

Agenda

1. What is jQuery?
2. JQuery Selectors
3. HTML/DOM manipulation
4. CSS manipulation
5. HTML event methods

What is JQuery?

Till now we deeply studied the 3 main parts of web HTML,CSS and Javascript.We structure our websites with HTML, style them with CSS, and add interactive functionality with JavaScript.

jQuery is the “Write Less, Do More” JavaScript library.It is not a programming language, but rather a tool used to make writing common JavaScript tasks more concise. jQuery has the added benefit of being cross-browser compatible, meaning you can be certain the output of your code will render as intended in any modern browser.

By comparing a simple “Hello, World!” program in both JavaScript and jQuery, we can see the difference of how they’re both written.

```
document.getElementById("demo").innerHTML = "Hello, World!";
```



```
$("#demo").html("Hello, World!");
```



This short example demonstrates how jQuery can achieve the same end result as plain JavaScript in a succinct manner.

Setting Up JQuery

jQuery is a JavaScript file that you will link to in your HTML. There are two ways to include jQuery in a project:

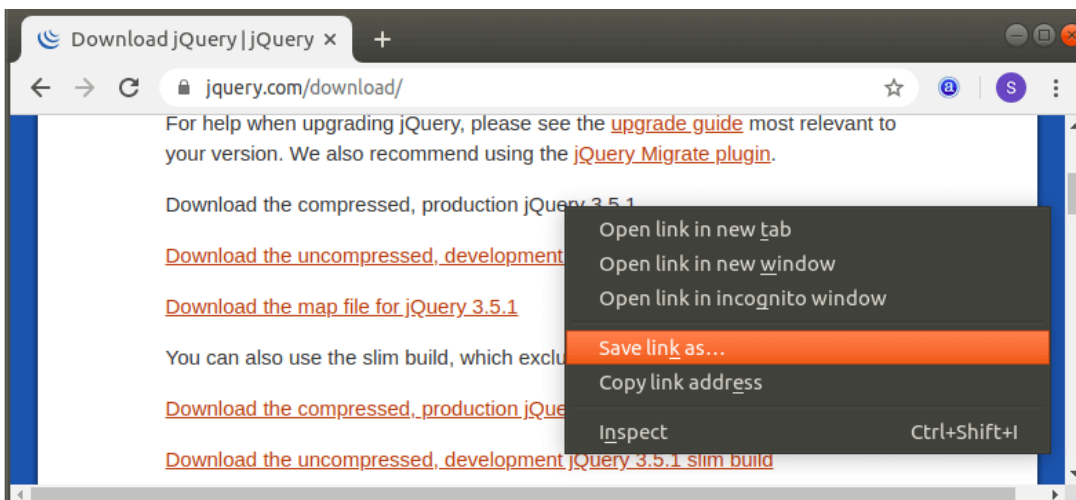
- Download a local copy.
- Link to a file via Content Delivery Network (CDN).

NOTE: A Content Delivery Network (CDN) is a system of multiple servers that deliver web content to a user based on geographical location. When you link to a hosted jQuery file via CDN, it will potentially arrive faster and more efficiently to the user than if you hosted it on your own server.

Download and Include jQuery File

If we want to add the jQuery to Html page by downloading the jQuery file, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

Step 1: Firstly, we have to download the jquery js file from the following official site of jQuery. <https://jquery.com/download/>



Step 2: When we have downloaded the file, then we have to open that Html file in which we want to add the jquery.

```
<!Doctype Html>
<Html>
<Head>
<Title>
Add the jQuery file into Html by downloading and Including file
</Title>
</Head>
<Body>
</Body>
</Html>
```



Step 3: After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the

```
<!Doctype Html>
<Html>
<Head>
<script type="text/javascript" src="jquery-3.5.1.min.js">
</script>
<Title>
Add the jQuery file into Html by downloading and Including file
</Title>
</Head>
<Body>
Hello User!
</Body>
</Html>
```



