SWE 363: Web Engineering & Development

Module 4-3

JQuery

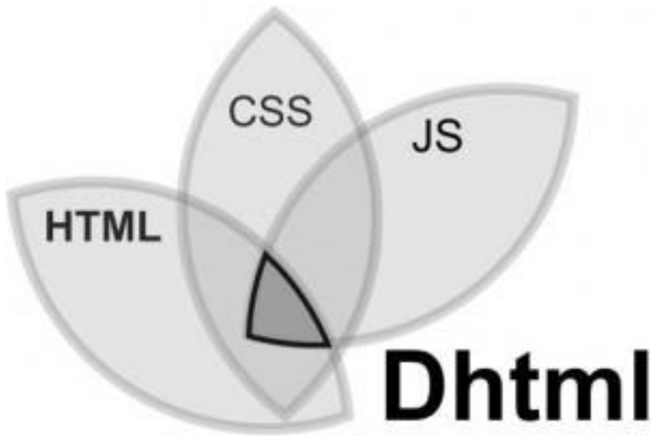
Objectives

- Describe the role played by jQuery
- Install/Configure jQuery to run on your web pages
- Become familiar with a page containing everything you need to get started with jQuery that integrates CSS (internal and external), CSS classes, standard JS function, and the jQuery '`ready()`' function.

Outline

- ❑ Introduction
- ❑ Syntax
- ❑ Selectors
- ❑ CSS manipulation
- ❑ HTML event methods
- ❑ Effects and animations
- ❑ jQuery Traversing

Remember that ..



- Manipulating the web page's structure is essential for creating a highly responsive UI

- Two main approaches
  - Manipulate page via plain JS
  - Manipulate page using JS + library (e.g., jQuery)

Introduction

- A JavaScript library is typically a file or collection of files >> each has a bunch JS functions.
  - These files have (usually) been widely tested and finessed to tackle some commonly encountered problems/issues in web page design.

- There are several JS libraries out there including:
  - Dojo
  - Prototype
  - jsPHP
  - And........ **jQuery**

- If you wish to use any of these libraries, you can easily download and/or link to any of these libraries in your web pages.

jQuery

- jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML
  - It simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.
- jQuery is free, open source software Dual-licensed under the MIT License and the GNU General Public License
- jQuery helps web developers to create simple pieces of interaction without being forced to write long, complex, book-length pieces of code.
- Advantages over just JavaScript
  - Much easier to use
  - Eliminates cross-browser problems

More advantages of using jQuery

- JQuery is intensively tested, revised and updated.
- With jQuery: most common JS actions can be written with fewer lines of code
- Lets you write JS quicker and easier.
  - JQuery uses the fast and best lines of code for accomplishing most tasks.
- Browser compatibility – you can write code which runs across browsers without having to know the various browser intricacies and won't break.
- Simplification of usually complicated operations – complex operations like <u>Ajax interactions</u>, <u>animation</u>, <u>event handling</u> etc. are handled by jQ with the best lines of code.

Why jQuery has become so popular?

- ❑ Each JS library has its own advantages and disadvantages.
- ❑ Some of the reasons that jQuery has become so popular:
  - Small size: Every time a user visits your page, along with the HTML file, CSS file(s), images and so on that will be needed to be downloaded to their browser, they will also need to download the library. Some versions of the jQuery library are less than 10K in size!
  - Popular - means that there is a large and active community constantly testing, updating, and improving the code.
    - ➢ Tested: Because of the widespread use of jQuery, bugs and other limitations are quickly revealed and fixed.
    - ➢ Plug-Ins: These are small add in programs that typically have some highly specialized use. The large jQuery community has resulted in the development of thousands of plug ins.
  - Free: jQuery is free.

How do I get it

- jQuery exists as a single external JS file. You can run jQuery one of two ways:
  - Download the jQuery library file to your computer
  - Link to a remotely hosted version of the library code from a reliable website

- Each option has its pros and cons
  - Download: The main benefit is that if you are offline, the file will still be there on your computer when you are developing your code.
  - Link: These externally linked files are typically the most up to date. However, it requires that the viewer of your page is online. It also takes a few extra milliseconds which can, in theory, slow down your site a little bit. Still, once your page goes 'live', we typically use the version linking to an external file.

- For using the externally linked version
  - There are many different servers from reputable companies such as Microsoft, Google, jQuery and others. You can easily find links to these online.

How do I get it..

