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

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

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

<https://jquery.com/download/>

https://code.jquery.com/jquery-3.6.3.js

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)

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-3.6.1.min.js"></script>  
</head>

jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

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".

The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

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

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

## jQuery Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](https://www.w3schools.com/cssref/css_selectors.asp), and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

The #id Selector

The jQuery #*id* selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the HTML element:

$("#test")

## 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 |  |

## Functions In a Separate File

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>  
<script src="my\_jquery\_functions.js"></script>  
</head>

## What are Events?

All the different visitors' actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

* moving a mouse over an element
* selecting a radio button
* clicking on an element

The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".

Here are some common DOM events:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse Events** | **Keyboard Events** | **Form Events** | **Document/Window Events** |
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave |  | blur | unload |

## Commonly Used jQuery Event Methods

**$(document).ready()**

The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained in the [jQuery Syntax](https://www.w3schools.com/jquery/jquery_syntax.asp) chapter.

**click()**

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

The following example says: When a click event fires on a <p> element; hide the current <p> element:

**dblclick()**

The dblclick() method attaches an event handler function to an HTML element.

The function is executed when the user double-clicks on the HTML element:

**mouseenter()**

The mouseenter() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer enters the HTML element:

**mouseleave()**

The mouseleave() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer leaves the HTML element:

**mousedown()**

The mousedown() method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

**mouseup()**

The mouseup() method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

**hover()**

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

**focus()**

The focus() method attaches an event handler function to an HTML form field.

The function is executed when the form field gets focus:

**blur()**

The blur() method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

## The on() Method

The on() method attaches one or more event handlers for the selected elements.

Attach a click event to a <p> element:

# **jQuery Effects - Hide and Show**

## jQuery hide() and show()

With jQuery, you can hide and show HTML elements with the hide() and show() methods:

## jQuery toggle()

You can also toggle between hiding and showing an element with the toggle() method.

Shown elements are hidden and hidden elements are shown:

# **jQuery Effects - Fading**

## jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

* fadeIn()
* fadeOut()
* fadeToggle()
* fadeTo()

jQuery fadeIn() Method

The jQuery fadeIn() method is used to fade in a hidden element.

**Syntax:**

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

jQuery fadeOut() Method

The jQuery fadeOut() method is used to fade out a visible element.

**Syntax:**

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeOut() method with different parameters:

## jQuery Animations - The animate() Method

The jQuery animate() method is used to create custom animations.

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

## jQuery Callback Functions

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.

To prevent this, you can create a callback function.

A callback function is executed after the current effect is finished.

Typical syntax: **$(*selector*).hide(*speed,callback*);**

## jQuery Method Chaining

Until now we have been writing jQuery statements one at a time (one after the other).

However, there is a technique called chaining, that allows us to run multiple jQuery commands, one after the other, on the same element(s).

**Tip:** This way, browsers do not have to find the same element(s) more than once.

To chain an action, you simply append the action to the previous action.

The following example chains together the css(), slideUp(), and slideDown() methods. The "p1" element first changes to red, then it slides up, and then it slides down:

Set Content - text(), html(), and val()

We will use the same three methods from the previous page to **set content**:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

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