in the example above).

Tip: Use the `:lt` selector to select elements index numbers lesser than the specified number.

COLOR PICKER



Row 0

Row 1

Row 2

Row 3

Row 4

Row 5

Row 6

jQuery Filters

- ❖ You can select elements that have a particular child using `:has()`, have no children using `:empty()`, or match a particular piece of text with `:contains()`.

jQuery Tutorial

[jQuery HOME](#)
[jQuery Intro](#)
[jQuery Get Started](#)
[jQuery Syntax](#)
[jQuery Selectors](#)
[jQuery Events](#)

jQuery Effects

[jQuery Hide/Show](#)
[jQuery Fade](#)
[jQuery Slide](#)
[jQuery Animate](#)
[jQuery stop\(\)](#)
[jQuery Callback](#)
[jQuery Chaining](#)

jQuery HTML

[jQuery Get](#)
[jQuery Set](#)
[jQuery Add](#)
[jQuery Remove](#)
[jQuery CSS Classes](#)
[jQuery css\(\)](#)
[jQuery Dimensions](#)



jQuery has() Method

[↳ jQuery Traversing Methods](#)

Example

Return all `<p>` elements that have a `` element inside of them:

```
$("p").has("span")
```

[Try it Yourself »](#)

Definition and Usage

The `has()` method returns all elements that have one or more elements inside of them, that matches the specified selector.

Tip: To select elements that have multiple elements inside of them, use comma (see example below).

Syntax



PREMIUM
SERVERS



\$879

PowerEdge T330
Tower Server[†]

Roll Over for Legal.

[Shop Now](#)



Intel® Xeon® processor

COLOR PICKER

jQuery Tutorial

[jQuery HOME](#)
[jQuery Intro](#)
[jQuery Get Started](#)
[jQuery Syntax](#)
[jQuery Selectors](#)
[jQuery Events](#)

jQuery Effects

[jQuery Hide/Show](#)
[jQuery Fade](#)
[jQuery Slide](#)
[jQuery Animate](#)
[jQuery stop\(\)](#)
[jQuery Callback](#)
[jQuery Chaining](#)

jQuery HTML

[jQuery Get](#)
[jQuery Set](#)
[jQuery Add](#)
[jQuery Remove](#)
[jQuery CSS Classes](#)
[jQuery css\(\)](#)
[jQuery Dimensions](#)



jQuery empty() Method

[« jQuery HTML/CSS Methods](#)

Example

Remove the content of all <div> elements:

```
$("button").click(function(){
    $("div").empty();
});
```

[Try it Yourself »](#)

Definition and Usage

The empty() method removes all child nodes and content from the selected elements.

Note: This method does not remove the element itself, or its attributes.



COLOR PICKER

jQuery :contains() Selector

[jQuery Selectors](#)

Example

Select all <p> elements containing "is":

```
$("p:contains(is)")
```

[Try it Yourself »](#)

**SAVE
25%**

on domains, websites,
email, & more

Use Code:
GETSTARTED

[GET STARTED](#)



Definition and Usage

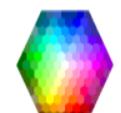
The :contains() selector selects elements containing the specified string.

The string can be contained directly in the element as text, or in a child element.

This is mostly used together with another selector to select the elements containing the text in a group (like in the example above).

Note: The text is case sensitive.

COLOR PICKER



Syntax

```
$(":contains(text)")
```

HOW TO

Tabs

Dropdowns

jQuery Tutorial

[jQuery HOME](#)

[jQuery Intro](#)

[jQuery Get Started](#)

[jQuery Syntax](#)

[jQuery Selectors](#)

[jQuery Events](#)

jQuery Effects

[jQuery Hide/Show](#)

[jQuery Fade](#)

[jQuery Slide](#)

[jQuery Animate](#)

[jQuery stop\(\)](#)

[jQuery Callback](#)

[jQuery Chaining](#)

jQuery HTML

[jQuery Get](#)

[jQuery Set](#)

[jQuery Add](#)

[jQuery Remove](#)

[jQuery CSS Classes](#)

[jQuery css\(\)](#)

[jQuery Dimensions](#)

jQuery Traversing

[jQuery Traversing](#)

