**JQUERY**

**HISTORY:-**



**jQuery** is a [JavaScript library](https://simple.wikipedia.org/wiki/JavaScript) created to make it easier and simpler to write [JavaScript](https://simple.wikipedia.org/wiki/JavaScript) and [HTML](https://simple.wikipedia.org/wiki/HTML). It works on most [web browsers](https://simple.wikipedia.org/wiki/Web_browsers)[[2]](https://simple.wikipedia.org/wiki/JQuery#cite_note-jquery.com-2) and was invented by [John Resig](https://simple.wikipedia.org/w/index.php?title=John_Resig&action=edit&redlink=1). The first [release](https://simple.wikipedia.org/wiki/Software_release) was in January 2006 at [BarCamp](https://simple.wikipedia.org/w/index.php?title=BarCamp&action=edit&redlink=1" \o "BarCamp (not yet started)) NYC. Today there is a [team](https://simple.wikipedia.org/wiki/Team) of [programmers](https://simple.wikipedia.org/wiki/Programmer) working on jQuery, which is led by Dave Methvin.

-jQuery is a JavaScript Library.

-jQuery greatly simplifies JavaScript programming.

-jQuery is easy to learn.

**jQuery Introduction**

-The purpose of jQuery is to make it much easier to use JavaScript on your website.

**What is jQuery?**

-jQuery is a lightweight, "**write less, do more**", JavaScript library.

-The purpose of jQuery is to make it much easier to use JavaScript on your website.

-jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

-jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

-The jQuery library contains the following features:

* HTML/DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities

**Tip:** In addition, jQuery has plugins for almost any task out there.

**Why jQuery?**

-There are lots of other JavaScript libraries out there, but jQuery is probably the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

* Google
* Microsoft
* IBM
* Netflix

## JQUERY FEATURES

Here is the list of important core features supported by jQuery −

* **DOM manipulation** − The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
* **Event handling** − The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
* **AJAX Support** − The jQuery helps you a lot to develop a responsive and featurerich site using AJAX technology.
* **Animations** − The jQuery comes with plenty of built-in animation effects which you can use in your websites.
* **Lightweight** − The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
* **Cross Browser Support** − The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
* **Latest Technology** − The jQuery supports CSS3 selectors and basic XPath syntax.

**SETTING UP JQUERY**

There are two ways to use jQuery.

1. **Local Installation** − You can download jQuery library on your local machine and include it in your HTML code.
2. **CDN Based Installation** − You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

## 1.jQuery - Local Installation

You can download latest version of jQuery on your web server and include the downloaded library in your code. We suggest you to download compressed version of the library for a better performance.

* Go to the <https://jquery.com/download/> to download the latest version available.
* Now put downloaded **jquery-3.6.0.min.js** file in a directory of your website, e.g. /jquery/**jquery-3.6.0.js**.
* Finally include this file in your HTML markup file as shown below.

### Example

Now you can include *jquery* library in your HTML file as given below. Try

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.min.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>JQuery Element Selector</h1>

    <p>This is p tag</p>

    <span>This is span tag</span>

    <div>This is div tag</div>

   <script>

       $(document).ready(function() {

            $("p").css("background-color", "yellow");

    });

   </script>

</body>

</html>

**Adding jQuery to Your Web Pages**

There are several ways to start using jQuery on your web site. You can:

* Download the jQuery library from jQuery.com
* Include jQuery from a CDN, like Google

## Downloading jQuery

There are two versions of jQuery available for downloading:

* Production version - this is for your live website because it has been minified and compressed
* Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from [jQuery.com](http://jquery.com/download/).

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

<head>  
<script src="jquery-3.7.1.min.js"></script>  
</head>

**Tip:** Place the downloaded file in the same directory as the pages where you wish to use it.

## 2.jQuery CDN

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Google is an example of someone who host jQuery:

### Google CDN:

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

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

    <title>Document</title>

</head>

<body>

    <h1>JQuery Using CDN</h1>

    <p>This is p tag</p>

    <span>This is span tag</span>

    <div>This is div tag</div>

   <script>

       $(document).ready(function() {

            $("p").css("background-color", "yellow");

    });

   </script>

</body>

</html>

-In You have to copy the **Google CDN** on jQuery Documentation.

**How to Call a jQuery Library Functions?**

-As almost everything, we do when using jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

-If you want an event to work on your page, you should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

To do this, we register a ready event for the document as follows −

$(document).ready(function() {

// do stuff when DOM is ready

});

-To call upon any jQuery library function, use HTML script tags as shown

Open Compiler

<!doctype html>

<html>

<head>

<title>The jQuery Example</title>

<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>

<script>

$(document).ready(function() {

$("div").click(function() {alert("Hello, world!");});

});

</script>

</head>

<body>

<div>Click on this to see a dialogue box.</div>

</body>

</html>

**Document Ready Event**

-Before we look into **jQuery Syntax**, let's try to understand what is **Document Ready Event**. Actually, before we execute any jQuery statement, we would like to wait for the document to be fully loaded. This is because jQuery works on DOM and if complete DOM is not available before executing jQuery statements, then we will not get desired result.

Following is basic syntax of a Document Ready Event:

$(document).ready(function(){

// jQuery code goes here...

});

-Alternatively you can also use the following syntax for document ready event:

$(function(){

// jQuery code goes here...

