

INTRODUCTION TO JQUERY:

- jQuery is a Client-side javascript library Created by John Resig in the year 2006.

▪ Definition:

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

Designed to simplify - HTML DOM traversal & manipulation, Event handling, CSS animation and AJAX.

- It is free, open source software.
- JQuery is a scripting language. Unlike traditional programming languages, it is interpreted, not executed.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.
- It also provides capabilities for developers to create plug-ins on top of the JavaScript library.

▪ Advantages of jQuery:

1. Simple and easy to use:

- jQuery library is built using simpler and shorter codes.
- It consists of a large number of predefined methods which can be directly used in our applications.
- With its open coding standards and simple syntax, web designers can shorten the time that it takes to deploy a site or application.

2. Compact and light weight library about 19KB in size.

3. Open source library:

- jQuery is an open source library that is free and supported well across different applications.

- This means that anyone can use this language in their applications without worrying about any licensing or compatibility issues.

4. Separates JavaScript and HTML:

- Instead of using HTML attributes to call JavaScript functions for event handling, jQuery can be used to handle events purely in JavaScript. Thus, the HTML tags and JavaScript can be completely separated.

5. Cross-browser compatibility:

- JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another.
- jQuery handles all these cross browser inconsistencies and provides a consistent interface that works across different browsers.

6. AJAX support:

- Enables a web page to make AJAX requests to a web server to add the data, without reloading the page.

7. Event handling:

- jQuery is tailor-made to respond to events in an HTML page. In jQuery, most DOM events have an equivalent jQuery method to handle them.

8. Custom animations and effects:

- jQuery provides a lot of built in methods to add effects like fading and sliding of elements.
- It also allows developer to add custom animations to web pages.

9. HTML/DOM manipulation:

- The DOM is a tree structure representation of all the elements of a webpage.
- The jQuery made it easy to select DOM elements, traverse them and modifying their content.
- jQuery methods like `html()`, `text()`, `val()` and `attr()` can be used for this purpose.

10. Extensibility:

- jQuery makes extending the framework very simple. New events, elements and methods can be easily added and then reused as plug-in.

11. Brevity and clarity:

jQuery promotes brevity and clarity with features like chainable functions and shorthand function name.

jQuery Syntax:

The jQuery syntax is used for selecting HTML elements and performing some action on the element(s).

Basic syntax is:

1. `$(selector).action()`
2. `$(selector).action(function(){
});`

- 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:

- `$("p").hide()` - hides all <p> elements.
- `$(this).hide()` - hides the current element.

The Document Ready Event:

➤ To prevent any jQuery code from running before the document is finished loading all

jQuery methods are written inside a document ready event:

➤ **Syntax:**

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

EVENTS:

- jQuery is Event-driven – “respond to events in an HTML page” .
- “ An event represents the precise moment when something happens” .
- A program contains necessary block of code known as “ event handler” , to handle an event.

• **Example:**

- Clicking of a mouse
- Loading of a web page
- Pressing a key on a keyboard
- Submitting a form

Event	Description
blur	Occurs when an element loses focus
change	Occurs when the value of an element has been changed
click	Occurs when an element is clicked
dblclick	Occurs when an element is double-clicked
focus	Occurs when an element gets focus
hover	Triggered when the mouse pointer hovers over the selected elements
keydown	Occurs when a keyboard key is pressed down
keypress	Occurs when a keyboard key is pressed down
keyup	Occurs when a keyboard key is released
load	Occurs when a specified element has been loaded
mousedown	Occurs when the left mouse button is pressed down over the selected element
mouseenter	Mouse pointer enters the selected element
mouseleave	Mouse pointer leaves the selected element
mouseout	Mouse pointer leaves the selected element
mousemove	Mouse pointer moves within the selected element
mouseover	Mouse pointer is over the selected element

Event	Description
mouseup	Left mouse button is released over the selected element
ready	Occurs when the DOM (Document Object Model) has been loaded
resize	Occurs when the browser window changes size
scroll	Occurs when the user scrolls in the specified element
select	Occurs when a text is selected in a text area or a text field
submit	Occurs when a form is submitted
unload	Occurs when the user navigates away from the page

