

# UNIT – 4

## jQuery

BCA SEM – 2  
WEB PROGRAMMING  
Code : CS-09

# Topics :

- What is jQuery?
- jQuery Syntax
- jQuery Selector
  1. Element Selector
  2. Class Selector
  3. id Selector
- jQuery Events
  1. click
  2. dbclick
  3. keypress
  4. keydown
  5. keyup
  6. submit
  7. change
  8. focus
- 9. blur
- 10. load
- 11. resize
- 12. scroll
- 13. unhide
- jQuery Effects
  1. hide
  2. show
  3. fade
  4. Slide
- jQuery Methods
  1. css
  2. height
  3. Width
  4. innerWidth
- 5. innerHeight
- 6. outerWidth
- 7. outerHeight
- 8. Html
- 9. Text
- 10. Append
- 11. prepend,
- 12. After
- 13. before
- 14. addClass
- 15. removeClass
- 16. toggleClass
- 17. Remove
- 18. empty

# jQuery :



- JQuery is Lightweight JavaScript Library .
- **"write less, do more"**.
- 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.
- 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.

# Why use JQuery :

- Short Selector

Ex : in JS to access class element be like :

`document.getElementsByClassName('class_name');`

Ex : in JQuery to access class element be like : `$('.class_name')`

- Variety of Animation functions

- Easy DOM Manipulation

- Easy CSS Styling

- Easy DOM Traversing

- Simple Ajax Code

- **3 STEPS TO IMPLEMENT JQuery in HTML :**

1. Download jquery.js file
2. Include jquery.js file in HTML file
3. Do JQuery code in `<script>` tag

## Jquery CDN : Content Delivery Network

# jQuery Syntax:

- A jQuery statement typically starts with the dollar sign (\$) and ends with a semicolon (;).
- In jQuery, the dollar sign (\$) is just an alias for jQuery. Let's consider the following example code which demonstrates the most basic statement of the jQuery.

- **SYNTAX :**

```
<script>
```

```
$(document).ready(function() {  
    // Some code to be executed...  
    alert("Hello World!");  
});
```

```
</script>
```

- The **<script>** element — Since jQuery is just a JavaScript library, so the jQuery code can be placed inside the **<script>** element. However, if you want to place it in an external JavaScript file, which is preferred, you just remove this part.
- The **\$(document).ready(handler);** — This statement is typically known as ready event. Where the handler is basically a function that is passed to the ready() method to be executed safely as soon as the document is ready to be manipulated i.e. when the DOM hierarchy has been fully constructed.
- The jQuery **ready()** method is typically used with an anonymous function.

# jQuery Syntax:

- The jQuery syntax is for selecting HTML elements and performing some action on the element(s).
- **syntax:-** `$(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)
- **The Document Ready Event :-**
- You might have noticed that all jQuery methods in our examples are inside a document ready event:

```
$(document).ready(function() {  
    // jQuery methods go here...  
    // here $(document) is selector it use to target Element  
    // and the ready(); is method is use for any work  
});
```

- This is to prevent any jQuery code from running before the document is finished loading (is ready).
- To wait for the document to be fully loaded and ready before working with it.
- This also allows you to have your JavaScript code before the body of your document, in the head section.
- 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
- ```
$(function() {  
    // jQuery methods go here...  
});
```

# jQuery Selector :

- jQuery selectors are one of the most important parts of the jQuery library.
  - jQuery selectors allow you to select and manipulate HTML element(s).
  - jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
  - All selectors in jQuery start with the dollar sign and parentheses: `$()`.
1. Element Selector
  2. Class Selector
  3. id Selector



# 1. Element Selector (Tag Name) :

- The jQuery element selector selects elements based on the element name. You can select all elements on a page like this:

`$("p")`

```
<html><head>
<script src="jquery.min.js"></script>
<script>
$(document).ready(function() {
    $("button").click(function() {
        $("p").hide();
    });
});
</script> </head>
<body>
<h2>This is a heading</h2><p>This is a paragraph. </p>
<button>Click me to hide paragraphs</button>
</body>
</html>
```

## 2. Class Selector :

- The jQuery .class selector finds elements with a specific class.
- To find elements with a specific class, write a period character, followed by the name of the class:

`$(".test")`

```
<html><head>
```

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

```
<script>
```

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

```
    $("button").click(function() {
```

```
        $(".test ").hide();
```

```
    });
```

```
});
```

```
</script> </head>
```

```
<body>
```

```
<h2>This is a heading</h2><p class="test">This is a paragraph. </p>
```

```
<button>Click me to hide paragraphs</button>
```

```
</body>
```

```
</html>
```

# 3. id Selector :

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element: \$("#test")

```
<html><head>
```

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

```
<script>
```

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

```
});
```

```
</script> </head>
```

```
<body>
```

```
<h2>This is a heading</h2><p id="test">This is a paragraph. </p>
```

```
<button>Click me to hide paragraphs</button>
```

```
</body>
```

```
</html>
```

# jQuery Events :

- What are Events? All the different visitors' actions that a web page can respond to are called events. An event represents the precise moment when something happens.
- Examples:
  - moving a mouse over an element
  - selecting a radio button
  - clicking on an element
- The term "fires/fired" is often used with events. Example: "The keypress event is fired, the moment you press a key". Here are some common DOM events:

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouenter	keyup	focus	scroll
mouseleave		blur	unload

- **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!!  
});
```

- **Used jQuery Event Methods :**

- `$(document).ready()`
- The `$(document).ready()` method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

## 1. **click()** :

- The click() method attaches an event handler function to an HTML element.
- The function is executed when the user clicks on the HTML element.
- The following example says: When a click event fires on a element; hide the current element:

- **Example :**

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

## 2. **dblclick()** :

- The dblclick() method attaches an event handler function to an HTML element.
- The function is executed when the user double-clicks on the HTML element:

- **Example :**

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

Presented By : Dhruvita Savaliya

```
<html><head>
<script
  src="jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("p").dblclick(function() {
    $(this).hide();
  });

  $("button").click(function() {
    $("p").show();
  });
});
</script>
</head>
```

```
<body>
<p>If you double-click on me, I
  will disappear.</p>
<button>Click me to show !
  </button>
<p>Click me too!</p>
</body>
</html>
```

### 3. mouseenter()

- The mouseenter() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer enters the HTML element:
- **Example :**

```
$(“h1”).mouseenter(function() {  
    alert("You entered h1!");  
});
```

### 4. mouseleave()

- The mouseleave() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer leaves the HTML element:

#### **Example :**

```
$(“h1”).mouseleave(function() {  
    alert("Bye! You now leave h1!");  
});
```



## 5. mousedown()

- The mousedown() method attaches an event handler function to an HTML element.
- The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

### Example :

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

## 6. mouseup()

- The mouseup() method attaches an event handler function to an HTML element.
- The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

- **Example :**

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

## 7. focus() :

- The **jQuery focus()** is an inbuilt method which is used to focus on an element. The element get focused by the mouse click or by the tab-navigating button.
- **Syntax:**
- `$(selector).focus(function)` Here selector is the selected element.
- **Parameter:** It accepts an optional parameter “function” which specifies the function to run when the focus event occurs.
- **Example :**

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

## 8. blur() :

- The **jQuery blur()** is an inbuilt method that is used to remove focus from the selected element. This method starts the blur event or it can be attached a function to run when a blur event occurs.
- **Syntax:**
- `$(selector).blur(function)` **Parameter:** It accepts an optional parameter “function”.
- **Example :**

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

## 9. keypress() :

1. keydown - The key is on its way down. `i = 0;`
  2. keypress - The key is pressed down. `$(document).ready(function() {`
  3. keyup - The key is released. `$("#input").keypress(function() {`  
`$("#span").text(i += 1);`  
`});`
- The keypress() method triggers the keypress event, or attaches a function to run when a keypress event occurs. `});`
  - The keypress event is similar to the keydown event. The event occurs when a button is pressed down. `</script>`  
`</head>`  
`<body>`
  - However, the keypress event is not fired for all keys (e.g. ALT, CTRL, SHIFT, ESC). Use the keydown() method to also check these keys. `Enter your name: <input type="text">`  
`<p>Keypresses: <span>0</span></p>`  
`</body>`  
`</html>`

`<html>`

`<head>`

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

## 10. **keydown()** :

- The keydown event occurs when a keyboard key is pressed down.
- The keydown() method triggers the keydown event, or attaches a function to run when a keydown event occurs.

## 11. **keyup()** :

- The keyup event occurs when a keyboard key is released.
- The keyup() method triggers the keyup event, or attaches a function to run when a keyup event occurs.

- **Example of keyup ()and keydown() :**

