

JQUERY

CHEAT SHEET

[jQuery Introduction](#)

[Linking jQuery](#)

[Select elements](#)

[Manipulate styles & add classes to elements](#)

[Ready\(\) function](#)

[Change text or html of tag](#)

[Manipulate attributes of element](#)

[Event handling](#)

[Add & remove elements](#)

[Add animation](#)

[Add Custom animation](#)

[AJAX Request](#)

[AJAX Error handling](#)

JQUERY - INTRODUCTION

- jQuery is a JavaScript library designed to simplify the code syntax for DOM tree traversal and manipulation, event handling, CSS animation, and Ajax.

Selecting elements

```
document.querySelector("h1")
```



```
jQuery("h1")
```



```
$( "h1" )
```

Changing Styles

```
document.querySelector("h1").  
style.color = "red";
```



```
$( "h1" ).css( "color", "red" );
```

Adding event listener

```
document.addEventListener(  
"keypress", function(){  
alert("keypress")  
})
```



```
$(document).keypress(function(){  
alert(" keypress ")  
})
```

LINKING JQUERY

Add the following code snippet before the end of the body tag →

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">  
</script>
```

SELECT ELEMENTS

`$(), jQuery()`

- You can use `$()` or `jQuery()` to select elements in jquery
- In the `()` you can specify the selector

// Selects the h1

```
$("h1")
```

// Selects all the buttons

```
$("button")
```

// Selects the entire document

```
$(document)
```

// Selects the ol inside of li

```
$("ol li");
```

MANIPULATE STYLES & ADD CLASSES TO ELEMENTS

css("property", "value")

- We can change styles in jQuery using the `css()` method.

// change color of h1 to red

```
$("h1").css("color", "red");
```

// change color of all buttons to red

```
$("button").css("color", "red");
```

// show the font-size value of h1

```
$("h1").css("font-size");
```

addClass("classes")

- It adds specified classes to elements class list

// Add big-title class to h1

```
$("h1").addClass("big-title");
```

// Add multiple classes to h1

```
$("h1").addClass("big-title margin-50");
```

removeClass("classes")

- It removes specified classes from elements class list

// Remove the big-title class from h1

```
$("h1").removeClass("big-title");
```

toggleClass("classes")

- It will add or remove class

// Toggle the big-tittle class from h1

```
$("h1").toggleClass("big-title");
```

hasClass("classes")

- It checks if a element has the specified class

// Check if h1 has the margin-50 class

```
$("h1").hasClass("margin-50")
```

READY() FUNCTION

ready()

- We can use the ready function to execute the jquery code only once the document is fully loaded
- It's a Callback function

// Change the color of buttons only once the document is fully loaded

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

CHANGE TEXT OR HTML OF TAG

text()

- We can change text of tag in jQuery using the `text()` method.

// Change text of h1 to bye

```
$("h1").text("Bye")
```

html()

- We can change inner html of tag in jQuery using the `html()` method.

// Change inner html of button to Hey

```
$("button").html("<em>Hey</em>")
```

MANIPULATE ATTRIBUTES OF ELEMENT

attr()

- We can use the attr() function to change the attribute value of a element in jQuery

// Shows the value of src attribute of image

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

// Change the value of href attribute of anchor tag

```
$("a").attr("href", "https://www.bing.com/")
```

EVENT HANDLING

event(function())

- We can use the event() method to handle events
- Here event refers to event name like click, keyup, mouseover, hover etc.

// Change the color of h1 to purple when clicked on h1

```
$("h1").click(function () {  
    $("h1").css("color", "purple");  
});
```

// Change the text of h1 to key value when any key pressed

```
$(document).keypress(function (event) {  
    $("h1").text(event.key);  
});
```

on("event", function())

- We can use the following method to handle events
- Here event refers to event name like click, keypress, mouseover etc.

// Change the color of h1 to purple when mouse overed on h1

