



Full Stack Web Development with Python and Django

jQuery

Study Material



Full Stack Web Development with Python and Django

Module-6: jQuery

Official website: jquery.com

jQuery is a fast, small, and rich JavaScript library.

We can use jQuery Library to grab and manipulate HTML elements, to perform event handling and AJAX.

jQuery supports multiple browsers, i.e. it provides support for cross-browser compatibility.

The main advantage of jQuery is it provides several methods and objects in the form of JavaScript files, so that we can use these directly and the developer's life gets simplified.

Note: In Plain old JavaScript (also known as Vanilla JavaScript), we have to write everything manually. But if we use jQuery, we are not required to write much code and we can use its library directly.

Sample code to change color of every h1 tag:

Vanilla JS code:

```
1) var myh1=document.querySelectorAll('h1')
2) for( h1 of myh1){
3) h1.style.color='red';
4) }
```

jQuery Code:

```
$('h1').css('color','red')
```

Advantages of jQuery:

1. It provides several built-in methods and objects. We can use these directly so that development will become very easy.
2. Clear and Shorter code
3. Ease of use
4. Cross Browser support
5. AJAX support



Limitations of jQuery:

1. What ever jQuery doing, we can implement everything without jQuery also. ie jQuery won't do any things extra.
2. The total library should be loaded compulsory, which creates performance problems.

Note: Because of above limitations, some part of the developer's community won't recommend jQuery usage.

Eg: youmightnotneedjquery.com

How to connect with jQuery:

We can make jQuery library available to our application in the following 2 ways

1. By Locally
2. By CDN

1. By Locally:

Download jQuery.js file from jquery.com

<https://code.jquery.com/jquery-3.3.1.js>

Download and place in application folder.

Inside head of html we have to write <script> tag as follows .

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

2. By using CDN:

www.code.jquery.com

```
1) <script  
2)   src="http://code.jquery.com/jquery-3.3.1.js"  
3)   integrity="sha256-2Kok7MbOyxpGUvVak/HJ2jigOSYS2auK4Pfzbn7uH60="<br>  
4)   crossorigin="anonymous"></script>
```

We can get several jquery CDNs from the google

Eg:

```
<script type="text/javascript"  
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```



How to check whether jQuery available or not in our application:

From the console just type
> jQuery or \$
We should not get any error

jQuery Selectors:

In Vanilla Javascript we have several methods to select/grab html elements like

- getElementById()
- getElementsByName()
- getElementsByTagName()
- querySelector()
- querySelectorAll()
- etc

But in jQuery we have only one way to select html elements.i.e to use \$ symbol

In jQuery \$ symbol is equivalent to querySelectorAll() in Vanilla Javascript

Demo Application:

demo.html:

```
1) <!DOCTYPE html>
2) <html lang="en" dir="ltr">
3) <head>
4) <script
5) src="http://code.jquery.com/jquery-3.3.1.js"
6) integrity="sha256-2Kok7MbOyxpqUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
7) crossorigin="anonymous"></script>
8) <meta charset="utf-8">
9) <title></title>
10) </head>
11) <body>
12) <h1>Hello First H1</h1>
13) <h2>Favourite Drink:</h2>
14) <ul>
15) <li>KingFisher</li>
16) <li>KnockOut</li>
17) <li>Foster</li>
18) <li id='special'>HumanBlood</li>
19) </ul>
20) <a href="https://google.com">Click Here to go to Google</a>
21) </body>
22) </html>
```




jQuery code:

```
$('#h1')  
$('#li')  
$('#special')  
$('#body')  
etc...
```

Manipulating HTML Elements:

Once we grab elements by using \$ symbol, we can manipulate by using `css()` method.

Syntax:

```
$(selector).css(property,value)
```

Examples:

```
$('#h1').css('color','white');  
$('#h1').css('background','red');  
$('#h1').css('border','5px solid green');
```

We can save selected html element by using variable

```
var x = $('#h1')  
x.css('color','white');  
x.css('background','red');  
x.css('border','5px solid green');
```

Instead of passing parameters one by one, we can create Object and pass that object directly.