```
$("#input").keydown(function() {  
    $("#input").css("background-color", "yellow");  
});  
$("#input").keyup(function() {  
    $("#input").css("background-color", "pink");  
});
```

## 12. submit() :

- The submit event occurs when a form is submitted.
- This event can only be used on elements.
- The submit() method triggers the submit event, or attaches a function to run when a submit event occurs.
- **Example :**

```
$("#form").submit(function() {  
    alert("Submitted");  
});
```

## 13. change() :

- The change event occurs when the value of an element has been changed (only works on , <input>, <textarea> and <select> elements).
- The change() method triggers the change event, or attaches a function to run when a change event occurs.
- Note: For select menus, the change event occurs when an option is selected. For text fields or text areas, the change event occurs when the field loses focus, after the content has been changed.
- **Example :**

```
$("#input").change(function() {  
    alert("The text has been changed.");    });
```

## 14. load() :

- The load() method attaches an event handler to the load event.
- The load event occurs when a specified element has been loaded.
- This event works with elements associated with a URL (image, script, frame, iframe), and the window object.
- Depending on the browser, the load event may not trigger if the image is cached (Firefox and IE).
- Note: There is also a jQuery AJAX method called load(). Which one is called, depends on the parameters.
- **Example :**

```
$("#img").load(function() {  
    alert("Image loaded.");  
});
```

## 15. `resize()` :

- The `resize` event occurs when the browser window changes size.
- The `resize()` method triggers the `resize` event, or attaches a function to run when a `resize` event occurs.
- **Example :**

```
<script>
```

```
var x = 0;
```

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

```
    $(window).resize(function() {
```

```
        $("span").text(x += 1);
```

```
    });
```

```
});
```

```
</script>
```

```
<body>
```

```
<p>Window resized <span>0</span> times.</p>
```

```
<p>Try resizing your browser window.</p>
```

## 16. scroll() :

- The scroll event occurs when the user scrolls in the specified element.
- The scroll event works for all scrollable elements and the window object (browser window).
- The scroll() method triggers the scroll event, or attaches a function to run when a scroll event occurs.
- **Example :**

```
<script>
```

```
var x = 0;
```

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

```
  $("div").scroll(function() {
```

```
    $("span").text( x+= 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.

```
</div>
```



## 17. unload() :

- The unload event occurs when the user navigates away from the page.
- The unload event is triggered when:
  - a link to leave the page is clicked
  - a new URL is typed in the address bar
  - the forward or back buttons are used
  - the browser window is closed
  - the page is reloaded
  - The unload() method specifies what happens when a unload event occurs.
  - The unload() method should only

be used on the window object.

### • Example :

```
<script>
$(document).ready(function() {
  $(window).unload(function() {
    alert("Goodbye!");
  });
});
</script> </head> <body>

<p>When you click <a
href="https://w3schools.com">
this link</a>, or
close the window, an alert box will
be triggered.</p>
```

# Jquery Effect :

1. hide
2. show
3. fade
4. Slide

## 1-2. Showing and Hiding Elements :

- The commands for showing and hiding elements are pretty much what we would expect:
- `show()` to show the elements in a wrapped set and `hide()` to hide them.

- **Syntax:**

`[selector].show( speed, [callback] );`

`[selector].hide( speed, [callback] );`

- Here is the description of all the parameters:
  1. **speed:** A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
  2. **callback:** This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

- **Example :**

```
<script
  src="jquery.min.js"></script>
<script type="text/javascript"
  language="javascript">
$(document).ready(function() {
$("#show").click(function () {
$(".mydiv").show( 1000 );
});
$("#hide").click(function () {
$(".mydiv").hide( 1000 );
});
});
</script>
<style>
```

```
.mydiv {
  margin:10px;padding:12px;
  border:2px solid #666;
  width:100px;
  height:100px;}
</style>
</head>
<body>
<div class="mydiv"> This is a
  SQUARE </div>
<input id="hide" type="button"
  value="Hide" />
<input id="show" type="button"
  value="Show" />
</body>
</html>
```

### 3. fade effect :

- With jQuery you can fade an element in and out of visibility. jQuery has the following fade methods:

1. `fadeIn()`
2. `fadeOut()`
3. `fadeToggle()`
4. `fadeTo()`

## 1. fadeIn() :

- The jQuery fadeIn() method is used to fade in a hidden element.
- Syntax:  
`$(selector).fadeIn(speed,callback);`
- The optional speed parameter specifies the duration of the effect.
- It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.

