

# jQuery

---

## What is jQuery

**jQuery is a lightweight JavaScript library.** It contains a lot of methods that can do tasks requiring many lines of JavaScript code, in just a single line of code. Examples of such tasks are - removing elements from HTML, adding elements to HTML, etc.

The benefits of using jQuery can be summarized below -

- Accelerates the speed of development
- More work in less code
- Writing code is simpler and easy
- Provides a lot of methods and features
- Works around multiple browsers

**Using jQuery does not mean that JavaScript is not required.** jQuery is just a library built using JavaScript that provides functions for easy use. You still need to use JavaScript to create logic and to use it with these functions.

### - Fetching Elements

In JavaScript, we have different methods to fetch elements using an id or by class name. But jQuery makes fetching elements very easy by using CSS selectors (just like the way we use CSS).

The syntax for using jQuery is - `$(selector).action()`

\$ - defines a **function to access jQuery**

**selector** - **CSS selector** to query HTML elements

**action** - **jQuery function** to be executed on selected elements

We have some examples of fetching different elements -

→ `$("p")` - this will return all the **p tags**

- \$(".prim") - this will return all the **elements with class 'prim'**
- \$("#cringe") - this will return the **element with id 'cringe'**
- \$("span.light") - this will return all the **span elements with class 'light'**
- \$("form#subscribe") - this will return the **form element with id 'subscribe'**
- \$(".hide .del") - this will return all the **elements with class 'hide' and 'del'**

**NOTE:** The above examples will fetch the jQuery object and not the DOM object.

## SOME PROPERTIES OF JQUERY

In this section, we will learn about some of the methods in jQuery. But let's first discuss one very important jQuery function - **ready()**.

This method is used to **prevent any code from running before the document is finished loading**. Any method that is written inside this method then starts executing only after the document has finished loading. The way to use this method is -

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

It is a good practice to wait for the document to be fully loaded before working on it. This makes it possible to use a script tag in the head section of the HTML document. Another method for document ready event is -

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

Alternatively, you can also include Javascript code at the end of the body tag. Basically when all your HTML content is done.

## - Hide Elements

The **hide()** method of jQuery is used to **hide elements from the HTML page**. It actually makes the **visibility of the element hidden**.

The syntax is - `$(selector).hide(speed, callback)`

*speed* - **optional parameter** to set the speed of hiding having values - 'slow', 'fast' or milliseconds.

*callback* - **optional parameter** that runs a callback function after the hide() method completes.

Eg., the below code will hide all the p tags with class 'hide' in 1 second -

```
$("#p.hide").hide(1000);
```

## - Remove Elements

The **remove()** method is used to **remove elements from the HTML page**. It deletes the selected element and its child elements from the web page.

The syntax is - `$(selector).remove()`

Eg., the below code will remove all the p tags with class 'del' -

```
$("#p.del").remove();
```

## Get/Set Element Content

The **html()** method is used to **set or return the content (i.e. innerHTML) of the selected elements**.

When this method is used to **return** content, it returns **the content of the first matched element**. The syntax is - `$(selector).html()`

When this method is used to **set** content, it **overwrites the content of all matched elements**. The syntax is - `$(selector).html(content)`

Eg., the below code will set the content of all the p tags with 'Hello world!' -

```
$("#p").html("Hello <b>world</b>!");
```

## - Get/Set Element Text

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

When this method is used to **return** content, it returns **the text content of all matched elements** and the child HTML element will not be returned.

When this method is used to **set** content, it **overwrites the content of all matched elements**. This will **remove the child HTML elements** of the selected elements.

Eg., the below code will set the text content of all the p tags with 'Hello world!' and remove any of its child elements-

```
$("#p").text("Hello world!");
```

## MODIFY CSS

Modifying the CSS has become very easy by using jQuery. It provides a very easy method to add one or more styles to the selected elements.

jQuery provides a css() method to **change the CSS properties of the element**.

To **set a specific property** use syntax - `css("propertyname", "value")`

To **set multiple properties** use syntax -

```
css({ "propertyname": "value", "propertyname": "value", ... })
```

Examples to modify the css properties are

```
$("#p").css("background-color", "red");
$("#p").css("backgroundColor", "red");
$("#p").css({ "background-color": "red", "font-size": "30px" });
$("#p").css({ "backgroundColor": "red", "fontSize": "30px" });
```

jQuery allows having **both types of formats for multiple-word properties**. It sets the same property for both of these multi-word formats.

- **addClass() method**

The **addClass()** method is used to **add class attributes to the selected elements**. It allows you to add multiple classes to the elements as well.

Examples are -

- `$("span, b, u").addClass("blue");` - it will add class 'blue' to all the selected elements
- `$("img").addClass("collage");` - it will add class 'collage' to all the image elements
- `$("#imp").addClass("important blue");` - it will add 'important' and 'blue' classes to the element with id 'imp'

- **removeClass() method**

The **removeClass()** method is used to **remove one or more classes from the selected elements**.

Examples are -

- `$("span, b, u").removeClass("blue");` - it will remove class 'blue' to all the selected elements
- `$("img").removeClass("collage");` - it will remove class 'collage' to all the image elements
- `$("#imp").removeClass("important blue");` - it will remove 'important' and 'blue' classes to the element with id 'imp'

- **toggleClass() method**

The **toggleClass()** method is used to **toggle between adding and removing classes from the selected elements**.

If the class is not present it will be added first and if it is present then it will be removed first.

Examples are -

- `$("span, b, u").toggleClass("blue");` - it will toggle the class 'blue' to all the selected elements
- `$("img").toggleClass("collage");` - it will toggle the class 'collage' to all the image elements
- `$("#imp").toggleClass("important blue");` - it will toggle 'important' and 'blue' classes separately for the element with id 'imp'

## EVENT HANDLING

jQuery has also made adding events to the elements easy. The syntax for jQuery event methods is somewhat similar to those of JavaScript.

All the JavaScript event methods are also written in jQuery. Some of them are -

- ***click()***
- ***mouseenter()***
- ***mousedown()***
- ***hover()***
- ***keyup()***

The syntax to add events to the element(s) is -

```
$(selector).event(function(e) {
    // Write something here
});
```

Ex., the below code will ***add a click event to a button with id 'submit-page'*** -

```
$
("#submit-page").click(function() {
    alert("Form has been submitted");
});
```

Just like JavaScript, there is also another method to add events to the elements, i.e. by using the 'on' method.

The syntax for it is -

```
$(selector).on(event, function(e) {  
    // Write something here  
});
```

Eg., you can modify the above code as -

```
$("#submit-page").on("click", function() {  
    alert("Form has been submitted");  
});
```