

Module 4

- JQUERY
 - Syntax,
 - Selectors
 - Events
 - JQuery Effects
 - jQuery hide/show
 - fade
 - slide
 - animate
 - stop
 - delay
 - JQuery HTML-get, set, add and remove,
 - JQuery CSS-before, after, append.

INTRODUCTION TO jQUERY

- jQuery is a small and lightweight JavaScript library.
- jQuery is cross-platform.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.
- It is designed to simplify the client-side scripting of HTML.
- It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.
- The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive.
- It is also used to add animation.

Important features of jQuery

- HTML manipulation
- DOM manipulation
- DOM element selection
- CSS manipulation
- Effects and Animations
- AJAX
- HTML event methods
- Extensibility through plug-ins

Need of jQuery

- It is very fast and extensible.
- It facilitates the users to write UI related function codes in minimum possible lines.
- It improves the performance of an application.
- Browser's compatible web applications can be developed.
- It uses mostly new features of new browsers.
- Out of many JavaScript frameworks, jQuery is the most popular and the most extendable.
- Some of the companies that use jQuery on the web are:
 - Microsoft
 - Google
 - IBM
 - Netflix

History of jQuery

Version No.	Release Date
1.0	26, August, 2006
1.1	14, January, 2007
1.2	10, September, 2007
1.3	14, January, 2009
1.4	14, January, 2010
1.5	31, January, 2011
1.6	3, May, 2011
1.7	3, November, 2011
1.8	9, August, 2012
1.9	15, January, 2013
1.10	24, May, 2013
1.11	24, January, 2014
2.0	18, April, 2013
2.1	24, January, 2014

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()**
 - \$ – It the shorthand for jQuery function.
 - (selector) – It defines the HTML element that you want to select.
 - action() – It is the jQuery method used to perform actions on the elements.

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"

\$(document).ready() and \$()

- The code inserted between `$(document).ready()` is executed only once when page is ready for JavaScript code to execute.
- In place of `$(document).ready()`, you can use shorthand notation `$()` only.

Example:

```
$(document).ready(function() {  
    $("p").css("color", "red");  
});
```

- The above code is equivalent to this code.

```
$(function() {  
    $("p").css("color", "red");  
});
```

There are two main ways to start using jQuery:

- **Include jQuery locally:** Download the jQuery library from jquery.com and include it in your HTML code.

```
<head>  
  <script src="/jquery/jquery-3.4.1.min.js"></script>  
  <script src="js/scripts.js"></script>  
</head>
```

- **Use a CDN:** Link to the jQuery library using a CDN (Content Delivery Network).

```
<head>  
  <script src =  
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
  <script src="js/scripts.js"></script>  
</head>
```


jQuery Selectors

- 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 from a DOM.
- 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: `$()`
- It is known as the factory function. It uses the three basic building blocks while selecting an element in a given document.

S.No.	Selector	Description
1)	Tag Name:	It represents a tag name available in the DOM. For example: <code>\$('p')</code> selects all paragraphs 'p' in the document.
2)	Tag ID:	It represents a tag available with a specific ID in the DOM. For example: <code>\$('#real-id')</code> selects a specific element in the document that has an ID of real-id.
3)	Tag Class:	It represents a tag available with a specific class in the DOM. For example: <code>\$('real-class')</code> selects all elements in the document that have a class of real-class.

How to use selectors

- The jQuery selectors can be used single or with the combination of other selectors. They are required at every step while using jQuery.
- They are used to select the exact element that you want from your HTML document.

S.No.	Selector	Description
1)	Name:	It selects all elements that match with the given element name.
2)	#ID:	It selects a single element that matches with the given id.
3)	.Class:	It selects all elements that matches with the given class.
4)	Universal(*)	It selects all elements available in a DOM.
5)	Multiple Elements A,B,C	It selects the combined results of all the specified selectors A,B and C.

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
 - In the below code, when the user clicks on a button, all <p> elements will be hidden

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

- Selects all elements of a specified type.