});

-You should always keep **Document Ready Event** block inside **<script>...</script>** tags and you can keep this script tag either inside the **<head>...</head>** tags or at the bottom of the page before closing **<body>** tag.

-You can use either of these two syntax to keep your jQuery code inside the block which will be executed only when complete DOM is downloaded and ready to be parses.

## jQuery Syntax

-Following is the basic syntax for selecting HTML elements and then performing some action on the selected element(s):

$(document).ready(function(){

$(selector).action()

});

-Any jQuery statement starts with a dollar sign **$** and then we put a **selector** inside the braces **()**. This syntax **$(selector)** is enough to return the selected HTML elements, but if you have to perform any action on the selected element(s) then **action()** part is required.

*-The factory function****$()****is a synonym of****jQuery()****function. So in case you are using any other JavaScript library where $ sign is conflicting with some thing else then you can replace****$****sign by jQuery name and you can use function****jQuery()****instead of****$()****.*

Basic syntax is: **$(*selector*).*action*()**

* 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:

$(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".

1.Example1

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.min.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>jQuery Basic Syntax</h1>

    <p>This is p tag</p>

    <p>This is another p tag</p>

    <span>This is span tag</span>

    <div>This is div tag</div>

   <script>

        $(document).ready(function(){

            $("p").hide();

        })

   </script>

</body>

</html>

2.Example2

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.min.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>jQuery Basic Syntax</h1>

    <p>This is p tag</p>

    <span>This is span tag</span>

    <div>This is div tag</div>

    <script>

        $(document).ready(function(){

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

        })

    </script>

</body>

</html>

## jQuery Selectors

jQuery Selectors are used to select HTML element(s) from an HTML document. Consider an HTML document is given and you need to select all the <div> from this document. This is where jQuery Selectors will help.

jQuery Selectors can find HTML elements (ie. Select HTML elements) based on the following:

* HTML element Name
* Element ID
* Element Class
* Element attribute name
* Element attribute value
* Many more criteria

The jQuery library harnesses the power of **Cascading Style Sheets** (CSS) selectors to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).

jQuery Selectors works in very similar way on an HTML document like an **SQL Select Statement** works on a Database to select the records.

## jQuery Selector Syntax

Following is the jQuery Selector Syntax for selecting HTML elements:

$(document).ready(function(){

$(selector)

});

-A jQuery selector starts with a dollar sign **$** and then we put a **selector** inside the braces **()**. Here **$()** is called **factory function**, which makes use of following three building blocks while selecting elements in a given document:

|  |  |
| --- | --- |
| Selector Name | Description |
| The element Selector | Represents an HTML element name available in the DOM. For example **$('p')** selects all paragraphs <p> in the document. |
| The #id Selector | Represents a HTML element available with the given ID in the DOM. For example **$('#some-id')** selects the single element in the document that has **some-id** as element Id. |
| The .class Selector | Represents a HTML elements available with the given class in the DOM. For example **$('.some-class')** selects all elements in the document that have a class of **some-class**. |

All the above selectors can be used either on their own or in combination with other selectors. All the jQuery selectors are based on the same principle except some tweaking.

## 1.The Element Selector

-The jQuery **element** selector selects HTML element(s) based on the element name. Following is a simple syntax of an element selector:

$(document).ready(function(){

$("Html Element Name")

});

Example:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.min.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>JQuery Element Selector</h1>

    <p>This is p tag</p>

    <span>This is span tag</span>

    <div>This is div tag</div>

   <script>

       $(document).ready(function() {

            $("p").css("background-color", "yellow");

    });

   </script>

</body>

</html>

## 2. The #id Selector

-The jQuery **#id** selector selects an HTML element based on the element **id** attribute.

-Following is a simple syntax of a **#id** selector:

$(document).ready(function(){

$("#id of the element")

});

-To use jQuery **#id** selector, you need to make sure that **id** attribute should be uniquely assigned to all the DOM elements. If your elements will have similar ids then it will not produce correct result.

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <script src="../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.min.js"></script>

    <title>Document</title>

  </head>

  <body>

*<!--! ID SELECTOR  -->*

    <h1>Using #Id Selector</h1>

    <p id="Para">

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Numquam culpa

      modi provident, sed accusamus a, cumque et cupiditate assumenda, officia

      eum dolores. Commodi, reprehenderit.

    </p>

    <button id="btn">Change Color</button>

    <script>

      $(document).ready(function () {

        $("#Para").css("color", "red");

      });

    </script>

  </body>

</html>

## 3. The .class Selector

-The jQuery **.class** selector selects HTML element(s) based on the element **class** attribute.

$(document).ready(function(){

$(".class of the element")

});

-Because a class can be assigned to multiple HTML elements with in an HTML document, so it is very much possible to find out multiple elements with a single **.class** selector statement.

Examples

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <script src="../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

  </head>

  <body>

    <p class="para">

      Lorem ipsum dolor sit amet consectetur, adipisicing elit. Modi, cupiditate

      est!

    </p>

    <p class="para">

      Harum voluptatem, esse repellat alias voluptate quis ipsum totam veniam

      modi?