Step 4: And, at last, save the Html file and the jQuery file is successfully added into our Html page

Include the jQuery by CDN

If we want to add the jQuery to Html page by the help of CDN, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

Step 1: Firstly, we have to open that Html file in which we want to add the jQuery using CDN.

```
<!Doctype Html>
<Html>
<Head>
<Title>
Add the jQuery file into Html by including CDN
</Title>
</Head>
<Body>
Hello User!
</Body>
</Html>
```



Step 2: After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the
And then, we have to type the following path in the src attribute.

```
http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js
```



Step 3: And, at last, save the Html file and the jQuery file is successfully added into our Html page

```
<!Doctype Html>
<Html>
<Head>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<Title>
Add the jQuery file into Html by including CDN
</Title>
```



```
</Head>
<Body>
Hello User!
</Body>
</Html>
```

Your JavaScript file (scripts.js) must be included *below* the jQuery library in the document or it will not work.

JQuery Selectors

Selectors are how we tell jQuery which elements we want to work on. Most jQuery selectors are the same as what you're familiar with in CSS, with a few jQuery-specific additions.

To access a selector, use the jQuery symbol \$, followed by parentheses ().

```
$("#selector")
```



Below is a brief overview of some of the most commonly used selectors.

- \$("*") - Wildcard: selects every element on the page.
- \$(this) - Current: selects the current element being operated on within a function.
- \$("p") - Tag: selects every instance of the tag.
- \$(".example") - Class: selects *every* element that has the example class applied to it.
- \$("#example") - Id: selects a *single instance* of the unique example id.
- \$("[type='text']") - Attribute: selects any element with text applied to the type attribute.
- \$("p:first-of-type") - Pseudo Element: selects the first .

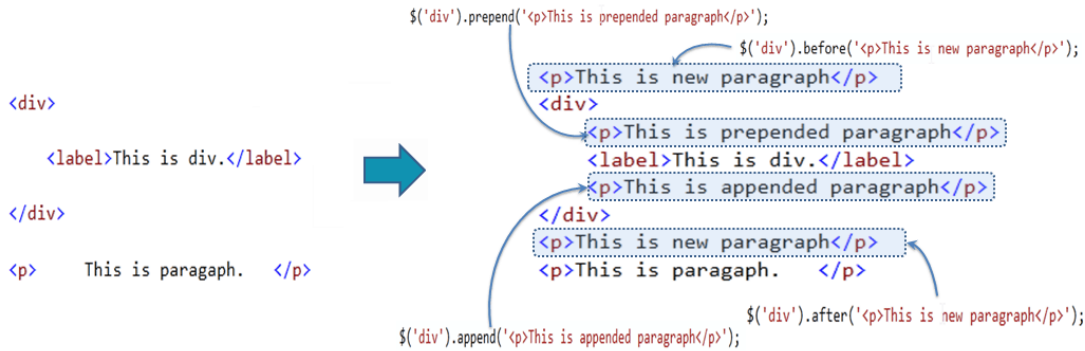
HTML/DOM Manipulation

jQuery provides various methods to add, edit or delete DOM element(s) in the HTML page.

The following table lists some important methods to add/remove new DOM elements.

Method	Description
append()	Inserts content to the end of element(s) which is specified by a selector.
before()	Inserts content (new or existing DOM elements) before an element(s) which is specified by a selector.
after()	Inserts content (new or existing DOM elements) after an element(s) which is specified by a selector.
prepend()	Insert content at the beginning of an element(s) specified by a selector.
remove()	Removes element(s) from DOM which is specified by selector.
replaceAll()	Replace target element(s) with specified element.
wrap()	Wrap an HTML structure around each element which is specified by selector.

The following figure shows how the DOM manipulation methods add new elements.



