1. **Question: What is jQuery?**

**Answer**: jQuery is a feature-rich JavaScript library that makes HTML document traversal and manipulation, event handling, animation, and Ajax much simpler and faster. jQuery has an easy-to-use API that works across many browsers. UI related functions can be written with minimal lines of code using jQuery.

1. **Question: What is the difference between JavaScript and jQuery?**

**Answer**: JavaScript is an interpreted programming language, whereas jQuery is a library with APIs built for JavaScript. jQuery simplifies the use of the JavaScript language.

1. **Question: What are the *effects* methods used in jQuery?**

**Answer**: jQuery enables us to add effects on a web page. jQuery effects can be categorized into fading, sliding, hiding/showing, and animation effects. jQuery provides many methods for effects on a web page

1. **These are the *effects* methods used in jQuery:**

* show() - It displays or shows the selected elements.
* hide() - It hides the matched or selected elements.
* toggle() - It shows or hides the matched elements. In other words, it toggles between the hide() and shows () methods.
* fadeIn() - It shows the matched elements by fading it to opaque. In other words, it fades into the selected elements.
* fadeOut() - It shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.

1. **Question: What are the features of jQuery?**

**Answer:** Some important features of jQuery are:

* Easy DOM manipulation using the Sizzle engine.
* Event handling & AJAX support.
* Built-in effects and animations.
* Lightweight library.
* Cross-browser compatibility.
* Supports CSS3, basic XPath, HTML5.

1. **Question: Explain the Advantages of jQuery?**

**Answer:** The advantages of jQuery are:

* Simple and easy to use.
* Has extensive documentation in the form of blogs, code snippets, tutorials, and other resources
* Simple and clean syntax.
* Open coding standards, intuitive.
* Deals with cross-browser issues without the developer having to worry about them.
* Lightweight with the core library being just 24kb.
* Open-source library.
* Exhaustive set of animations and effects.
* Extensible and fast.
* It can be optimized for search engines for better SEO.

1. **Question: What is jQuery Ajax?**

**Answer**: AJAX is an acronym standing for Asynchronous JavaScript and XML, and this technology helps us load data and exchange data with the server without a browser page refresh. JQuery is a great tool that provides a rich set of AJAX methods to develop next-generation web applications.

1. **Question: What does the ajax() method do?**

**Answer:**This method sends an asynchronous HTTP request to the server.

1. **Question: What does the ajax method load() do?**

**Answer:**The load() method sends an HTTP request to load the HTML or text content from the server and adds them to the DOM element(s).

1. **Question: What are the jQuery Ajax Events?**

**Answer:** The jQuery library also includes events that will be fired based on the *Ajax request state*; these are called Ajax events.

1. **Question: What does the jQuery Ajax event method ajaxComplete() do?**

**Answer**: Whenever an Ajax request completes, jQuery triggers the *ajaxComplete* event. Any handlers that have been registered with the .*ajaxComplete*() method are executed at this time.

1. **Question: What does the jQuery ajax event method ajaxStart() do?**

**Answer:**Whenever an Ajax request is about to be sent, jQuery checks whether there are any other outstanding Ajax requests. If none are in progress, jQuery triggers the *ajaxStart* event. Any and all handlers that have been registered with the .*ajaxStart*() method are executed at this time.

1. **Question: What are Events in jQuery?**

**Answer:**Responding to user actions on a webpage is called events. jQuery provides simple methods for attaching event handlers to selections. When an event occurs, the provided function is executed.

1. **Question: What are the categories in jquery Events?**

**Answer**: The**common DOM events are as follows**

* Form
* Keyboard
* Mouse
* Browser
* Document Loading

1. **Question: What is the use of css() method in JQuery?**

**Answer:** The jQuery CSS() method is used to *get (return*)or *set* style properties or values for selected elements. It facilitates you to get one or more style properties.

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

1. **Question: What is the difference between *find* and *children* methods?**

**Answer:**Both the methods are used to filter the child of the matched elements. The find method is used to find all levels down the DOM tree, but the children method searches only a single level down the DOM tree.

$("#testChildren").click(function () {

$('div').css('background','white');

$('.B1').children('.child').css('background','red');

});

$("#testFind").click(function () {

$('div').css('background','white');

$('.B1').find('.child').css('background','red');

});

1. **Question: What are the selectors in jQuery, and what are the types of Selectors?**

**Answer:** If you would like to work with an element on the web page, first you need to find or select it. Selectors find the HTML elements using jQuery.