- Download the jQuery library from jQuery.com
  - The jQuery library is a single JavaScript file, put it in your website folder and reference it with the HTML <script> tag:

```
<head>
<script src="jquery-3.2.1.min.js"></script>
</head>
```

- Reference the online page. e.g. Google

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</script>
</head>
```

  - As discussed, other can be utilized http://jquery.com/download/

How can I learn it

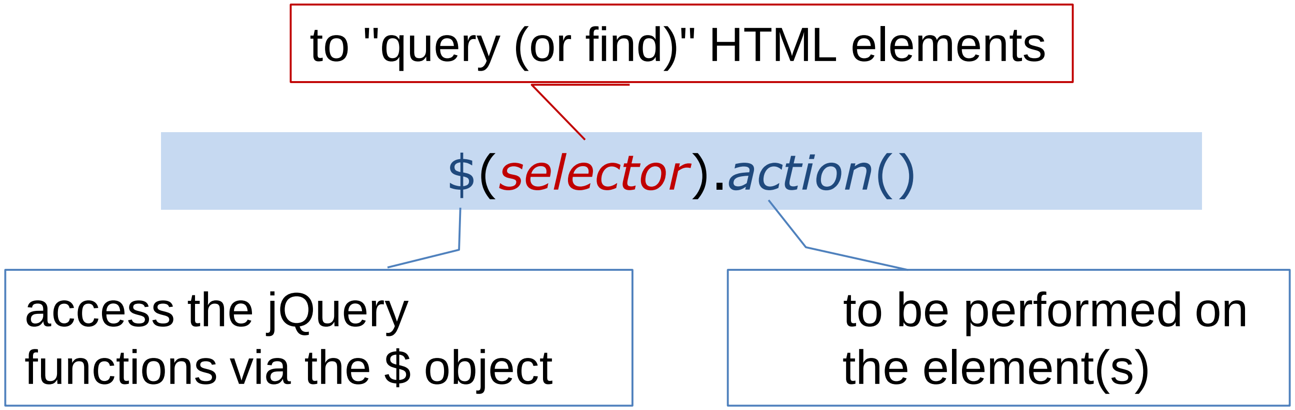
- jQuery in Action by Bibeault & Katz, Manning
- jQuery Visual Quickstart Guide by Holzner, Peachpit
- www.jquery.com
- http://learn.jquery.com
- www.javascripttoolbox.com/jquery
- http://www.w3schools.com/jquery/jquery_examples.asp
- https://www.tutorialspoint.com/jquery/index.htm
- https://www.codeschool.com/courses/try-jquery

5 Things jQuery Provides

- Select DOM elements on a page – <u>one element</u> or a <u>group of them</u>
- Set properties of DOM elements, in groups (“Find something, do something with it”)
- Manipulate DOM elements: Creates, deletes, shows, hides DOM elements
- Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
- AJAX calls

jQuery Syntax

- JQuery syntax is employed for **selecting** HTML elements and **performing** some **action** on the element(s).



to "query (or find)" HTML elements

$(*selector*).*action*()

access the jQuery functions via the $ object

to be performed on the element(s)

- Examples:
  - $(this).hide() - hides the current element.
  - $("p").hide() - hides all <p> elements.
  - $(".test").hide() - hides all elements with class="test".
  - $("#test").hide() - hides the element with id="test".

Staring example

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
<script>
    function func(){
       $(".test").hide();
       }
</script>
</head>

<body>
      <h2 class="test"> Heading 2 </h2>
      <p onclick="func()"> click here to hide Heading 2, P2 and P3 </p>
      <p class="test"> Paragraph 2 </p>
      <p class="test"> Paragraph 3 </p>
      <p onclick="$(this).hide();"> click here to hide me  </p>
</body>
```

The `'ready()'` function

- jQuery takes into account that: DOM can't be used before constructing the page
- The 'ready' function executes ***after*** the *entire web page has finished loading* in the visitor's browser
- You will typically include this function on *every* page in which you are using jQuery.

```
<script>
   $(document).ready(
      function()
      {
         // Some jQuery code will
         // go in here...
      }
   );
</script>
```

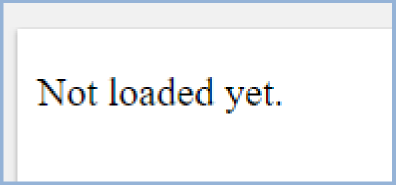
or

```
<script>
    $(
     function() {
      // Handler for
    //.ready() called.
      }
    );
    </script>
```

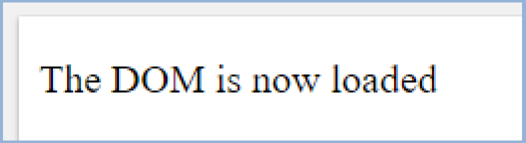
- This is to prevent any jQuery code from running before the document is finished loading (is ready).

