What is jQuery

jQuery is a lightweight JavaScript library that simplifies programming with JavaScript.

A JavaScript library is a library of pre-written JavaScript which allows for easier development of JavaScript-based applications, especially for AJAX and other web-centric technologies.

According to jQuery.com

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

Advantages of using jQuery over raw JavaScript

The use of JQuery has several benefits over using the raw javascript.

1. jQuery is cross-browser
2. jQuery is a lot more easy to use than raw javascript
3. jQuery simplifies and has rich AJAX support
4. jQuery has large development community and many plugins. Example autocomplete textbox plugin.

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:
HTML/DOM manipulation
CSS manipulation
HTML event methods
Effects and animations
AJAX
Utilities

In addition, jQuery has plugins for almost any task out there.

There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:
Google
Microsoft
IBM
Netflix

The jQuery team knows all about cross-browser issues, and they have written this knowledge into the jQuery library. jQuery will run exactly the same in all major browsers.

Adding jQuery to Your Web Pages
There are several ways to start using jQuery on your web site. You can:

Download the jQuery library from jQuery.com

Include jQuery from a CDN, like Google

jQuery CDN

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Both Google and Microsoft host jQuery.

To use jQuery from Google or Microsoft, use one of the following:

Google CDN:

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```

Microsoft CDN:

```
<head>
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>
</head>
```

One big advantage of using the hosted jQuery from Google or Microsoft: Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(*selector*).*action*()**

A $ sign to define/access jQuery
A (*selector*) to "query (or find)" HTML elements
A jQuery *action*() 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".

all jQuery methods in our examples, are inside a document ready event:

```
$(document).ready(function(){

  // jQuery methods go here...

});
```

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

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){
  // jQuery methods go here...
});
```

The element Selector

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

$("p")

Example:

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

The #id Selector

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

```
$(document).ready(function(){
   $("button").click(function(){
      $("#test").hide();
   });
});
```

The .class Selector

The jQuery class selector finds elements with a specific class.

```
$(document).ready(function(){
   $("button").click(function(){
      $(".test").hide();
   });
});
```

More Examples of jQuery Selectors

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute

Functions In a Separate File

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script src="my_jquery_functions.js"></script>
</head>
```

jQuery Event Methods

jQuery Syntax For Event Methods

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!!
});
```

Commonly Used jQuery Event Methods

$(document).ready()

click()

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

dblclick()

The dblclick() method attaches an event handler function to an HTML element.

```
$("#p1").mouseenter(function(){
   alert("You entered p1!");
});

$("#p1").mouseleave(function(){
   alert("Bye! You now leave p1!");
});
```

```
$("#p1").mouseup(function(){
   alert("Mouse up over p1!");
});

$("#p1").hover(function(){
   alert("You entered p1!");
},
function(){
   alert("Bye! You now leave p1!");
});

$("input").focus(function(){
   $(this).css("background-color", "#cccccc");
});

$("input").blur(function(){
   $(this).css("background-color", "#ffffff");
});
```

dblclick

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script
>
<script>
$(document).ready(function(){
   $("p").dblclick(function(){
      $(this).hide();
   });
});
</script>
</head>
<body>

<p>Double click me, if you want me to disappear</p>

</body>
</html>
```

Mouseenter

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></scrip
t>
<script>
$(document).ready(function(){
   $("#p1").mouseenter(function(){
      alert("You entered p1!");
   });
});
</script>
</head>
<body>

<p id="p1">Enter this paragraph.</p>

</body>
</html>
```

The on() Method

The on() method attaches one or more event handlers for the selected elements.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
      $(this).css("background-color", "lightblue");
    },
    click: function(){
      $(this).css("background-color", "yellow");
    }
  });
});
</script>
</head>
<body>
<p>Click or move the mouse pointer over this paragraph.</p>
</body>
</html>
```

Points to remember :

1. ready() function ensures that the DOM is fully loaded.
2. $ is a shortcut for jQuery.
3. All three of the following syntaxes are equivalent:

$(document).ready(handler)
$().ready(handler) (this is not recommended)
$(handler)

What is $(document).ready(function() in jquery:

$(document).ready is a jQuery event. It fires as soon as the DOM is loaded and ready to be manipulated by script. This is the earliest point in the page load process where the script can safely access elements in the page's html DOM. This event is fired before all the images, css etc.. are fully loaded.