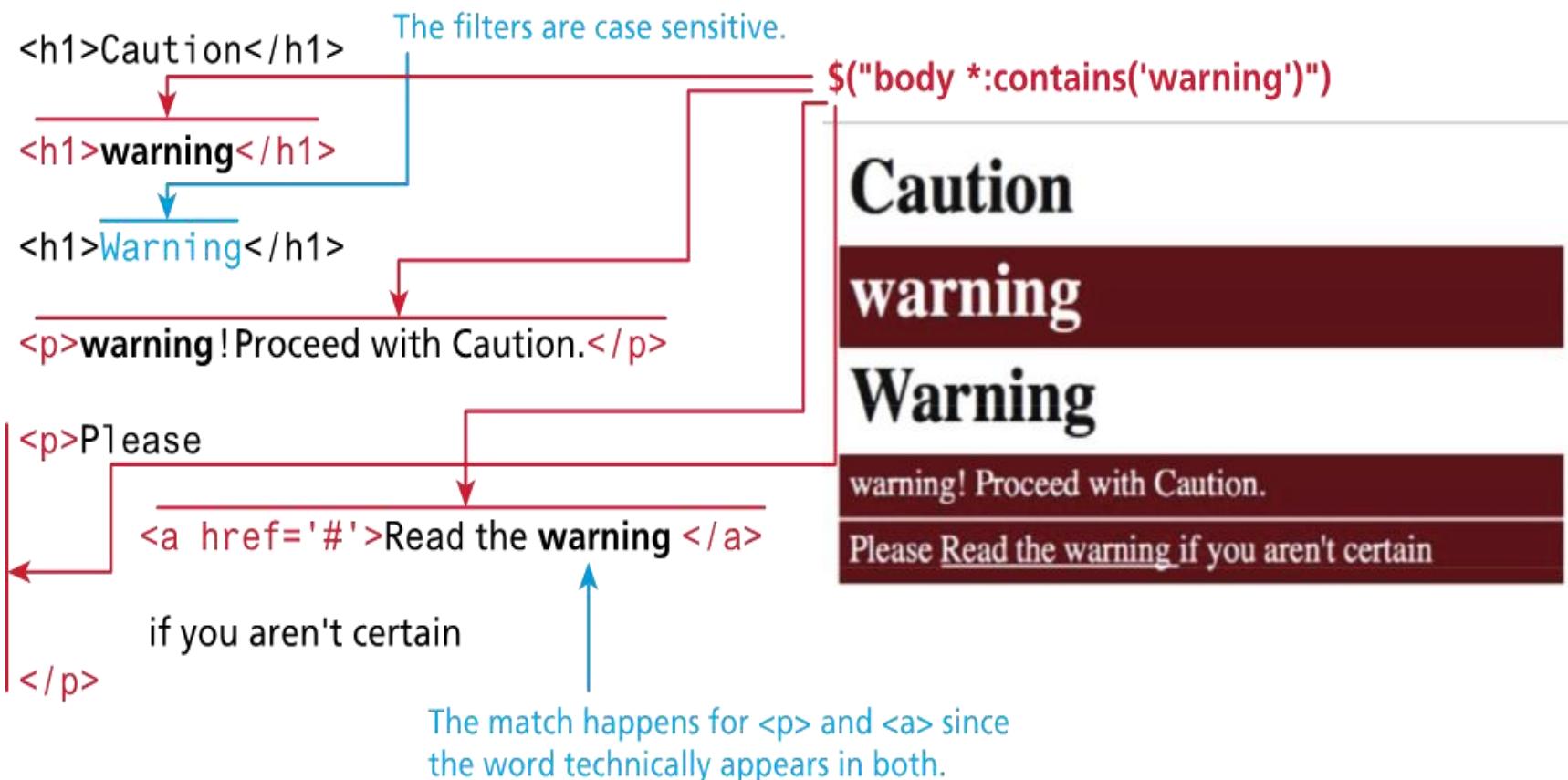
jQuery Filters

- ❖ Consider the following example:

```
var allWarningText = $("body *:contains('warning')");
```

- ❖ It will return a list of all the DOM elements with the word **warning** inside of them.
- ❖ You might imagine how we may want to highlight those DOM elements by coloring the background red as shown in Figure 10.3 with one line of code:

jQuery Filters



Form Selectors

- ❖ Since form HTML elements are frequently used to collect and transmit data, there are jQuery selectors written especially for them.
- ❖ These selectors, listed in Table 10.1 (next 2 slides), allow for quick access to certain types of field as well as fields in certain states.