Example

```
<html>
<head>
<script>
 document.getElementById("p1").innerHTML = "The DOM is now loaded" ;
</script>
 </head>
<body>
 <p id="p1">Not loaded yet.</p>
</body>
</html>
```


Not loaded yet.

```
<head>
<script src="https://code.jquery.com/jquery-1.10.2.js"></script>
<script>
$(function() {
    $( "p" ).text( "The DOM is now loaded" );
  });
</script>
 </head>
<body>
 <p id="p1">Not loaded yet.</p>
</body>
```


The DOM is now loaded

Example

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(
                        function(){
                         $("h2").slideToggle();}
                      );
});
</script>
</head>

<body>
     <h2> This is a new </h2>
     <p>This is a paragraph.</p>
     <button>Toggle between slide up and slide down for a h2
       element</button>
</body>
```

Click on this link for another example of slideToggle()

Selecting Elements

- $(selector)

$('*')	select all elements	
$('#id')	id of element	`getElementById("id")`
$('p')	tag name	`getElementsByTagName("tag")`
$('.class')	CSS class	
$('p.class')	<p> elements having the CSS class	
$('p:first')	Select the first <p> element, $('p:last') $('p:odd') $('p:even')	
$('p:eq(2)')	gets the 2nd <p> element (1 based)	
$('p')[1]	gets the 2nd <p> element (0 based)	
$('p:nth-child(3))	gets the 3rd <p> element of the parent.	
$('p:nth-child(5n+1)')	gets the 1st element after every 5th one	
$('p a')	<a> elements, descended from <p>	

Selecting Elements

$('p>a')	<a> elements, direct child of a <p>
$('p, a')	<p> and <a> elements
$('li:has(ul)')	<li> elements that have at least one <ul> descendent
$('p:hidden')	only <p> elements that are hidden
$('p:empty')	<p> elements that have no child elements
$('img'[alt])	<img> elements having an *alt* attribute
$('a'[href^=http://])	<a> elements with an href attribute starting with 'http://'
$('a'[href$=.pdf])	<a> elements with an href attribute ending with '.pdf'
$('a'[href*=ntpcug])	<a> elements with an href attriute containing 'ntpcug'

- http://api.jquery.com/category/selectors/
- http://www.w3schools.com/jquery/jquery_selectors.asp
- https://www.w3schools.com/jquery/jquery_ref_selectors.asp

jQuery Event Methods

- Event methods **trigger** or **attach** a function to an event handler for the selected elements.

- In jQuery, most DOM events have an equivalent jQuery method.

- To assign a click event to all paragraphs on a page, you can do this:

```
$("p").click();
```

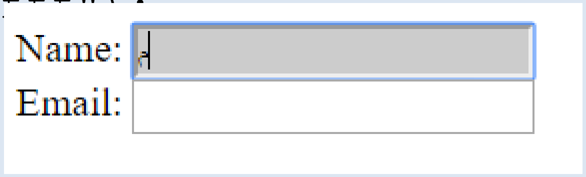
- The next step is to define what should happen when the event fires. You must pass a function to the event:

```
$("p").click(
function(){
  // action goes here!!
}
);
```

https://www.w3schools.com/jquery/jquery_ref_events.asp

jQuery Event Methods
example

```
<!DOCTYPE html>
<html> <head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"
></script>
<script>
$(document).ready(function(){
    $("input").focus(
        function(){
        $(this).css("background-color", "#cccccc");}
        );
    $("input").blur(function(){
        $(this).css("background-color", "#ffffff");
    });
});
</script> </head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">
</body> </html>
```



https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_focus_blur

To attach event handlers?

- on() method- attaches one or more event handlers for the selected elements.
  - Attach a click event to a <p> element:

```
$("p").on("click", function(){
    $(this).hide();
});
```

  - Attach multiple event handlers to a <p> element:

```
$("p").on({
    mouseenter: function(){
        $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
        $(this).css("background-color", "lightblue");
    },
    click: function(){
        $(this).css("background-color", "yellow");
    }
});
```

Common JQuery Effects..

- With jQuery you can fade elements in and out of visibility.
  - jQuery Fading Methods
    - fadeIn(): is used to fade in (show) a hidden element.
    - fadeOut() : is used to fade out (hide) a visible element. It works on opacity
    - fadeToggle() : to toggle between the fadeIn() and fadeOut() methods.
    - fadeTo(): allows fading to a given opacity (value between 0 and 1).

- The jQuery slide methods slide elements up and down.
  - jQuery Sliding Methods
    - slideDown(): Slide-down (show) hidden element(s). It works on element height
    - slideUp(): Slide-up (hide) element(s)
    - slideToggle(): Toggle between slideUp() and slideDown()

https://www.bogotobogo.com/JQuery/JQuery_effects_hide_show_slideUp_slideDown_fadeOut_fadeIn_toggle_slideToggle_fadeToggle.php

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_eff_fadeout_fadein

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_eff_slideup_slidedown

JQuery Animations –
The animate() Method

- The jQuery animate() method is used to create custom animations.