    </p>

    <span class="bar">This is bar span tag</span>

    <div class="bill">This is bill div tag</div>

    <script>

        $(document).ready(function () {

          $(".para").css("color", "White");

          $(".para").css("background-color", "black");

        });

*// You can also use the click event to change the color of the text and the background color of the span and div tags.*

    </script>

  </body>

</html>

**More Examples of jQuery Selectors**

|  |  |  |
| --- | --- | --- |
| Syntax | Description | Example |
| $("\*") | Selects all elements | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_all2) |
| $(this) | Selects the current HTML element | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_this) |
| $("p.intro") | Selects all <p> elements with class="intro" | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_pclass) |
| $("p:first") | Selects the first <p> element | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_pfirst) |
| $("ul li:first") | Selects the first <li> element of the first <ul> | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_ullifirst) |
| $("ul li:first-child") | Selects the first <li> element of every <ul> | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_ullifirstchild) |
| $("[href]") | Selects all elements with an href attribute | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattr) |
| $("a[target='\_blank']") | Selects all <a> elements with a target attribute value equal to "\_blank" | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattrblank) |
| $("a[target!='\_blank']") | Selects all <a> elements with a target attribute value NOT equal to "\_blank" | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattrnotblank) |
| $(":button") | Selects all <button> elements and <input> elements of type="button" | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_button2) |
| $("tr:even") | Selects all even <tr> elements | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_even) |
| $("tr:odd") | Selects all odd <tr> elements | [Try it](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_odd) |

**1.$("\*")**

-Selects all elements.

-It will target all the element in the document.

**Syntax**:-

$(document).ready(function(){

$("\*”)

});

Example:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="./INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>$("\*") Selector</h1>

    <h2>This is a heading</h2>

    <p>This is a paragraph.</p>

    <p>This is another paragraph.</p>

    <script>

        $(document).ready(function() {

            $("\*").css("color", "red");

        });

    </script>

</body>

</html>

**2.$("this")**

-Selects the current HTML element

**Syntax**:-

$(document).ready(function(){

$("this”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>$("\*") Selector</h1>

    <h2>This is a heading</h2>

    <p>This is a paragraph.</p>

    <p>This is another paragraph.</p>

    <script>

        $(document).ready(function(){

          $("p").click(function(){

            $(this).hide();

          });

        });

        </script>

</body>

</html>

**3.$("p.intro")**

-Selects all <p> elements with class="intro".

**Syntax**:-

$(document).ready(function(){

$("this”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>Selects all <p> elements with class="intro"</h1>

    <h2 class="intro">This is a heading</h2>

    <p class="intro">This is a paragraph.</p>

    <p>This is another paragraph.</p>

    <button>Click me</button>

    <script>

        $(document).ready(function(){

            $("button").click(function(){

                $("p.intro").css("color","red")

            })

        })

    </script>

</body>

</html>

**4.$("p:first")**

-Selects the first <p> element

**Syntax**:-

$(document).ready(function(){

$("p.first”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h2>This is a heading</h2>

    <p>This is a paragraph.</p>

    <p>This is another paragraph.</p>

    <button>Click me</button>

    <script>

    $(document).ready(function(){

        $("button").click(function(){

            $("p:first").hide();

        });

    });

    </script>

</body>

</html>

**5.$("ul li:first")**

Selects the first <li> element of the first <ul>.

**Syntax**:-

$(document).ready(function(){

$("ul li:first”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h1>List 1:</h1>

    <ul>

    <li>Coffee</li>

    <li>Milk</li>

    <li>Tea</li>

    </ul>

    <p>List 2:</p>

    <ul>

    <li>Coffee</li>

    <li>Milk</li>

    <li>Tea</li>

    </ul>

    <button>CLICK ME</button>

    <script>

        $(document).ready(function(){

        $("button").click(function(){

            $("ul li:first").css("color","red");

        });

    });

    </script>

</body>

</html>

**6.$("ul li:first")**

Selects the first <li> element of the first <ul>.

**Syntax**:-

$(document).ready(function(){

$("[href]”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <h2>This is a heading</h2>

    <p>This is a paragraph.</p>

    <p>This is another paragraph.</p>

    <p><a href="https://www.w3schools.com/html/">HTML Tutorial</a></p>

    <p><a href="https://www.w3schools.com/css/">CSS Tutorial</a></p>

    <button>Click me</button>

    <script>

        $(document).ready(function(){

        $("button").click(function(){

            $("[href]").hide();

        });

    });

    </script>

</body>

</html>

**7.$("** **p:even ")**

Selects the first <li> element of the first <ul>.

**Syntax**:-

$(document).ready(function(){

$(" p:even]”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad, nam vel.</p>

    <p>Corporis aliquid, illum et iusto nihil nemo quod velit ullam earum.</p>

    <p>In ipsam at error eos ex possimus nam aspernatur, dolores debitis.</p>

    <p>Magni nobis, corrupti perspiciatis molestias sunt eos quam nam aut aspernatur?</p>

    <p>Necessitatibus sequi omnis reiciendis corrupti earum debitis quisquam hic vero quod.</p>

    <p>Maiores, molestias dignissimos? Commodi inventore, excepturi nulla ducimus magni aperiam nesciunt?</p>

    <p>Ratione voluptatibus consequuntur reprehenderit consequatur suscipit eaque itaque non voluptatum quam.</p>

    <p>Itaque, velit dignissimos. Excepturi dolorem rerum at perferendis praesentium impedit quidem?</p>

    <p>Nulla, voluptate inventore quidem omnis error fugiat amet quas? Fugit, sit.</p>

    <p>Amet maxime dicta, consequuntur autem voluptatum quod veritatis tenetur dolor perspiciatis.</p>

    <script>

        $(document).ready(function(){

            $("p:even").css("background-color", "yellow");

        });

    </script>

</body>

</html>

**8.$("** **p:odd ")**

Selects the first <li> element of the first <ul>.