The following example works, because the jquery code that adds the event handler to the button is inside the ready() function, which ensures that the DOM is fully loaded before this piece of code is executed, so the JavaScript can find the button element in the DOM and adds the click event handler.

```
<html>
<head>
 <script src="jquery-3.3.1.js"> </script>
  <script type="text/javascript">
       $(document).ready(function () {
          $('#button1').click(function () {
             alert('Welcome to jQuery');
          });
       });
    </script>
</head>
<body>
  <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

In the following example, we have removed the ready() method. When you click the button now, you don't get the alert. This is because the jQuery code is present before the button element, so by the time the jQuery code is executed the button element is not loaded into DOM.

```
<html>
<head>
 <script src="jquery-3.3.1.js"> </script>
  <script type="text/javascript">

        $('#button1').click(function () {
           alert('Welcome to jQuery' );
        });

   </script>
</head>
<body>
  <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

To make this example work, you have 2 options

1. Place your jQuery code in $(document).ready function OR
2. Place your script at the bottom of the page just before the closing </body> element

```
<html>
<head>
 <script src="jquery-3.3.1.js"> </script>
</script>
</head>
<body>
  <input id="button1" type="button" value="Click Me" />
  <script type="text/javascript">
            $('#button1').click(function () {
 alert('Welcome to jQuery');
            });
    </script>
</body>
</html>
```

In most cases, the script can be run as soon as the DOM hierarchy has been fully constructed. So ready() is usually the best place to write your JavaScript code.
For example, let's say we want to display the actual image dimensions (Height and Width). To get the actual image dimensions, we will have to wait until the image is fully loaded, so the jQuery code to get the height and width should be in $(document).ready event.

jQuery HTML Method Main Tips
jQuery offers various methods for manipulating the HTML and CSS.
The method **html()** is used to set or return the HTML content of the selected elements.

```
<html>
<head>
   <title></title>
    <script src="jquery-3.1.1.js"> </script>

   <script type="text/javascript">
      $(document).ready( function () {
         $('#div1').html("Height = " + $('#Image1').height()
            + "<br/>" + "Width = " + $('#Image1').width())
      });
   </script>
</head>
<body>
   <div id="div1"></div>
   <img src="image/1.jpg" id="Image1" />
</body>
</html>
```

Benefits of using CDN

CDN stands for **Content Delivery Network**. A CDN is a system of distributed servers that hosts resources such as images, CSS, JavaScript files etc.

Companies like Microsoft, Google, Yahoo etc have a free public CDN from which we can load jQuery instead of hosting it on our own web server.

Microsoft jQuery CDN
http://www.asp.net/ajax/cdn#jQuery_Releases_on_the_CDN_0

Google jQuery CDN
https://developers.google.com/speed/libraries/devguide#jquery

```
<html>
<head>
<title></title>
<!-- <script src="jquery-3.1.1.js"> </script> -->
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.js"> </script>
</head>
<body>
<script type="text/javascript">
   jQuery('document').ready(function () {
      jQuery('#button1').click(function () {
         alert("Welcome to jQuery");
      });
   });
</script>
<input type="button" value="Click Me" id="button1" />
</body>
</html>
```

Advantages of using a CDN

1. **Distributed CDN servers** : The jQuery file can be downloaded from the CDN server that is closest to the user

2. **Browser Caching** : jQuery is used on many popular websites. If a user has already visited a webpage that uses jQuery from a CDN, and then if he arrives at your page, the jQuery file has already been cached by the browser so there is no need to download it again.

3. **Parallel Downloads** : There is a browser limit on how many files can be concurrently downloaded from a given domain. This number varies from browser to browser. For example, if the browser allows only 2 concurrent downloads from a given domain, the 3rd download is blocked until one of the previous files has been fully downloaded. Since the jQuery file is on a CDN, it is being downloaded from a different domain. So this means the browser allows another 2 parallel downloads from the CDN server.