```
$("button").click(function(){
    $("div").animate({left: '250px'});
});
```

- Notice that multiple properties can be animated at the same time:

```
$("button").click(function(){
    $("div").animate({
        left: '250px',
        opacity: '0.5',
        height: '150px',
        width: '150px'
    });
});
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation1_multicss

JQuery Animations –
The animate() Method

- Using Relative Values –
  - This is done by putting += or -= in front of the value:

```
$("button").click(function(){
    $("div").animate({
        left: '250px',
        height: '+=150px',
        width: '+=150px'
    });
});
```

  - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation1_relative

- Using Pre-defined Values
  - You can even specify a property's animation value as "show", "hide", or "toggle":

```
$("button").click(function(){
    $("div").animate({
        height: 'toggle'
    });
});
```

JQuery Animations – The animate() Method

- Uses Queue Functionality
  - if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.

```
$("button").click(function(){
    var div = $("div");
    div.animate({height: '300px', opacity: '0.4'}, "slow");
    div.animate({width: '300px', opacity: '0.8'}, "slow");
    div.animate({height: '100px', opacity: '0.4'}, "slow");
    div.animate({width: '100px', opacity: '0.8'}, "slow");
});
```

  - Another example below first moves the <div> element to the right, and then increases the font size of the text:

```
$("button").click(function(){
    var div = $("div");
    div.animate({left: '100px'}, "slow");
    div.animate({fontSize: '3em'}, "slow");
});
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation2

JQuery DOM Manipulation

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

Get Content - text(), html(), and val()

- Three simple, but useful, jQuery methods for DOM manipulation are:
  - text() - Sets or returns the text content of selected elements
  - html() - Sets or returns the content of selected elements (including HTML markup)
  - val() - Sets or returns the value of form fields

```
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
```

```
$("#btn1").click(function(){
    alert("Text: " + $("#test").text());
$("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
});
```

output

```
Text: This is some bold text in a paragraph.
```

```
HTML: This is some <b>bold</b> text in a paragraph.
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_html_get

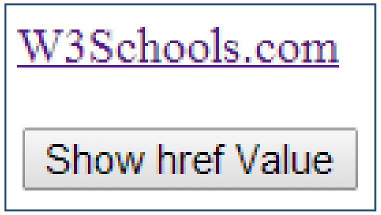
Get Attributes - attr()

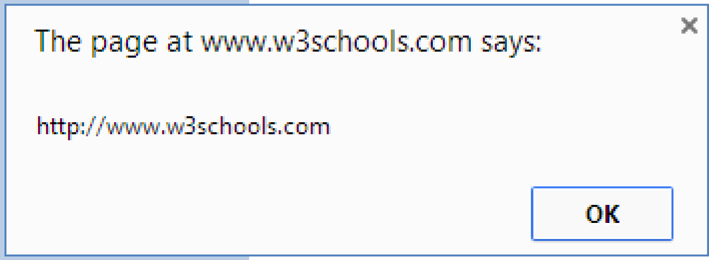
- The jQuery attr() method is used to get attribute values.

```
</head>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>

</head>

<body>
    <p><a id="w3s" href="http://www.w3schools.com">
      W3Schools.com</a></p>
    <button>Show href Value</button>
</body>
```




The page at www.w3schools.com says:

http://www.w3schools.com

http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_attr_get

Set Content - text(), html(), and val()

- We will use the same three methods from the previous page to set content:
  - text() - Sets or returns the text content of selected elements
  - html() - Sets or returns the content of selected elements (including HTML markup)
  - val() - Sets or returns the value of form fields

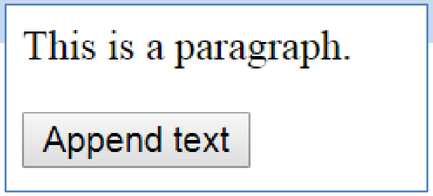
```
$("#btn1").click(function(){
    $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
});
```

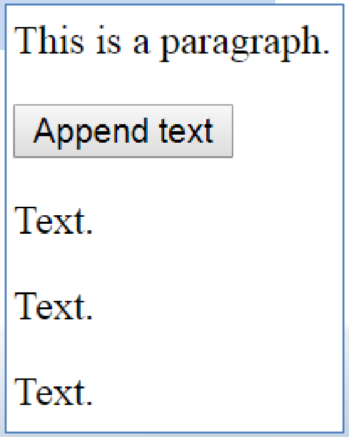
https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_html_set

Add new HTML content

- With jQuery, it is easy to add new elements/content.
  - append() - Inserts content at the end of the selected elements
  - prepend() - Inserts content at the beginning of the selected elements
  - after() - Inserts content after the selected elements
  - before() - Inserts content before the selected elements

```
function appendText() {
    var txt1 = "<p>Text.</p>";               // Create element with HTML
    var txt2 = $("<p></p>").text("Text.");   // Create with jQuery
    var txt3 = document.createElement("p");  // Create with DOM
    txt3.innerHTML = "Text.";
    $("body").append(txt1, txt2, txt3);      // Append the new elements
}
```


This is a paragraph.


This is a paragraph.

Text.

