

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+3OJU5yExlq6GSYGS7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
```

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- jQuery below -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"
      integrity="sha256-/xUj+3OJU5yExlq6GSYGS7tPXikynS7ogEvDej/m4="
      crossorigin="anonymous"></script>
  </head>
  <body>

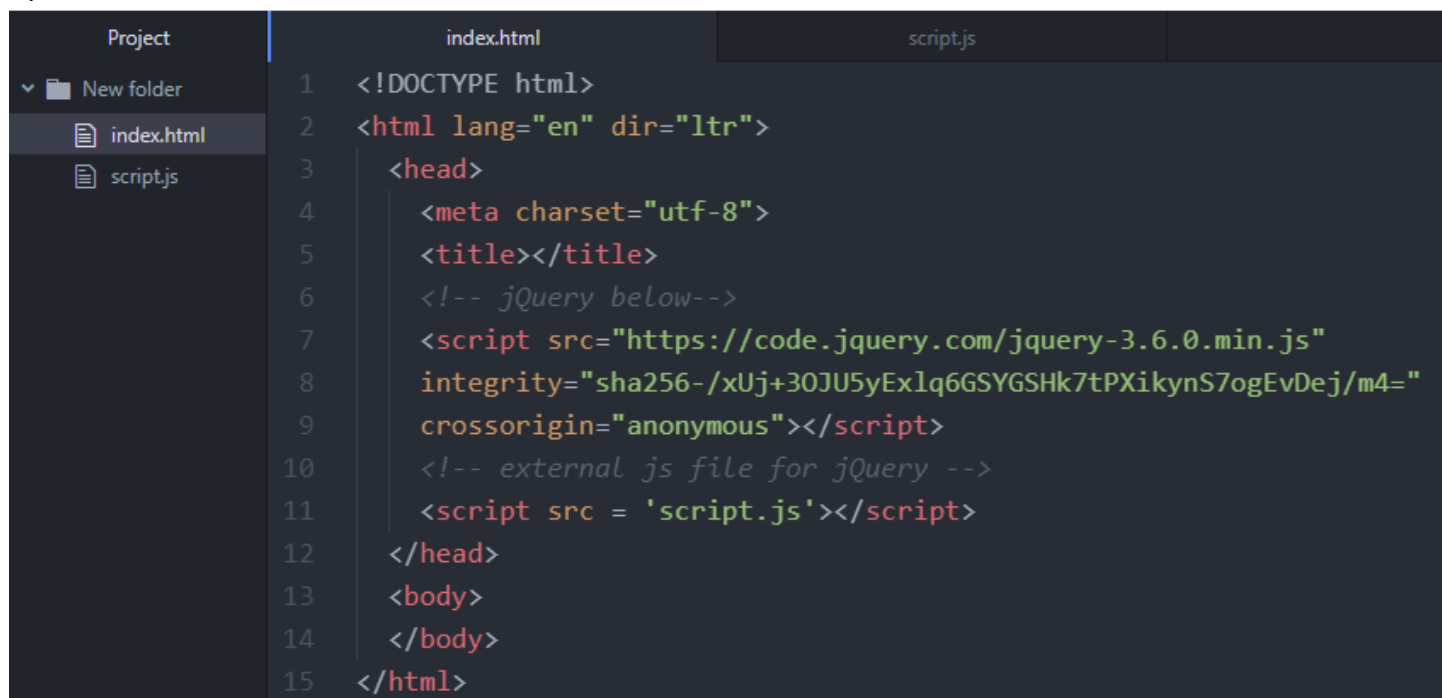
</body>
</html>
```

Writing jQuery Code

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

```
index.html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- jQuery below -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"
      integrity="sha256-/xUj+3OJU5yExlq6GSYGS7tPXikynS7ogEvDej/m4="
      crossorigin="anonymous"></script>
  </head>
  <body>
    <script>
      //jQuery code in here
    </script>
  </body>
</html>
```

Option 2: External file



```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6     <!-- jQuery below-->
7     <script src="https://code.jquery.com/jquery-3.6.0.min.js"
8       integrity="sha256-/xUj+30JU5yExlq6GSYGS7Hk7tPXikynS7ogEvDej/m4="
9       crossorigin="anonymous"></script>
10    <!-- external js file for jQuery -->
11    <script src = 'script.js'></script>
12  </head>
13  <body>
14  </body>
15 </html>
```

Document Ready Event

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

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

or a shorthand but less explicit way is:

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

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

```
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
```

The id Selector

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

```
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
```

The class Selector

Syntax: `$(".class_name")` This selects all elements with the corresponding class

```
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
```

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:

```
$("p").click();
```

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

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

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.

```
$("p").on({
    mouseenter: function(){
        $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
        $(this).css("background-color", "lightblue");
    },
    click: function(){
        $(this).css("background-color", "yellow");
    }
});
```

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

```
$("button").click(function(){
    $("p").hide();
    $("p").hide('slow');
    $("p").hide(1000);
});
```

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

```
$("button").click(function(){
    $("p").toggle();
    $("p").toggle(1000);
});
```

fadeIn() and fadeOut()

```
$("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
});
```

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.

```
$("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
});
```

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.

```
$("button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo(1000, 0.7);
});
```

slideDown() and slideUp()

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

```
$("#flip").click(function(){
    $("#panel").slideDown();
    $("#panel").slideDown('slow');
    $("#panel").slideDown(1000);
});
```

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.

```
$("#flip").click(function(){
    $("#panel").slideToggle();
    $("#panel").slideToggle('slow');
    $("#panel").slideToggle(100);
});
```