Example:

```

<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
        $(document).ready(function()
        {
            $("#p1").click(function()
            {
                alert("You clicked on the paragraph");
            });
            $("#p2").dblclick(function()
            {
                alert("You double clicked on the paragraph");
            });
            $("#p3").hover(function()
            {
                alert("mouse moved over the paragraph");
            });
        });
    </script>
</head>
<body>
<p id="p1">Click on this paragraph</p>

```

```
<br>
<p id="p2">Double Click on this paragraph</p>
<br>
<p id="p3">Move cursor over this paragraph</p>
</body>
</html>
```

Keyboard events:

List of keyboard events:

1. keydown
2. keypress
3. keyup

Example:

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
        $(document).ready(function()
        {
            $("input").keydown(function()
            {
                $(this).css("background-color","yellow");
            });
            $("input").keyup(function()
            {
                $(this).css("background-color","pink");
            });
        });
    </script>
</head>
<body>
<form method="post">Enter name:<input type="text" name="t1"><br>
</form>
```

```
</body>
</html>
```

Mouse events:

List of Mouse events:

1. mousedown
2. mouseenter
3. mouseleave
4. mousemove
5. mouseout
6. mouseover
7. mouseup

Example:

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function()
{
$("#i1").mouseenter(function()
{
$("#i1").css("background-color","yellow");
});
$("#i1").mouseout(function()
{
$("#i1").css("background-color","pink");
});
$("#i2").mousedown(function()
{
$("#i2").css("background-color","blue");
});
$("#i2").mouseup(function()
```

```

        {
        $("#i2").css("background-color","green");
        });
        $("#i3").mouseover(function()
        {
        alert("cursor is over this heading");
        });
        });
    </script>
</head>
<body>
<h1 id="i1">Mouse Cursor over this heading 1 to trigger mouse events</h1>
<h1 id="i2">Mouse Cursor over this heading 2 to trigger mouse events</h1>
<h1 id="i3">Mouse Cursor over this heading 3 to trigger mouse events</h1>
</body>
</html>

```

Form events:

List of form events:

1. Submit
2. Change
3. Focus
4. Blur

Example:

```

<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function()
{
$("#form").submit(function()
{
alert("Form is submitted");

```



```
});  
$("input").focus(function()  
{  
$(this).css("background-color","yellow");  
});  
$("input").blur(function()  
{  
$(this).css("background-color","pink");  
});  
$("input").change(function()  
{  
alert("Text is changed");  
});  
$("input").select(function()  
{  
alert("Text is selected");  
});  
});  
</script>
```

</head>

<body>

<form method="post">

Enter name:<input type="text" name="t1">

Enter password:<input type="password" name="t2">

<button type="submit">Submit</button>

</form>

</body>

</html>

Document/Window Events:

List of Document/Window Events:

1. Load
2. Resize
3. Scroll
4. Unload

Example:

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $(window).resize(function(){
            alert("Window is resized");
        });
    });
</script>
</head>
<body>
<p>Resize Window</p>
</body>
</html>
```

EFFECTS:

- The jQuery library provides several techniques for adding animation to a web page.
- It contains various methods to apply simple, standard animations that are frequently used, and also sophisticated custom effects.

Category	Method	Description
Hide/Show	hide()	Hides the selected elements
	show()	Shows the selected elements
	toggle()	Toggles between the hide() and show() methods

Category	Method	Description
Fading	fadeIn()	Fades in the selected elements
	fadeOut()	Fades out the selected elements
	fadeTo()	Fades in/out the selected elements to a given opacity
	fadeToggle()	Toggles between the fadeIn() and fadeOut() methods
Sliding	slideUp()	Slides-up (hides) the selected elements
	slideDown()	Slides-down (shows) the selected elements
	slideToggle()	Toggles between the slideUp() and slideDown() methods
Animation	animate()	Runs a custom animation on the selected elements
	stop()	Stops the currently running animation for the selected elements
	delay()	Sets a delay for all queued functions on the selected elements
	Example: <code>\$("#h1").delay("slow").fadeIn();</code>	

Showing and Hiding of elements:

- **hide()** - hide() method hides the selected elements.

- **Syntax:**

`$(selector).hide(speed,callback);`

speed - Specifies the speed of the hide/show effect.

Possible values: milliseconds, "slow", "fast".

callback - A function to be executed after the method is completed.

- **show()** - shows the hidden, selected elements.

- **Syntax:**

`$(selector).show(speed,callback);`

- **toggle()** - toggles between hide() and show() for the selected elements.

- **show()** is run if an element is hidden.

- **hide()** is run if an element is visible

- **Syntax:**

`$(selector).toggle(speed,callback);`

Example:

```
<html>
<head>
<scriptsrc="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
        $(document).ready(function(){
            $("#b1").click(function(){
                $("img").hide(1000);
            });
            $("#b2").click(function(){
                $("img").show("slow");
            });
            $("#b3").click(function(){
                $("img").toggle("fast");
            });
        });
    </script>
</head>
<body>

<button id="b1"> Hide </button>
<button id="b2"> Show </button>
<button id="b3"> Toggle </button>
</body>
</html>
```

Fading effects:

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

- **fadeIn()** - used to fade in a hidden element.

▪ Syntax:

```
$(selector).fadeIn(speed,callback);
```

- **fadeOut()** - used to fade out a visible element.

- **Syntax:**

`$(selector).fadeOut(speed,callback);`

- **fadeToggle()** – toggles between the fadeIn() and fadeOut() methods.

- If the elements are fadedout, fadeToggle() will fade them in.
- If the elements are fadedin, fadeToggle() will fade them out.

- **Syntax:**

`$(selector).fadeToggle(speed,callback);`

- **fadeTo()** - allows fading to a given opacity (value between 0 and 1).

- **Syntax:**

`$(selector).fadeTo(speed,opacity,callback);`

- **Opacity:** Specifies the opacity to fade to. Must be a number between 0.00 and 1.00.

Example:

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#b1").click(function(){
$("#img").fadeOut(1000);
});
$("#b2").click(function(){
$("#img").fadeIn("slow");
});
$("#b3").click(function(){
$("#img").fadeToggle("fast");
});
$("#b4").click(function(){
$("#img").fadeTo("slow",0.3);
});
});
```

```

        </script>

    </head>

    <body>

        <br><br>

        <button id="b1"> Fade Out </button>

        <button id="b2"> Fade In </button>

        <button id="b3"> Fade Toggle </button>

        <button id="b4"> Fade To </button>

    </body>

</html>

```

Sliding effects:

With jQuery you can create a sliding effect on elements. jQuery slide methods slide elements up and down.

- **slideUp()** – used to slide up an element.

- **Syntax:**

```
$(selector).slideUp(speed,callback);
```

- **slideDown()** - used to slide down an element.

- **Syntax:**

```
$(selector).slideDown(speed,callback);
```

- **slideToggle()** – toggles between the slideDown() and slideUp() methods.

- If the elements have been slidedown, slideToggle() will slide them up.
- If the elements have been slideup, slideToggle() will slide them down.

- **Syntax:**

```
$(selector).slideToggle(speed,callback);
```

Program for sliding effects:

```

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("#b1").click(function(){

        $("h1").slideUp(1000);
    });
});

```

```

    });
    $("#b2").click(function(){
    $("h1").slideDown("slow");
    });
    $("#b3").click(function(){
    $("h1").slideToggle("fast");
    });
    });
</script>

</head>
<body>

<h1 style="background-color:orange">Click on the button to see sliding effect</h1>

<button id="b1"> Slide up </button>
<button id="b2"> Slide down </button>
<button id="b3"> Slide Toggle </button>

</body>
</html>

```

Animate() and stop():

- **animate()**

- animate() method is used to create custom animations.
- multiple (CSS) properties can be animated at the same time using animate()
- changes an element from one state to another with CSS styles.

- **Syntax:**

```
$(selector).animate({params},speed,callback);
```

- **params** - required parameter defines the CSS properties to be animated.

- **Stop() –**

- The jQuery stop() method is used to stop an animation or effect before it is finished.
- works for all jQuery effect functions, including sliding, fading and custom animations.

- **Syntax:**

```
$(selector).stop();
```

- kills the current animation being performed on the selected element.

- **Program for animate and stop:**

```
<html>
<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
        $(document).ready(function(){
            $("#b1").click(function(){
                $("div").animate({left:'850px', height:'+=250',width:'+=250px'},1000);
            });
            $("#b2").click(function(){
                $("div").stop();
            });
        });
    </script>
</head>
<body>
<button id="b1">Start Animation</button>
<button id="b2">Stop Animation</button>
<br><br>
<div style="background-color:blue;width:300px;height:300px;position:absolute"></div>
</body>
</html>
```

****Thank You****

And I'll keep on updating this file as per the requirements of the organizations