There are many types of selectors in the jQuery library. Some basic selectors are:

* **Name**: It is used to select all elements which match with the given element Name.
* **#ID**: It is used to select a single element which matches with the given ID
* .**Class**: It is used to select all elements which match with the given Class.
* **Universal** (\*): It is used to select all elements available in a DOM.
* **Multiple Elements E, F, G**: It is used to select the combined results of all the specified selectors E, F, or G.
* **Attribute Selector**: It is used to select elements based on its attribute value.

1. **Question: What is the difference between the ID selector and class selector in jQuery?**

**Answer**: The ID selector and class selector are the same as they are in CSS. ID selector uses ID while the class selector uses a class to select elements. You use an ID selector to select just one element. If you want to select a group of elements, you can use the same CSS class to use the class selector.

1. **Question: What are the Advantages of jQuery Ajax methods?**

**Answer:** The advantages of using jQuery Ajax methods are

* Cross-browser support
* Simple methods to use
* Ability to send GET and POST requests
* Ability to Load JSON, XML, HTML or Scripts

1. **Question: What is the difference between onload() and document.ready() methods?**

**Answer:**Body.Onload() event will be called only after the DOM and associated resources like images get loaded, but jQuery's document.the ready() event will be called once the DOM is loaded, and it does not wait for the resources such as images to be loaded.

1. **Question: What is jQuery connect?**

**Answer**: A ‘ jQuery connect’ is a plug-in used to connect or bind a function with another function. Connect is used to execute a function whenever a function from another object or plug-in is executed.

1. **Question: Is jQuery required for bootstrap?**

**Answer:**[Bootstrap](https://hackr.io/tutorials/learn-bootstrap?ref=blog-post)uses jQuery for JavaScript plugins (like models, tooltips, etc.). However, if you just use the CSS part of Bootstrap, you don't need jQuery.

1. **Question: What is jQuery Mobile?**

**Answer: j**Query Mobile is an HTML5-based user interface system designed to make responsive websites and apps accessible on all smartphones, tablets, and desktop devices.

1. **Question: What is the difference between jquery.min.js and jquery.js?**

**Answer:**jquery.min.js is a compressed version of jquery.js(whitespaces and comments are removed, shorter variable names are used, and so on) to preserve bandwidth. In terms of functionality, they are absolutely the same. It is recommended to use this compressed version in the production environment. The efficiency of a web page increases when the minimized version of jQuery is used.

1. **Question: Is it possible that jQuery HTML works for both HTML and XML documents?**

**Answer:**No, jQuery HTML only works for HTML documents. It doesn't work for XML documents.

1. **Question: What is jQuery UI?**

**Answer:**[jQuery UI](http://jqueryui.com/) is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. jQuery UI works well for highly interactive web applications with many controls or for a simple page with a date picker control.

1. **Question: What is a Data Table plug-in for jQuery?**

**Answer**: DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, built upon the foundations of progressive enhancement, which adds advanced features to any HTML table.

**Question: Explain the various Ajax functions available in jQuery?**

**Answer:**There are many methods like:

* .ajaxStart() - register the handler to be called when the first Ajax request begins.
* .ajaxStop() - register the handler to be called when all requests are complete.
* .ajaxSuccess() - register the handler to be called when an Ajax request is successfully completed.

Check all the methods on the[official jQuery documentation page](https://api.jquery.com/category/ajax/), which explains each method with an example.

**Question: What is the difference between width() vs css(‘width’) in jQuery?**

**Answer:**CSS(‘width’) returns the width value in pixels, whereas width() returns the integer (without the unit values). For example:

**div**{  
width: 20cm;  
}

If you print the values:

$(**this**).width();  
$(**this**).css(‘width’);

you will get the values like 756 and 756px, respectively. Note that though we specified width in cm, it is converted to pixel (px) for output purposes.

**Question: What is the difference between bind() vs live() vs delegate() methods in jQuery?**

**Answer:**

**bind():** this method registers the event handler directly to the required DOM element. E.g.:

$(“#members a”).bind(“click”, function(f){….});

This means any matching anchors will have this event handler attached!

**live():** this method attaches the event handler to the root of the document. This means one handler can be used for all events that propagated to the root. The handler is thus attached only once.

**delegate():** in this method, you can choose where to attach the handler. This is the most efficient and robust method for delegation.