**Syntax**:-

$(document).ready(function(){

$(" p:even]”)

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="../../../INSTALLATION/LOCAL INSTALLATION/jquery-3.7.1.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad, nam vel.</p>

    <p>Corporis aliquid, illum et iusto nihil nemo quod velit ullam earum.</p>

    <p>In ipsam at error eos ex possimus nam aspernatur, dolores debitis.</p>

    <p>Magni nobis, corrupti perspiciatis molestias sunt eos quam nam aut aspernatur?</p>

    <p>Necessitatibus sequi omnis reiciendis corrupti earum debitis quisquam hic vero quod.</p>

    <p>Maiores, molestias dignissimos? Commodi inventore, excepturi nulla ducimus magni aperiam nesciunt?</p>

    <p>Ratione voluptatibus consequuntur reprehenderit consequatur suscipit eaque itaque non voluptatum quam.</p>

    <p>Itaque, velit dignissimos. Excepturi dolorem rerum at perferendis praesentium impedit quidem?</p>

    <p>Nulla, voluptate inventore quidem omnis error fugiat amet quas? Fugit, sit.</p>

    <p>Amet maxime dicta, consequuntur autem voluptatum quod veritatis tenetur dolor perspiciatis.</p>

    <script>

        $(document).ready(function(){

            $("p:odd ").css("background-color", "yellow");

        });

    </script>

</body>

</html>

**jQuery Event Methods**

jQuery Event Methods

-A jQuery Event is the result of an action that can be detected by jQuery (JavaScript). When these events are triggered, you can then use a custom function to do pretty much whatever you want with the event. These custom functions are called **Event Handlers**.

-The jQuery library provides methods to handle all the DOM events and make complete event handling considerably easier than what we have available in JavaScript.

Following are the examples of some common events −

* A mouse click
* A web page loading
* Taking mouse over an element
* Submitting an HTML form
* A keystroke on your keyboard, etc.

The following table lists some of the important DOM events.

|  |  |  |  |
| --- | --- | --- | --- |
| Mouse Events | Keyboard Events | Form Events | Document Events |
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| hover | keyup | select | scroll |
| mousedown |  | blur | unload |
| mouseup |  | focusin | ready |

## jQuery Event Binding Syntax

Consider a situation when you want to click a <div> in an HTML document and then you want to perform some action against this click. To achieve this you will have to bind a jQuery **click** event with the <div> element and then define an action against the click event.

Following is jQuery syntax to bind a click event with all the <div> elements available in an HTML document:

$("div").click();

The next step is to define an action against the click event. Following is the syntax to define a function which will be executed when **click** event will be fired. This function is called **jQuery Event Handler**

$("div").click(function(){

// jQuery code goes here

});

Following is another syntax to bind a click event with any of the DOM elements:

$("div").bind('click', function(){

// jQuery code goes here

});