Text.

Text.

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_append2

Add new HTML content..

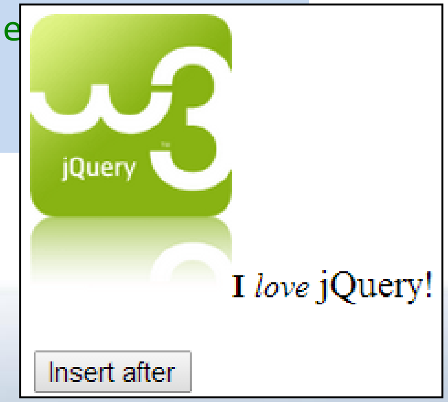
- jQuery after() and before() Methods- to inserts content AFTER or BEFORE the selected HTML elements.

```
$("img").after("Some text after");

$("img").before("Some text before");
```

- Both the after() and before() methods can take an infinite number of new elements as parameters.

```
function afterText() {
    var txt1 = "<b>I </b>";                      // Create element with HTML
    var txt2 = $("<i></i>").text("love ");       // Create with jQuery
    var txt3 = document.createElement("b");      // Create with DOM
    txt3.innerHTML = "jQuery!";
    $("img").after(txt1, txt2, txt3);            // Insert new ele
<img>
}
```



https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_after2

Remove Elements/Content

- remove() - Removes the selected element (and its child elements)

```
$("#div1").remove();
```

- empty() - Removes the child elements from the selected element

```
$("#div1").empty();
```

- The jQuery remove() method **allows you to filter the elements to be removed.**

```
$("p").remove(".test"); //removes all <p> elements with class="test"
```

```
$("p").remove(".test, .demo"); //removes all <p> elements with class="test" and class="demo":
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_remove3

jQuery Manipulating CSS

- jQuery has several methods for CSS manipulation

  - addClass() - **Adds one or more classes to the selected elements**

```
.important {
    font-weight: bold;
    font-size: xx-large; }

.blue {
    color: blue; }
```

```
$("button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");  });
```

```
$("button").click(function(){
    $("#div1").addClass("important blue"); });
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_addclass2

  - removeClass() - **Removes one or more classes from the selected elements**

```
$("button").click(function(){
    $("h1, h2, p").removeClass("blue"); });
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_removeclass

  - toggleClass() - **Toggles between adding/removing classes from selected elements**

```
$("button").click(function(){
    $("h1, h2, p").toggleClass("blue");  });
```

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_toggleclass

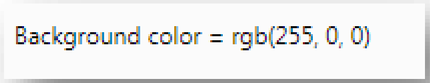
jQuery Manipulating CSS

- css() - sets or returns one or more style properties for the selected elements.

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

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

    ```
    $("p").css("background-color");
    ```

    
    Background color = rgb(255, 0, 0)

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

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

    ```
    $("p").css("background-color", "yellow");
    ```

  - To set multiple CSS properties, use the following syntax:

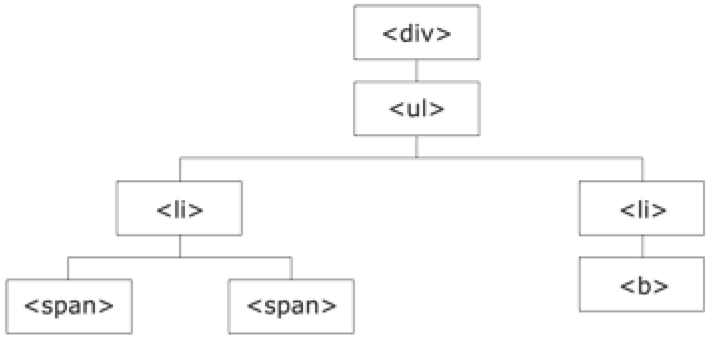
    ```
    $("p").css({"background-color": "yellow", "font-size": "200%"});
    ```

https://www.w3schools.com/jquery/jquery_css.asp

jQuery Traversing

What is Traversing?

- jQuery traversing is used to "find" (or select) HTML elements based on their relation to other elements
  - **Start with one selection and move through that selection until you reach the elements you desire..**

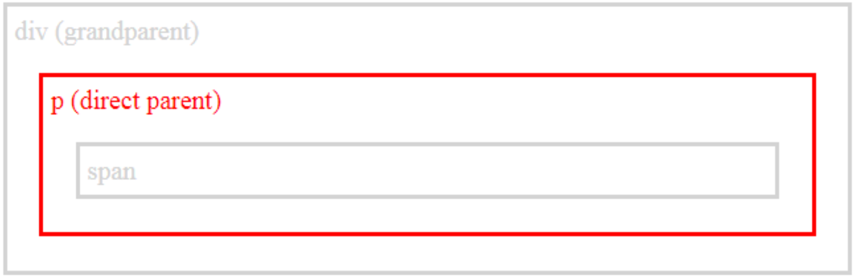


- The <div> element is the **parent** of <ul>, and an **ancestor** of everything inside of it
- The <ul> element is the **parent** of both <li> elements, and a **child** of <div>
- The left <li> element is the **parent** of <span>, **child** of <ul> and a **descendant** of <div>
- The <span> element is a **child** of the left <li> and a **descendant** of <ul> and <div>
- The two <li> elements are **siblings** (they share the same parent)
- The right <li> element is the **parent** of <b>, **child** of <ul> and a **descendant** of <div>
- The <b> element is a **child** of the right <li> and a **descendant** of <ul> and <div>

jQuery Traversing: Ancestors

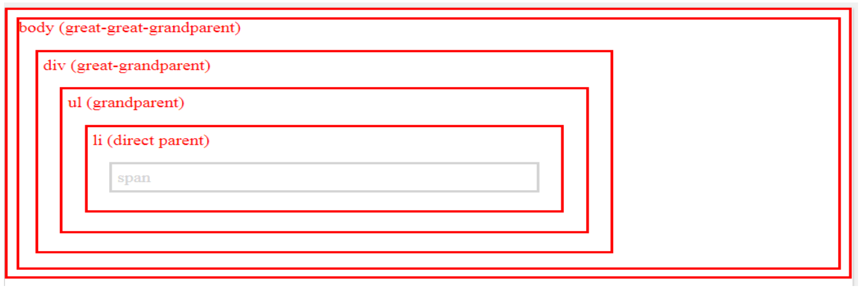
The **parent()** method returns the **direct parent element** of the selected element.