Let's have a quick overview of important DOM manipulation methods.
 Example:

```

<!DOCTYPE html>
<html>
  <head>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"
      integrity="sha256-/xUj+30JU5yEx1q6GSYGS5Hk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
    </script>
    <script>
      $(document).ready(function () {
        //After
        $('#div1').after('<div style="background-color:yellow"> After div </div>');

        //Before
        $('#div1').before('<div style="background-color:yellow"> Before div </div>');

        //Append
        $('#div1').append(' Appended text!');

        //Remove
        $('#div2').remove();

        //Replace All
        $('<span>This is a span replacing p</span>').replaceAll('p');

        //Wrap
        $('span').wrap('<p></p>');
      });
    </script>
    <style>
      div {
        border: 1px solid;
        background-color: red;
        margin: 2px 0 2px 0;
      }
    </style>
  </head>
  <body>
    <h1>Demo: jQuery methods demo </h1>
    <div id="div1">div 1</div>
    <div id="div2">div 2</div>
    <p>Text after this is</p>
  </body>
</html>

```

OUTPUT:

Demo: jQuery methods demo

Before div
div 1 Appended text!
After div

This is a span replacing p

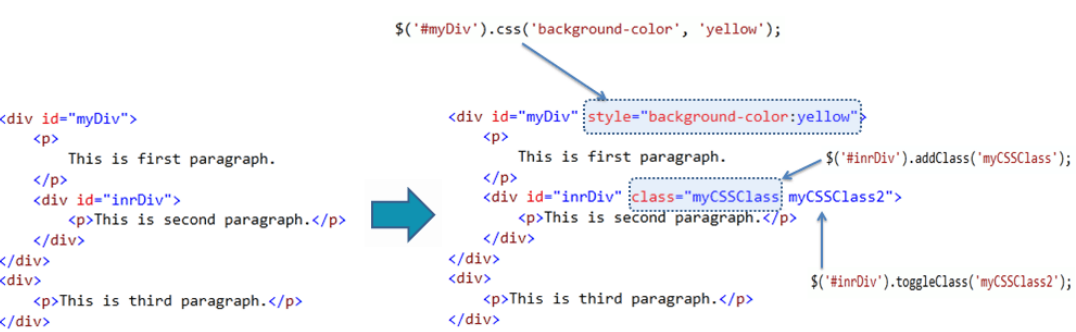
CSS Manipulation

The jQuery library includes various methods to manipulate style properties and CSS class of DOM element(s).

The following table lists jQuery methods for styling and css manipulation.

jQuery Methods	Description
css()	Get or set style properties to the specified element(s).
addClass()	Add one or more class to the specified element(s).
hasClass()	Determine whether any of the specified elements are assigned the given CSS class.
removeClass()	Remove a single class, multiple classes, or all classes from the specified element(s).
toggleClass()	Toggles between adding/removing classes to the specified elements

The following figure shows how jQuery methods changes style and css class of the DOM elements.



Let's have an overview of important jQuery methods for style and css manipulation.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"
      integrity="sha256-/xUj+30JU5yEx1q6GSYGSk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
    <script>
      $(document).ready(function () {
```



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

//Add Class
$('p:nth-of-type(2)').addClass('yellowDiv');

$('p:nth-of-type(3)').addClass('impPrg');

//Toggle Class
$('p:nth-of-type(2)').toggleClass('yellowDiv');

});
</script>
<style>
    .yellowDiv {
        margin: 10px 0 0 0;
        height: 50px;
    }

    .impPrg {
        font-size: 16px;
        font-weight: bold;
    }
</style>
</head>

<body>
    <h1>Demo: jQuery methods demo</h1>

    <p>This is first paragraph.</p>
    <p>This is second paragraph.</p>
    <p>This is third paragraph.</p>
</body>

</html>
```

OUTPUT:

Demo: jQuery methods demo

This is first paragraph.

This is second paragraph.

This is third paragraph.