E.g.:

$(“#members”).delegate(“ul li a”, “click”, function(f){….});

**Question: Describe the use of the param() method in jQuery?**

**Answer:**The param() method outputs a serialized representation of an object or array.

For example:

student = **new** Object();  
student.name = “Mary”;  
student.marks = 67;  
$("div").text($.param(student);

When an event occurs that calls this code, the method will give the following output:

name=Mary&marks=67

**Question: Explain the difference between $(this) and this in jQuery?**

**Answer:**$() is the jQuery constructor function, whereas this is a reference to the DOM element. To use jQuery methods, we use $(this).

**Question: Explain the difference between jquery.size() and jquery.length?**

**Answer:**Both return the number of elements. But length is faster. As of jQuery 1.8, size() has been deprecated.

Question: What are the four parameters used for the jQuery Ajax method?

**Answer:**The four parameters are:

* **URL:** URL for sending the request
* **Type:** GET/POST request
* **Success:**the callback function when the request is successful
* **DataType:**return data type – HTML, XML, text etc.

**Question: What are all the ways to include jQuery on a page?**

**Answer:**

1. You can use <script> to add the library in the HTML <head> or <body> tag: <script src='jquery-3.2.1.min.js'></script>
2. Write the code within the HTML document inside the <script> tag, here we have used cdn link.

<**script** src='http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.js'></**script**>  
<**script** type = “text/javascript”>  
$(document)……… <jQuery code>  
</**script**>

1. Include the .js file, which has the jQuery code into the HTML document.

<script src='script.js' type="text/javascript"></script>

**Question: What is the use of the css() method in JQuery?**

**Answer: css**() sets style properties for all the selected elements. It also returns the first matched element of the specified CSS property.

<**p**>Welcome to styling</**p**>  
<**p**>I will be styled just as the previous paragraph</**p**>  
<**button**>click me to change the style</**button**>  
<**script**>  
$(document).ready(**function**(){  
$("button").click(**function**(){  
$("p").css("color", "blue");  
});  
});  
</**script**>

**Question: What is jQuery Datepicker in jQuery?**

**Answer:**It is a plugin/widget that adds datepicker functionality in HTML pages. It is highly configurable and can be customized for date format, language, restricting date selection, etc. Refer to this[jQuery documentation](https://api.jqueryui.com/datepicker/) for datepicker options.

**Question: Define slideToggle() effect?**

**Answer:**It is used to toggle between sliding up and sliding down for the selected elements.

<**h2**>This is a paragraph.</**h2**>  
<**button**>show me toggle</**button**>  
<**script**>  
$(document).ready(**function**(){  
$("button").click(**function**(){  
$("h2").slideToggle();  
});  
});  
</**script**>

**Question: How can you use an array with jQuery?**

**Answer:**To create an array use $.makeArray(<object>)

**var** myObj = [“John”, “Jake”, “Jack”, “King”];  
**var** myArr = $.makeArray(myobj);

You can search for specific element in array using $.inArray()

$.inArray(“Jack”, myArr);

To merge two arrays, use $.merge() method

**var** arr1 = [“John”, “Jake”, “Jack”, “King”];  
**var** arr2 = [“Mary”, “Katy”, “Jill”, “Queen”];  
**var** mergeArr = $.merge(arr1, arr2);

**Question: What are jQuery plugins?**

**Answer:**Plugins are simply methods that enable developers to extend jQuery’s prototype objects. Plugins are written in a standard javascript file. jQuery provides a lot of plugins that you can download from their [repository link](https://jquery.com/plugins).[You can include plugins in the code using](https://learn.jquery.com/plugins/)<script src = "jquery.plugin.js" type = "text/javascript"></script>

**Question: Difference between Map and Grep function in jQuery?**

**Answer:**Map function translates a set of elements into another set of values in a jQuery array that may or may not have the elements. The map is called:

$(“<element>”).map(<**function** to execute **for** elements **in** the object>)

Grep, on the other hand, finds an element in an array.

jQuery.grep(myArr, **function**(){}

var items = $.map(items, function(item) {

if (item == 'A')

return 'A'+'B';

return item;

});

output will be ['AB', 'B', 'C', 'AB']

$.grep is used for filtering

var items = $.grep(items, function(item) {

return item != 'A';

});

items is now ['B','C']

[**https://www.c-sharpcorner.com/UploadFile/97fc7a/diiffernece-between-map-and-grep-function-in-jquery/**](https://www.c-sharpcorner.com/UploadFile/97fc7a/diiffernece-between-map-and-grep-function-in-jquery/)

So, we can see here the difference is **grep() method** filters an array and returns the filtered array, while **map() method** applies a function to each item in the array, thus returning a modified array.