```
1) var x = $('#h1')  
2) var myCSS={  
3)   color:'white',  
4)   background:'green',  
5)   border:'red 5px solid'  
6) }  
7) x.css(myCSS);
```

Note: We can use \$ symbol to select and `css()` method to manipulate html elements.



How to select a particular html element instead of all matched elements:

By default \$('element') selects all matched html elements.

But by using the following ways we can get particular matched element.

\$('element:first')
\$('element:last')
\$('element:first-of-type')
\$('element:nth-of-type(n)')
etc...

Eg:

1. \$('h1')==>Selects all h1 tags
2. \$('h1:first')==>Selects only first h1 tag
3. \$('h1').first()==>selects only first h1 tag
4. \$('h1:first-of-type')==>Selects only first h1 tag
5. \$('h1:nth-of-type(2)')==>Selects second h1 tag
6. \$('h1:last')==>Selects only last h1 tag
7. \$('h1').last()==>selects only last h1 tag

Q1. Write Vanilla Javascript and jQuery codes to change all h1 tags text color as white and background as red.

Vanilla Javascript code:

```
1) var allh1s=document.querySelectorAll('h1');  
2) for(h1 of allh1s){  
3)   h1.style.color='white';  
4)   h1.style.background='red';  
5) }
```

jQuery Code:

```
1) var mystyles={  
2)   color:'white',  
3)   background:'red'  
4) };  
5) $('h1').css(mystyles)  
6)  
7) Instead of this we can write directly as follows  
8)  
9) $('h1').css({  
10)   color:'white',  
11)   background:'red'  
12) });
```



Q2. Write Vanilla Javascript and jQuery codes to set all li tags font size as 20px.

Vanilla Javascript code:

```
1) var x=document.querySelectorAll('li');  
2) for(li of x){  
3)   li.style.fontSize='20px';  
4) }
```

jQuery Code:

```
$('li').css('fontSize','20px')
```

Note: The biggest advantage of jQuery is we can do more things with less code(Write less,do More)

Demo Application:

demo.html:

```
1) <!DOCTYPE html>  
2) <html lang="en" dir="ltr">  
3) <head>  
4)   <script  
5)     src="http://code.jquery.com/jquery-2.2.4.js"  
6)     integrity="sha256-IT6Q9iMJYUQiMWNd9IDyBUSTlq/8PuOW33aOqmvFpql="&br/>7)     crossorigin="anonymous"></script>  
8)  
9)   <meta charset="utf-8">  
10)  <title></title>  
11) </head>  
12) <body>  
13)  <p>This is First Paragraph</p>  
14)  <p id="second">This is Second Paragraph</p>  
15)  <p class="remaining">This is Third Paragraph</p>  
16)  <p class="remaining">This is Fourth Paragraph</p>  
17) </body>  
18) </html>
```

Case-1: Select all <p> tags and set background as green
`$('.p').css('background','green')`

Case-2: Select all <p> tags with class 'remaining' and make them 200px wide(width)
`$('.remaining').css('width','200px')`



Case-3: Select all <p> tags with id 'second' and give red solid 10px border.
`$('#second').css('border','10px solid red')`

case-4: Select only first <p> tag and change text color as white
`$('#p:first').css('color','white')`

case-4: Select only third <p> tag and change font-size as 30px
`$('#p:nth-of-type(3)').css('fontSize','30px')`

The Most Commonly used jQuery Methods:

The following are the most commonly used jQuery methods

1. text()
2. html()
3. attr()
4. val()
5. addClass()
6. removeClass()
7. toggleClass()

1. text():

We can use this method to get and set text of the matched elements. i.e this method acts as both getter and setter method.

`text()`====>To get text of all matched elements including child tags.In this case this method acts as getter method

`text(content)`====>To set provided text content for every matched element. In this case this method acts as setter method.

Note: If any method developed to acts as both getter and setter method then it is said to be this method follows getter and setter paradigm.

Eg:

```
1) <!DOCTYPE html>
2) <html lang="en" dir="ltr">
3) <head>
4) <script
5) src="http://code.jquery.com/jquery-2.2.4.js"
6) integrity="sha256-IT6Q9iMjYUqIMWnD9IDyBUSTlq/8PuOW33aOqmvFpqI="
7) crossorigin="anonymous"></script>
8) <meta charset="utf-8">
9) <title></title>
10) </head>
11) <body>
```




```
12) <h1>This is H1 Data</h1>
13) <h2>Choose Your Favourite Subject:</h2>
14) <ul>
15) <li>HTML</li>
16) <li>JavaScript</li>
17) <li>CSS</li>
18) <li>jQuery</li>
19) </ul>
20) </body>
21) </html>
```

1. To get text content of h1:

`$('#h1').text()`

2. To get text of all li tags

`$('#li').text()` or
`$('#ul').text()`

3. To get text of only first li

`$('#li').first().text()`

4. To set h1 text content as : DURGA SOFTWARE SOLUTIONS

`$('#h1').text('DURGA SOFTWARE SOLUTIONS')`

5. To set every li tag text content as : KINGFISHER

`$('#li').text('KINGFISHER')`

2. html():

We can use this method to get and set html content.

If we are not passing any argument then this method acts as getter method, which returns HTML contents of the first matched element.

Eg: `$('#h1').html()`

If we are passing argument then this method acts as setter method, which sets the HTML contents of every matched element.

Eg: `$('#li').html('AMAZON')`

3. attr():

We can use this method to get and set attribute values.

`attr(attributename)`--->To get the value of specified attribute of the first matched element.

`attr(attributename, attributevalue)`

-->If the specified attribute already there then old value replaced with new value. If the specified attribute not already available then a new attribute will be added for every matched tag.



Demo for getting and setting src attribute of img tag:

```
  
  

```

1. To set width and height of every image properly

```
$('.img').css({width:'150px',height:'150px'});
```

2. To get src attribute value of the first image:

```
$('.img').attr('src')
```

3. To set src attribute of all images with our new image

```
$('.img').attr('src',"https://images.unsplash.com/photo-1484406566174-9da000fda645?ixlib=rb-0.3.5&ixid=eyJhcHBfaWQiOjEyMDd9&s=a439fe04a06e1457297643e219add900&auto=format&fit=crop&w=400&q=60");
```

4. To change only first image src attribute value

```
$('.img').first().attr('src',"https://images.unsplash.com/photo-1504350440606-81847d413a13?ixlib=rb-0.3.5&ixid=eyJhcHBfaWQiOjEyMDd9&s=bdf251e852bd65abcc522fc4e903da02&auto=format&fit=crop&w=400&q=60")
```

Demo for getting and setting type attribute of input tag:

User Input: `<input type="text" name="" value="">`

1. To get the value of type attribute of input tag

```
$('.input').attr('type')
```

2. To set the value of type attribute with 'color'

```
$('.input').attr('type','color')
```

3. To set the value of type attribute with 'checkbox'

```
$('.input').attr('type','checkbox')
```

4.val() method:

We can use val() method to get value of the first matched element.

Eg: value entered in the text box

which radio button selected

which value selected from dropdown box



We can also use `val()` method to set value for every matched element.

`val()`==>getter method

`val(text)`==>setter method

Eg 1: with input tag

Enter Name:<input type="text" name="user1" value="" placeholder="Your Name">

Enter Name:<input type="text" name="user2" value="" placeholder="Your Name">

`$('#input').val()`==>It returns the value entered by the end user in the first text box.

`$('#input').val('durga')`==>To set value durga for every text box.

Usease: While implementing reset button functionality to clear all input fields this method is helpful.

Eg 1: with dropdown menu

Choose Your Required Course:

```
1) <select class="" name="">
2) <option value="CorePython">Core Python</option>
3) <option value="AdvPython">Adv Python</option>
4) <option value="DJango">DJango</option>
5) <option value="Flask">Flask</option>
6) </select>
```

`$('#select').val()`==>Returns Selected Value

`$('#select').val('DJango')` ==>To set value as DJango

addClass(),removeClass() &toggleClass():

addClass(): We an use this method to add specified class/classes to the set of matched elements.

removeClass(): Remove a single class/classes from each matched element.

toggleClass(): We can use this method to add and remove classes from the matched elements.

If the specified class already set then it will be removed. It is not already set then the class will be added for every matched element.

Eg:

```
1) <!DOCTYPE html>
2) <html lang="en" dir="ltr">
3) <head>
4) <script
```




```
5) src="http://code.jquery.com/jquery-2.2.4.js"
6) integrity="sha256-IT6Q9iMJYUQiMWNd9IDyBUStIq/8PuOW33aOqmvFpql="
7) crossorigin="anonymous"></script>
8) <meta charset="utf-8">
9) <title></title>
10) <style >
11) .high{
12)   color:white;
13)   background:red;
14) }
15) .low{
16)   color:white;
17)   background: blue;
18) }
19) .completed{
20)   text-decoration: line-through;
21) }
22) </style>
23) </head>
24) <body>
25) <h1>List of Activities: </h1>
26) <ul>
27) <li>Meeting Sunny</li>
28) <li>Reading Django</li>
29) <li>Taking Classes</li>
30) <li>Visiting US</li>
31) <li>Watching Movie</li>
32) <li>Playing Cricket</li>
33) <li>Swimming </li>
34) </ul>
35) </body>
36) </html>
```

1. \$('li').addClass('high')

It will add 'high' class to every li tag

2. \$('li:nth-of-type(even)').removeClass('high')

It will remove 'high' class from every even numbered li tag.

3. \$('li:nth-of-type(2)').addClass('completed')

It will add 'completed' class only for 2nd li

4. \$('li').toggleClass('low')

It will add 'low' class for every li tag(because this class is not already set)

5. \$('li').toggleClass('low')

It will remove 'low' class from every li tag(because this class is already set)



Event Handling by using jQuery:

We can implement event handling to make our html elements interactive.

jQuery defines several methods for event handling. We can get complete jquery event handling related methods from below link:

<https://api.jquery.com/category/events/>

The top 3 most commonly used jQuery event methods:

1. click()
2. keypress()
3. on()

1. click():

jQuery click() method can be used to add a click listener to the elements.

Eg 1: To raise alert message when ever we are clicking h1 tag.

```
1) $('h1').click(function(){  
2)   alert('h1 tag got clicked');  
3) });
```

Eg 2: To raise alert message and to change background color of last button on click event

```
1) $('button:last').click(function(){  
2)   alert('Hello Stupid Dont Sleep I will Kill You!!!');  
3)   $(this).css('background','red')  
4) });
```

Note:

In Vanilla Java Script, 'this' always represent current element. But in jQuery we have to use \$(this)

Demo Application:

```
1) <!DOCTYPE html>  
2) <html lang="en" dir="ltr">  
3)   <head>  
4)     <script  
5)       src="http://code.jquery.com/jquery-2.2.4.js"  
6)       integrity="sha256-iT6Q9iMUyQiMWNd9IDyBUStIq/8PuOW33aOqmvFpql="  
7)       crossorigin="anonymous"></script>  
8)     <meta charset="utf-8">  
9)     <title></title>  
10)  </head>
```



```
11) <body>
12) <h1>jQuery Event Handler Demo </h1>
13) <button type="button" name="button">Dont Sleep First Warning</button>
14) <button type="button" name="button">Dont Sleep Second Warning</button>
15) <button type="button" name="button">Dont Sleep Third Warning</button>
16) <script type="text/javascript" src="demo.js">
17) </script>
18) </body>
19) </html>
```

demo.js:

```
1) $('h1').click(function(){
2)   alert('h1 tag clicked')
3) })
4) $('button:first').click(function(){
5)   alert('Hello Dont Sleep');
6)   $(this).css('background','yellow')
7) });
8) $('button:nth-of-type(2)').click(function(){
9)   alert('Dont Sleep I will beat you');
10)  $(this).css('background','orange')
11) });
12) $('button:last').click(function(){
13)  alert('Hello Stupid Dont Sleep I will Kill You!!!');
14)  $(this).css('background','red')
15) });
```

2. keypress():

We can use this method to add keypress listener to elements.

i.e whenever we are pressing the key if we want to do any activity automatically then we should go for keypress() method.

Eg: To raise alert message for every character typed in text box

Enter Name:

jQuery code:

```
1) $('input').keypress(function(){
2)   alert('Inserted one character!!!');
3) })
```



To know the pressed character:

In the case of keypress event, the total information is available in the event object. For every character the corresponding key-code is available.

char-code, unicode, ascii code, which all represents same key-code

The complete key-code information is available in the following link

<https://www.cambiaresearch.com/articles/15/javascript-char-codes-key-codes>

If we use event.which, then we will get the corresponding key-code

To raise alert message whenever we are pressing x or X:

```
1) $('input').keypress(function(event){
2)   if (event.which==88 || event.which==120)
3)   {
4)     alert('Hello You are pressing x or X. You are under monitoring !!!!')
5)   }
6) })
```

Note: The key-code for 'x' is :120 where as for 'X' is :88

To raise alert message with typed content whenever we are pressing enter key:

```
1) $('input').keypress(function(event){
2)   if (event.which==13)
3)   {
4)     alert('Hello just pressed enter key and your current content is:'+$(this).val())
5)   }
6) })
```

Difference between keypress and keydown, keyup events:

keydown and keyup provides a code indicating which key is pressed, where as keypress indicates that which character was entered.

Eg: For shift+a

1. keydown for 'shift'
2. keydown for 'a'
3. keyup for 'a'
4. keyup for 'shift'
5. key press for 'A'



Event handling by using on() method:

on() is the most commonly used method to perform event handling in jQuery.

It is similar to Vanilla JavaScript addEventListener() method.

click()→applicable only for click event

keypress()→applicable only for keypress event

on()→applicable for all events including click and keypress.

Eg 1: Whenever mouseover event happens, the text of h1 tag should be changed to 'Bangalore City' with red background and white text.

Whenever mouseout event happens, the text of h1 tag should be changed to 'Hyderabad City' with green background and white text.

```
1) $('h1').on('mouseover',function(){
2)   $(this).text('BANGALORE CITY')
3)   $(this).css({background:'red',color:'white'});
4) })
5) $('h1').on('mouseout',function(){
6)   $(this).text('HYDERABAD CITY')
7)   $(this).css({background:'green',color:'white'});
8) })
```

Eg 2: For button

Single Click: alert message as Hello Stupid dont click

Double Click: alert message as Hello Animal I will kill you

html:

```
1) <button type="button" name="button">Click Here to get Greeting Message</button><br>
   <br>
2) <button type="button" name="button">Double Click Here to get Greeting Message</button>
```

jQuery:

```
1) $('button:first').on('click',function(){
2)   alert('Hello Stupid Dont Click!!!')
3) })
4)
5) $('button:last').on('dblclick',function(){
6)   alert('Hello Animal I Will Kill You!!!')
7) })
```




jQuery Effects:

jQuery provides several in-built effects. But main important effects are:

1. Fading Effects
2. Sliding Effects

<https://api.jquery.com/category/effects/>

1. Fading Effects:

jQuery defines the following methods for fading purposes

1. fadeOut():

Hide the matched elements by fading them to transparent.

2. fadeIn():

Display the matched elements which are fadeout

3. fadeToggle():

Display or hide the matched elements
if already fadeOut then fadeIn will be performed.
If already fadeIn then fadeOut will be performed.

Various sample codes:

```
1) $('div').fadeOut();  
2) $('div').fadeOut(2000);  
3) $('div').fadeOut(2000,function(){  
4)   console.log('Fadeout of the element completed');  
5) });
```

2000 is the time in milliseconds

Similarly we can use these combinations for fadeIn() and fadeToggle() also.

Demo Application:

demo.html:

```
1) <!DOCTYPE html>  
2) <html lang="en" dir="ltr">  
3)  
4)   <head>  
5)     <style>  
6)       div{  
7)         height: 100px;  
8)         width: 100px;
```



```
9)    background: red;
10)   color:white;
11)   margin:20px;
12)   text-align: center;
13)   float: left;
14)   }
15)   </style>
16)   <script
17)   src="http://code.jquery.com/jquery-2.2.4.js"
18)   integrity="sha256-IT6Q9iMJYUQiMWNd9IDyBUStIq/8PuOW33aOqmvFpql="
19)   crossorigin="anonymous"></script>
20)   <meta charset="utf-8">
21)   <title></title>
22) </head>
23) <body>
24)   <button type="button" name="button">Click Here</button>
25)   <div>First Div</div>
26)   <div>Second Div</div>
27)   <div>Third Div</div>
28)   <div>Fourth Div</div>
29) </body>
30) <script type="text/javascript" src="demo.js">
31)
32) </script>
33) </html>
```

demo.js:

```
1) $('button').on('click',function(){
2)   $('div').fadeToggle(2000);
3) })
```

Note: Whenever we perform fadeout just elements will be hidden but won't be removed. But based on requirement we can remove the matched elements also.

```
1) $('button').on('click',function(){
2)   $('div').fadeToggle(2000,function(){
3)     $(this).remove();
4)   });
5) })
```



Sliding Effects:

jQuery defines the following methods for sliding effects purposes.

1. slideUp():

Hide the matched elements with a sliding motion.

2. slideDown():

Display the matched elements with a sliding motion.

3. slideToggle():

Display or hide the matched elements with a sliding motion.

Eg:

```
$('#button').on('click',function(){  
    $('#div').slideToggle(2000);  
})
```

Note: passing function as argument, remove matched elements are exactly same as fading effects.

Assignments:

1. Connect4 Game
2. Todo List Application
3. Implement atleast 2 case studies from w3schools howto
<https://www.w3schools.com/howto/>