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".

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

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")

**Example**

When a user clicks on a button, the element with id="test" will be hidden:

Example

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

## **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")

**Example**

When a user clicks on a button, the elements with class="test" will be hidden:

### Example

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

jQuery is a library that’s built upon JavaScript, using the DOM. Takes a 100k of data to download and store on the phone or computer and needs to be called up every time you use the website.

The advantages of using jQuery are:

* **Separates Javascript and HTML:** Instead of using HTML attributes to call Javascript functions for event handling, jQuery allows all event-handling functions to be done purely in Javascript. Thus, the HTML tags and Javascript can be completely separated.
* **Brevity and Clarity:** jQuery provides various syntaxes that promotes brevity and clarity, e.g. chaining effects and actions, shorthand methods.
* **Eliminates cross-browser incompatibilities:** The Javascript engines of different browsers differ slightly, so Javascript code that works for one browser may not work on the other. jQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.
* **Extensible:** jQuery makes extending the framework very simple. New events, elements and methods can be easily added and then reused as a plugin.
* document.body.style.background = "#ccc"; --- **JavaScript**
* $("body").css("background", "#ccc"); --- **jQuery**

**jQuery Library** | [Top](http://cis2.oc.ctc.edu/oc_apps/Westlund/CIS255-xbook/xbook.php?unit=07&proc=page&numb=1#top)

The jQuery library is a single JavaScript file, containing all of its common DOM, event, effects, and Ajax functions. It can be included within a web page by linking to a local copy, or to one of the many copies available from public servers. jQuery has a CDN hosted by MaxCDN (moved from MediaTemple and, before that, Amazon). Google and Microsoft host it as well. [Wikipedia] Since IE 8 is still relatively common, we recommend using the 1.x version unless you are certain no IE 6/7/8 users are visiting the site. [jQuery] [View jQuery libraries](http://jquery.com/download/).

The jQuery library can be located on your server or from many places on the Web. If your sever, you know what is there. If the user is far away, the jQuery file may take longer to load since a closer copy may be available from another server being managed by jQuery or Google.

**Differences Between JavaScript and jQuery** | [Top](http://cis2.oc.ctc.edu/oc_apps/Westlund/CIS255-xbook/xbook.php?unit=07&proc=page&numb=1#top)

The examples below do exactly the same thing. Note how the developer is abstracted from the JavaScript code in the jQuery example.

**jQuery show/hide example:** Note the reference by id in the HTML and jQuery code in managing the events and the lack of any JavaScript code!

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#hide").click(function(){

$("#demo").hide();

});

$("#show").click(function(){

$("#demo").show();

});

});

</script>

<p>

<a id="hide" href="#">Hide</a>

<a id="show" href="#">Show</a>

</p>

<p id="demo">Watch this space!</p>

Top of Form



Bottom of Form

**JavaScript show/hide example:** Note the use of onClick in the link and the use of getElementById() to identify the targets. Note the use of the DOM in JavaScript to facilitate the events.

<script>

function hide(id) {

document.getElementById(id).style.display = "none";

}

function show(id) {

document.getElementById(id).style.display = "block";

}

</script>

<p>

<a id="hide" href="#" onClick="hide('demo')">Hide</a>

<a id="show" href="#" onClick="show('demo')">Show</a>

</p>

<p id="demo">Watch this space!</p>

Top of Form



Bottom of Form

In programming, a variable is a container for a value. You can think of variables as little containers for information that live in a computer’s memory. Information stored in variables, such as a username, account number, or even personalized greeting can then be found in memory.

Variables also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves.

In short, variables label and store data in memory. There are only a few things you can do with variables:

1. Create a variable with a descriptive name.
2. Store or update information stored in a variable.
3. Reference or “get” information stored in a variable.

It is important to distinguish that variables are not values; they contain values and represent them with a name. Observe the diagram with the colored boxes. Each box represents variables; the values are represented by the content, and the name is represented with the label.

In this lesson, we will cover how to use the var, let, and const keywords to create variables.