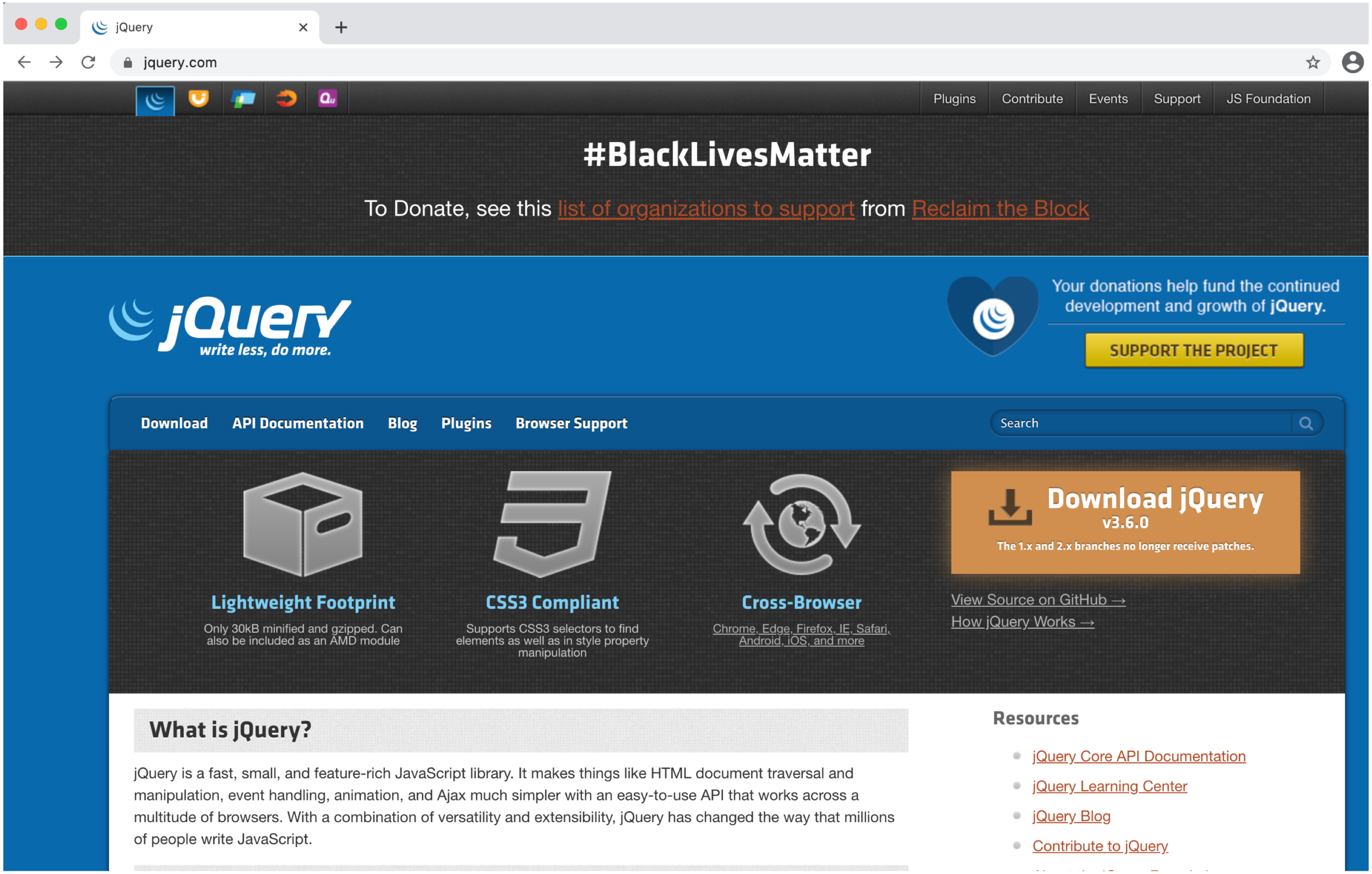
jQuery

jQuery

- Resources:
  - https://jquery.com/
  - https://www.w3schools.com/jquery/default.asp

Latest Version: 3.6.0



#BlackLivesMatter

To Donate, see this list of organizations to support from Reclaim the Block

Your donations help fund the continued development and growth of **jQuery.**

Lightweight Footprint
Only 30kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser
Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more

Download jQuery
v3.6.0
The 1.x and 2.x branches no longer receive patches.

View Source on GitHub →
How jQuery Works →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Resources

- jQuery Core API Documentation
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

jQuery

- **jQuery** is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.
- **jQuery is designed to change the way that you write JavaScript.**
- **jQuery** provides several methods for AJAX functionality (Same job but easier with different browsers).

