**jQuery Notes**

You can write both javascript and jquery code since they are the same language

**Adding jQuery**

You can download it or use CDN, I will use a CDN

CDN:

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

Scroll down to Using jQuery with a CDN

Click <https://code.jquery.com/>

Choose which version of jQuery you want to use

We will use jQuery 3.x and click minified

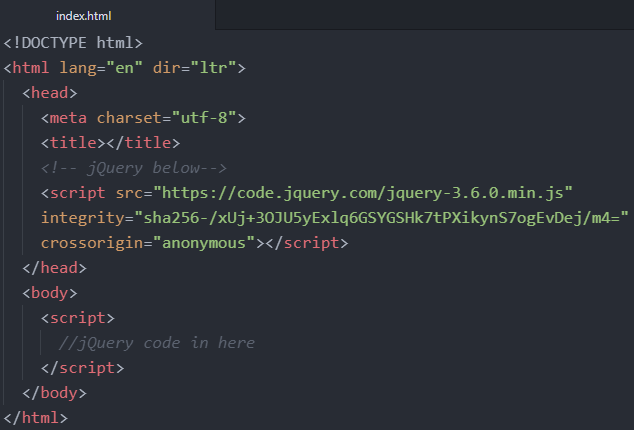
Copy the following:

<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

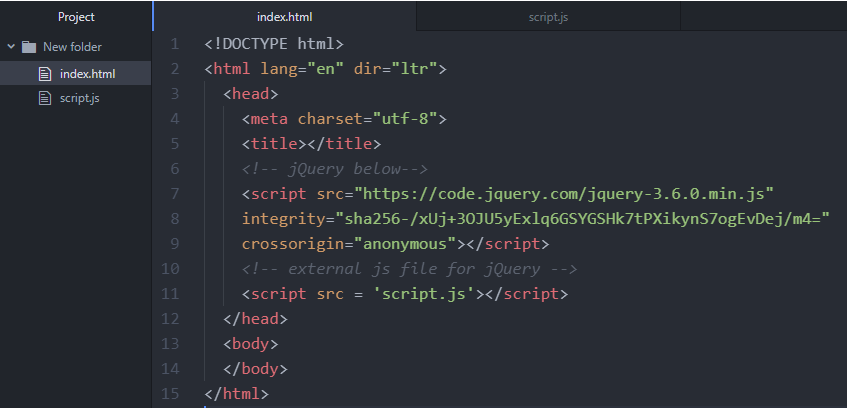


**Writing jQuery Code**

Option 1: Write jQuery code through a script tag in the body (makes sure it comes after the jQuery CDN link)

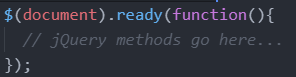


Option 2: External file

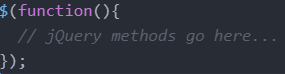


**Document Ready Event**

A document ready event prevents any jQuery code from running before the document is finished loading (is ready).



or a shorthand but less explicit way is:



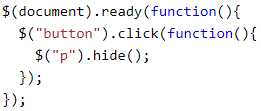
**Selectors**

jQuery selectors allow you to select and manipulate HTML element(s) based on their name, id, classes, etc.

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

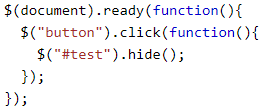
The Element Selector

Syntax: $("p") This selects all paragraph elements



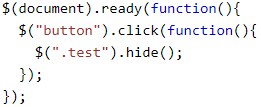
The id Selector

Syntax: $("#id\_name") This selects a specific element with the corresponding id



The class Selector

Syntax: $(".class\_name") This selects all elements with the corresponding class



**Event Methods**

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

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 |

click()

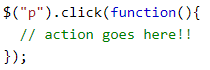
The click() method attaches an event handler function to an HTML element.

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

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



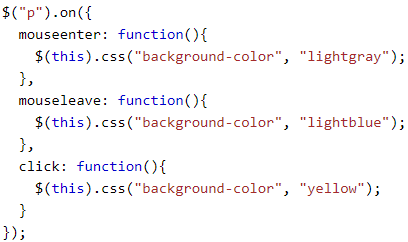
The next step is to define what should happen when the event fires. You must pass a function to the event:



on()

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

Attach a click event to a <p> element and attach multiple event handlers to it.



link to all jQuery events

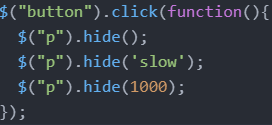
<https://api.jquery.com/category/events/>

**Effects**

hide() and show()

hide and show HTML elements with the hide() and show() methods

Specify a speed of hiding/showing by adding a number inside the brackets that represents the time in milliseconds

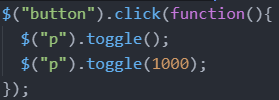


toggle()

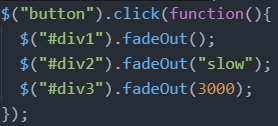
You can also alternate between hiding and showing an element with the toggle() method.