|  |  |
| --- | --- |
| Method / Property | Description |
| [bind()](https://www.w3schools.com/jquery/event_bind.asp) | Deprecated in version 3.0. Use the [on()](https://www.w3schools.com/jquery/event_on.asp) method instead. Attaches event handlers to elements |
| [blur()](https://www.w3schools.com/jquery/event_blur.asp) | Attaches/Triggers the blur event |
| [change()](https://www.w3schools.com/jquery/event_change.asp) | Attaches/Triggers the change event |
| [click()](https://www.w3schools.com/jquery/event_click.asp) | Attaches/Triggers the click event |
| [dblclick()](https://www.w3schools.com/jquery/event_dblclick.asp) | Attaches/Triggers the double click event |
| [delegate()](https://www.w3schools.com/jquery/event_delegate.asp) | Deprecated in version 3.0. Use the [on()](https://www.w3schools.com/jquery/event_on.asp) method instead. Attaches a handler to current, or future, specified child elements of the matching elements |
| [die()](https://www.w3schools.com/jquery/event_die.asp) | Removed in version 1.9. Removes all event handlers added with the live() method |
| [error()](https://www.w3schools.com/jquery/event_error.asp) | Removed in version 3.0. Attaches/Triggers the error event |
| [event.currentTarget](https://www.w3schools.com/jquery/event_currenttarget.asp) | The current DOM element within the event bubbling phase |
| [event.data](https://www.w3schools.com/jquery/event_data.asp) | Contains the optional data passed to an event method when the current executing handler is bound |
| [event.delegateTarget](https://www.w3schools.com/jquery/event_delegatetarget.asp) | Returns the element where the currently-called jQuery event handler was attached |
| [event.isDefaultPrevented()](https://www.w3schools.com/jquery/event_isdefaultprevented.asp) | Returns whether event.preventDefault() was called for the event object |
| [event.isImmediatePropagationStopped()](https://www.w3schools.com/jquery/event_isimmediatepropagationstopped.asp) | Returns whether event.stopImmediatePropagation() was called for the event object |
| [event.isPropagationStopped()](https://www.w3schools.com/jquery/event_ispropagationstopped.asp) | Returns whether event.stopPropagation() was called for the event object |
| [event.namespace](https://www.w3schools.com/jquery/event_namespace.asp) | Returns the namespace specified when the event was triggered |
| [event.pageX](https://www.w3schools.com/jquery/event_pagex.asp) | Returns the mouse position relative to the left edge of the document |
| [event.pageY](https://www.w3schools.com/jquery/event_pagey.asp) | Returns the mouse position relative to the top edge of the document |
| [event.preventDefault()](https://www.w3schools.com/jquery/event_preventdefault.asp) | Prevents the default action of the event |
| [event.relatedTarget](https://www.w3schools.com/jquery/event_relatedtarget.asp) | Returns which element being entered or exited on mouse movement |
| [event.result](https://www.w3schools.com/jquery/event_result.asp) | Contains the last/previous value returned by an event handler triggered by the specified event |
| [event.stopImmediatePropagation()](https://www.w3schools.com/jquery/event_stopimmediatepropagation.asp) | Prevents other event handlers from being called |
| [event.stopPropagation()](https://www.w3schools.com/jquery/event_stoppropagation.asp) | Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event |
| [event.target](https://www.w3schools.com/jquery/event_target.asp) | Returns which DOM element triggered the event |
| [event.timeStamp](https://www.w3schools.com/jquery/event_timestamp.asp) | Returns the number of milliseconds since January 1, 1970, when the event is triggered |
| [event.type](https://www.w3schools.com/jquery/event_type.asp) | Returns which event type was triggered |
| [event.which](https://www.w3schools.com/jquery/event_which.asp) | Returns which keyboard key or mouse button was pressed for the event |
| [focus()](https://www.w3schools.com/jquery/event_focus.asp) | Attaches/Triggers the focus event |
| [focusin()](https://www.w3schools.com/jquery/event_focusin.asp) | Attaches an event handler to the focusin event |
| [focusout()](https://www.w3schools.com/jquery/event_focusout.asp) | Attaches an event handler to the focusout event |
| [hover()](https://www.w3schools.com/jquery/event_hover.asp) | Attaches two event handlers to the hover event |
| [keydown()](https://www.w3schools.com/jquery/event_keydown.asp) | Attaches/Triggers the keydown event |
| [keypress()](https://www.w3schools.com/jquery/event_keypress.asp) | Attaches/Triggers the keypress event |
| [keyup()](https://www.w3schools.com/jquery/event_keyup.asp) | Attaches/Triggers the keyup event |
| [live()](https://www.w3schools.com/jquery/event_live.asp) | Removed in version 1.9. Adds one or more event handlers to current, or future, selected elements |
| [load()](https://www.w3schools.com/jquery/event_load.asp) | Removed in version 3.0. Attaches an event handler to the load event |
| [mousedown()](https://www.w3schools.com/jquery/event_mousedown.asp) | Attaches/Triggers the mousedown event |
| [mouseenter()](https://www.w3schools.com/jquery/event_mouseenter.asp) | Attaches/Triggers the mouseenter event |
| [mouseleave()](https://www.w3schools.com/jquery/event_mouseleave.asp) | Attaches/Triggers the mouseleave event |
| [mousemove()](https://www.w3schools.com/jquery/event_mousemove.asp) | Attaches/Triggers the mousemove event |
| [mouseout()](https://www.w3schools.com/jquery/event_mouseout.asp) | Attaches/Triggers the mouseout event |
| [mouseover()](https://www.w3schools.com/jquery/event_mouseover.asp) | Attaches/Triggers the mouseover event |
| [mouseup()](https://www.w3schools.com/jquery/event_mouseup.asp) | Attaches/Triggers the mouseup event |
| [off()](https://www.w3schools.com/jquery/event_off.asp) | Removes event handlers attached with the on() method |
| [on()](https://www.w3schools.com/jquery/event_on.asp) | Attaches event handlers to elements |
| [one()](https://www.w3schools.com/jquery/event_one.asp) | Adds one or more event handlers to selected elements. This handler can only be triggered once per element |
| [$.proxy()](https://www.w3schools.com/jquery/event_proxy.asp) | Takes an existing function and returns a new one with a particular context |
| [ready()](https://www.w3schools.com/jquery/event_ready.asp) | Specifies a function to execute when the DOM is fully loaded |
| [resize()](https://www.w3schools.com/jquery/event_resize.asp) | Attaches/Triggers the resize event |
| [scroll()](https://www.w3schools.com/jquery/event_scroll.asp) | Attaches/Triggers the scroll event |
| [select()](https://www.w3schools.com/jquery/event_select.asp) | Attaches/Triggers the select event |
| [submit()](https://www.w3schools.com/jquery/event_submit.asp) | Attaches/Triggers the submit event |
| [toggle()](https://www.w3schools.com/jquery/event_toggle.asp) | Removed in version 1.9. Attaches two or more functions to toggle between for the click event |
| [trigger()](https://www.w3schools.com/jquery/event_trigger.asp) | Triggers all events bound to the selected elements |
| [triggerHandler()](https://www.w3schools.com/jquery/event_triggerhandler.asp) | Triggers all functions bound to a specified event for the selected elements |
| [unbind()](https://www.w3schools.com/jquery/event_unbind.asp) | Deprecated in version 3.0. Use the [off()](https://www.w3schools.com/jquery/event_off.asp) method instead. Removes an added event handler from selected elements |
| [undelegate()](https://www.w3schools.com/jquery/event_undelegate.asp) | Deprecated in version 3.0. Use the [off()](https://www.w3schools.com/jquery/event_off.asp) method instead. Removes an event handler to selected elements, now or in the future |
| [unload()](https://www.w3schools.com/jquery/event_unload.asp) | Removed in version 3.0. Use the [on()](https://www.w3schools.com/jquery/event_on.asp) or [trigger()](https://www.w3schools.com/jquery/event_trigger.asp) method instead. Attaches an event handler to the unload event |

**MOUSE EVENTS**

**1.click()**

-The click event occurs when an element is clicked.