**Question: What is the method chaining in jQuery, and what are the advantages?**

**Answer:**With chaining, multiple jQuery commands on a particular element can be executed in one go. It helps in implementing various actions on an element at once rather than executing them one after the other.

$("#h2").css("color","blue").animate((left: '100px'}).slideDown(1000);

**Question: Difference between jQuery.get() and jQuery.ajax()?**

**Answer:**In the get() method, we have to pass individual arguments, whereas the ajax() methods get all those arguments as an object.

**jQuery**.ajax({  
url: 'mydoc.txt',  
dataType: 'text',  
type: “GET”,  
success: function(data) {  
console.log(data);  
}  
});

get() method accepts arguments. The three main arguments passed are explained below:

jQuery.get('mydoc.txt',**function**(data){  
console.log(data)  
},'text');

In this, the first argument is the url, the second is the callback function, and the third (‘text’) is the return type.

**Question: Difference between prop and attr?**

**Answer:**Both attr() and prop() can be used to set or get an element’s value, however attr() returns the original (default) value whereas prop() returns the most recent (current) value. For example, if a text input had an initial value of ‘Male,’ and later it was changed by the user to ‘female,’ attr() will return the value as ‘Male’ whereas prop() will return the value as ‘female.’

**Question: What is the use of the toggle() method in JQuery?**

**Answer:**If there is a click event, toggle() attaches functions to toggle. So, on first click first action occurs, on second, second action and so on.

<**button**>Change my color on each click</**button**>  
<**script**>  
$(document).ready(**function**(){  
$("button").toggle(  
**function**(){$("button").css({"color": "blue"});},  
**function**(){$("button").css({"color": "yellow"});},  
**function**(){$("button").css({"color": "red"});  
});  
});  
</**script**>

**Question: What is CDN?**

**Answer:**CDN is the short form for Cloud Delivery Network. It is a system of servers on the network (distributed) that provide specific web content like graphics, images, text, etc. to the user based on their geographic locations, the origin of the page, and the content delivery server. CDN provides high availability and performance.

**Question: How to add and remove CSS classes to an element using jQuery?**

**Answer:**You can use the addClass() and removeClass() methods to do the same.

$("h1").addClass("myclass");  
$("h1").removeClass("myclass");

**Question: Can you write a jQuery code to select all links inside the paragraph?**

**Answer:**

$('a:visible').css('text-transform', 'uppercase');

**Question: What is the purpose of the fade toggle() method in JQuery?**

**Answer:**It is used to toggle between the fadeIn() and fadeOut() functions. Here is an example to show the same:

<div id="div1" style="color:orange;”>My text</div>  
<button>fade in/out</button><br><br>  
<script>  
$("button").click(function(){  
$("#div1").fadeToggle();  
}  
</script>

**var and let** are both used for variable declaration in **javascript** but the difference between them is that **var** is function scoped and **let** is block scoped.

There are 3 ways to declare a JavaScript variable:

* Using var
* Using let
* Using const

The let keyword was introduced in [ES6 (2015)](https://www.w3schools.com/js/js_es6.asp).

Variables defined with let cannot be Redeclared.

Variables defined with let must be Declared before use.

Variables defined with let have Block Scope.

### Function-scope

function myFn() {

var foo = 'peekaboo!';

console.log(foo); // 'peekaboo!'

}

console.log(foo); // ReferenceError: foo is not defined

### Block-scope

if (true) {

var foo = 'peekaboo!';

let bar = 'i see u';

const baz = 'baby blue!';

console.log(foo); // 'peekaboo!';

console.log(bar); // 'i see u';

console.log(baz); // 'baby blue!';

}

console.log(foo); // 'peekaboo!';

console.log(bar); // ReferenceError: bar is not defined

console.log(baz); // ReferenceError: baz is not defined

How many method in string

Length

let str = "Please locate where 'locate' occurs!";  
str.indexOf("locate")    // Returns 7

let str = "Please locate where 'locate' occurs!";  
str.lastIndexOf("locate")    // Returns 21

let str = "Please locate where 'locate' occurs!";  
str.search("locate")     // Returns 7

let str = "Apple, Banana, Kiwi";  
str.slice(7, 13)     // Returns Banana

let str = "Apple, Banana, Kiwi";  
substring(7, 13)    // Returns Banana

let text = "Please visit Microsoft!";  
let newText = text.replace("Microsoft", "W3Schools");