Category	jQuery Method	DOM Event
Form events	blur	onblur
	change	onchange
	focus	onfocus
	focusin	onfocusin
	select	onselect
	submit	onsubmit
Keyboard events	keydown	onkeydown
	keypress	onkeypress
	keyup	onkeyup
	focusout	
Mouse events	click	onclick
	dblclick	ondblclick
	focusout	
	hover	
	mousedown	onmousedown
	mouseenter	onmouseenter
	mouseleave	onmouseleave
	mousemove	onmousemove
	mouseout	onmouseout
	mouseover	onmouseover
	mouseup	onmouseup
	Toggle	
Browser events	Resize	onresize
	Scroll	onscroll
Document loading	Ready	

Let us see some important methods

jQuery Mouse Events

Example:

```
<!DOCTYPE html>
<html>

  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
    </script>
    <script>
      $(document).ready(function () {
```



```

//Click
$('#myDiv1').click(function (eventObj) {
    alert('CLICKED!!');
});

//Mouse Enter
$('#myDiv1').mouseenter(function (data) {
    $(this).css('background-color', 'green');
    $(this).append('<span>Mouse Entered</span>');
});

//Mouse Leave
$('#myDiv1').mouseleave(function (data) {
    $(this).css('background-color', 'red');
    $('span').remove();
});

//Hover
$('#myDiv2').hover(function (data) {
    $(this).css('background-color', 'green');
    $(this).append('<span>Hovered</span>');
}, function (data) {
    $(this).css('background-color', 'red');
    $('span').remove();
});
});
</script>
</head>

<body>
    <h1>Demo: jQuery click,mouseenter() , mouseleave() & hover methods </h1>

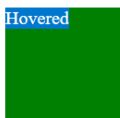
    <div id="myDiv1" style="width:100px;height:100px;background-color:red;">
        <h5>CLICK ME</h5>
    </div>
    <div id="myDiv2" style="width:100px;height:100px;background-color:red;margin-top:100px"></div>
</body>

</html>

```

OUTPUT:

Demo: jQuery click,mouseenter() , mouseleave() & hover methods



Example:

```
<!DOCTYPE html>
<html>

  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script>
      $(document).ready(function () {

        //Focus
        $("#second").focus();

        //Submit
        $("form").submit(function (event) {
          event.preventDefault(); // prevent the page from reloading
          $(this).append('<p>Form Submitted</p>')
        });
      });
    </script>
  </head>

  <body>

    <form>
      First name: <input id="first" type="text" name="FirstName" value="Mickey"><br>
      Last name: <input id="second" type="text" name="LastName" value="Mouse"><br>
      <input type="submit" value="Submit">
    </form>

  </body>

</html>
```

OUTPUT:

First name:	<input type="text" value="Mickey"/>
Last name:	<input type="text" value="Mouse"/>
<input type="submit" value="Submit"/>	

Form Submitted

JQuery Keyboard Events

Example:

```
<!DOCTYPE html>
<html>

  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```

<script>
    let i = 0;
    $(document).ready(function () {

        //Keydown
        $("input").keydown(function () {
            $("input").css("background-color", "yellow");
        });

        //Keyup
        $("input").keyup(function () {
            $("input").css("background-color", "pink");
        });

        //Keypress
        $("input").keypress(function () {
            $("span").text(i += 1);
        });
    });
</script>
</head>

<body>

    Enter your name: <input type="text">

    <p>Enter your name in the input field above. It will change background color on keydown and keyup.</p>
    <p>Keypresses: <span>0</span></p>

</body>

</html>>

```

OUTPUT:

Enter your name:

Enter your name in the input field above. It will change background color on keydown and keyup.