`$('p')` // Selects all `<p>` elements

- Class Selector: Selects all elements with a specific class.

`$('.className')` // Selects all elements with class 'className'

- ID Selector: Selects an element with a specific ID.

`$('#Id')` // Selects the element with ID 'Id'

- Attribute Selector: Selects elements with a specific attribute.

`$('input[type="text"]')` // Selects all `<input>` elements with type 'text'

The #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")`
- Example: When a user clicks on a button, the element with id="test" will be hidden:

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

The .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")`
- Example:
- To find elements with a specific class, write a period character, followed by the name of the class:

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

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code><p></code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code><p></code> element
<code>\$("ul li:first")</code>	Selects the first <code></code> element of the first <code></code>
<code>\$("ul li:first-child")</code>	Selects the first <code></code> element of every <code></code>
<code>\$("[href]")</code>	Selects all elements with an href attribute
<code>\$("a[target='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value equal to <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code>
<code>\$("tr:even")</code>	Selects all even <code><tr></code> elements
<code>\$("tr:odd")</code>	Selects all odd <code><tr></code> elements

jQuery Events

- jQuery is tailor-made to respond to events in an HTML page.
- *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".

- Following are some of the DOM events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	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!!  
});
```

Commonly Used jQuery Event Methods

- **`$(document).ready()`** : This method allows us to execute a function when the document is fully loaded
- **`click()`**: This 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 `<p>` element; hide the current `<p>` element

- **Example**

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

- **Example**

```
$('#button').click(function() {  
    alert('Button Clicked!');  
});
```

- **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

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

mouseenter(): The function is executed when the mouse pointer enters the HTML element

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

- **mouseleave()**: The function is executed when the mouse pointer leaves the HTML element.

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

mousedown(): The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element.

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

- **mouseup():** The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element.

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

- **focus():** The function is executed when the form field gets focus.

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

blur(): The function is executed when the form field loses focus

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

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

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

- Attach multiple event handlers to a <p> element