toLowerCase

toUpperCase

Concat

Number Method

let x = 123;  
x.toString();            // returns 123 from variable x  
(123).toString();        // returns 123 from literal 123  
(100 + 23).toString();   // returns 123 from expression 100 + 23

let x = 9.656;  
x.toFixed(0);           // returns 10  
x.toFixed(2);           // returns 9.66  
x.toFixed(4);           // returns 9.6560  
x.toFixed(6);           // returns 9.656000

parseInt("-10");        // returns -10  
parseInt("-10.33");     // returns -10  
parseInt("10");         // returns 10  
parseInt("10.33");      // returns 10  
parseInt("10 20 30");   // returns 10  
parseInt("10 years");   // returns 10  
parseInt("years 10");   // returns NaN

parseFloat("10");        // returns 10  
parseFloat("10.33");     // returns 10.33  
parseFloat("10 20 30");  // returns 10  
parseFloat("10 years");  // returns 10  
parseFloat("years 10");  // returns NaN

Math.round(4.9);    // returns 5  
Math.round(4.7);    // returns 5  
Math.round(4.4);    // returns 4  
Math.round(4.2);    // returns 4  
Math.round(-4.2);    // returns -4

Math.ceil(4.9);     // returns 5  
Math.ceil(4.7);     // returns 5  
Math.ceil(4.4);     // returns 5  
Math.ceil(4.2);     // returns 5  
Math.ceil(-4.2);     // returns -4

Math.floor(4.9);    // returns 4  
Math.floor(4.7);    // returns 4  
Math.floor(4.4);    // returns 4  
Math.floor(4.2);    // returns 4  
Math.floor(-4.2);    // returns -5

Arrow function

var hello;

hello = () => "Hello World!";

**Wildcard Selectors (\*, ^ and $) in CSS for classes**

**Syntax:**

[attribute\*="value"] {

// CSS property

}

1. $("input[id^='code']");//id attribute all input tags starting with code
2. $("input[id$='code']");//id attribute all input tags ending with code
3. $("input[id\*='code']");//id attribute contains all the input tags of code
4. $("input[name^='code']");//name attribute all input tags starting with code
5. $("input[name$='code']");//name attribute all input tags ending with code
6. $("input[name\*='code']");//name attribute contains all the input tags of code

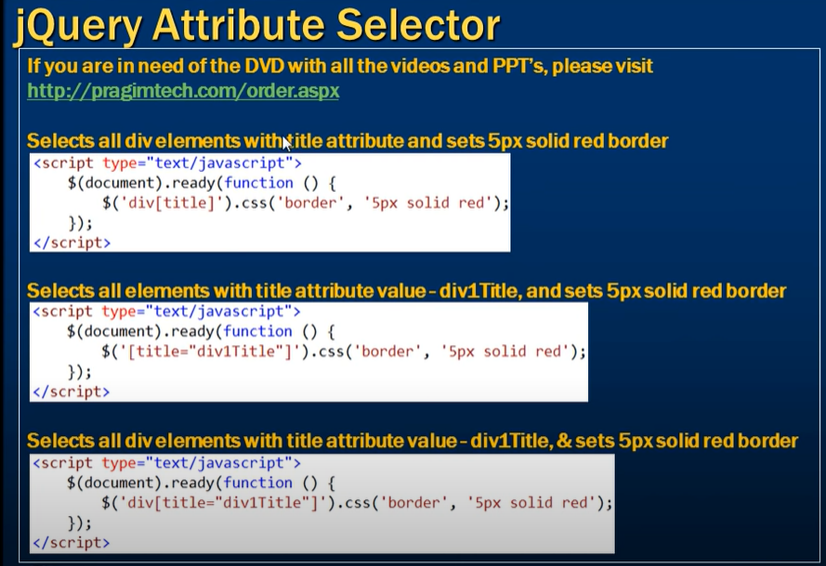
$.ajax({  
            url: '/home/update',  
            type: 'POST',  
            dataType: 'json',  
            data: comicBookJSON,  
            contentType: 'application/json; charset=utf-8',  
        });

