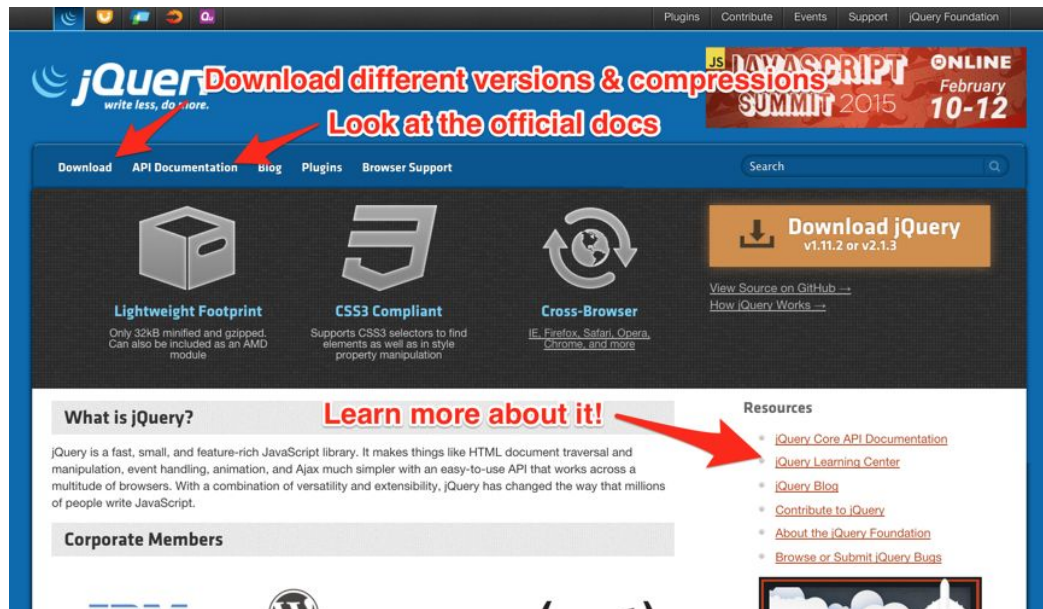


jQuery

An Introduction

jQuery Website



jquery.com/

Including jQuery

Do one of the following:

1. Download and include in your project folder as a relative linked js file.
2. Use a CDN to get jQuery - allows caching across multiple web sites.

Including jQuery

How to choose?

- For development, use it local and don't use the minified version.
- For production (live code), use the CDN and the minified version.

Examples

Including the development version:

```
<script src="jquery-2.1.3.js"></script>
```

To find a CDN: <https://code.jquery.com/>

```
<script src="https://code.jquery.com/jquery-2.1.4.min.js">  
</script>
```

Why use a CDN?

1. Decreased latency – CDNs have servers world wide and automatically target the closest one.
2. Increased parallelism – Browsers only allow so many connections to a server to avoid overload.
3. Better caching – If another site previously used jQuery, then it won't be downloaded again.

What does minified mean?

Removes all unnecessary characters - mainly whitespace & comments - from source code without changing the functionality.

It also typically does things like reducing the size of variable names to single characters.

```

/*!
 * jQuery JavaScript Library v2.1.3
 * http://jquery.com/
 *
 * Includes Sizzle.js
 * http://sizzlejs.com/
 *
 * Copyright 2005, 2014 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2014-12-18T15:11Z
 */

(function( global, factory ) {

    if ( typeof module === "object" && typeof module.exports === "object" ) {
        // For CommonJS and CommonJS-like environments where a proper `window`
        // is present, execute the factory and get jQuery.
        // For environments that do not have a `window` with a `document`
        // (such as Node.js), expose a factory as module.exports.
        // This accentuates the need for the creation of a real `window`.
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info.
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            };
    }

```

Development jQuery 2.1.3 – 242 KB file size


```

/*! jQuery v2.1.3 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?
module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new Error("jQuery
requires a window with a document");return b(a)}:b(a)}("undefined"!=typeof window?
window:this,function(a,b){var c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h=
{},i=h.toString,j=h.hasOwnProperty,k={},l=a.document,m="2.1.3",n=function(a,b){return new
n.fn.init(a,b)},o=/^\s\uFEFF\xA0+|[\s\uFEFF\xA0]+$/g,p=/^ms-/,q=/-([\da-
z])/gi,r=function(a,b){return b.toUpperCase()};n.fn=n.prototype=
{jquery:m,constructor:n,selector:"",length:0,toArray:function(){return
d.call(this)},get:function(a){return null!=a?0>a?
this[a+this.length]:this[a]:d.call(this)},pushStack:function(a){var
b=n.merge(this.constructor(),a);return
b.prevObject=this,b.context=this.context,b},each:function(a,b){return
n.each(this,a,b)},map:function(a){return this.pushStack(n.map(this,function(b,c){return
a.call(b,c,b)}))},slice:function(){return
this.pushStack(d.apply(this,arguments))},first:function(){return
this.eq(0)},last:function(){return this.eq(-1)},eq:function(a){var b=this.length,c=+a+(0>a?
b:0);return this.pushStack(c>0&&b>c?[this[c]]:[])},end:function(){return
this.prevObject||this.constructor(null)},push:f,sort:c.sort,splice:c.splice},n.extend=n.fn.
extend=function(){var a,b,c,d,e,f,g=arguments[0]||
{},h=1,i=arguments.length,j=!1;for("boolean"==typeof g&&(j=g,g=arguments[h]||
{}),h++),"object"==typeof g||n.isFunction(g)|| (g={},h===i&&(g=this,h--));i>h;h++)if(null!=
(a=arguments[h]))for(b in a)c=g[b],d=a[b],g!=d&&(j&&d&&(n.isPlainObject(d)||
(e=n.isArray(d)))?(e?(e=!1,f=c&&n.isArray(c)?c:[]):f=c&&n.isPlainObject(c)?c:
{}),g[b]=n.extend(j,f,d)):void 0!==d&&(g[b]=d));return g},n.extend({expando:"jQuery"+
(m+Math.random()).replace(/\D/g,""),isReady:!0,error:function(a){throw new
Error(a)},noop:function(){},isFunction:function(a)
{return"function"===n.type(a)},isArray:Array.isArray,isWindow:function(a){return
null!=a&&a===a.window},isNumeric:function(a){return!n.isArray(a)&&a-
parseFloat(a)+1>0},isPlainObject:function(a)

