**INTRODUCTION**

jQuery is a JavaScript Library.

jQuery greatly simplifies JavaScript programming.

jQuery is easy to learn.

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

**What You Should Already Know**

Before you start studying jQuery, you should have a basic knowledge of:

* [HTML](https://www.w3schools.com/html/default.asp)
* [CSS](https://www.w3schools.com/css/default.asp)
* [JavaScript](https://www.w3schools.com/js/default.asp)

The jQuery library contains the following features:

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

# Get Started

**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

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

## jQuery CDN

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.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...*  
  
});

# jQuery Selectors

jQuery selectors are one of the most important parts of the jQuery library.

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.

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

**The element Selector**

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

$("p")

**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")

**The .class Selector**

The jQuery *.class* selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

$(".test")

**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

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 |

**jQuery Syntax For Event Methods**

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

$("p").click();

**(or)**

$("p").click(function(){  
  // action goes here!!  
});

## 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.

**dblclick()**

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

**mouseenter()**

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

**mouseleave()**

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

**mousedown()**

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

**mouseup()**

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

**hover()**

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

**focus()**

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

**blur()**

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

## The on() Method

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

## jQuery Effects

# Hide and Show

**Syntax:**

$(*selector*).hide(*speed,callback*);  
  
$(*selector*).show(*speed,callback*);

## jQuery toggle()

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

**Syntax:**

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

# 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*);

**jQuery fadeToggle() Method**

The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods.

If the elements are faded out, fadeToggle() will fade them in.

If the elements are faded in, fadeToggle() will fade them out.

**Syntax:**

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

**jQuery fadeTo() Method**

The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).

**Syntax:**

$(*selector*).fadeTo(*speed,opacity,callback*);

# Sliding

**jQuery Sliding Methods**

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

* slideDown()
* slideUp()
* slideToggle()

**jQuery slideDown() Method**

The jQuery slideDown() method is used to slide down an element.

**Syntax:**

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

**jQuery slideUp() Method**

The jQuery slideUp() method is used to slide up an element.

**Syntax:**

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

**jQuery slideToggle() Method**

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.

If the elements have been slid down, slideToggle() will slide them up.

If the elements have been slid up, slideToggle() will slide them down.

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

# Animation

**jQuery Animations - The animate() Method**

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

**Syntax:**

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

# jQuery Stop Animations

**jQuery stop() Method**

The jQuery stop() method is used to stop an animation or effect before it is finished.

The stop() method works for all jQuery effect functions, including sliding, fading and custom animations.

**Syntax:**

$(*selector*).stop(*stopAll,goToEnd*);

## 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).

## jQuery HTML

## jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

## Get Attributes - attr()

The jQuery attr() method is used to get attribute values.

**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

## Set Attributes - attr()

The jQuery attr() method is also used to set/change attribute values.

# jQuery - Add Elements

**Add New HTML Content**

We will look at four jQuery methods that are used to add new content:

* append() - Inserts content at the end of the selected elements
* prepend() - Inserts content at the beginning of the selected elements
* after() - Inserts content after the selected elements
* before() - Inserts content before the selected elements

**jQuery append() Method**

The jQuery append() method inserts content AT THE END of the selected HTML elements.

## jQuery prepend() Method

The jQuery prepend() method inserts content AT THE BEGINNING of the selected HTML elements.

## jQuery after() and before() Methods

The jQuery after() method inserts content AFTER the selected HTML elements.

The jQuery before() method inserts content BEFORE the selected HTML elements.

**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

**jQuery remove() Method**

The jQuery remove() method removes the selected element(s) and its child elements.

## jQuery empty() Method

The jQuery empty() method removes the child elements of the selected element(s).

# jQuery - Get and Set CSS Classes

**jQuery Manipulating CSS**

jQuery has several methods for CSS manipulation. We will look at the following methods:

* addClass() - Adds one or more classes to the selected elements
* removeClass() - Removes one or more classes from the selected elements
* toggleClass() - Toggles between adding/removing classes from the selected elements
* css() - Sets or returns the style attribute

**Example Stylesheet**

The following stylesheet will be used for all the examples on this page:

.important {  
  font-weight: bold;  
  font-size: xx-large;  
}  
  
.blue {  
  color: blue;  
}

**jQuery css() Method**

The css() method sets or returns one or more style properties for the selected elements.

**Return a CSS Property**

To return the value of a specified CSS property, use the following syntax:

css("*propertyname*");

**Set a CSS Property**

To set a specified CSS property, use the following syntax:

css("*propertyname*","*value*");

**Set Multiple CSS Properties**

To set multiple CSS properties, use the following syntax:

css({"*propertyname*":"*value*","*propertyname*":"*value*",...});

# jQuery - Dimensions

**jQuery Dimension Methods**

jQuery has several important methods for working with dimensions:

* width()
* height()
* innerWidth()
* innerHeight()
* outerWidth()
* outerHeight()

**jQuery Dimensions**



## jQuery width() and height() Methods

The width() method sets or returns the width of an element (excludes padding, border and margin).

The height() method sets or returns the height of an element (excludes padding, border and margin).

## jQuery innerWidth() and innerHeight() Methods

The innerWidth() method returns the width of an element (includes padding).

The innerHeight() method returns the height of an element (includes padding).

## jQuery outerWidth() and outerHeight() Methods

The outerWidth() method returns the width of an element (includes padding and border).

The outerHeight() method returns the height of an element (includes padding and border).

# jQuery - AJAX Introduction

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

## What is AJAX?

AJAX = Asynchronous JavaScript and XML.

## What About jQuery and AJAX?

jQuery provides several methods for AJAX functionality.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

# jQuery - AJAX load() Method

**jQuery load() Method**

The jQuery load() method is a simple, but powerful AJAX method.

The load() method loads data from a server and puts the returned data into the selected element.

**Syntax:**

$(*selector*).load(*URL,data,callback*);

The optional callback parameter specifies a callback function to run when the load() method is completed. The callback function can have different parameters:

* responseTxt - contains the resulting content if the call succeeds
* statusTxt - contains the status of the call
* xhr - contains the XMLHttpRequest object

# jQuery - AJAX get() and post() Methods

The jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request.

**HTTP Request: GET vs. POST**

Two commonly used methods for a request-response between a client and server are: GET and POST.

* **GET** - Requests data from a specified resource
* **POST** - Submits data to be processed to a specified resource

GET is basically used for just getting (retrieving) some data from the server. **Note:** The GET method may return cached data.

POST can also be used to get some data from the server. However, the POST method NEVER caches data, and is often used to send data along with the request.

**jQuery $.get() Method**

The $.get() method requests data from the server with an HTTP GET request.

**Syntax:**

$.get(*URL,callback*);

**jQuery $.post() Method**

The $.post() method requests data from the server using an HTTP POST request.

**Syntax:**

$.post(*URL,data,callback*);

## jQuery Misc

# jQuery - The noConflict() Method

## jQuery and Other JavaScript Frameworks

As you already know; jQuery uses the $ sign as a shortcut for jQuery.

If two different frameworks are using the same shortcut, one of them might stop working.

The jQuery team have already thought about this, and implemented the noConflict() method.

**The jQuery noConflict() Method**

The noConflict() method releases the hold on the $ shortcut identifier, so that other scripts can use it.

There are many other popular JavaScript frameworks like: Angular, Backbone, Ember, Knockout, and more.

## jQuery Filters

Use jQuery to filter/search for specific elements.