```
$(document).ready(function(){
    $("span").parent();
});
```



The **parents()** method returns **all ancestor elements** of the selected element

```
$(document).ready(function(){
    $("span").parents();
});
```



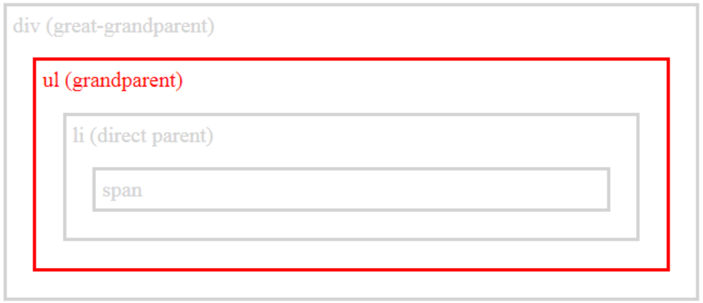
http://www.w3schools.com/jquery/jquery_traversing_ancestors.asp

jQuery Traversing: Ancestors..

You can also use an optional parameter to filter the search for ancestors.

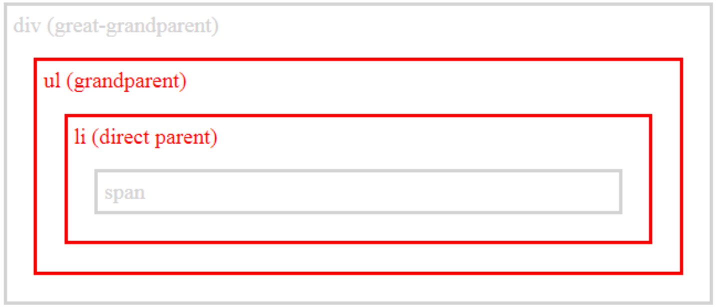
To returns all ancestors of all <span> elements that are <ul> elements

```
$(document).ready(function(){
    $("span").parents("ul");
});
```



The **parentsUntil()** method **returns all ancestor elements between two given arguments**.

```
$(document).ready(function(){
    $("span").parentsUntil("div");
});
```

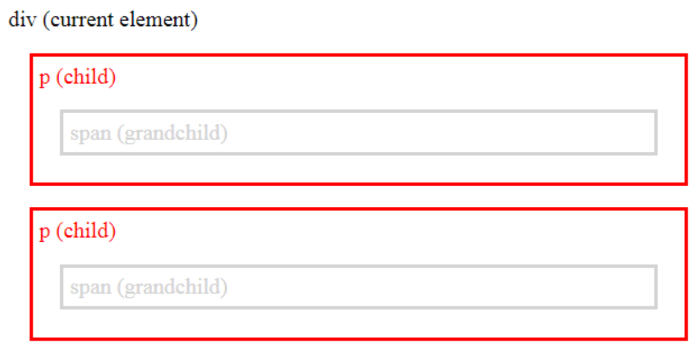


http://www.w3schools.com/jquery/jquery_traversing_ancestors.asp

jQuery Traversing: Descendants

The **children()** method **returns all direct children of the selected element**.

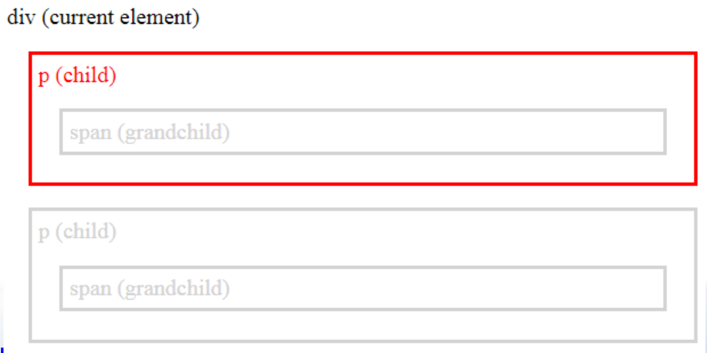
```
$(document).ready(function(){
    $("div").children();
});
```



You can also use an optional parameter to filter the search for children.

e.g. The returns all <p> elements with the class name "first", that are direct children of <div>