Shown elements are hidden and hidden elements are shown.

Specify a speed of the toggle by adding a number inside the brackets that represents the time in milliseconds



fadeIn() and fadeOut()

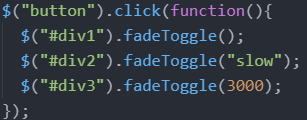


fadeToggle()

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

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

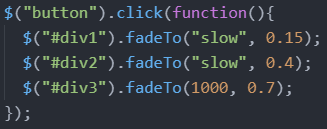
If the elements are faded in, fadeToggle() will fade them out.



fadeTo()

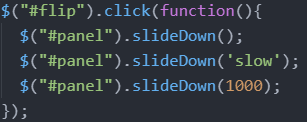
The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).

Closer to 0, more opaque. Closer to 1, more solid.



slideDown() and slideUp()

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

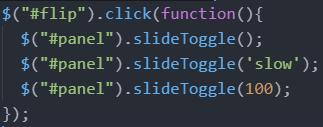


slideToggle()

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.



animate()

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

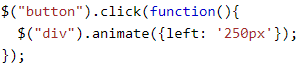
Syntax: $(selector).animate({params},speed,callback);

The required params parameter defines the CSS properties to be animated.

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

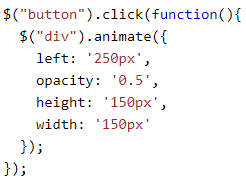
The optional callback parameter is a function to be executed after the animation completes.

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!

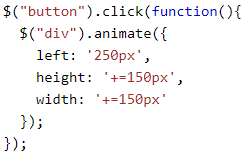


Multiple properties can be animated at the same time

All property names must be camel-cased when used with the animate() method: You will need to write paddingLeft instead of padding-left, marginRight instead of margin-right, and so on.

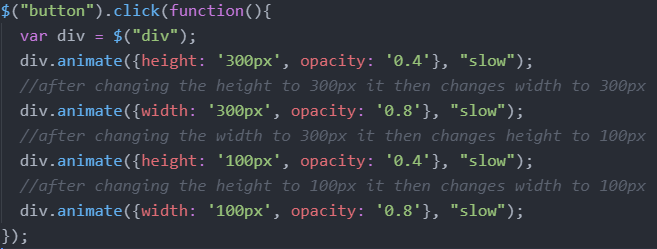


It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value.



By default, jQuery comes with queue functionality for animations.

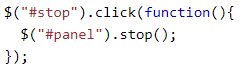
This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animation calls ONE by ONE.



stop()

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

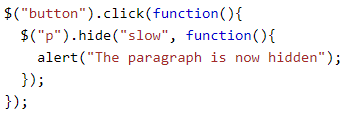
The stop() method works for all jQuery effect functions, including sliding, fading and custom animations.



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.

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

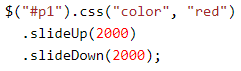


Chaining

allows us to run multiple jQuery commands, one after the other, on the same element(s)



The lines can get quite long so we can also do the following (jQuery throws away extra whitespace and executes the lines above as one long line of code)



**HTML**

jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

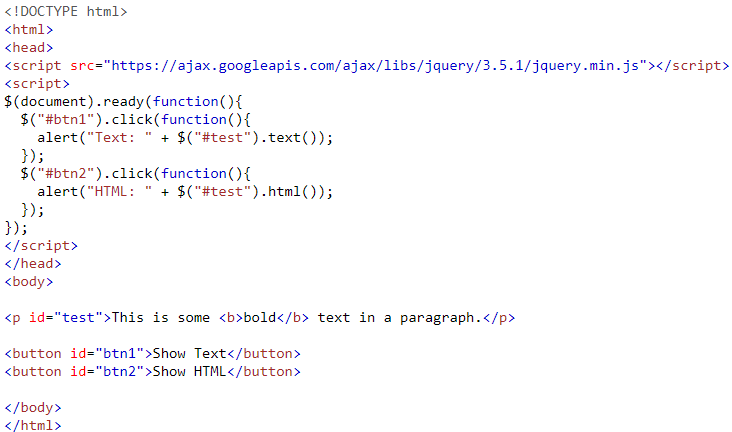
Get/Set

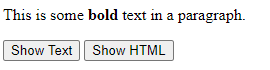
text() - Sets or returns the text content of selected elements