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

Adding jQuery to Your Web Pages

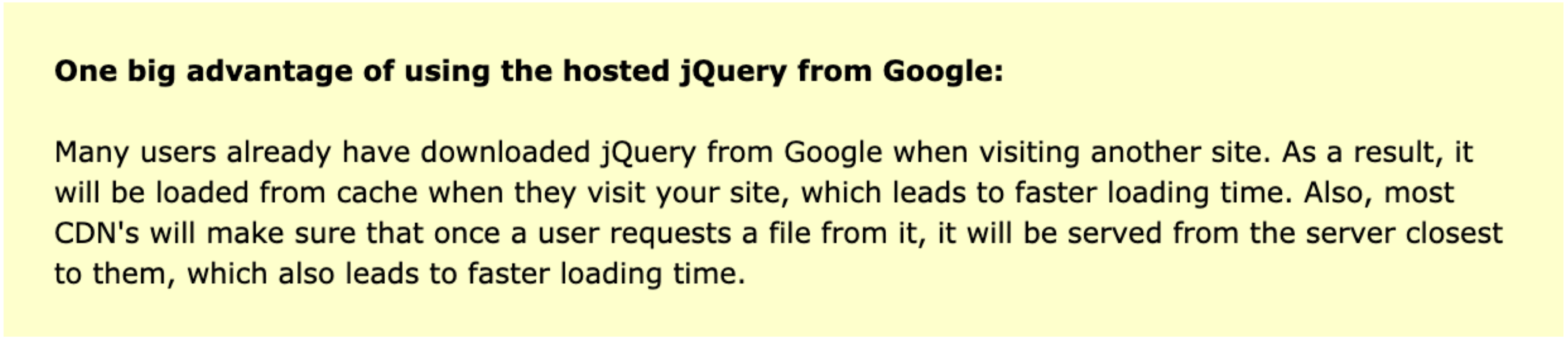
- jQuery library is a single JavaScript file that can be downloaded from jQuery.com
  - Place the downloaded file in the same directory as the pages where you wish to use it.
  - Reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section)

```
<script src="jquery-3.6.0.min.js"></script>
```

- If you don't want to download and host jQuery yourself, you can include it from a CDN such as Google CDN:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp



One big advantage of using the hosted jQuery from Google:

Many users already have downloaded jQuery from Google when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

jQuery Syntax

- Basic syntax is: **$(*selector*).*action*()**
  - A $ sign to define/access jQuery
  - A (*selector*) to "query (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".

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

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 id, classes, types, attributes, values of attributes and much more.
- It's based on the existing **CSS Selectors.**
- Examples:

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

jQuery Events

- All the different visitor's **actions** that a web page can respond to are called **events.**
- Here are some common jQuery 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

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

Single Event

- After choosing a selector and an event, define what should happen when the event fires.

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

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

Multiple Events

- You can attach multiple event handlers to one element by using the on() method
  - Example

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

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

The Document Ready Event

- All jQuery methods are put inside a document ready event:

```
$(document).ready(function(){

   // ALL jQuery methods go here...

});
```

  - Example:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jq
uery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
}); </script>
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

The Document Ready Event

- This prevents jQuery code from running before the document is finished loading.

- Examples of actions that can fail if run before the document is fully loaded:
  - Hiding an element that is not created yet
  - Getting the size of an image that is not loaded yet

jQuery with Ajax

$(selector).load(url, data, callback)	
Loads data from the server and puts it into the selected element.	
$.get(url, data, callback, type)	Returns: XMLHttpRequest
Load data from the server using an HTTP GET request.	
$.post(url, data, callback, type)	Returns: XMLHttpRequest
Load data from the server using an HTTP POST request.	
$.ajax(options)	Returns: XMLHttpRequest
Load data from the server using an HTTP request.	

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$(selector).load

Loads data from the server and puts it in the selected element

load(url, [data], [callback])

url	The URL of the file to load.
data (optional)	A set of query string key/value pairs that will be sent along with the request to the server.
callback (optional)	A function to be executed whenever the data is loaded (parameters: response text, status of the call, and the XMLHttpRequest object).

```
$("#div1").load("demo_test.txt");
```

```
$('#results').load('mypage.php', { name : this.value });
```

```
$("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){
    if(statusTxt == "success")
      alert("External content loaded successfully!");
    if(statusTxt == "error")
      alert("Error: " + xhr.status + ": " + xhr.statusText);
 });
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

Examples

https://www.w3schools.com/jquery/jquery_ajax_load.asp

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$.get

$.get(url, [data], [callback], [type])

- An easy way to send a simple GET request to a server.
- A callback function will be executed when the request is complete (and only if the response has a successful response code).

url	The URL of the page to load.
data (optional)	Key/value pairs that will be sent to the server with the request.
callback (optional)	A function to be executed whenever the data is loaded successfully.
type (optional)	Type of data to be returned to callback function: "xml", "html", "script", "json", or "text".

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$.get

- Example:

```
$.get("test.php",
      { name: "John", time: "2pm" },
      function(data){
            alert("Data Loaded: " + data);
      },
      “text”
);
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$.post

$.post(url, [data], [callback], [type])

- Very similar to $.get()

```
$.post("test.php",
     { name: "John", time: "2pm" },
     function(data){
          alert("Data Loaded: " + data);
     },
     “text”
);
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$.ajax

All jquery ajax methods use this code. Used for requests where other methods cannot be used.

$.ajax(options)

`options`: A set of key/value pairs that configure the Ajax request. All options are optional.

- Example:

```
$.ajax({
      type: "POST",
      url: "some.php",
      data: "name=John&location=Boston",
      success: function(msg){
            alert( "Data Saved: " + msg );
      }
});
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

$.ajax

- type: Specifies the type of request. (GET or POST)
- url: Specifies the URL to send the request to. Default is the current page
- data: Specifies data to be sent to the server
- success(result,status,xhr): A function to be run when the request succeeds

- Example:
  - http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_ajax

```html
<!DOCTYPE html>
<html>
<body>

<div id="demo"><h2>Let AJAX change this
text</h2></div>

<button type="button" onclick="loadDoc()">Change
Content</button>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200)
{
      document.getElementById("demo").innerHTML =
xhttp.responseText;
    }
  }
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

w3schools jQuery Tutorial:
https://www.w3schools.com/jquery/default.asp

```html
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/
1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.ajax({url: "demo_ajax_load.txt", async: true,
success: function(result){
            $("div").html(result);
        }});
    });
});
</script>
</head>
<body>

<div><h2>Let AJAX change this text</h2></div>

<button>Change Content</button>

</body>
</html>
```