```
$(document).ready(function(){
    $("div").children("p.first");
});
```

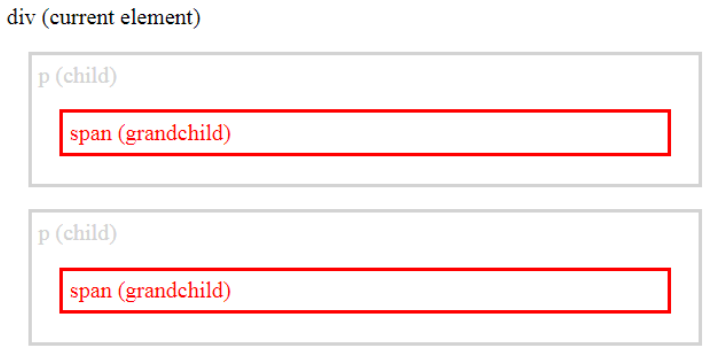


http://www.w3schools.com/jquery/jquery_traversing_descendants.asp

jQuery Traversing: Descendants..

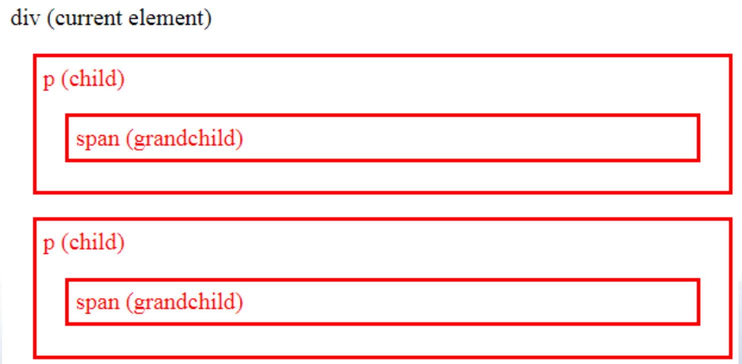
The **find()** method returns **descendant elements of the selected element, all the way down to the last descendant.**

```
$(document).ready(function(){
    $("div").find("span");
});
```



The following example returns all descendants of <div>:

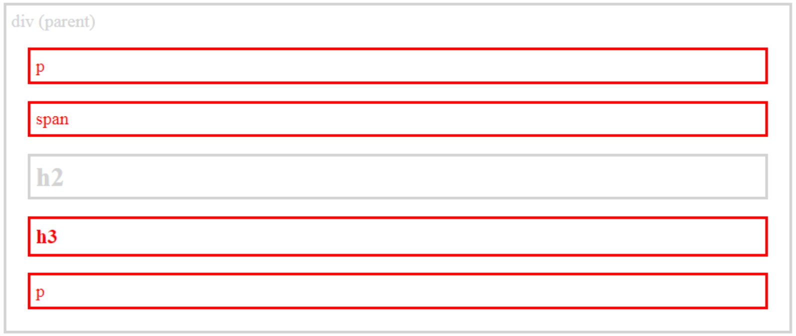
```
$(document).ready(function(){
    $("div").find("*");
});
```



jQuery Traversing - Siblings

The **siblings()** method **returns all sibling elements of the selected element.**

```
$(document).ready(function(){
    $("h2").siblings();
});
```



You can also use an optional parameter to filter the search for siblings.

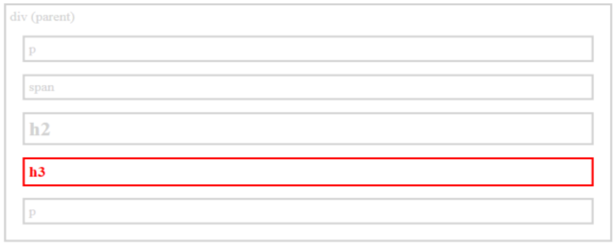
```
$("h2").siblings("p");
```

http://www.w3schools.com/jquery/jquery_traversing_siblings.asp

jQuery Traversing - Siblings

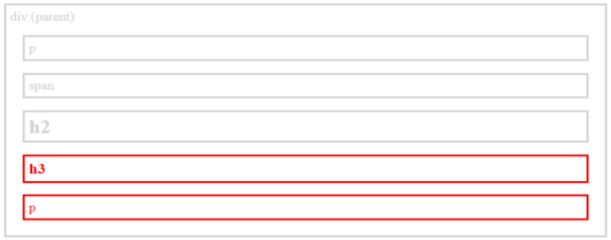
The **next()** method **returns the next sibling element of the selected element**.