Keypresses: 10

JQuery Browser Events

Example:

```

<!DOCTYPE html>
<html>
    <head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script>
            let x = 0;
            let y = 0;

```



```

$(document).ready(function () {
    $("div").scroll(function () {
        $("#scroll").text(x += 1);
    });
    $(window).resize(function () {
        $("#resize").text(y += 1);
    });
});
</script>
</head>
<body>
    <p>Try the scrollbar in the div</p>

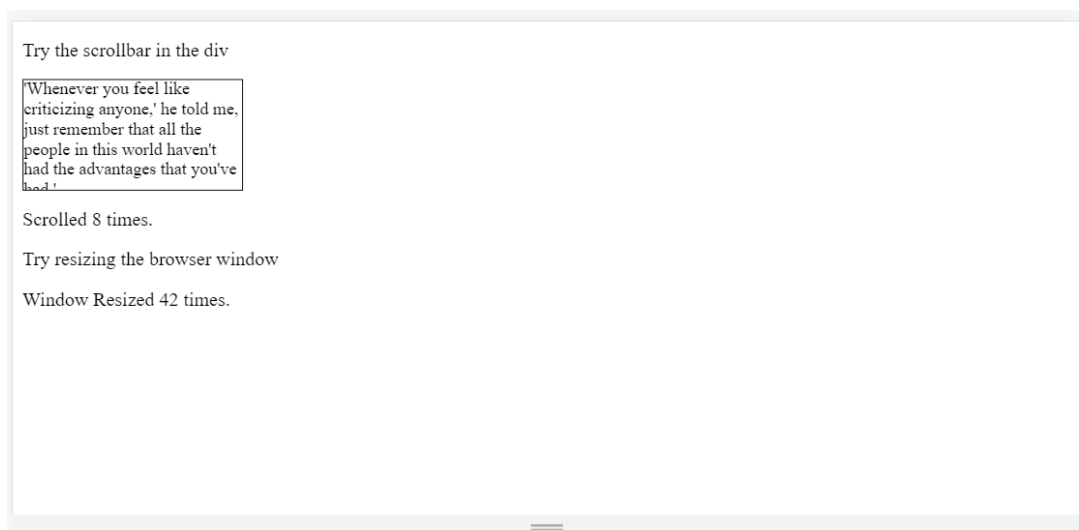
    <div style="border:1px solid black;width:200px;height:100px;overflow:scroll;">In my younger and more vulnerable
        years my father gave me some advice that I've been turning over in my mind ever since.
        <br><br>
        'Whenever you feel like criticizing anyone,' he told me, just remember that all the people in this world
        haven't had the advantages that you've had.'
    </div>

    <p>Scrolled <span id="scroll">0</span> times.</p>

    <p>Try resizing the browser window</p>
    <p>Window Resized <span id="resize">0</span> times.</p>
</body>
</html>

```

OUTPUT:



Conclusion:

So, till now we have learned the fundamentals of JQuery and how easy it is to use and make your web page more interactive.

In this module we have learned about:

- What is jQuery and how to use it
- Different jQuery selectors
- CSS and DOM Manipulation using JQuery
- JQuery Events

So, now from the next module we will be learning about Asynchronous Communication.

Interview Questions

| What are the effects methods used in jQuery?

These are some 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 in the selected elements.
- `fadeOut()` - It shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.

What is the use of `css()` method in JQuery?

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. The jQuery `CSS()` provides two ways:

Return a CSS property

It is used to get the value of a specified CSS property.

```
$(document).ready(function(){  
    $("button").click(function(){  
        alert("Background color = " + $("p").css("background-color"));  
    });  
});
```



Set a CSS property

This property is used to set a specific value for all matched element.

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").css("background-color", "violet");  
    });  
});
```



What is the use of `delay()` method in JQuery?

The jQuery `delay()` method is used to delay the execution of functions in the queue. It is the best method to make a delay between the queued jQuery effects. The jQuery `delay()` method sets a timer to delay the execution of the next item in the queue.

```
$(selector).delay (speed, queueName)
```



speed: It is an optional parameter. It specifies the speed of the delay. Its possible values are `slow`, `fast` and `milliseconds`.

queueName: It is also an optional parameter. It specifies the name of the queue. Its default value is `"fx"` the standard queue effect.

Thank You !

Thank You !