-The click() method triggers the click event, or attaches a function to run when a click event occurs.

**Syntax**

Trigger the click event for the selected elements:

$(*selector*).click()

Attach a function to the click event:

$(*selector*).click()

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Click on this paragraph.</p>

    <script>

        $(document).ready(function(){

            $("p").click(function(){

                alert("The paragraph was clicked.");

            });

        });

    </script>

</body>

</html>

**2.click()**

-The dblclick event occurs when an element is double-clicked.

-The dblclick() method triggers the dblclick event, or attaches a function to run when a dblclick event occurs.

**Tip:** The dblclick event also generates a click event. This can cause problems if both events are applied to the same element.

**Syntax**

Trigger the dblclick event for the selected elements:

$(*selector*).dblclick()

Attach a function to the click event:

$(*selector*).dblclick(*function*)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Click on this paragraph.</p>

    <script>

        $(document).ready(function(){

            $("p").click(function(){

                alert("The paragraph was clicked.");

            });

        });

    </script>

</body>

</html>

**3.hover()**

-The hover() method specifies two functions to run when the mouse pointer hovers over the selected elements.

-This method triggers both the [mouseenter](https://www.w3schools.com/jquery/event_mouseenter.asp) and [mouseleave](https://www.w3schools.com/jquery/event_mouseleave.asp) events.

**Syntax**

$(*selector*).hover(*inFunction,outFunction*)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Hover the mouse pointer over this paragraph.</p>

    <script>

        $(document).ready(function(){

            $("p").hover(function(){

                $(this).css("background-color", "yellow");

            }, function(){

                $(this).css("background-color", "white");

            });

        })

    </script>

</body>

</html>

**4.mouseup()**

-The mouseup event occurs when the left mouse button is released over the selected element.

-The mouseup() method triggers the mouseup event, or attaches a function to run when a mouseup event occurs.

**Tip:** This method is often used together with the [mousedown()](https://www.w3schools.com/jquery/event_mousedown.asp) method.

**Syntax**

$(*selector*).mouseup()

Attach a function to the click event:

$(*selector*).mouseup(*function*)

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<div>Press down and release the mouse button over this div element.</div>

<script>

    $(document).ready(function(){

      $("div").mouseup(function(){

        $(this).after("<p style='color:green;'>Mouse button released.</p>");

      });

      $("div").mousedown(function(){

        $(this).after("<p style='color:purple;'>Mouse button pressed down.</p>");

      });

    });

    </script>

</body>

</html>

**5.mousedown()()**

-The mousedown event occurs when the left mouse button is pressed down over the selected element.

-The mousedown() method triggers the mousedown event, or attaches a function to run when a mousedown event occurs.

**Tip:** This method is often used together with the [mouseup()](https://www.w3schools.com/jquery/event_mouseup.asp) method.

**Syntax**

$(*selector*).mousedown()

Attach a function to the click event:

$(*selector*).mousedown(*function*)

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<div>Press down and release the mouse button over this div element.</div>

<script>

    $(document).ready(function(){

      $("div").mouseup(function(){

        $(this).after("<p style='color:green;'>Mouse button released.</p>");

      });

      $("div").mousedown(function(){

        $(this).after("<p style='color:purple;'>Mouse button pressed down.</p>");

      });

    });

    </script>

</body>

</html>

**6.mouseenter()**

-The mouseenter event occurs when the mouse pointer is over (enters) the selected element.

-The mouseenter() method triggers the mouseenter event, or attaches a function to run when a mouseenter event occurs..

**Note:** Unlike the [mouseover](https://www.w3schools.com/jquery/event_mouseover.asp) event, the mouseenter event only triggers when the mouse pointer enters the selected element. The mouseover event is triggered if a mouse pointer enters any child elements as well. See the example at the end of the page for a demonstration.

**Tip:** This event is often used together with the [mouseleave](https://www.w3schools.com/jquery/event_mouseleave.asp) event.

**Syntax**

$(*selector*).mouseenter()

Attach a function to the click event:

$(*selector*).mouseenter(*function*)

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<div>Press down and release the mouse button over this div element.</div>

<script>

    $(document).ready(function(){

      $("div").mouseup(function(){

        $(this).after("<p style='color:green;'>Mouse button released.</p>");

      });

      $("div").mousedown(function(){

        $(this).after("<p style='color:purple;'>Mouse button pressed down.</p>");

      });

    });

    </script>

</body>

</html>

**7.mouseleave()**

-The mouseleave event occurs when the mouse pointer leaves the selected element.

-The mouseleave() method triggers the mouseleave event, or attaches a function to run when a mouseleave event occurs.

**Note:** Unlike the [mouseout](https://www.w3schools.com/jquery/event_mouseout.asp) event, the mouseleave event only triggers when the mouse pointer leaves the selected elements. The mouseout event is triggered if a mouse pointer leaves any child elements as well as the selected element.  See the example at the end of the page for a demonstration.