4. **Reduced server load** : The HTTP request for jQuery file is handled by the CDN server, so the load on your web server is reduced. This also means there is a saving on your website bandwidth consumption which in turn will reduce your hosting cost.

Disadvantages of using a CDN

Your clients may block the CDN. So you may have to request your clients to whitelist the CDN.

What if the required jQuery file cannot be downloaded from CDN

Let assume that, the CDN is down or because of some network issue we are not able to download jQuery from CDN. In this case we will have to fallback to use jQuery file that we hosted on our own server.

Here is the code that falls back to use jQuery on your web server, if it can't be downloaded from CDN. If jQuery is successfully downloaded, jQuery property is added to the window object. If this property is not found then jQuery is not downloaded. So in this case we are writing a script tag to fallback to the local jQuery file.

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.js"> </script>

<script>
   window.jQuery || document.write("<script src='jquery-3.1.1.js '><\/script>");
</script>
```

```
<html>
<head>
<title></title>
<!--<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.js"> </script> -->
<script>
 window.jQuery || document.write("<script src='jquery-3.1.1.js'><\/script>");
</script>
</head>
<body>
<script type="text/javascript">
    jQuery('document').ready(function () {
        jQuery('#button1').click(function () {
            alert("Welcome to jQuery");
        });
    });
</script>
<input type="button" value="Click Me" id="button1" />
</body>
</html>
```

jQuery Selectors

One of the most important concept in jQuery is selectors. jQuery selectors allow you to select and manipulate HTML elements.

Different selectors in jQuery

jQuery Selectors allow you to select html elements in the DOM by

1. Element ID
2. Element Tag Name
3. Element Class Name
4. Element attribute
5. Element Attribute Values

Id Selector in jQuery

To find an HTML element by ID, use the jQuery #id selector

Example : The following example finds button with ID **button1** and attaches the click event handler.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         $('#button1').click(function () {
            alert('Welcome to jQuery');
         });
      });
   </script>
</head>
<body>
   <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

Changes the background colour of the button to yellow

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         $('#button1').css('background-color','yellow')
      });
   </script>
</head>
<body>
   <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

Important points to remember about jQuery #id selector

1. **jQuery #id selector** uses the JavaScript **document.getElementById()** function

2. **jQuery #id selector** is the most efficient among all jQuery selectors. If you know the id of an element that you want to find, then always use the #id selector.

3. HTML element IDs must be unique on the page. **jQuery #id selector** returns only the first element, if you have 2 or more elements with the same ID.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.3.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         $('#button1').css('background-color','yellow')
      });
   </script>
</head>
```

```
<body>
   <input id="button1" type="button" value="Click Me" />
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

4. JavaScript's **document.getElementById()** function throws an error if the element with the given id is not found, where as **jQuery #id selector** will not throw an error. To check if an element is returned by the #id selector use length property.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         document.getElementById("button2").style.backgroundColor = 'yellow';
         // $('#button1').css('background-color','yellow')
      });
   </script>
</head>
```

```
<body>
   <input id="button1" type="button" value="Click Me" />
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

In jQuery:

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            if ($('#button1').length > 0) {
               alert('Element found')
            }
            else {
               alert('Element not found')
            }
         });
   </script>
</head>
```

```
<body>
   <input id="button2" type="button" value="Click Me" />
</body>
</html>
```

5. JavaScript's **document.getElementById()** and **jQuery(#id) selector** are not the same. **document.getElementById()** returns a **raw DOM object** where as **jQuery('#id') selector** returns a jQuery object that wraps the DOM object and provides jQuery methods. This is the reason you are able to call jQuery methods like css(), click() on the object returned by jQuery. To get the underlying DOM object from a jQuery object write $('#id')[0]

6. **document.getElementById()** is faster than **jQuery('#id') selector**. Use document.getElementById() over jQuery('#id') selector unless you need the extra functionality provided by the jQuery object.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            alert($('#button2')[0]);
         });
   </script>
</head>
<body>
   <input id="button2" type="button" value="Click Me" />
   <input id="button2" type="button" value="Click Me" />
</body>
</html>
```

jQuery Element Selector

jQuery Element Selector is used for selecting elements by tag name.

Syntax : $(element)

```
$('td')   // Selects all td elements
$('div a')    // Select all anchor elements that are descendants of div element
$('div, span, a')   // Selects all div, span and anchor elements
```

Example: Alerts the total count of td elements on the page

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         alert($('td').length);
      });
   </script>
</head>
```

```
<body>
   <table style="border:1px solid black; border-collapse:collapse">
      <tr>
         <td>C#</td>
         <td>ASP.NET</td>
         <td>SQL Server</td>
      </tr>
      <tr>
         <td>ADO.NET</td>
         <td>jQuery</td>
         <td>JavaScript</td>
      </tr>
      <tr>
         <td>AJAX</td>
         <td>CSS</td>
         <td>HTML</td>
      </tr>
```

```
 <tr>
        <td>VB</td>
        <td>Dot NET</td>
        <td>Visual C++</td>
      </tr>
      <tr>
        <td>Oracle</td>
        <td>Java</td>
        <td>J2EE</td>
      </tr>
   </table>
</body>
</html>
```

Example: Selects all the tr elements on the page and changes their background colour to red

```
<script type="text/javascript">
      $(document).ready(function () {
         $('tr').css('backgroundColor', 'red');
      });
   </script>
```

Example: Alerts the HTML content of the table

```
<script type="text/javascript">
      $(document).ready(function () {
         alert($('table').html());
      });
   </script>
```

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

css() Method

The following example demonstrates how to get the value of an input field with the

jQuery val() method:

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      alert("Value: " + $("#test").val());
   });
});
</script>
</head>
<body>

<p>Name: <input type="text" id="test" value="Hello LPU"></p>

<button>Show Value</button>
</body>
</html>
```

jQuery html() and text() method:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#btn1").click(function(){
      alert("Text: " + $("#test").text());
   });
   $("#btn2").click(function(){
      alert("HTML: " + $("#test").html());
   });
});
</script>
</head>
<body>

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

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```

jQuery css() Method

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

Return a CSS Property

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      alert("Background color = " + $("p").css("background-color"));
   });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<button>Return background-color of p</button>

</body>
</html>
```

Set Multiple CSS Properties

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("p").css({"background-color": "yellow", "font-size": "200%"});
   });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set multiple styles for p</button>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

Set a CSS Property

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("p").css("background-color", "yellow");
   });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set background-color of p</button>

</body>
</html>
```

Example: Alerts the HTML content of each table row

```
<script type="text/javascript">
      $(document).ready(function () {
         $('table tr').each(function () {
            alert($(this).html());
         });
      });
   </script>
```

Example: Select and change the background colour of all the div, span and anchor elements

```
<script type="text/javascript">
      $(document).ready(function () {
         $('div, span, a').css('backgroundColor', 'yellow');
      });
   </script>
```

Example: Select all anchor elements that are descendants of div element

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            $('div a').css('backgroundColor', 'yellow');
         });
   </script>
</head>
<body>
   <div>
      <a href="http://google.com">Google</a>
   </div>
   <br />
   <a href="http://microsoft.com">Microsoft</a>
</body>
</html>
```

Example: Changes the background color of even rows to green and odd rows to yellow on both the tables.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            $('tr:even').css('backgroundColor', 'green');
            $('tr:odd').css('backgroundColor', 'yellow');
         });
   </script>
</head>
<body>
   <table id="table1" border="1">
      <tr>
         <td>C#</td>
         <td>ASP.NET</td>
         <td>SQL Server</td>
      </tr>
```

```
<tr>
        <td>ADO.NET</td>
        <td>jQuery</td>
        <td>JavaScript</td>
      </tr>
      <tr>
        <td>AJAX</td>
        <td>CSS</td>
        <td>HTML</td>
      </tr>
      <tr>
        <td>VB</td>
        <td>Dot NET</td>
        <td>Visual C++</td>
      </tr>
      <tr>
        <td>Oracle</td>
        <td>Java</td>
        <td>J2EE</td>
      </tr>
  </table>
  <br />
```

```
<table id="table2" border="1">
    <tr>
      <td>Rahul</td>
      <td>Rohit</td>
      <td>Virat</td>
    </tr>
    <tr>
      <td>Pujara</td>
      <td>Rahane</td>
      <td>Murli</td>
    </tr>
    <tr>
      <td>Saha</td>
      <td>Jadeja</td>
      <td>Ashwin</td>
    </tr>
```

```
<tr>
        <td>Ishant</td>
        <td>Shami</td>
        <td>Umesh</td>
    </tr>
    <tr>
        <td>Nair</td>
        <td>Shikhar</td>
        <td>Bhuvi</td>
    </tr>
  </table>
</body>
</html>
```

Example: To change the background color of even rows to green and odd rows to yellow just for one of the table, use #id selector along with element selector.

```
<script type="text/javascript">
        $(document).ready(function () {
            $('#table1 tr:even').css('backgroundColor', 'green');
            $('#table1 tr:odd').css('backgroundColor', 'yellow');
    });

    </script>
```

jQuery Class Selector

jQuery Class Selector is used for selecting elements using their class name .

Syntax : $('.class')

jQuery Class Selectors uses JavaScript's native **getElementsByClassName**() function if the browser supports it.

```
$('.small')              // Selects all elements with class small.
$('.small,.big')         // Selects all elements with class small or big.
$('div.small,.big')      // Selects div elements with class small and any element with class big.
```

Example: Selects all elements **with class "small"** and sets 5px solid red border.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
```

```
<script type="text/javascript">
      $(document).ready(function () {
         $('.small').css('border', '5px solid red');
      });
   </script>
</head>
<body>
   <span class="small">
      Span 1
   </span>
   <br /><br />
   <div class="small">
      Div 1
   </div>
   <br />
   <span class="big">
      Span 2
   </span>
   <p class="big">Lovely Professional University</p>
</body>
</html>
```

Example: Selects all elements **with class "small" and all span elements with class "big"** and sets 5px solid red border.

```
<script type="text/javascript">
      $(document).ready(function () {
         $('.small, span.big').css('border', '5px solid red');
      });
   </script>
```

Example: Selects all elements with **class small that are nested in a an element with id=div2** and sets 5px solid red border.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            $('#div2 .small').css('border', '5px solid red');
         });
   </script>
</head>
```

```
<body>
   <div id="div1" class="small">
      DIV 1
   </div>
   <br />
   <div id="div2">
      Div 2
      <br />
      <div class="small">
         DIV 3
      </div>
      <br />
      <span class="small">
         SPAN
      </span>
   </div>
</body>
</html>
```

Example: Selects all elements **with class small** and sets 5px solid red border. Notice div1 has 2 classes - small and big.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
      <script type="text/javascript">
         $(document).ready(function () {
            $('.small').css('border', '5px solid red');
         });
   </script>
</head>
<body>
   <div class="small big">
      DIV 1
   </div>
   <br />
   <div class="small">
      DIV 2
   </div>
</body>
</html>
```

Example: Selects all elements **that has both the classes - small and big**. There should be no space between the class names.

```
<script type="text/javascript">
        $(document).ready(function () {
           $('.small.big').css('border', '5px solid red');
        });
   </script>
```

NB: If you have a **space between the two class names then we are trying to find descendants**, i.e. find elements with class big that are descendants of an element with class small.

```
<html>
<head>
   <title></title>
   <script src="jquery-3.1.1.js"></script>
        <script type="text/javascript">
           $(document).ready(function () {
              $('.small .big').css('border', '5px solid red');
           });
   </script>
</head>
```

```
<body>
   <div class="small big">
      DIV 1
   </div>
   <br />
   <div class="small">
      DIV 2
      <div class="big">
         DIV 3
      </div>
   </div>
</body>
</html>
```

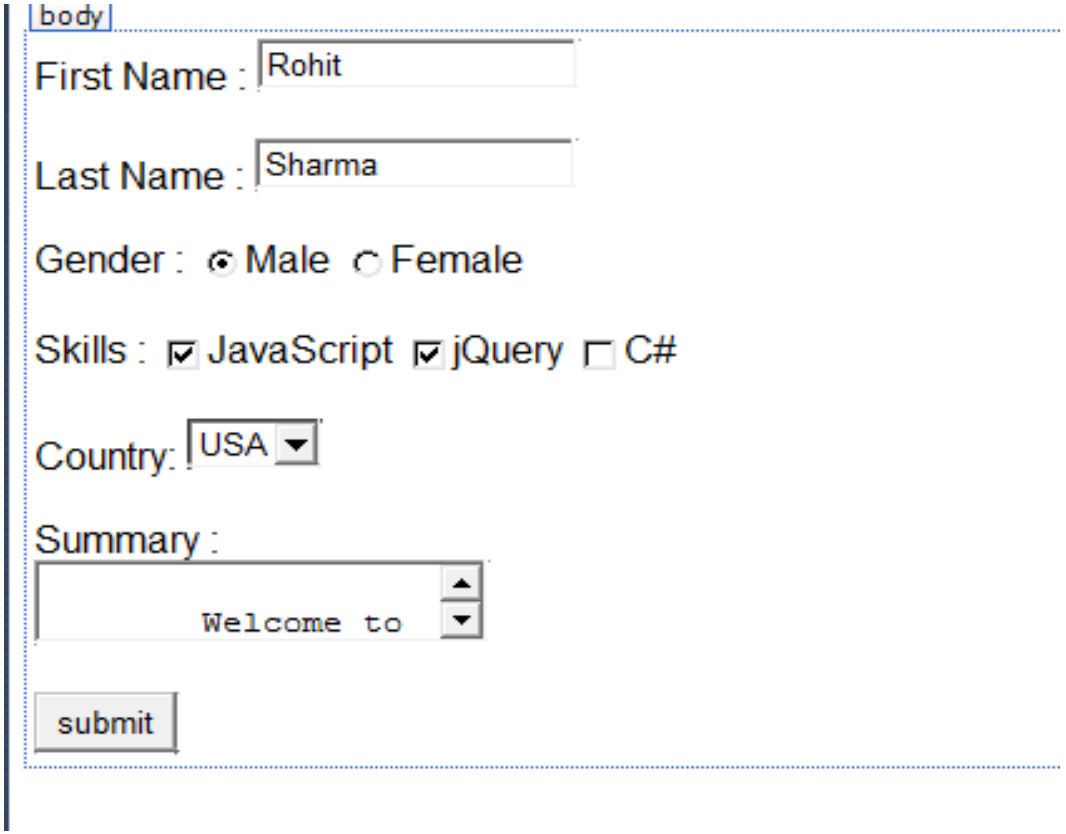
NB: **Another way to select all elements that has both the classes - small and big is by using filter method**. But this approach is slower because it will first create a list of objects with class "small" and then removes elements that does not have class "big“

```
<script type="text/javascript">
            $(document).ready(function () {
               $('.small').filter('.big').css('border', '5px solid red');
            });
   </script>
```

jQuery input vs :input

$(':input') selects all input, textarea, select and button elements where as $('input') just selects elements with an input tag.

Consider the web page below



```
<html>
<head>
    <title></title>
    <script src="jquery-3.3.1.js"></script>
<script>
$(document).ready(function(){
$(":input").css("background","orange");
});
</script>
</head>
<body style="font-family:Arial">
    First Name : <input type="text" value="Rohit" />
    <br /><br />
    Last Name : <input type="text" value="Sharma" />
    <br /><br />
    Gender :
    <input type="radio" name="gender" checked="checked" value="Male">Male
    <input type="radio" name="gender" value="Female">Female
    <br /><br />
    Skills :
    <input type="checkbox" name="skills" checked="checked"
          value="JavaScript" />JavaScript
    <input type="checkbox" name="skills" checked="checked"
          value="jQuery" />jQuery
    <input type="checkbox" name="skills" value="C#" />C#
    <br /><br />
```

```
Country:
   <select>
      <option selected="selected" value="USA">USA</option>
      <option value="India">India</option>
      <option value="UK">UK</option>
   </select>
   <br /><br />
   Summary :
   <br />
   <textarea>
      Welcome to Lovely Professional University
   </textarea>
   <br /><br />
   <input type="submit" value="submit" />
</body>
</html>
```

jQuery code to get textbox value using $(input)

```
<html>
<head>
<script src="jquery-3.3.1.js"></script>
<script type="text/javascript">
      $(document).ready(function () {
         $('input[type="text"]').each(function () {
            alert($(this).val());
         });
      });
</script>
</head>
<body>
<input type="text" value="Samir"/>
</body>
</html>
```

jQuery code to get textbox value using $(:input)

```
<html>
<head>
<script src="jquery-3.3.1.js"></script>
<script type="text/javascript">
      $(document).ready(function () {
         $(':input[type="text"]').each(function () {
            alert($(this).val());
         });
      });
</script>
</head>
<body>
<input type="text" value="Samir"/>
<input type="text" value="Rohan"/>

</body>
</html>
```

Each Function

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("li").each(function(){
            alert($(this).text())
        });
    });
});
</script>
</head>
<body>

<button>Alert the value of each list item</button>

<ul>
  <li>Coffee</li>
  <li>Milk</li>
  <li>Soda</li>
</ul>

</body>
</html>
```

Which one is better for performance $('input[type="text"]') or $(':input[type="text"]')

$('input[type="text"]') is better for performance over $(':input[type="text"]').

This is because $(':input[type="text"]') needs to scan all input elements, textarea, select etc, where as $('input[type="text"]') scans only input elements.

So if you want to find elements with an input tag, it is always better to use $('input[type="text"]') over $(':input[type="text"]')

jQuery Attribute Selector and Attribute Value Selector

jQuery Attribute Selector is used for selecting elements
1. That have specified attribute
2. That have specified attribute values

Syntax :

```
$('[attribute]')
$('[attribute="value"]')
```

```
$('[title]') // Selects all elements that have title attribute
$('div[title]') // Selects all div elements that have title attribute
$('[title="divTitle"]') // Selects all elements that have title attribute value - divTitle
$('div[title="divTitle"]') // Selects all div elements that have title attribute value - divTitle
```

Example : Selects all elements with title attribute and sets 5px solid red border

```
<html>
<head>
   <script src="jquery-3.3.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         $('[title]').css('border', '5px solid red');
      });
   </script>
</head>
<body>
   <div title="div1Title">
      DIV 1
   </div>
   <br />
   <div title="div2Title">
      DIV 2
   </div>
```

```
 <p title="pTitle">
      This is a paragraph
   </p>
   <span title="span1Title">
      SPAN 1
   </span>
   <br /><br />
   <span>
      SPAN 2
   </span>
</body>
</html>
```

Example: Selects all div elements with title attribute and sets 5px solid red border

```
<script type="text/javascript">
      $(document).ready(function () {
         $('div[title]').css('border', '5px solid red');
      });
   </script>
```

Example :Selects all elements with title attribute value - div1Title, and sets 5px solid red border

```
<script type="text/javascript">
      $(document).ready(function () {
         $('[title="div1Title"]').css('border', '5px solid red');
      });
   </script>
```

Example: Selects all div elements with title attribute value - div1Title, and sets 5px solid red border

```
<script type="text/javascript">
      $(document).ready(function () {
         $('div[title="div1Title"]').css('border', '5px solid red');
      });
   </script>
```

Example: Selects all div elements with both title and style attributes, and sets 5px solid black border

```
<html>
<head>
   <script src="jquery-3.3.1.js"></script>
   <script type="text/javascript">
      $(document).ready(function () {
         $('div[title][style]').css('border', '5px solid black');
      });
   </script>
</head>
<body>
   <div title="divTitle" style="background-color:red">
      Red DIV
   </div>
   <br />
   <div title="divTitle" style="background-color:green">
      Green DIV
   </div>
   <br />
```

```
<div title="divTitle">
      Normal Div
   </div>
   <br />
   <div>
      Normal Div without any attributes
   </div>
</body>
</html>
```

Example: Selects all div elements with title attribute value - divTitle, and style attribute value - background-color:red, and sets 5px solid black border.

```
<script type="text/javascript">
      $(document).ready(function () {
         $('div[title="divTitle"][style="background-color:red"]')
         .css('border', '5px solid black');
      });
   </script>
```

Example: Selects all div elements with either title or style attributes, and sets 5px solid black border

```
<script type="text/javascript">
      $(document).ready(function () {
         $('div[title],[style]').css('border', '5px solid black');
      });
   </script>
```