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!

```
$("#button").click(function(){
    $("#div").animate({left: '250px'});
});
```

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.

```
$("#button").click(function(){
    $("#div").animate({
        left: '250px',
        opacity: '0.5',
        height: '150px',
        width: '150px'
    });
});
```

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.

```

$("button").click(function(){
    $("div").animate({
        left: '250px',
        height: '+=150px',
        width: '+=150px'
    });
});

```

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.

```

$("button").click(function(){
    var div = $("div");
    div.animate({height: '300px', opacity: '0.4'}, "slow");
    //after changing the height to 300px it then changes width to 300px
    div.animate({width: '300px', opacity: '0.8'}, "slow");
    //after changing the width to 300px it then changes height to 100px
    div.animate({height: '100px', opacity: '0.4'}, "slow");
    //after changing the height to 100px it then changes width to 100px
    div.animate({width: '100px', opacity: '0.8'}, "slow");
});

```

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.

```

$("#stop").click(function(){
    $("#panel").stop();
});

```

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);`

```

$("button").click(function(){
    $("p").hide("slow", function(){
        alert("The paragraph is now hidden");
    });
});

```

Chaining

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

```

$("#p1").css("color", "red").slideUp(2000).slideDown(2000);

```

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)

```

$("#p1").css("color", "red")
    .slideUp(2000)
    .slideDown(2000);

```

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:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```

This is some **bold** text in a paragraph.

Show Text

Show HTML

An embedded page on this page says

Text: This is some bold text in a paragraph.

OK

An embedded page on this page says

HTML: This is some bold text in a paragraph.

OK

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

```
$("#btn1").click(function(){
  alert("Value: " + $("#test").val());
});
```

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

```
$("#btn1").click(function(){
    $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
});
```

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.

```
<script>
function appendText() {
    var txt1 = "<p>Text.</p>";           // Create text with HTML
    var txt2 = $("<p></p>").text("Text."); // Create text with jQuery
    var txt3 = document.createElement("p");
    txt3.innerHTML = "Text.";           // Create text with DOM
    $("body").append(txt1, txt2, txt3); // Append new elements
}
</script>
```

Below is how the output looks:

This is a paragraph.

Append text

Text.

Text.

Text.

Remove Elements

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

```
$("#div1").remove();
```

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":

```
$("p").remove(".test, .demo");
```

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

```
$("#div1").empty();
```


CSS

Get and Set Classes

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

```
$("#button").click(function(){
  //all h1, h2, and p tags are part of the blue class
  $("h1, h2, p").addClass("blue");
  //all div tags are part of the important class
  $("div").addClass("important");
});
```

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

```
$("#button").click(function(){
  $("#div1").addClass("important blue");
});
```

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

```
$("#button").click(function(){
  $("h1, h2, p").removeClass("blue");
});
```

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:

```
$("#p").css("background-color");
```

Set Multiple CSS Properties

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

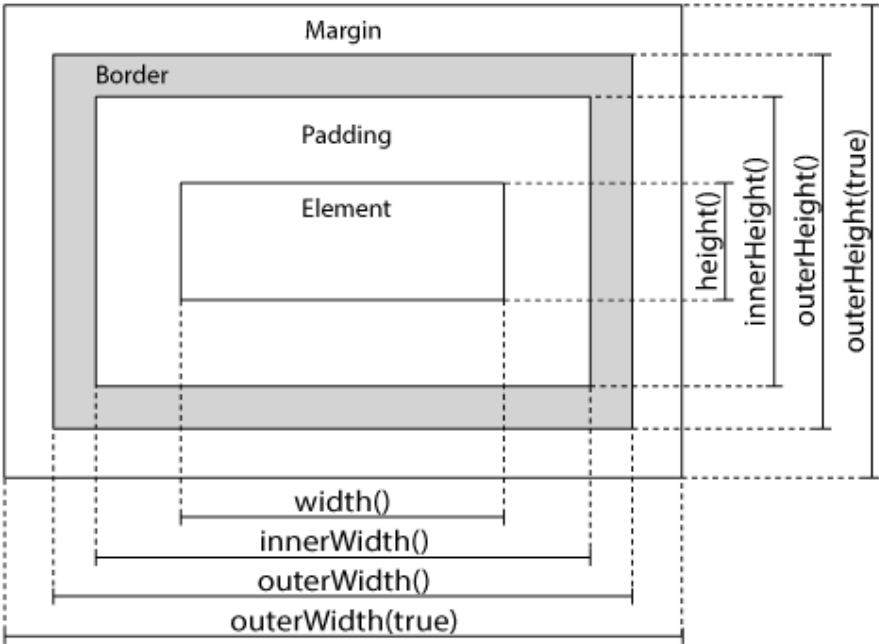
The following example will set a background-color and a font-size for **ALL** matched elements:

```
$("#p").css({"background-color": "yellow", "font-size": "200%"});
```

Dimensions

Get

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



```
$("#button").click(function(){
    var txt = "";
    txt += "Outer width (+margin): " + $("#div1").outerWidth(true) + "</br>";
    txt += "Outer height (+margin): " + $("#div1").outerHeight(true);
    $("#div1").html(txt);
});
```

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

```
$("#button").click(function(){
    var txt = "";
    txt += "Document width/height: " + $(document).width();
    txt += "x" + $(document).height() + "\n";
    txt += "Window width/height: " + $(window).width();
    txt += "x" + $(window).height();
    alert(txt);
});
```

Set

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

```
$("#button").click(function(){
    $("#div1").width(500).height(500);
});
```