**Tip:** This event is often used together with the [mouseenter](https://www.w3schools.com/jquery/event_mouseenter.asp) event.

**Syntax**

$(*selector*).mouseleave()

Attach a function to the click event:

$(*selector*).mouseleave(*function*)

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<div>Press down and release the mouse button over this div element.</div>

<script>

    $(document).ready(function(){

      $("div").mouseup(function(){

        $(this).after("<p style='color:green;'>Mouse button released.</p>");

      });

      $("div").mousedown(function(){

        $(this).after("<p style='color:purple;'>Mouse button pressed down.</p>");

      });

    });

    </script>

</body>

</html>

**8.mouseout()**

-The mouseout event occurs when the mouse pointer leaves the selected element.

-The mouseout() method triggers the mouseout event, or attaches a function to run when a mouseout event occurs.

**Note:** Unlike the [mouseleave](https://www.w3schools.com/jquery/event_mouseleave.asp) event, the mouseout event is triggered if a mouse pointer leaves any child elements as well as the selected element. The mouseleave event only triggers when the mouse pointer leaves the selected element. See the example at the end of the page for a demonstration.

**Tip:** This event is often used together with the [mouseover](https://www.w3schools.com/jquery/event_mouseover.asp) event.

**Syntax**

$(*selector*).mouseout()

Attach a function to the click event:

$(*selector*).mouseout(*function*)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Move the mouse pointer over this paragraph.</p>

    <script>

        $(document).ready(function(){

          $("p").mouseover(function(){

            $("p").css("background-color", "yellow");

          });

          $("p").mouseout(function(){

            $("p").css("background-color", "lightgray");

          });

        });

    </script>

</body>

</html>

**9.mouseover()**

-The mouseover event occurs when the mouse pointer is over the selected element.

-The mouseover() method triggers the mouseover event, or attaches a function to run when a mouseover event occurs.

**Note:** Unlike the [mouseenter](https://www.w3schools.com/jquery/event_mouseenter.asp) event, the mouseover event triggers if a mouse pointer enters any child elements as well as the selected element. The mouseenter event is only triggered when the mouse pointer enters the selected element. See the example at the end of the page for a demonstration.

**Tip:** This event is often used together with the [mouseout](https://www.w3schools.com/jquery/event_mouseout.asp) event.

**Syntax**

$(*selector*).mouseover()

Attach a function to the click event:

$(*selector*).mouseover(*function*)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Move the mouse pointer over this paragraph.</p>

    <script>

        $(document).ready(function(){

          $("p").mouseover(function(){

            $("p").css("background-color", "yellow");

          });

          $("p").mouseout(function(){

            $("p").css("background-color", "lightgray");

          });

        });

    </script>

</body>

</html>

**10.mousemove()**

-The mousemove event occurs whenever the mouse pointer moves within the selected element.

-The mousemove() method triggers the mousemove event, or attaches a function to run when a mousemove event occurs.

**Note:** Each time a user moves the mouse one pixel, a mousemove event occurs. It takes system resources to process all mousemove events. Use this event carefully.

**Syntax**

$(*selector*).mousemove()

Attach a function to the click event:

$(*selector*).mousemove(*function*)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <title>Document</title>

</head>

<body>

    <p>Mouse is at coordinates: <span></span>.</p>

    <script>

        $(document).ready(function(){

          $(document).mousemove(function(*event*){

            $("span").text(*event*.pageX + ", " + *event*.pageY);

          });

        });

    </script>

</body>

</html>

# **jQuery - DOM Manipulation**

-jQuery provides a number of methods to manipulate DOM in efficient way. You do not need to write big and complex code to set or get the content of any HTML element.

**jQuery DOM Manipulation**

jQuery provides methods such as **attr()**, **html()**, **text()** and **val()** which act as getters and setters to manipulate the content from HTML documents.

*Document Object Model (DOM) - is a W3C (World Wide Web Consortium) standard that allows us to create, change, or remove elements from the HTML or XML documents.*

Here are some basic operations which you can perform on DOM elements with the help of jQuery standard library methods −

* Extract the content of an element
* Change the content of an element
* Adding a child element under an existing element
* Adding a parent element above an existing element
* Adding an element before or after an existing element
* Replace an existing element with another element
* Delete an existing element
* Wrapping content with-in an element

We have already covered **attr()** method while discussing [jQuery Attributes](https://www.tutorialspoint.com/jquery/jquery-attributes.htm) and remaining DOM content manipulation methods **html()**, **text()** and **val()** will be discussed in this chapter.

**jQuery - Remove Elements**

-jQuery provides **remove()** and **empty()** methods to remove existing HTML elements from an HTML document.

**jQuery remove() Method**

-The jQuery **remove()** method removes the selected element(s) and it's child elements from the document.

Following is the syntax of the **remove()** method:

$(selector).remove();

You should use **remove()** method when you want to remove the element itself, as well as everything inside it.