- **Example :**

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

```
<script>
```

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

```
{
```

```
$("#btn").click(function () {
```

```
$("#h1").fadeIn(2000);
```

```
});
```

```
});
```

```
</script>
```

```
<body style="text-align:center;">
```

```
<h1
```

```
style="color:red;display:none;">
```

```
HELLO FYBCA
```

```
</h1><br>
```

```
<button id="btn">Fade
```

```
In</button>
```

```
</body>
```

## 2. fadeOut() :

- The jQuery fadeOut() method is used to fade out a visible element.
- **Syntax:**  
\$(selector).fadeOut(speed,callback);
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.
- **Example :**

```
<script  
  src="jquery.min.js"></script>  
<script>  
  $(document).ready(function () {
```

```
    $("#btn").click(function () {  
      $("h1").fadeIn(2000);  
    });  
  });
```

```
</script>
```

```
</head>
```

```
<body style="text-align:center;">
```

```
  <h1
```

```
    style="color:red;display:none;">
```

```
    HELLO FYBCA
```

```
  </h1><br>
```

```
  <button id="btn">Fade
```

```
  Out</button>
```

```
</body>
```

### 3. **fadeToggle()** :

- The jQuery `fadeToggle()` method toggles between the `fadeIn()` and `fadeOut()` methods.
- If the elements are faded out, `fadeToggle()` will fade them in.
- If the elements are faded in, `fadeToggle()` will fade them out.
- Syntax: `$(selector).fadeToggle(speed,callback);`
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.



## 4. fadeTo() :

- The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).
- Syntax: \$(selector).
- fadeTo(speed, opacity, callback);
- The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).
- The optional callback parameter is a function to be executed after the function completes

### 3. jQuery Sliding Methods :

- With jQuery you can create a sliding effect on elements.
- jQuery has the following slide methods:
  1. `slideDown()`
  2. `slideUp()`
  3. `slideToggle()`

## 1. slideDown() :

- The jQuery slideDown() method is used to slide down an element.
- **Syntax:** \$(selector).slideDown(speed,callback);
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

## 2. slideUp()

- The jQuery slideUp() method is used to slide up an element.
- **Syntax:** \$(selector).slideUp(speed,callback);
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

- **3. slideToggle() :**
- The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.
- If the elements have been slid down, slideToggle() will slide them up.
- If the elements have been slid up, slideToggle() will slide them down.
- **Syntax:-** \$(selector).slideToggle(speed,callback);
- The optional speed parameter can take the following values: "slow", "fast", milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes

- **slideDown():**

```
<script src="jquery.min.js"></script>
<script>
    $(document).ready(function() {
        $(".btn").click(function() {
            $("h1").slideDown();
        });
    });
</script>
</head>

<body>
    <h1 style = "display:none" >
        HELLO FYBCA
    </h1>

    <button class="btn">
        Click on button to slide down
    </button>
</body>
```

- **slideUp() :**

```
<script src="jquery.min.js"></script>
<script>
    $(document).ready(function() {
        $(".btn").click(function() {
            $("h1").slideUp();
        });
    });
</script>
</head>

<body>
    <h1>
        HELLO FYBCA
    </h1>
    <button class="btn">
        Click on button to slide UP
    </button>
</body>
```

# jQuery Methods :

1. Css
2. height
3. width
4. innerWidth
5. innerHeight
6. outerWidth
7. outerHeight
8. html
9. text
10. append
11. prepend
12. After
13. before
14. addClass
15. removeClass
16. toggleClass
17. remove
18. empty

## 1.css() :

- The **css() method** in **jQuery** is used to change the style property of the selected element. This method can be used in different ways. The **css()** method can be used to get/return the present value of the property for the selected element.

- **Syntax:**

- `$(selector).css(property)` or

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

- **Example :**

```
<html>
```

```
<head>
```

```
  <script src="jquery.min.js">
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
  <button>
```

Click here and it will change the

color of paragraph element

```
</button>
```

```
<p>
```

We come to gfg!.

```
</p>
```

```
</body>
```

```
<script>
```

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

```
    $("button").click(function () {
```

```
      $("p").css("color",
```

```
      "red");
```

```
    });
```

```
  });
```

```
</script>
```

```
</html>
```

## 2.height() – 3.width() :

- The **height() method** is an inbuilt method in **jQuery** that is used to check the **height and width** of an element but it will not check the padding, border, and margin of the element.
- **Syntax:**
- `$("#Selector").height();`
- `$("#Selector").width();`
- **Parameters:** This function does not accept any parameter.
- **Return value:** It returns the height of the selected element.
- **Example :**

```
<script src="jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $("#button").click(function () {
            var msg = "";
            msg += "height of div: " +
                $("#demo").height();
            $("#demo").html(msg);
        });
    });
```

```
});
</script>
<style>
    #demo {
        height: 100px;
        width: 300px;
        border: 1px solid blue;
        background-color:
            lightgreen;
    }
</style>
```

```
<body>
    <div id="demo"></div>
    <button>Click Me!!!</button>
    <p>
        Click on the button and check the
        height & width of the element(excluding
        padding).
    </p>
</body>
</html>
```



#### 4. **innerHeight()** – 5. **innerWidth()** :

- The **innerHeight** and **innerWidth** is an inbuilt method in jQuery which is used to return the width of the first matched element.

##### **Syntax:**

- `$(selector).innerHeight()`
- `$(selector).innerWidth()`
- Here selector is the selected element.

**Parameters:** It does not accept any parameters.

**Return value:** It returns the width of the selector.

## Example of innerHeight ()& innerWidth() <style>

```
:
<html>
<head>
  <script src="jquery.min.js">
  </script>
  <script>
$(document).ready(function () {
  $("button").click(function () {
    var msg = "";
    msg += "Inner Height of div: " +
    $("#demo").
    innerHeight() + "</br>";
    $("#demo").html(msg);
    msg += "Inner width of div: " +
    $("#demo").
    innerWidth() + "</br>";
    $("#demo").html(msg);
  });
});
</script>
</head>
```

```
#demo {
  height: 150px;
  width: 350px;
  padding: 10px;
  margin: 3px;
  background-color: lightgreen;
}
</style>
<body>
  <div id="demo"></div>
  <button>Click Me!!!</button>
  <p>
    Click on the button and check the
    innerHeight & innerWidth
    of an element(includes padding).
  </p>
</body>
</html>
```

## 6. `outerHeight()` – 7. `outerWidth()` :

- The **`outerHeight()`** and **`outerWidth()`** method in jQuery is used to find the outer height of the specified element. Outer height and Outer width of an element includes padding and border.
- **Syntax:**
- `$(selector).outerHeight(includeMargin)` or
- `$(selector).outerWidth(includeMargin)`
- **Parameters:** This method accepts single parameter **`includeMargin`** which is optional. It contains boolean value and used to specify whether the margin to be included or not. If `includeMargin` is set to true then margin included otherwise margin not included. By default the `includeMargin` set to false.

- **Example :**

```

<html>
<head>
    <script src="../../jquery.min.js"> </script>
    <script>
$(document).ready(function() {
    $("button").click(function() {
        alert("Outer height of div: “
        + $("div").outerHeight());
    });
});
</script>
<style>
    .outer{
        height: 80px;
        width: 200px;
        padding: 5px;
        margin: 5px;
        border: 2px solid black;
        background-color: green;
        text-align: center;
    }
</style>
</head>
<body>
    <div class="outer">
        FYBCA
    </div>
    <button>Click Here to display outer
    height</button>
</body>
</html>

```

## 8. `html()` :

- The **`html()` Method** in **jQuery** is used to set or return the innerHTML content of the selected element.
- **Syntax:**
- It returns the content of the first matched element.
- `$(selector).html()` It sets the content of the matched element.
- `$(selector).html(content)` It sets the content using a function.
- `$(selector).html(function(index, currentcontent))`**Parameters:** This method accepts two parameters as mentioned above and described below:
- **content:** It is a mandatory parameter that specifies the new content for the selected elements.
- **function(index, currentcontent):** It is an optional parameter that specifies a function that returns the new content for the selected element.
  - **index:** It is used to return the index position of the element in the set.
  - **currentcontent:** It is used to return the current HTML content of the selected element.

- **Example :**

```
<html>
<head>
  <title>
    jQuery html() Method
  </title>
  <script src="../../jquery.min.js">
  </script>
</head>

<body style="text-align:center;">
  <h1 style="color:green;">
    hello bca
  </h1>
  <h2>
    Here code<br>using JQuery
    html()
```

```
</h2>
<button>Click</button>

<script>
  $(document).ready(function ()
  {
    $("button").click(function () {
      $("h2").html("Hello
  <u>students!</u>");
    });
  });
</script>
</body>

</html>
```

## 9. text() :

- This method is used to **set** or **return** the text content of the element. While setting the content, it overwrites the content of all the matched elements. The returned content of text method() is used to return the text content of all matched elements.
- **Syntax:**
- **Return text syntax:** \$(selector).text()
- **Set text syntax:** \$(selector).text(content)
- **Set text using a function:** \$(selector).text(function(index, currentcontent))
- **Property Values:**
- **Content Required:** It is used to set the new text content for the element.
- **Function(index, currentcontent):** It is used to specify a function that will return the new text content for the selected elements.
  - **index:** It returns the index position of the element.
  - **currentcontent:** It returns the current content of element.
- **Example :**

```
$("#button").click(function() {  
    $("#p").text("Jquery_text_method");  
});
```

## 10. append() :

- The **append() method** in jQuery is used to insert some content at the end of the selected elements.
- **Syntax:**
- `$(selector).append( content, function(index, html)`  
**)Parameters:** This method accepts two parameters as mentioned above and described below:
  - **content:** It is a required parameter and is used to specify the content which is to be inserted at the end of selected elements. The possible value of contents are HTML elements, jQuery objects, and DOM elements.
  - **function(index, html):** It is an optional parameter and is used to specify the function that will return the content to be inserted.
    - **index:** It is used to return the index position of the element.
    - **html:** It is used to return the current HTML of the selected element.



- **Example :**

```
$(document).ready(function () {  
    $("#btn1").click(function () {  
        $("p").append(" <b>BCA</b>.");  
    });  
  
    $("#btn2").click(function () {  
        $("ol").append("<li>Append FY</li>");  
    });  
});
```

```
<p>Hello</p>  
<p>Jquery</p>  
<ol>  
    <li>FY 1</li>  
    <li>FY 2</li>  
    <li>FY 3</li>  
</ol>  
<button id="btn1">Append to paragraph tag</button>  
<button id="btn2">Append to list</button>
```

## 11. prepend() :

- The prepend() method is an inbuilt method in jQuery which is used to insert a specified content at the beginning of the selected element.
- **Syntax:**
- `$(selector).prepend(content, function)`**Parameters:** This method accept two parameters as mentioned above and described below:
- **content:** It is required parameter which is used to specify the content need to be inserted.
- **function:** It is optional parameter which is used to specify the function to perform after call.
- **Return Value:** This method returns the selected element with the specified changes made by prepend() method.
- **Example :**

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("p").prepend("Welcome to ");  
    });  
});
```

## 12. after() :

- The **after() method** is an inbuilt function in **jQuery** which is used to insert content, specified by the parameter for each selected element in the set of matched elements.
- **Syntax:**
- `$(selector).after(A);`
- **Parameter:** It accepts a parameter “A” which is either a content or function passed to the method.
- **Return Value:** It returns the selected element with the modification.
- **Example :**

```
$(document).ready(function()  
{  
    $("button").click(function() {  
        $("p").after("<b><h2>Hello bca</h2></b>");  
    });  
});
```

### 13. before() :

- The **before() Method** in **jQuery** is used to add the content before the selected element.
- **Syntax:**
- `$(selector).before( content, function(index) )`
- **Parameters:** This method accepts two parameters as mentioned above and described below:
- **content:** This parameter holds the content to be inserted before the element. The possible value of content can be HTML Elements, DOM Elements, jQuery Elements.
- **function(index):** It is an optional parameter and is used to specify a function that returns the content to insert before the element and the index returns the index positioning of the element.
- **Example :**

```
$(document).ready(function()  
{  
    $("button").click(function(){  
        $("p").after("<b><h2>Hello bca</h2></b>");  
    });  
});
```

## 14. addClass() :

- The **addClass()** method is an inbuilt method in **jQuery** which is used to add more properties to each selected element. It can also be used to change the property of the selected element.
- This method can be used in two different ways:
  - 1) **By adding a Class name directly:** Here, the Class name can be used directly with the element which is going to be selected.
- **Syntax:**
- `$(selector).addClass(className);`
- **Parameters:** It accepts the parameter “className” which is the name of the class that is going to be added.
- **Return Value:** It returns the selected elements with the added new class.

- **Example :**

```
<html>
<head>
  <script src="jquery.min.js">
  </script>
  <script>
    $(document).ready(function()
    {
      $("button").click(function() {
        $("p").addClass("add");
      });
    });

  </script>
  <style>
    .add {
```

```
font-size:30px;
color:red;
```

```
    }
  </style>
</head>

<body>
  <p>I would like to say: hello
  </p>
  <p>I would like to say: bca</p>
  <button>click</button>

</body>

</html>
```

## 15. removeClass() :

- The removeClass() method is an inbuilt method in jQuery which is used to remove one or more class names from the selected element.
- **Syntax:**
- `$(selector).removeClass(class_name, function(index, current_class_name))`
- **Parameters:** This function accepts two parameters as mentioned above and described below:
- **class\_name:** It is optional parameter which is used to specify the class name (one or more class) to remove. Multiple class name separated with space.
- **function:** It is optional parameter and it returns one or more class name which need to be removed.
  - **index:** This parameter is used to return index of element.
  - **current\_class\_name:** This parameter returns the class name of selected elements.
- **Return Value:** This method returns the selected element with specified removed class name.

- **Example :**

```
<html>
```

```
<head>
```

```
  <script src="jquery.min.js">
```

```
  </script>
```

```
  <script>
```

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

```
    {
```

```
    $("button").click(function() {
```

```
      $("p").removeClass("add");
```

```
    });
```

```
  });
```

```
</script>
```

```
<style>
```

```
  .add {
```

```
    font-size:30px;
```

```
    color:red;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <p class="add">I would like to  
  say: hello </p>
```

```
  <p>I would like to say: bca</p>
```

```
<button>click</button>
```

```
</body>
```

```
</html>
```



## 16. toggleClass() :

- The toggleClass() method is an inbuilt method in jQuery which is used to toggle or change the class which attached with selected element.
- **Syntax:**
- `$(selector).toggleClass(class, function, switch)`**Parameters:** This method accepts three parameters as mentioned above and described below:
  - **class:** It is the required parameter and used to specify the class name need to replace.
  - **function:** It is optional parameter and used to specify a function that returns one or more class names. This parameter contains the index position of element and class name of element.
  - **switch:** It is optional parameter and used to specify either true or false. By default it is true.
- **Return Value:** This method return the selected element with the specified changes made by toggleClass() method.

- **Example :**

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <script>
```

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

```
        $("button").click(function() {
```

```
          $("div").toggleClass("gfg");
```

```
        });
```

```
      });
```

```
    </script>
```

```
    <style>
```

```
      .gfg {
```

```
        font-size: 25px;
```

```
        background-color: yellow;
```

```
        min-height: 120px;
```

```
      }
```

```
      div {
```

```
        width: 200px;
```

```
        min-height: 120px;
```

```
        background-color: lightgreen;
```

```
        padding: 20px;
```

```
        font-weight: bold;
```

```
        font-size: 20px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <div>
```

```
      <p>Hello!</p>
```

```
      <p>Welcome to Kamani  
      Science.!</p>
```

```
      <button>Click Here!</button>
```

```
    </div>
```

```
  </body>
```

```
</html>
```

## 17. remove() :

- The **remove()** method in **jQuery** used to remove all the selected elements including all the text. This method also remove data and all the events of the selected elements.

### Syntax:

- `$(selector).remove()`
- **Return Value:** It will return all the data of the selected elements deleted.
- **Example :**

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

## 18. empty() :

- The empty() method is an inbuilt method in jQuery which is used to remove all child nodes and its content for the selected elements.
- **Syntax:**
- \$(selector).empty()**Parameter:** This method does not accept any parameter.
- **Return Value:** This method returns the selected element with the specified changes made by empty() method.
- **Example :**

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("div").empty();  
    });  
});
```

```
<div>
```

```
    Hello !
```

```
    <p>Welcome to <span>this div will be empty after click on button</span>.</p>
```

```
</div>
```

```
<p>outside of div</p>
```

```
<button>Click here to empty div..!</button>
```