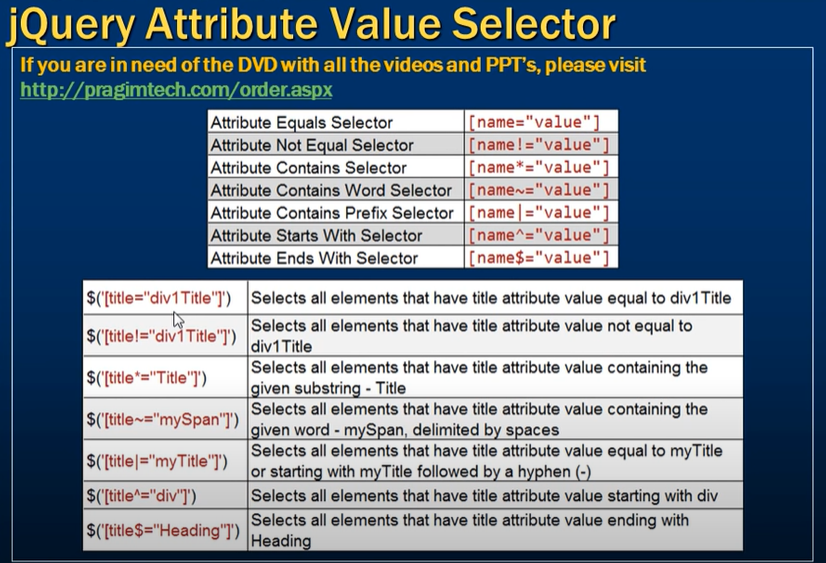
Document.Ready

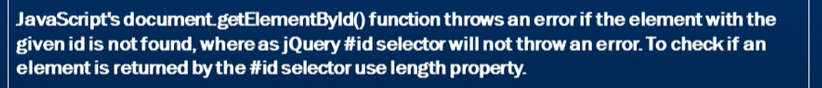


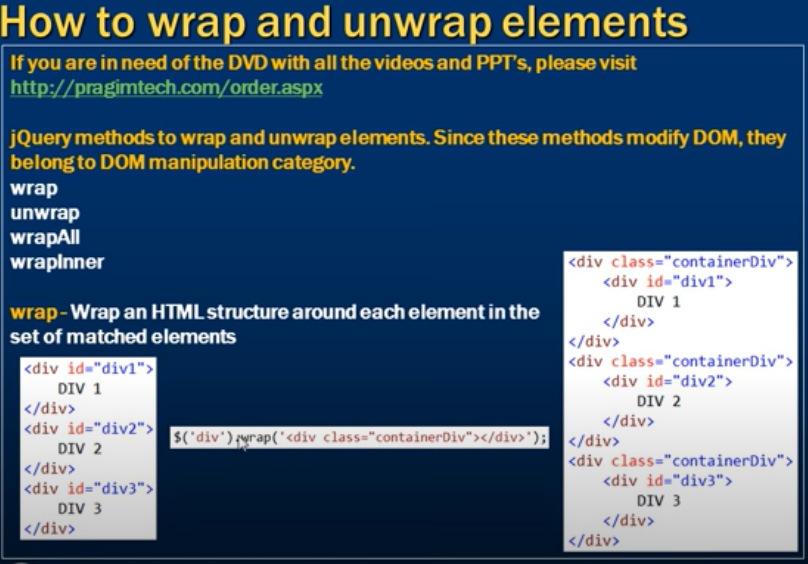
# **jQuery attribute selector**

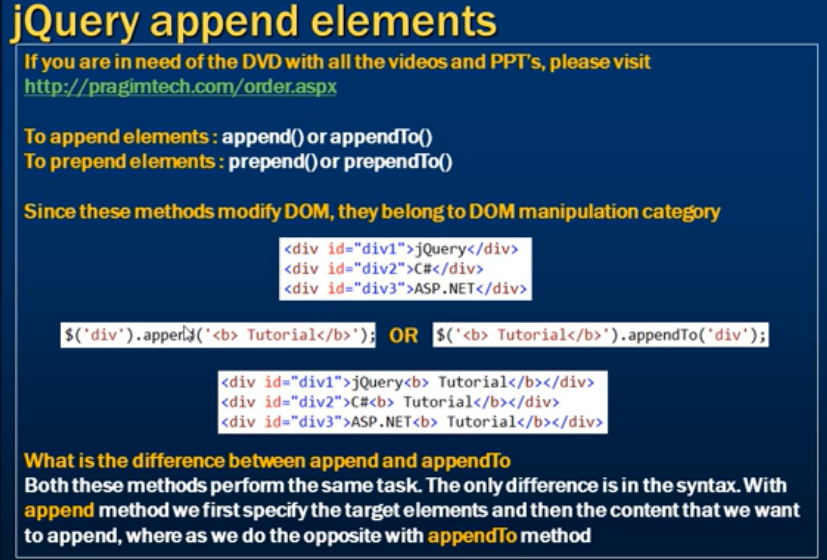
$('[title]').css('border', '1px solid black');