```
$(document).ready(function(){
    $("h2").next();
});
```



The **nextAll()** method **returns all next sibling elements of the selected element.**

```
$(document).ready(function(){
    $("h2").nextAll();
});
```

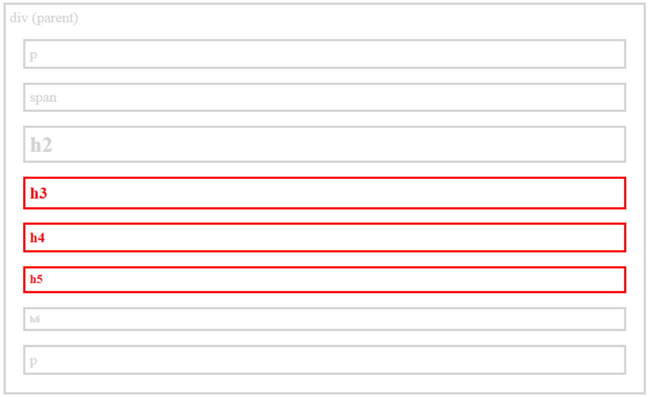


http://www.w3schools.com/jquery/jquery_traversing_siblings.asp

jQuery Traversing - Siblings

The nextUntil() method **returns all next sibling elements between two given arguments**.

```
$(document).ready(function(){
    $("h2").nextUntil("h6");
});
```



The prev(), prevAll() and prevUntil() methods work just like the methods above but with reverse functionality: they return previous sibling elements (traverse backwards along sibling elements in the DOM tree, instead of forward).

http://www.w3schools.com/jquery/jquery_traversing_siblings.asp

jQuery Traversing - Filtering

The first() method **returns the first element of the specified elements.**

```
$(document).ready(function(){
    $("div").first();  });
```

```
$(document).ready(function(){
  $("div").first().css("background-color", "yellow");
});
```

The last() method **returns the last element of the specified elements**.

```
$(document).ready(function(){
    $("div").last();  });
```

The eq() method **returns an element with a specific index number of the selected elements**.

```
$(document).ready(function(){
    $("p").eq(1);  });
```

The filter() method **lets you specify a criteria**.

```
$(document).ready(function(){
    $("p").filter(".intro");  });
```

The not() method **returns all elements that do not match the criteria.**

```
$(document).ready(function(){
    $("p").not(".intro");   });
```

http://www.w3schools.com/jquery/jquery_traversing_filtering.asp

Useful jQuery Functions

.each()	iterate over the set
.size()	number of elements in set
.end()	reverts to the previous set
.get(n)	get just the nth element (0 based)
.eq(n)	get just the nth element (0 based) also .lt(n) & .gt(n)
.slice(n,m)	gets only nth to (m-1)th elements
.not(‘p’)	don’t include ‘p’ elements in set
.add(‘p’)	add <p> elements to set
.remove()	removes all the elements from the page DOM
.empty()	removes the contents of all the elements
.filter(fn/sel)	selects elements where the func returns true or sel
.find(selector)	selects elements meeting the selector criteria
.parent()	returns the parent of each element in set
.children()	returns all the children of each element in set
.next()	gets next element of each element in set
.prev()	gets previous element of each element in set
.siblings()	gets all the siblings of the current element

https://www.tutorialspoint.com/jquery/jquery-utilities.htm

More about Event Binding

Using the jQuery Event Model, we can establish event handlers on DOM elements with the bind() method

$(‘img’).bind(‘click’, function(event){alert(‘Howdy’;});
$(‘img’).bind(‘click’, imgclick(event));
Allows unbinding the function
$(‘img’).unbind(‘click’, imgclick());
$(‘img’).unbind(‘click’);
$(‘img’).one(‘click’, imgclick(event));
Only works once
$(‘img’).click(imgclick);
$(‘img’).toggle(click1, click2);
$(‘img’).hover(mouseover, mouseout);

The bind() method was deprecated in version 3.0. Use the on() method instead.

https://www.tutorialspoint.com/jquery/jquery-events.htm

Widgets

- Accordion
  - Enable to collapse the content, that is broken into logical sections.
- Autocomplete
  - Enable to provides the suggestions while you type into the field.
- Datepicker
  - It is to open an interactive calendar in a small overlay.
- Progressbar
  - It shows the progress information.
- Tabs
  - It is used to swap between content that is broken into logical sections.