```
$("#p").on({  
    mouseenter: function(){  
        $(this).css("background-color", "lightgray");  
    },  
    mouseleave: function(){  
        $(this).css("background-color", "lightblue");  
    },  
    click: function(){  
        $(this).css("background-color", "yellow");  
    }  
});
```

- **hide()**: to hide a particular element
- **show()**: to show a particular element

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

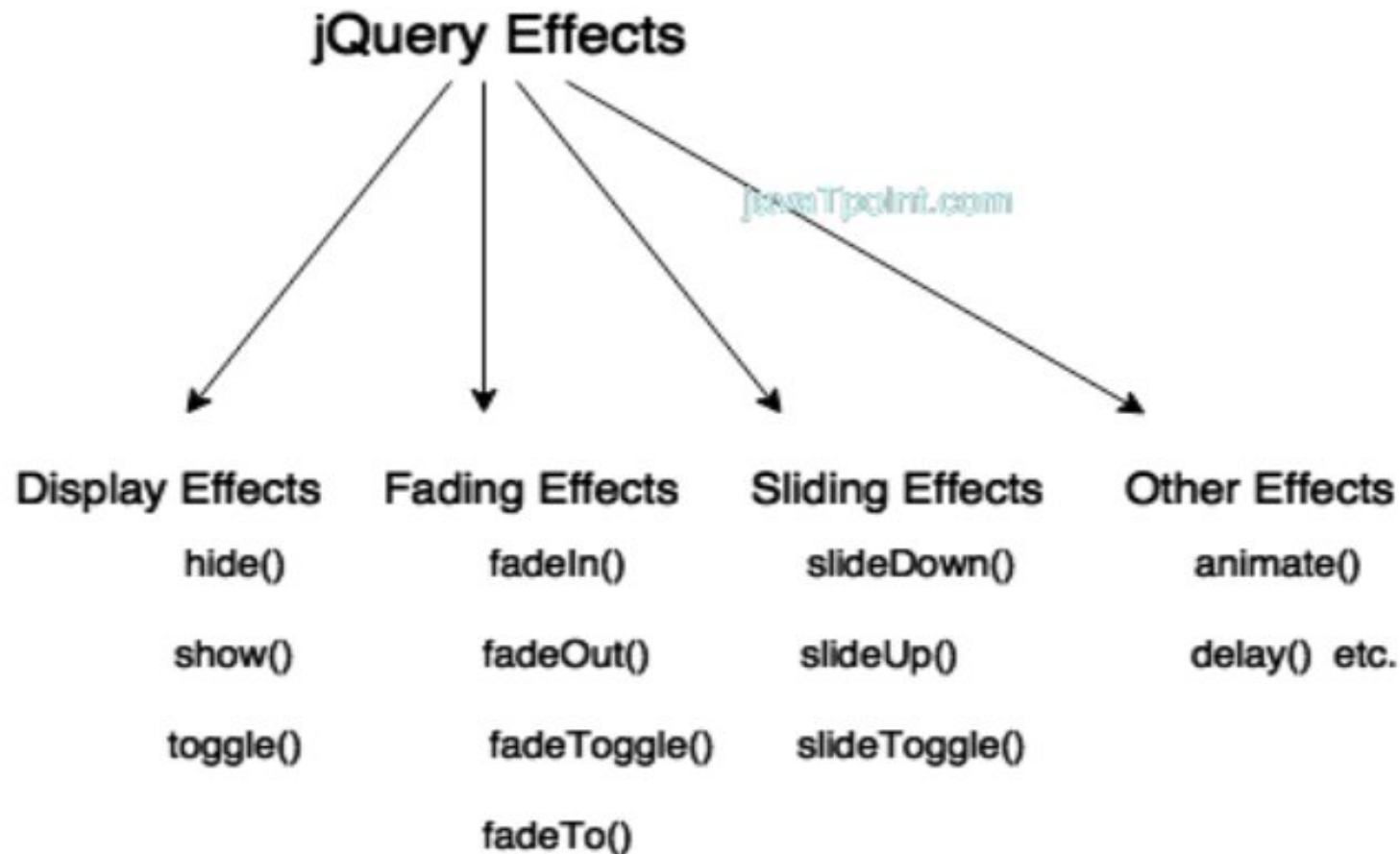
```
$("#show").click(function(){  
    $("#p").show();  
});
```

To Write jQuery in separate file and include it

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"> </script>  
<script src="my_jquery_functions.js"></script>  
</head>
```

jQuery Effects

- jQuery enables us to add effects on a web page.
- jQuery effects can be categorized into fading, sliding, hiding/showing and animation effects.



jQuery methods for effects on webpage

No.	Method	Description
1)	<code>animate()</code>	performs animation.
2	<code>clearQueue()</code>	It is used to remove all remaining queued functions from the selected elements.
3)	<code>delay()</code>	sets delay execution for all the queued functions on the selected elements.
4	<code>dequeue()</code>	It is used to remove the next function from the queue, and then execute the function.
5)	<code>fadein()</code>	shows the matched elements by fading it to opaque. In other words, it fades in the selected elements.

6)	<code>fadeOut()</code>	shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.
7)	<code>hide()</code>	hides the matched or selected elements.
8)	<code>queue()</code>	shows or manipulates the queue of methods i.e. to be executed on the selected elements.
9)	<code>show()</code>	displays or shows the selected elements.
10)	<code>slideDown()</code>	shows the matched elements with slide.

jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

- fadeIn() method

`$(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.
- ```
$("#button").click(function(){
 $("#div1").fadeIn();
 $("#div2").fadeIn("slow");
 $("#div3").fadeIn(3000);
});
```

- fadeout() method:
- `$("#button").click(function(){  
 $("#div1").fadeOut();  
 $("#div2").fadeOut("slow");  
 $("#div3").fadeOut(3000);  
});`
- fadeToggle() method:
- `$("#button").click(function(){  
 $("#div1").fadeToggle();  
 $("#div2").fadeToggle("slow");  
 $("#div3").fadeToggle(3000);  
});`

# jQuery HTML

- jQuery `html()` method is used to change the entire content of the selected elements. It replaces the selected element content with new contents.
- Note: It is a very useful function but works in a limited area because of its API documentation. The API documentation of the jQuery `html` function consists of three method signatures.
- The first method signature has no argument, so it just returns the HTML within that element. The remaining two signatures take a single argument: i.e. a string or a function that returns a string.