```
$("h1").on("mouseover", function () {  
    $("h1").css("color", "purple");  
});
```

ADD & REMOVE ELEMENTS

before()

- The following method will add the specified element before the specified selector opening tag

// Add a button before the h1 element tag

```
$( "h1" ).before( "<button>New</button>" );
```

after()

- The following method will add the specified element after the specified selector closing tag

// Add a button after the h1 element tag

```
$( "h1" ).after( "<button>New</button>" );
```

prepend()

- The following method will add the specified element just before the content of the specified selector

// Add a button just before the content of h1

```
$( "h1" ).prepend( "<button>New</button>" );
```

append()

- The following method will add the specified element just after the content of the specified selector

// Add a button just after the content of h1

```
$( "h1" ).append( "<button>New</button>" );
```

remove()

- The following method will remove the specified element

// Remove all buttons elements

```
$( "button" ).remove();
```

ADD ANIMATION

hide() <ul style="list-style-type: none">• Hides specified element	// Hide the h1 \$("h1").hide();	// This will hide the button \$("button").click(function(){ \$(this).hide(); });
toggle() <ul style="list-style-type: none">• Toggles specified element	// Toggle the h1 \$("h1").toggle();	
fadeOut() <ul style="list-style-type: none">• Hides specified element slowly	// Fade the h1 \$("h1").fadeOut();	
fadeIn() <ul style="list-style-type: none">• Shows specified element slowly	// Show the h1 slowly \$("h1").fadeIn();	
fadeToggle() <ul style="list-style-type: none">• Toggles specified element slowly	// Show or fade the element \$("h1").fadeToggle();	
slideUp() <ul style="list-style-type: none">• collapses specified element	// Collapse the h1 \$("h1").slideUp();	
slideDown() <ul style="list-style-type: none">• uncollapses specified element	// Uncollapse the h1 \$("h1").slideDown();	
slideToggle() <ul style="list-style-type: none">• Collapses or uncollapses element	// Collapse or Uncollapse the h1 \$("h1").slideToggle();	

ADD CUSTOM ANIMATION

animate()

- We can use the animate() method to add custom animation to elements
- Note that we only use properties that only have numeric values in this method

// Change the opacity of h1

```
$( "h1" ).animate({ opacity:0.5});
```

// Give the h1 the margin of 20px

```
$( "h1" ).animate({ margin:"20px" });
```

Chaining animations →

// adding multiple animations to h1

```
$( "h1" ).slideUp().slideDown().animate({ opacity: 0.5 });
```

AJAX REQUEST

ajax(url [, settings])

- The ajax() method sends asynchronous HTTP requests to the server.
- **URL** – url to which the request is sent.
- **Method** – HTTP method to use for request
- **Data** – data to be sent to the server
- **Success** – Callback function to be executed when AJAX request succeeds.
- The data argument in success function represents the data you are getting from the server

// Make a GET REQUEST

```
$ajax({  
    url: "https://dog.ceo/api/breeds/image/random",  
    method: "get",  
    success: function (data) {  
        var imageURL = data.message;  
        $(".Image").attr("src", imageURL);  
    },  
})
```

httpMethod()

- The above syntax can also be used for sending async requests to the server.
- Here httpMethods refers to method names like get, post

// Make a GET REQUEST

```
$.get("https://dog.ceo/api/breeds/image/random", function (data) {  
    var imageURL = data.message;  
    $(".Image").attr("src", imageURL);  
})
```

AJAX ERROR HANDLING

fail()

- The fail() method provides AJAX error handling in jQuery.
- The handler can receive the following 3 arguments :-
 - xhr - Jquery xhr object
 - textStatus - says error if there is a error
 - errorThrown

// Handle errors in case request fails

```
$get("https://dog.ceo/api/breeds/image/random", function (data) {  
    var imageURL = data.message;  
    $(".Image").attr("src", imageURL);  
}) .fail(function (xhr, textStatus, errorThrown) {  
    console.log("Request Failed");  
});
```