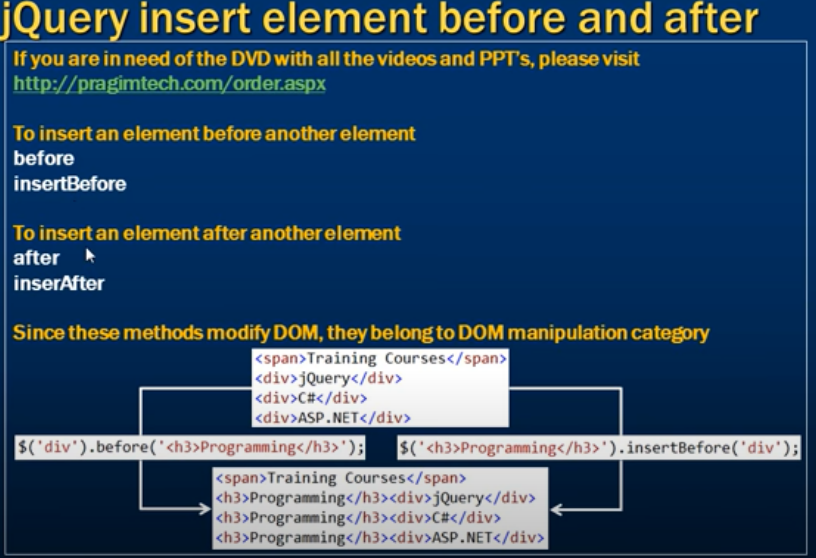




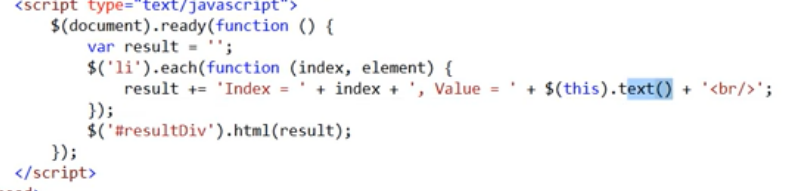


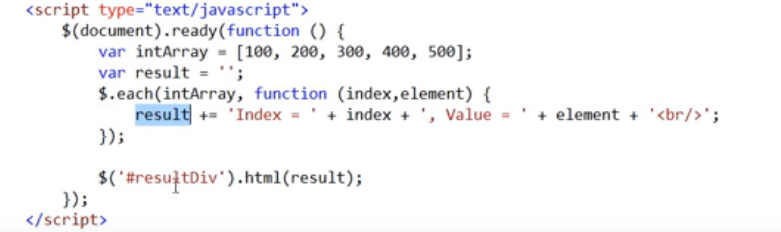


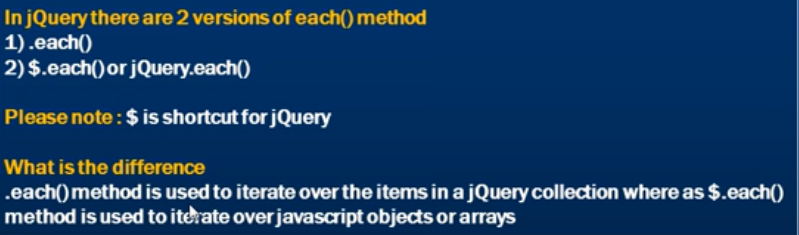




$each and each in jquery

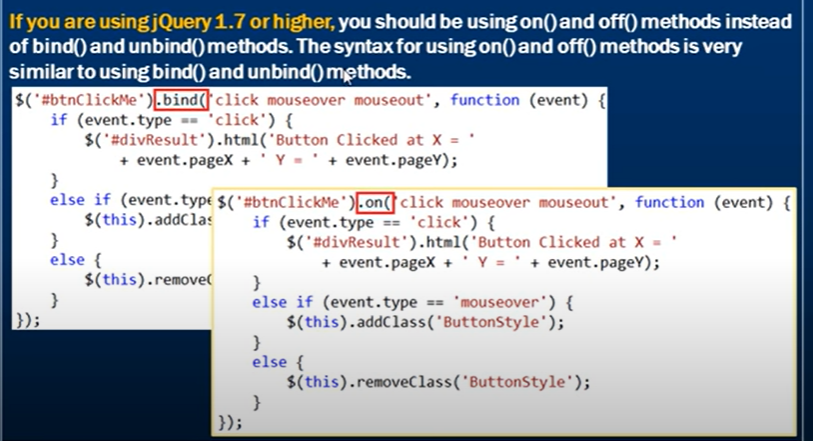


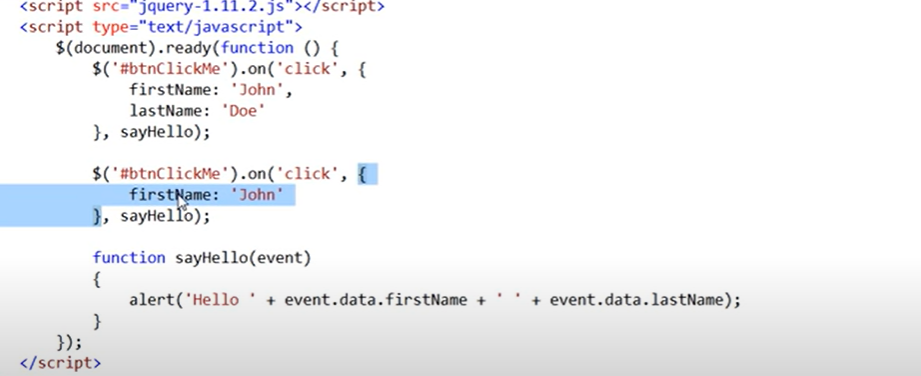


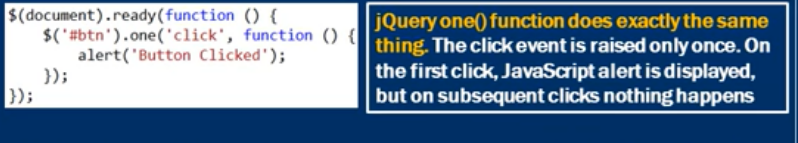










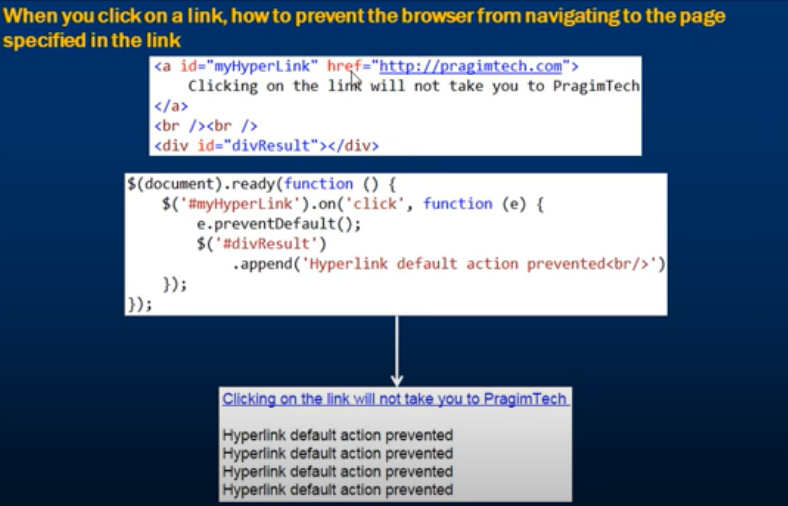


//$('li').css('color', 'blue');

//$('li').slideUp(1000);

//$('li').slideDown(1000);

//$('li').attr('title', 'MY TITLE');



## **Traversing Sideways in The DOM Tree**

There are many useful jQuery methods for traversing sideways in the DOM tree:

* siblings()
* next()
* nextAll()
* nextUntil()
* prev()
* prevAll()
* prevUntil()

<script>

$(document).ready(function(){

$("h2").siblings().css({"color": "red", "border": "2px solid red"});

});

</script>

</head>

<body class="siblings">

<div>div (parent)

<p>p</p>

<span>span</span>

<h2>h2</h2>

<h3>h3</h3>

<p>p</p>

</div>

## **The first(), last(), eq(), filter() and not() Methods**

The most basic filtering methods are first(), last() and eq(), which allow you to select a specific element based on its position in a group of elements.

Other filtering methods, like filter() and not() allow you to select elements that match, or do not match, a certain criteria.

<script>

$(document).ready(function(){

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

});

</script>

</head>

<body>

<h1>Welcome to My Homepage</h1>

<p>My name is Donald.</p>

<p class="intro">I live in Duckburg.</p>

<p class="intro">I love Duckburg.</p>

<p>My best friend is Mickey.</p>

</body>