```

Minified jQuery version 2.1.3 – 83 KB file size

Minification Tools

If you are writing JavaScript for a public website, you should definitely minify your code.

One example from Google is their Closure Compiler: <https://developers.google.com/closure/compiler/>

Using jQuery

The jQuery API documentation is an excellent reference.

Take a look at the docs and get familiar with using them. There is also a “Learning Center”. Highly recommend using it.

jQuery function a.k.a \$()

jQuery()

Categories: [Core](#)

Return a collection of matched elements either found in the DOM based on passed argument(s) or created by passing an HTML string.

Contents:

- [jQuery\(selector \[, context \] \)](#)
 - [jQuery\(selector \[, context \] \)](#)
 - [jQuery\(element \)](#)
 - [jQuery\(elementArray \)](#)
 - [jQuery\(object \)](#)
 - [jQuery\(selection \)](#)
 - [jQuery\(\)](#)
- [jQuery\(html \[, ownerDocument \] \)](#)
 - [jQuery\(html \[, ownerDocument \] \)](#)
 - [jQuery\(html, attributes \)](#)
- [jQuery\(callback \)](#)
 - [jQuery\(callback \)](#)

Selecting elements

There are many ways to select elements. See API docs:
<http://api.jquery.com/category/selectors/>

Selecting Elements by ID

```
1 | $( "#myId" ); // Note IDs must be unique per page.
```

Selecting Elements by Class Name

```
1 | $( ".myClass" );
```

Selecting Elements by Attribute

```
1 | $( "input[name='first_name']" ); // Beware, this can be very slow in older browsers
```

Selecting Elements by Compound CSS Selector

```
1 | $( "#contents ul.people li" );
```

Once you've made a selection, you'll often want to know whether you have anything to work with. A common mistake is to use:

```
1 | // Doesn't work!  
2 | if ( $( "div.foo" ) ) {  
3 |     ...  
4 | }
```

This won't work. When a selection is made using `$()`, an object is always returned, and objects always evaluate to `true`. Even if the selection doesn't contain any elements, the code inside the `if` statement will still run.

The best way to determine if there are any elements is to test the selection's `.length` property, which tells you how many elements were selected. If the answer is 0, the `.length` property will evaluate to `false` when used as a boolean value:

```
1 | // Testing whether a selection contains elements.  
2 | if ( $( "div.foo" ).length ) {  
3 |     ...  
4 | }
```

Did I get anything

Got an element, now what?

- `children()` - gets the children
- `text()` - gets the text
- `text(str)` - sets the text
- `val()` - gets the value
- `val(str)` - sets the value
- `attr(name)` - gets an attribute value
- `removeAttr(name)` - removes attribute

.text(text)

Returns: [jQuery](#)

Description: *Set the content of each element in the set of matched elements to the specified text.*

 **.text(text)**

version added: 1.0

text

Type: [String](#) or [Number](#) or [Boolean](#)

The text to set as the content of each matched element. When Number or Boolean is supplied, it will be converted to a String representation.

 **.text(function)**

version added: 1.4

function

Type: [Function](#)([Integer](#) index, [String](#) text) => [String](#)

A function returning the text content to set. Receives the index position of the element in the set and the old text value as arguments.

Set text the easy way!

Description: *Binds a function to be executed when the DOM has finished loading.*

jQuery(callback)

version added: 1.0

callback

Type: [Function\(\)](#)

The function to execute when the DOM is ready.

This function behaves just like `$(document).ready()`, in that it should be used to wrap other `$()` operations on your page that depend on the DOM being ready. While this function is, technically, chainable, there really isn't much use for chaining against it.

Examples:

Example: *Execute the function when the DOM is ready to be used.*

```
1 | $(function() {  
2 |   // Document is ready  
3 | });
```

Do something on window (page) load