Table 10.1 jQuery Form Selectors and Their CSS Equivalents When Applicable

Selector	CSS Equivalent	Description
<code>\$(":button")</code>	<code>button, input</code> <code>[type='button']</code>	Selects all buttons.
<code>\$(":checkbox")</code>	<code>[type=checkbox]</code>	Selects all checkboxes.
<code>\$(":checked")</code>	No equivalent	Selects elements that are checked. This includes radio buttons and checkboxes.
<code>\$(":disabled")</code>	No equivalent	Selects form elements that are disabled. These could include <code><button></code> , <code><input></code> , <code><optgroup></code> , <code><option></code> , <code><select></code> , and <code><textarea></code> .
<code>\$(":enabled")</code>	No equivalent	Opposite of <code>:disabled</code> . It returns all elements where the disabled attribute=false as well as form elements with no disabled attribute.
<code>\$(":file")</code>	<code>[type=file]</code>	Selects all elements of type <code>file</code> .
<code>\$(":focus")</code>	No equivalent	The element with focus.
<code>\$(":image")</code>	<code>[type=image]</code>	Selects all elements of type <code>image</code> .

<code>\$(":input")</code>	No equivalent	Selects all <code><input></code> , <code><textarea></code> , <code><select></code> , and <code><button></code> elements.
<code>\$(":password")</code>	<code>[type=password]</code>	Selects all <code>password</code> fields.
<code>\$(":radio")</code>	<code>[type=radio]</code>	Selects all <code>radio</code> elements.
<code>\$(":reset")</code>	<code>[type=reset]</code>	Selects all the <code>reset</code> buttons.
<code>\$(":selected")</code>	No equivalent	Selects all the elements that are currently selected of type <code><option></code> . It does not include checkboxes or radio buttons.
<code>\$(":submit")</code>	<code>[type=submit]</code>	Selects all submit input elements.
<code>\$(":text")</code>	No equivalent	Selects all input elements of type <code>text</code> . <code>\$('[type=text]')</code> is almost the same, except that <code>\$(":text")</code> includes <code><input></code> fields with no type specified.

Common Element Manipulations in jQuery

- ❖ With all of the selectors described in this chapter, you can select any set of elements that you want from a web page.
- ❖ Once selected, you can then manipulate them in a wide variety of ways.

Common Element Manipulations in jQuery

- ❖ The **html()** method is an easy way retrieve and manipulate the HTML contents (the part between the <> and </> tags associated with the innerHTML property in JavaScript) of a selected element.

```
// retrieve the content
```

```
var content = $("#sample").html();
```

```
// modify the content of an element
```

```
$("#sample").html("brand new content");
```

```
// modify the content of ALL <p> elements
```

```
 $("p").html("jQuery is fun");
```

Common Element Manipulations in jQuery

- ❖ 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://funwebdev.com
```

```
$(“a”).attr(“href”, “http://funwebdev.com”);
```

```
// change the class for all images on the page to fancy
```

```
$(“img”).attr(“class”, “fancy”);
```

Common Element Manipulations in jQuery

- ❖ 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
```

Common Element Manipulations in jQuery

- ❖ jQuery provides the extremely intuitive **css()** method.

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

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

Event Handling in jQuery

- ❖ Just like JavaScript, jQuery supports creation and management of listeners/handlers for JavaScript events.
- ❖ While pure JavaScript uses the **addEventListener()** method, jQuery has **on()** and **off()** methods as well as shortcut methods to attach events.

Listing 10.5 Event handling in jQuery versus JavaScript equivalents

```
<button id="example">Click me</button>
<span id="message"></span>

<script>
// javascript version
document.getElementById("example").addEventListener("click",
    function () {
        document.getElementById("message").innerHTML = "you clicked";
    });
// jquery version
$("#example").on("click", function () {
    $("#message").html("you clicked");
});
// alternate jquery version using defined function instead of anonymous one
$("#example").on("click", clicker);

function clicker() {
    $("#message").html("you clicked");
}

// alternate jquery version using click() shortcut method
$("#example").click(function () {
    $("#message").html("you clicked");
});</script>
```

C:\Users\Chris\Desktop\686\FunWebDev\0134481747_ci-268317\code-listings\chapter10\listing10-05.html - Notepad++

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

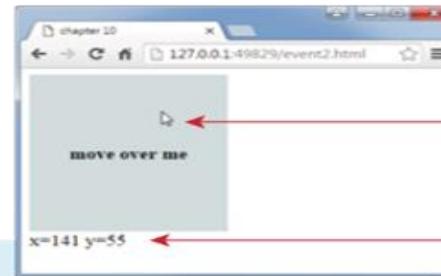
Chapter09-project01.html listing09-07.js chapter08-project01.html example.js styles.css styles.css figure10-04.html listing10-05.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Chapter 10</title>
6     <script src="http://code.jquery.com/jquery-3.2.1.min.js" ></script>
7
8   </head>
9   <body>
10
11   <button id="example">Click me</button>
12   <span id="message"></span>
13   <script>
14
15     // javascript version
16     /*document.getElementById("example").addEventListener("click", function () {
17       document.getElementById("message").innerHTML = "you clicked";
18     });*/
19
20     // jquery version
21     /*$("#example").on("click", function () {
22       $("#message").html("you clicked");
23     });
24
25     // alternate jquery version
26     $("#example").click(function () {
27       $("#message").html("you clicked");
28     });
29   </script>
30   </body>
31 </html>
```

Event Handling in jQuery

Notice that we are chaining together multiple event handlers in one statement. This is a common programming style used by jQuery programmers.

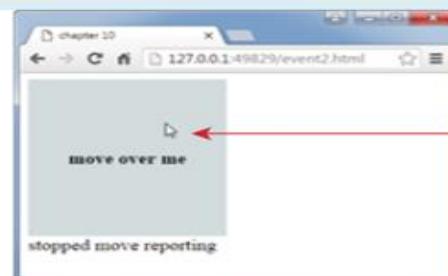
```
$(".panel")  
  .on("mousemove", function (e) {  
    $("#message").html("x=" + e.pageX + " y=" + e.pageY);  
  })  
  
  .on("mouseleave", function (e) {  
    $("#message").html("goodbye!");  
  })  
  
  .on("click", function () {  
    $("#message").html("stopped move reporting");  
    $(".panel").off("mousemove");  
  });
```



When user moves mouse over element, then display x, y coordinates.



But even though the mouse is gone, the panel is still listening for future mouse over events.



However, when the user clicks on the panel, we turn off its listener for mouse moves. Thus future moves will not trigger the mouse move event.



```
1 <html>
2 <head>
3   <title>chapter 10</title>
4   <script src="http://code.jquery.com/jquery-3.2.1.min.js"></script>
5   <style>
6     .panel { width: 200px; height: 200px; background-color: #CFD8DC; font-weight: bold; font-size: 20px;
7       display: flex; align-items: center; justify-content: center; }
8     span { font-size: 20px; }
9   </style>
10 </head>
11 <body>
12   <div class="panel">move over me</div>
13   <span id="message"></span>
14
15 <script>
16   // chaining handlers and
17   $(".panel")
18   .on("mousemove", function (e) {
19     $("#message").html("x=" + e.pageX + " y=" + e.pageY);
20   })
21   .on("mouseleave", function (e) {
22     $("#message").html("goodbye!");
23   })
24   .on("click", function () {
25     $("#message").html("stopped move reporting");
26     $(".panel").off("mousemove");
27   });
28 </script>
29 </body>
30 </html>
```

Event Handling in jQuery: Page Loading

- ❖ Recall that, in JavaScript, using **window.addEventListener("load", ...)** ensures the entire page and all DOM elements are loaded before trying to attach listeners to them.
- ❖ With jQuery we do the same thing but use the **\$(document).ready()** event as shown in Listing 10.6. (next slide)

Frame Events (Revisited)

- ❖ Frame events (see Table 9.10, next slide) are the events related to the browser frame that contains your web page.
- ❖ The most important event is the **load** event, which tells us an object is loaded and therefore can be manipulated via the DOM.
- ❖ In fact, every nontrivial event listener you write requires that the HTML be fully loaded.

Table 9.10 Frame Events in JavaScript

Event	Description
<code>abort</code>	An object was stopped from loading
<code>error</code>	An object or image did not properly load
<code>load</code>	When a document or object has been loaded
<code>resize</code>	The document view was resized
<code>scroll</code>	The document view was scrolled
<code>unload</code>	The document has unloaded

Frame Events (Revisited)

- ❖ As mentioned earlier, a problem can occur if the JavaScript tries to programmatically reference a DOM element that has not yet been loaded.
- ❖ If the code attempts to set up a listener on this not-yet-loaded element then an error will be triggered.
- ❖ For this reason, it is common practice to use the **load** event of the window object to trigger the execution of the rest of the page's scripts, as shown below:

```
window.addEventListener("load", function() {  
  // the DOM can be safely manipulated within this function  
  // ...  
});
```

C:\Users\Chris\Desktop\686\FunWebDev\20134481747_ci-268317\code-listings\chapter10\listing10-06.html - Notepad++

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

Chapter09-project01.html listing09-07.js chapter08-project01.html example.js styles.css styles.css figure10-04.html listing10-05.html listing10-06.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Chapter 10</title>
6     <script src="http://code.jquery.com/jquery-3.2.1.min.js" ></script>
7   <script>
8     $(document).ready(function() {
9       // set up listeners knowing page loads before this runs
10      $("#example").click(function () {
11        $("#message").html("you clicked");
12      });
13    });
14  </script>
15 </head>
16 <body>
17
18  <button id="example">Click me</button>
19  <span id="message"></span>
20
21 </body>
22 </html>
```

Event Handling in jQuery: Page Loading

- ❖ It is worth noting here that there is no **document.ready** event in the regular JavaScript DOM: this is a jQuery-only addition.
- ❖ Since the jQuery **ready()** method can only be used in the context of the document object, it is common to use the simpler equivalent shorthand:

```
$(function () {  
    ...  
});
```



Your donations help fund the continued development and growth of **jQuery**.

SUPPORT THE PROJECT

Download API Documentation Blog Plugins Browser Support

Search



- Ajax
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
- Deprecated 1.3

.ready()

Categories: [Events](#) > [Document Loading](#)

Returns: [jQuery](#)

Description: Specify a function to execute when the DOM is fully loaded.

↳ **.ready(handler)**

version added: 1.0

handler

Type: [Function\(\)](#)

A function to execute after the DOM is ready.

The `.ready()` method offers a way to run JavaScript code as soon as the page's Document Object Model (DOM) becomes safe to manipulate. This will often be a good time to perform tasks that are needed before the user views or interacts with the page, for example to add event handlers and initialize plugins. When multiple functions are added via successive calls to this method, they run when the DOM is ready in the order in which they are added. As of jQuery 3.0,

jQuery ready() Method

[jQuery Event Methods](#)

Take
your
website
today

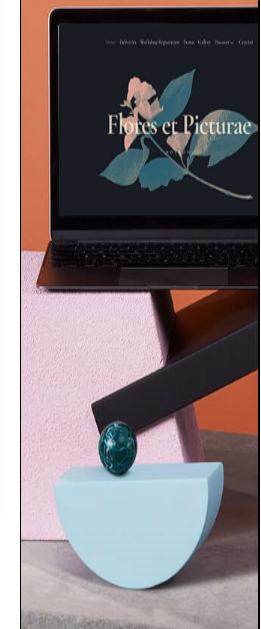
[Start Now](#)

Example

Use ready() to make a function available after the document is loaded:

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").slideToggle();
    });
});
```

[Try it Yourself »](#)

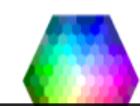


Definition and Usage

The ready event occurs when the DOM (document object model) has been loaded.

COLOR PICKER

Because this event occurs after the document is ready, it is a good place to have all other jQuery events and functions. Like in the example above.



DOM Manipulation: Creating Nodes