### Synopsis

Consider the following HTML content:

<div class="container">

<h2>jQuery remove() Method</h2>

<div class="hello">Hello</div>

<div class="goodbye">Goodbye</div>

</div>

Now if we apply the **remove()** method as follows:

$( ".hello" ).remove();

It will produce following result:

<div class="container">

<h2>jQuery remove() Method</h2>

<div class="goodbye">Goodbye</div>

</div>

If we had any number of nested elements inside <div class="hello">, they would be removed, too.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>

    <title>Document</title>

</head>

<body>

    <div class="container">

        <h2>jQuery remove() Method</h2>

        <div class="hello">Hello</div>

        <div class="goodbye">Goodbye</div>

    </div>

    <br>

    <button>Remove Text</button>

    <script>

        $(document).ready(function() {

            $("button").click(function(){

                $( ".hello" ).remove();

            });

        });

    </script>

</body>

</html>

## Remove with Filter

We can also include a selector as an optional parameter for the **remove()** method. For example, we could rewrite the previous DOM removal code as follows:

Let's try the following example and verify the result:

Open Compiler

<!doctype html>

<html>

<head>

<title>The jQuery Example</title>

<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>

<script>

$(document).ready(function() {

$("button").click(function(){

$("div").remove(".hello");

});

});

</script>

</head>

<body>

<div class="container">

<h2>jQuery remove(selector) Method</h2>

<div class="hello">Hello</div>

<div class="goodbye">Goodbye</div>

</div>

<br>

<button>Remove Text</button>

</body>

</html>

[Explore our](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)**[latest online courses](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)**[and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)

## jQuery empty() Method

The jQuery **empty()** method is very similar to **remove()** which removes the selected element(s) and it's child elements from the document. This method does not accept any arguments.

Following is the syntax of the **empty()** method:

$(selector).empty();

You should use **empty()** method when you want to remove the element itself, as well as everything inside it.

### Synopsis

Consider the following HTML content:

<div class="container">

<h2>jQuery empty() Method</h2>

<div class="hello">Hello</div>

<div class="goodbye">Goodbye</div>

</div>

Now if we apply the **empty()** method as follows:

$( ".hello" ).empty();

It will produce following result:

<div class="container">

<h2>jQuery empty() Method</h2>

<div class="goodbye">Goodbye</div>

</div>

### Example

Let's try the following example and verify the result:

Open Compiler

<!doctype html>

<html>

<head>

<title>The jQuery Example</title>

<script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>

<script>

$(document).ready(function() {

$("button").click(function(){

$( ".hello" ).empty();

});

});

</script>

</head>

<body>

<div class="container">

<h2>jQuery empty() Method</h2>

<div class="hello">Hello</div>

<div class="goodbye">Goodbye</div>

</div>

<br>

<button>Remove Text</button>

</body>

</html>

**jQuery - Plugins**

-A plug-in is piece of code written in a standard JavaScript file. These files provide useful jQuery methods which can be used along with jQuery library methods.

-There are plenty of jQuery plug-in available which you can download from repository link at <https://jquery.com/plugins>.

**How to use Plugins**

-To make a plug-in's methods available to us, we include plug-in file very similar to jQuery library file in the <head> of the document.

-We must ensure that it appears after the main jQuery source file, and before our custom JavaScript code.

Following example shows how to include **jquery.plug-in.js** plugin −

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript"

src = "https://www.tutorialspoint.com/jquery/jquery-3.6.0.js">

</script>

<script src = "jquery.plug-in.js" type = "text/javascript"></script>

<script src = "custom.js" type = "text/javascript"></script>

<script type = "text/javascript" language = "javascript">

$(document).ready(function() {

.......your custom code.....

});

</script>

</head>

<body>

.............................

</body>

</html>

**How to develop a Plug-in / Custom**

-This is very simple to write your own plug-in. Following is the syntax to create a a method −

jQuery.fn.methodName = methodDefinition;

-Here *methodNameM* is the name of new method and *methodDefinition* is actual method definition.

The guideline recommended by the jQuery team is as follows −

* Any methods or functions you attach must have a semicolon (;) at the end.
* Your method must return the jQuery object, unless explicity noted otherwise.
* You should use this.each to iterate over the current set of matched elements - it produces clean and compatible code that way.
* Prefix the filename with jquery, follow that with the name of the plugin and conclude with .js.
* Always attach the plugin to jQuery directly instead of $, so users can use a custom alias via noConflict() method.

For example, if we write a plugin that we want to name *debug*, our JavaScript filename for this plugin is −

jquery.debug.js

The use of the **jquery.** prefix eliminates any possible name collisions with files intended for use with other libraries.

[Explore our](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)**[latest online courses](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)**[and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](https://www.tutorialspoint.com/latest/courses?utm_source=tutorialspoint&utm_medium=tutorials_3p&utm_campaign=internal" \t "_blank)

## Example

Following is a small plug-in to have warning method for debugging purpose. Keep this code in *jquery.debug.js* file −

jQuery.fn.warning = function() {

return this.each(function() {

alert('Tag Name:"' + $(this).prop("tagName") + '".');

});

};

Here is the example showing usage of warning() method. Assuming we put *jquery.debug.js* file in same directory of html page.

[Live Demo](http://tpcg.io/1wF90g)

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript"

src = "https://www.tutorialspoint.com/jquery/jquery-3.6.0.js">

</script>

<script src = "jquery.debug.js" type = "text/javascript">

</script>

<script type = "text/javascript" language = "javascript">

$(document).ready(function() {

$("div").warning();

$("p").warning();

});

</script>

</head>

<body>

<p>This is paragraph</p>

<div>This is division</div>

</body>

</html>

This would alert you with following result −

This is paragraph

This is division