

# JQuery

- \* JQuery is JS library
- \* JQuery is developed completely based on JS
- \* JQuery is open source JS library
- \* JQuery is not replacement of JS. It simplifies the JS task
- \* JQuery is a cross platform JS library designed to simplify the client-side scripting of HTML
- \* JQuery is used to overcome the load on server. Many task can be performed in client side using JQuery.
- \* Its punch line is "Write less do more..."

Inventor of JQuery - John Resig  
Created in 2006

## Features of JS

- \* Simple
- \* DOM manipulation
- \* DOM Traversing
- \* Event Handling
- \* Styles Modifications      Extensible Markup language ↑
- \* Effects and Animation
- \* AJAX Support (Asynchronous JS & XML)
- \* JQuery - UI (User Interface)
- \* CSS3 Compliant
- \* Light Weight Library
- \* Cross Browser Support
- \* Interpreted

Ex- `$(document).ready(function(){  
 alert("Welcome to JQuery.");  
});`

Explanation of above code -

① \$ -

- \* \$ is jquery function. It is used to begin the coding in jquery.
- \* \$ function is used for DOM traversing.
- \* \$ function is used to select an object from \$.
- \* \$ is short form of "jquery()" function.

② \$(document) -

- \* document is a built-in object of DOM (whole webpage)
- \* \$(document) means selecting object of whole page

③ ready() -

- \* ready is an event of jquery.
- \* This event is used to run some code after page load / page ready
- \* ready() event is equivalent to "onload" event of JS

Note - In jQuery, when we use an event then we have to specify event handler function to perform that events. For event handlers we can use named or unnamed function both.

Ex - `function demo()`

```
        {  
            alert("Welcome to our page");  
        }  
$(document).ready(functiondemo());
```

DOM Accessing in jQuery:-

- A We can access the object of any tag from webpage by using jQuery.
- A To access/select an object we have to use "`$()`" junction.

Syntax - `$( "Selector" )`

Note :- We have main 3 types of selectors-  
Id Selector (#), class Selector (.) and  
tag based selector (without any symbol).

`$( )` function returns the object of associated tag

## DOM manipulation in jQuery -

We can read and modify (write) the content of any HTML tag by using jQuery.

To perform DOM manipulation JQuery provides below methods -

- 1- `val()` → value property in JS
- 2- `text()` → innerText property in JS
- 3- `html()` → innerHTML property in JS
- 4- `prop()`
- 5- `attr()`

### Syntax of val() / text() / html()

#### ① To get

`var name = Object of tag.FunctionName();`

Ex -

`var name = $("#txtname").val();`

`var para = $("#mypara").text();`

ANSWER

#### ② To Set -

`Object of tag.FunctionName(value/variable);`

Ex -

`$("#txtname").val("Arijit Singh");`

`$("#mydv").text("My Name is Raj");`

`$("#mydu").html("My Name is Arijit");`

④ fprop() ⇒ This function is used to get/set the property of a tag

⇒ This property junction get/set the property in real format (not in string format).

⇒ We can use this junction to work with checked, disabled, readonly property.

Syntax to get:

varname = Object-of-tag.fprop("Property Name");

Ex- var x = \$\("#btn").fprop("disabled");

Syntax to set:

Object of tag.fprop("prop-name", "prop-value");

Ex- \$\("Hchf").fprop("checked", true);

⑤ attr() ⇒

① This function is used to get/set the attributes of HTML tags

② We can use this function to read/write src, title (of img tag), rows, cols (textarea), type (of input tag) etc.

③ This function reads or write data in string format (not real format).

Its syntax is same as "fprop()", "junction"

## Managing Styles in jQuery

- ① We can read and manage CSS properties by using jQuery
- ② jQuery provides below some methods-

- ① css()
- ② addClass()
- ③ removeClass()
- ④ toggleClass()

### ① css()

This function is used to get/set the CSS property

Syntax - to get :-

var name = Obj. of tag.css("property\_name");

Ex -

var col = \$("#dv").css("color");

Syntax to set :-

obj. of tag.css("property\_name", "value");

obj. of tag.css({ "property": "value", "property": "value", ... })

for  
multiple  
CSS

\$("#dv").css("background-color", "blue");

Note: `addClass()` function is used to add/affix a CSS class in selected object during run time.

`removeClass()` function is used to remove the specific CSS class from selected tag.

`toggleClass` function creates toggle effect (add/remove).

### Syntax of `addClass()`:

Object of tag `addClass("Name of CSS class")`,  
Ex. `$("#txt").addClass("form-control");`

Syntax of `toggleClass()` & `removeClass()` is same as above.

### Some of DOM Events

<code>click</code>	Mouse Events	Keyboard Events
<code>dblclick</code>		
<code>mouseenter</code>	<code>keypress</code>	
<code>mouseleave</code>	<code>keydown</code>	
	<code>keyup</code>	

<code>submit</code>	Form Events	Document Events
<code>change</code>		
<code>focus</code>	<code>load</code>	
<code>blur</code>	<code>resize</code>	
	<code>scroll</code>	
	<code>unload</code>	

Some others are `hide()`, `show()`, `toggle()`, `fadeIn()`, `fadeOut()`, `fadeToggle()`.

slide Down() - used to slide down an element  
slide Up() - used to slide up an element  
slideToggle() - used to slide up/down both.  
animate({fx:frames}, speed, callback);  
stop() - used to stop an animation

### jQuery Chaining

With jQuery, you can chain together actions/methods.

Chaining allows us to run multiple jQuery methods within a single statement.

Ex- `$("#pi").css("color", "red").slideUp(2000).slideDown(2000);`

### Add New HTML Content

append() - Inserts content at the end of selected items.

prepend() - Inserts content at the beginning of the selected items.

after() - Inserts content after the selected elements (generally used withing)

before() - Inserts content before the selected items

## Remove Elements/Contents

remove() - Removes the selected element (and its child elements)

empty() - Removes the child elements from the selected element.

## jQuery Dimensions Methods

width()

height()

innerWidth()

innerHeight()

outerWidth()

outerHeight()

## Working with collection in jQuery

We can perform get/set operations in collection by using jQuery.

each() function is used to work with collections.

Syntax -

```
CollectionObject.each(function(), {
    // statements
});
```

Note - We can access current object from collection inside each function by using `$(this).each()` function works like a loop.

## Effects and Animations using jquery

jQuery provides features to create effect and animations.

To implement effect and animation, we have to use below functions -

- |   |                |    |              |
|---|----------------|----|--------------|
| 1 | show()         | 7  | fadeIn()     |
| 2 | hide()         | 8  | fadeOut()    |
| 3 | toggle()       | 9  | fadeToggle() |
| 4 | slide Down()   | 10 | animate()    |
| 5 | slide Up()     |    |              |
| 6 | slide Toggle() |    |              |



⇒ We have to pass duration of effect/animation inside above functions. We can pass this duration in milliseconds or in string format.

⇒ Duration of string format can be -

slow

normal

fast

⇒ Duration in milliseconds can be 1000, 1800, 4000 etc

Syntax of first 9 functions-

obj- of - `fg. functionName(duration);`

Ex- `$ ("#dv"). fadeIn ("slow");`  
`$ ("#bth"). hide(2300);`

### ⑩ animate()

This function is used to create custom animation

Syntax-

object of `fg.animate(Effects, duration);`

Ex-

`$ ("#im"). animate({width: 200px}, 3000);`