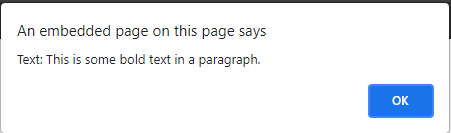
html() - Sets or returns the content of selected elements (including HTML markup)

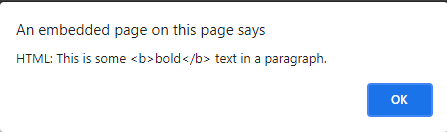
val() - Sets or returns the value of form fields

The following example demonstrates how to get content with the jQuery text() and html() methods:

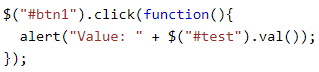




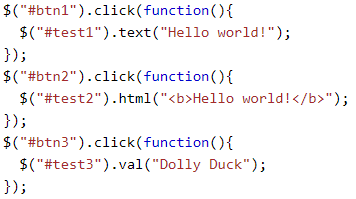




The following example demonstrates how to get the value of an input field with the jQuery val() method:



The following example demonstrates how to set content with the jQuery text(), html(), and val() methods



Adding Elements

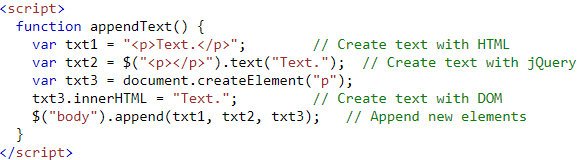
append() - Inserts content at the end of the selected elements

prepend() - Inserts content at the beginning of the selected elements

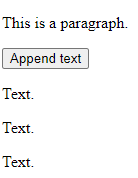
after() - Inserts content after the selected elements

before() - Inserts content before the selected elements

The below example creates a new html p element using html, jquery and javascript.



Below is how the output looks:



Remove Elements

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



The jQuery remove() method also accepts one parameter, which allows you to filter the elements to be removed. The parameter can be any of the jQuery selector syntaxes.

This example removes all <p> elements with class="test" and class="demo":



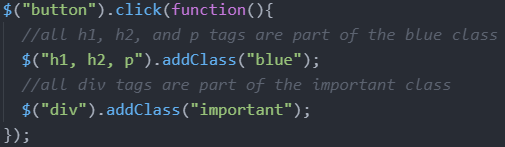
empty() - Removes the child elements from the selected element



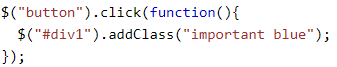
**CSS**

Get and Set Classes

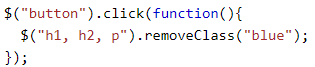
addClass() - Adds one or more classes to the selected elements



You can also specify multiple classes within the addClass() method



removeClass() - Removes one or more classes from the selected elements



css() methods

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

Syntax: css("propertyname","value");

The following example will return the background-color value of the FIRST matched element:



Set Multiple CSS Properties

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

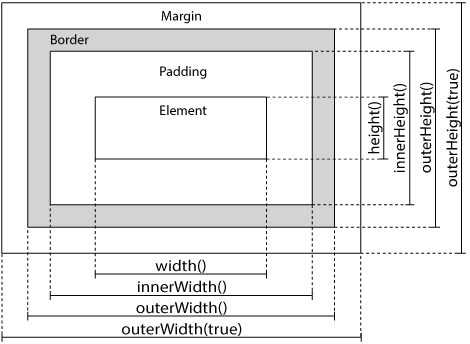
The following example will set a background-color and a font-size for ALL matched elements:

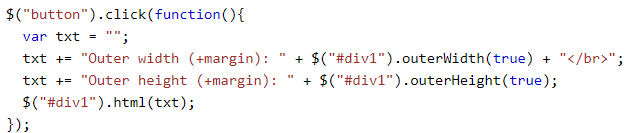


**Dimensions**

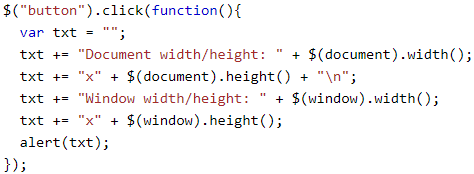
Get

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





The following example returns the width and height of the document (the HTML document) and window (the browser viewport):



Set

The following example sets the width and height of a specified <div> element:

