IDV460

INTERACTIVE DATA VIZ SPRING 16

jQUERY

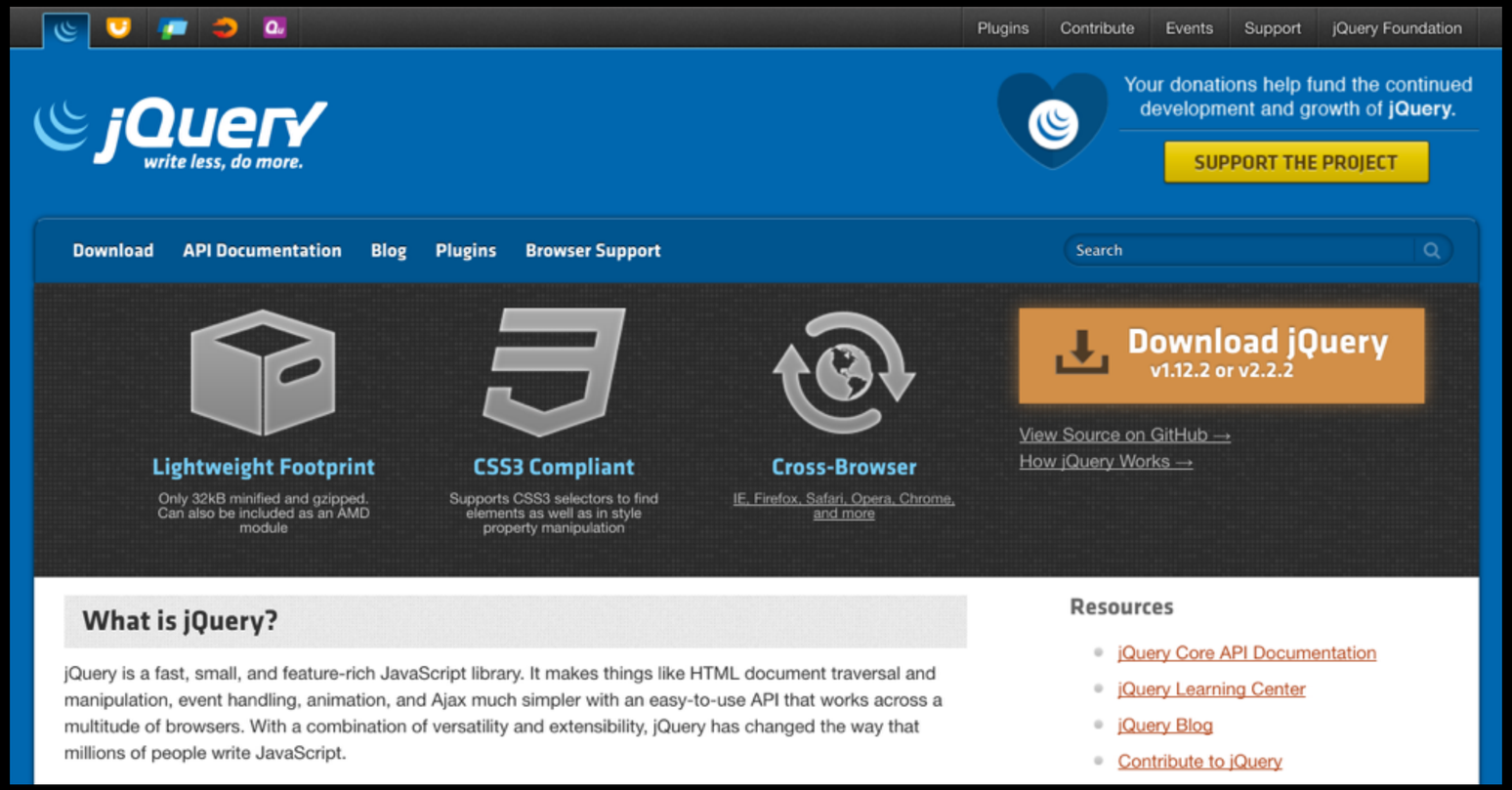
WHAT IS jQUERY?

jQuery is a Javascript library that lets you jump-start your programming by handling many of the messy details of Javascript programming.

A single line of code can thus achieve what could require dozens of lines of pure Javascript. Think of it as a collection of prewritten functions.

WHAT IS jQUERY?

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into **methods** that you can call with a single line of code. With jQuery, you can manipulate the DOM and your CSS, as well as create a number of effects and animations.

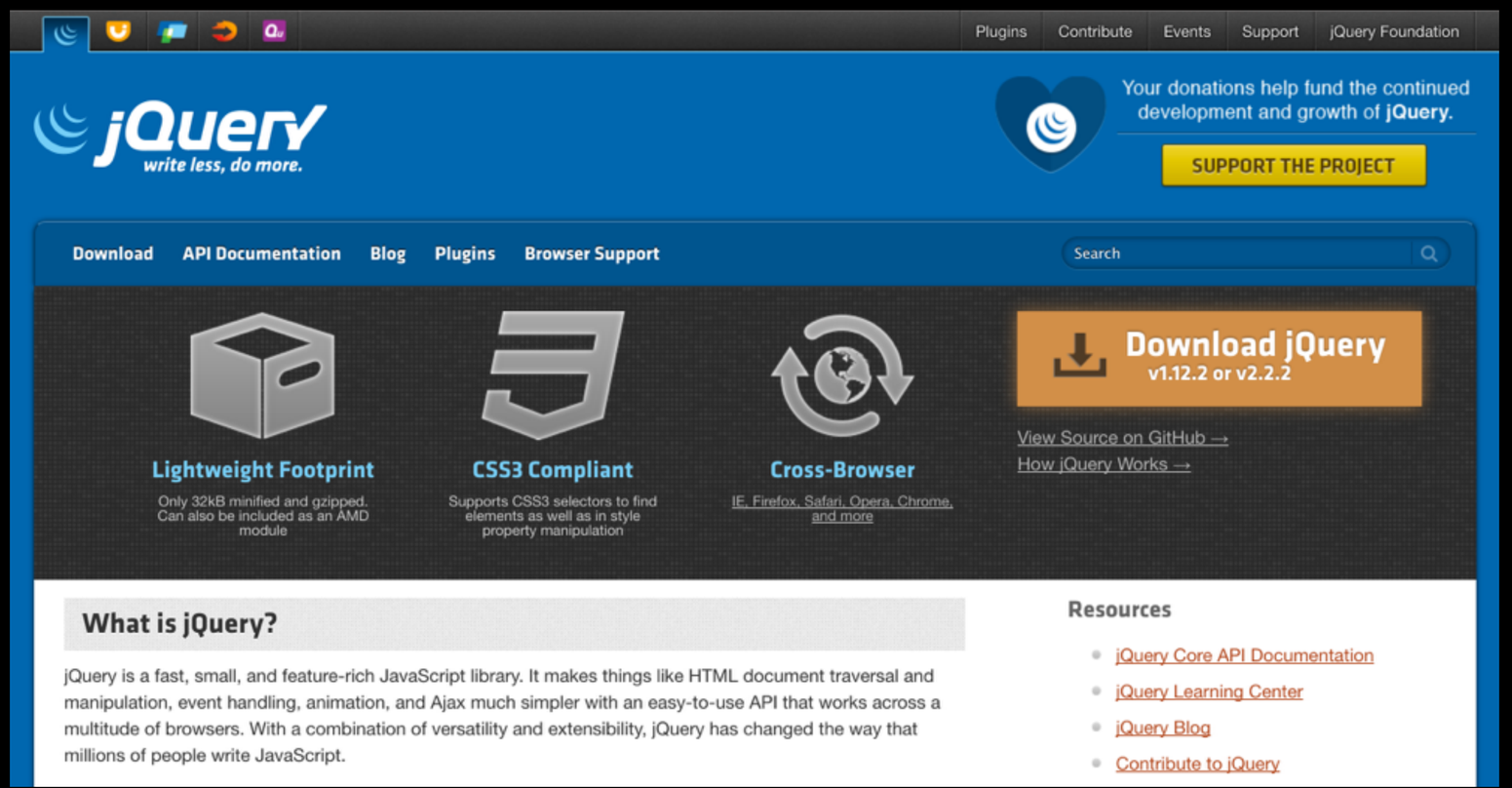


GETTING jQUERY

```
<script src="js/jquery-2.2.1.min.js"></script>
```

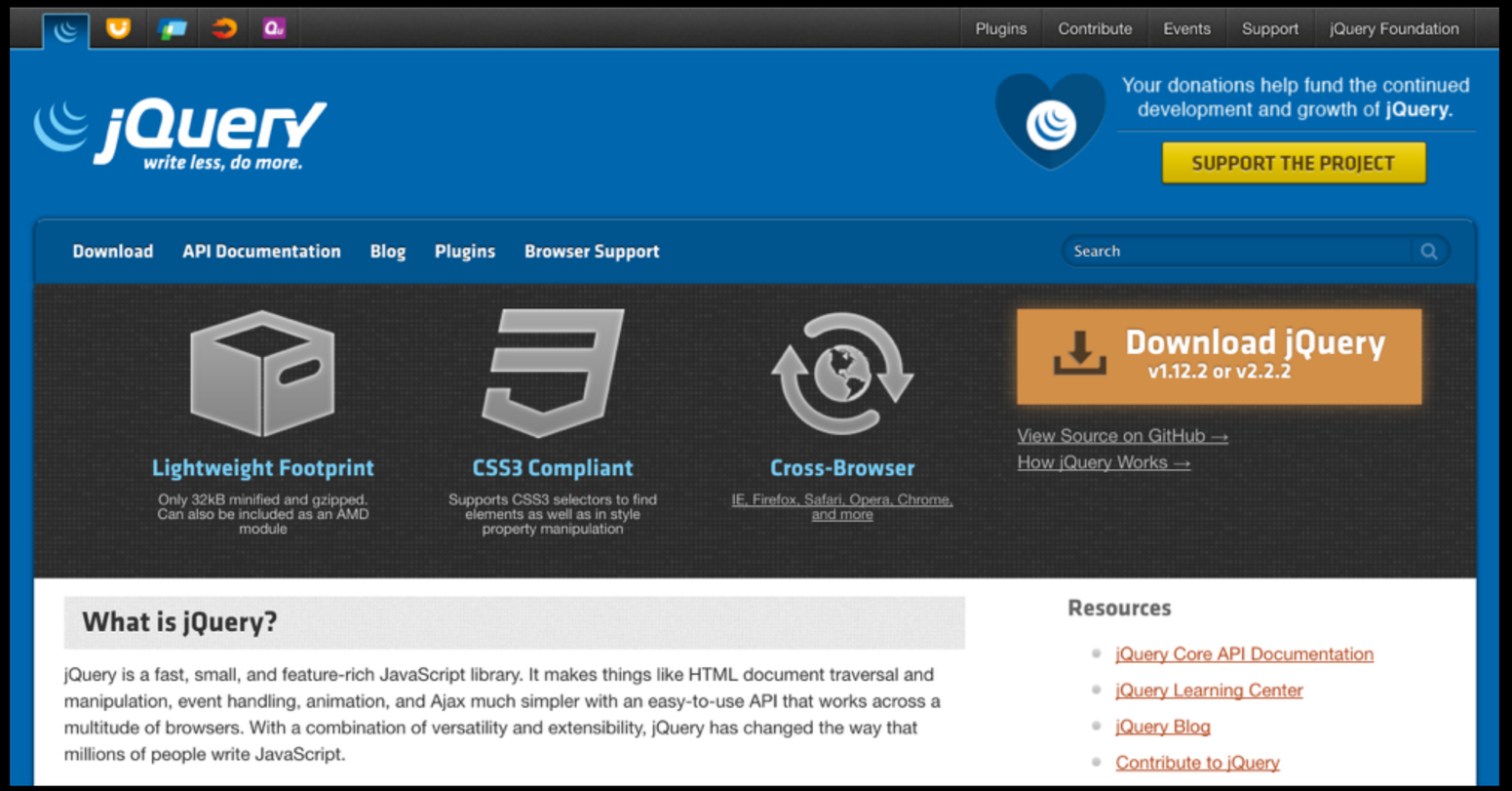
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
1.12.0/jquery.min.js"></script>
```

You add jQuery to your pages as a script tag in the head (usually) of your document, as we did with our Highcharts exercises. You can download your own versions (as we did), or use a CDN (content delivery network) to link to.



GETTING jQUERY

jQuery comes in two versions: a minified production version, and an uncompressed development version. The former is smaller and more lightweight, the latter is recommended if you wish to actually read the code. We will use the uncompressed version today (2.2.2).



http://jquery.com/download/

jQuery syntax

The syntax of jQuery is made to select HTML elements and then perform some action on those elements.

```
$('selector').action();
```

jQuery syntax

The syntax of jQuery is made to select HTML elements and then perform some action on those elements.

```
$('selector').action();
```

The dollar sign is used to define and access jQuery. This function creates the jQuery object — it is the same as writing …

```
jQuery('selector').action();
```

jQuery syntax

The selector is used to find HTML elements; it can be a tag, a class or an ID (using CSS notation).

```
$('selector').action();
```

```
$('button').action();
```

```
$('button.myButton').action();
```

```
$('button#myButton').action();
```

jQuery syntax

The jQuery action is then called to act upon the selected element(s). There are a number of built-in jQuery effects that you will find very useful in adding interactivity and motion to your page.

```
$('selector').action();
```

```
$('button').slideDown();
```

```
$('button.myButton').hide();
```

```
$('button#myButton').toggle();
```

jQuery syntax

jQuery methods are frequently inside an event called document.ready. This is to ensure that the jQuery code does not run until the document has finished loading.

```
$(document).ready(function(){
   $('button.myButton').hide();
});
```

A simple example

We will begin with a simple illustration of how jQuery works, using a very simple HTML document. Use you basic template to create a simple HTML document called **hideseek.html**. Add four circles in a row under a simple headline and paragraph.

```
$(document).ready(function(){
   $(this).hide();
});
```

New JS + jQuery things

We already know the equal sign in Javascript is an **assignment operator**. It does NOT mean equal to, as it would in 2 + 2 = 4, but that you are assigning a value to something (like a variable).

```
var x = 5;
```

New JS + jQuery things

Two equal signs mean equal to in Javascript. Often you would use this to check a value, like checking to see if the value of x is equal to 5. This creates a boolean, returning true or false.

```
var x = 5;
var x = 5;
document.getElementById("demo").
innerHTML = (x == 5);
```

New JS + jQuery things

Three equal signs are also used to as a comparison selector, and are used to check whether something has equal value and is equal type. It's like checking to see if one variable is the exact same thing as another.

```
var x = 5;
var x = 5;
document.getElementById("demo").
innerHTML = (x == 5);

if (index === currentIndex) {
  $button.addClass('active');
}
```

New JS + jQuery things

The code today includes several examples as well of **logical operators**. These are used to determine the logic between variables or values.

Two ampersands means “and.”

```
(x < 10 && y > 1)
```

(This would return true if the value of x is less than 10 AND the value of y is more than 1; otherwise, it would return false.)

New JS + jQuery things

The code today includes several examples as well of **logical operators**. These are used to determine the logic between variables or values.

An exclamation point means “NOT.”

```
!(x === y)
```

(Essentially, the NOT value would return true for false statements, and vice versa. In our code today, we will use this to check whether the current link has a class of “active.”.)

New JS + jQuery things

The code today includes several examples as well of **logical operators**. These are used to determine the logic between variables or values.

Two vertical lines means “OR.”

```
(x === 5 || y === 5)
```

(This would check whether either x OR y are five. If either are, it would return TRUE.)

New JS + jQuery things

Today, we will use several additional built-in jQuery methods — those that you can use without additional code if you have jQuery loaded into your page. Here is what they do.

`find();`

The **find** method returns descendant elements of the selected element.

`preventDefault();`

Based on events; if this method is called, the default action of the event will not be triggered.

`hash();`

this.hash reads the href attribute of **this**, and gets the part of the URL beginning with #

New JS + jQuery things

Today, we will use several additional built-in jQuery methods — those that you can use without additional code if you have jQuery loaded into your page. Here is what they do.

`is();`

Checks if one of the selected elements matches the selectorElement.

`parent();`

Returns the direct parent element of the selected element.

`attr();`

Sets or returns attributes and values of the selected elements. It returns the value of the FIRST matched element, or sets one or more attribute/value pairs for the set of matched elements.

New JS + jQuery things

Today, we will use several additional built-in jQuery methods — those that you can use without additional code if you have jQuery loaded into your page. Here is what they do.

`eq();`

Returns an element with a specific index number of the selected elements.

`push();`

Adds a new item to an array. (The opposite is pop( ), BTW.)

`appendTo();`

Inserts HTML elements at the end of the selected elements.