```
// pure JavaScript way
```

```
var jsLink = document.createElement("a");
jsLink.href = "http://www.funwebdev.com";
jsLink.innerHTML = "Visit Us";
jsLink.title = "JS";
```

```
// jQuery version 1
```

```
var link1 = $('<a href="http://funwebdev.com"
              title="jQuery">Visit Us</a>');
```

Listing 10.7 A comparison of node creation in JS and jQuery

```
// pure JavaScript way

var jsLink = document.createElement("a");
jsLink.href = "http://www.funwebdev.com";
jsLink.innerHTML = "Visit Us";
jsLink.title = "JS";

// jQuery version 1

var link1 = $('<a href="http://funwebdev.com" title="jQuery">Visit
    Us</a>');

// jQuery version 2

var link2 = $("<a></a>");
link2.attr("href", "http://funwebdev.com");
link2.attr("title", "jQuery verbose");
link2.html("Visit Us");

// version 3 ... also not creating a temporary variable which
// will be more typical once we start chaining methods (see next
// section)

$('<a>', {
    href: 'http://funwebdev.com',
    title: 'jQuery',
    text: 'Visit Us'
})
```

C:\Users\Chris\Desktop\686\FunWebDev\0134481747_ci-268317\code-listings\chapter10\listing10-07.html - Notepad++

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

tweets.txt obama.txt listing10-07.html data.js styles.css chapter08-project01.html data.js functions.js chapter08-project01.js

```
15 jsLink.href = "http://www.funwebdev.com";
16 jsLink.innerHTML = "Visit Us";
17 jsLink.title = "JS";
// document.getElementById("example").appendChild(jsLink);
19
20 // jQuery version 1
21 var link1 = $("<a href='http://funwebdev.com' title='jQuery'>Visit Us</a>");
22 //$("#example").append(link1);
23
24 // jQuery version 2
25 var link2 = $("<a></a>");
26 link2.attr("href", 'http://funwebdev.com');
27 link2.attr("title", "jQuery verbose");
28 link2.html("Visit Us");
29
30 //$("#example").append(link2);
31
32 // version 3 ... also not creating a variable which will be more common once we start chaining
33 $('<a>', {
34     href: 'http://funwebdev.com',
35     title: 'jQuery',
36     text: 'Visit Us'
37 }
38 ).appendTo($('#example'));
39
40
41 </script>
42
43 </body>
44 </html>
```

DOM Manipulation: Creating Nodes

```
// jQuery version 2  
var link2 = $('</a>');  
link2.attr("href","http://funwebdev.com");  
link2.attr("title","jQuery verbose");  
link2.html("Visit Us");
```

DOM Manipulation: Creating Nodes

```
// version 3
$('<a>', {
    href: 'http://funwebdev.com',
    title: 'jQuery',
    text: 'Visit Us'
});
```

DOM Manipulation: Adding Nodes

```
<div class="dest">  
existing content  
</div>
```

```
$(".dest").append(link);  
  
<div class="dest">  
existing content  
<a href="http://funwebdev.com">Fun</a>  
</div>
```

```
link.appendTo($(".dest"));  
  
<div class="dest">  
existing content  
<a href="http://funwebdev.com">Fun</a>  
</div>
```

```
$(".dest").before(link);  
  
<a href="http://funwebdev.com">Fun</a>  
<div class="dest">  
existing content  
</div>
```

```
link.insertBefore($(".dest"));  
  
<a href="http://funwebdev.com">Fun</a>  
<div class="dest">  
existing content  
</div>
```

```
var link = $('Fun');
```

```
$(".dest").prepend(link);
```

```
<div class="dest">  
<a href="http://funwebdev.com">Fun</a>  
existing content  
</div>
```

```
link.prependTo($(".dest"));
```

```
<div class="dest">  
<a href="http://funwebdev.com">Fun</a>  
existing content  
</div>
```

```
$(".dest").after(link);
```

```
<div class="dest">  
existing content  
</div>  
<a href="http://funwebdev.com">Fun</a>
```

```
link.insertAfter($(".dest"));
```

```
<div class="dest">  
existing content  
</div>  
<a href="http://funwebdev.com">Fun</a>
```