## Syntax:

- `$(selector).html()`
  - It is used to return content.
- `$(selector).html(content)`
  - It is used to set content.
- `$(selector).html(function (index, currentcontent))`
  - It is used to set content by calling function.

Note: The jQuery `html()` method is used either to set the content or return the content of the selected elements.

- **To set content:** When you use this method to set content, it overwrites the content of the all matched elements.
- **To return content:** When you use this method to return content, it returns the content of the first matched element.

The `text()` method is used to set or return only the text content of the selected elements.

# Parameters of jQuery html() method

| Parameter                        | Description                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content                          | It is an essential parameter. It is used to specify the new content for the selected elements. It can also contain HTML tags.                                                                                                                                                                                                          |
| Function (index, currentcontent) | <p>It is an optional parameter. It specifies a function that returns the new content for the selected elements.</p> <ul style="list-style-type: none"><li>○ <b>Index:</b> It shows the index position of the element in the set.</li><li>○ <b>Currentcontent:</b> It shows the current HTML content of the selected element.</li></ul> |

- jQuery text():
- The jQuery text() method is used to set or return the text content of the selected elements.
- **To return content:** When this method is used to return content, it returns the combined text content of all matched elements without the HTML markup.
- **To set content:** When this method is used to set content, it overwrites the content of all matched elements.
- The jQuery text() method is used to set or return html content without HTML markup while, html() method is used to set or return the innerHtml (text + HTML markup).
- The jQuery text() method can be used in both XML and HTML document while jQuery html() method can't.



# jQuery before()

- The jQuery before() method is used to insert the specified content before the selected elements. It adds the content specified by the parameter, before each element in the set of matched elements.
- **Syntax:**
- `$(selector).before(content, function(index))`

| Parameter        | Description                                                                                                                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content          | It is a mandatory parameter. It specifies the content to insert. Its possible values are: <ul style="list-style-type: none"><li>◦ HTML elements</li><li>◦ jQuery objects</li><li>◦ DOM elements</li></ul> |
| Function (index) | It specifies a function that returns the content which is used to insert. <ul style="list-style-type: none"><li>◦ <b>Index:</b> It provides the index position of the element in the set.</li></ul>       |

# jQuery after() method

- The jQuery after() method is used to insert specified content after the selected element. It is just like jQuery append() method.
- If you want to insert content before the selected element, you should use jQuery before() method.

| Parameter        | Description                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content          | <p>It is a mandatory parameter. It specifies the content to insert. Its possible values are:</p> <ul style="list-style-type: none"><li>◦ HTML elements</li><li>◦ jQuery objects</li><li>◦ DOM elements</li></ul> |
| Function (index) | <p>It specifies a function that returns the content which is used to insert.</p> <ul style="list-style-type: none"><li>◦ <b>index:</b> It provides the index position of the element in the set.</li></ul>       |

# jQuery DOM Manipulation

- jQuery provides various methods to add, edit or delete DOM element(s) in the HTML page.
- The following table lists some important methods to add/remove new DOM elements.

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

# jQuery css() method

- The css() method sets or returns one or more style properties for the selected elements.
- To return the value of a specified CSS property, use the syntax:
- `css("propertyname");`
- Ex: `$("#p").css("background-color");`
  - It returns the background-color value of the First matched element.
- To set a specified CSS property, use the following syntax:
- `css("propertyname","value");`
- Ex: `$("#p").css("background-color","yellow");`
- To set multiple properties, use the following syntax:
- `Css({"propertyname":"value","propertyname":"value",....});`