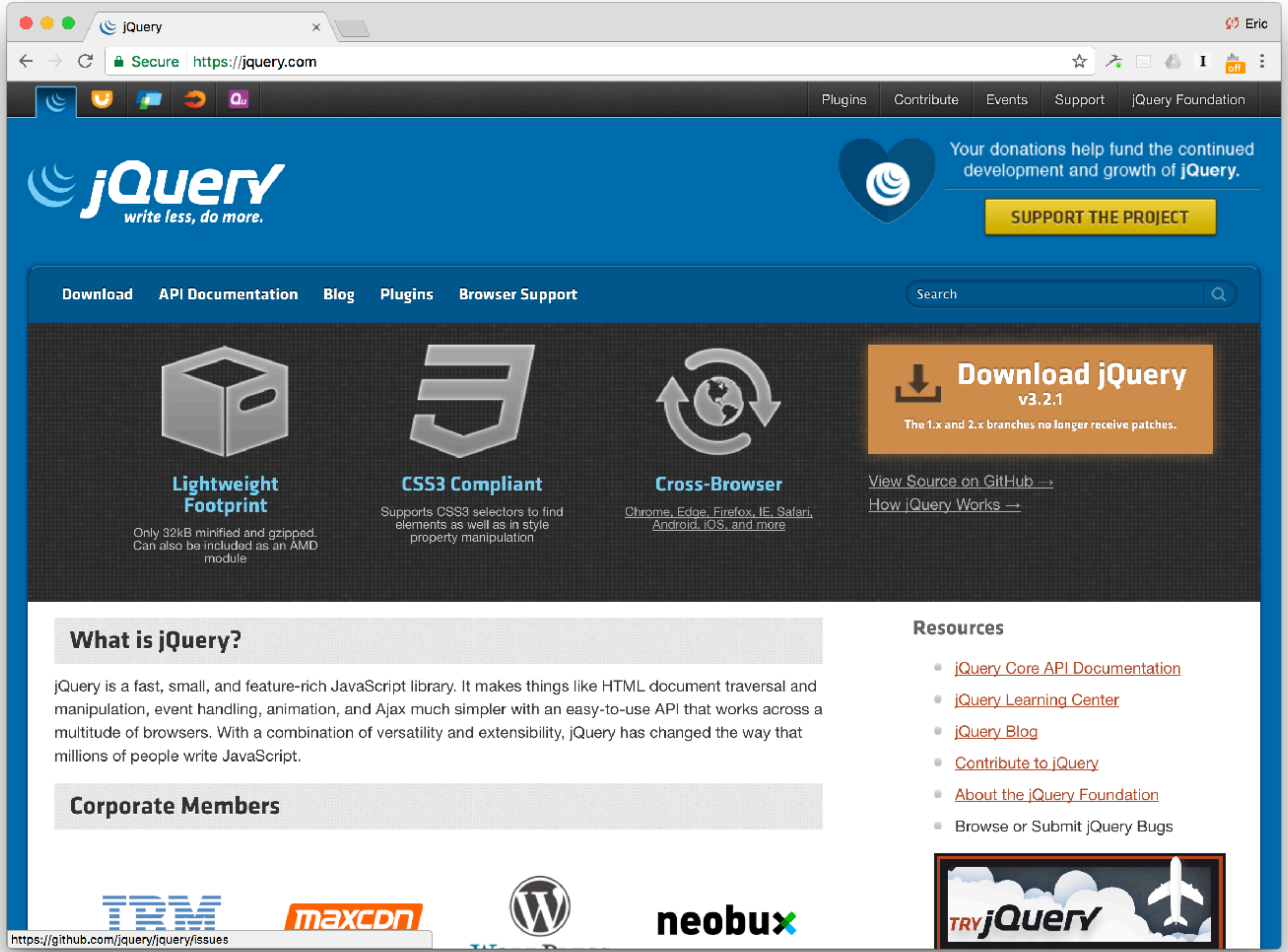
jQuery

jQuery is a library, or set of helpful add-ons, to the JavaScript programming language jQuery is much better at giving you immediate, visual results than regular JavaScript.

You’ll most likely be using a mix of basic JavaScript and jQuery together to get the effects you want.



jQuery

Secure https://jquery.com

Eric

Plugins | Contribute | Events | Support | jQuery Foundation

jQuery
write less, do more.

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

SUPPORT THE PROJECT

Download | API Documentation | Blog | Plugins | Browser Support

Search

Lightweight Footprint
Only 32kB 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
v3.2.1
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.

Corporate Members

IBM maxcdn neobux

Resources

- jQuery Core API Documentation
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery
- About the jQuery Foundation
- Browse or Submit jQuery Bugs

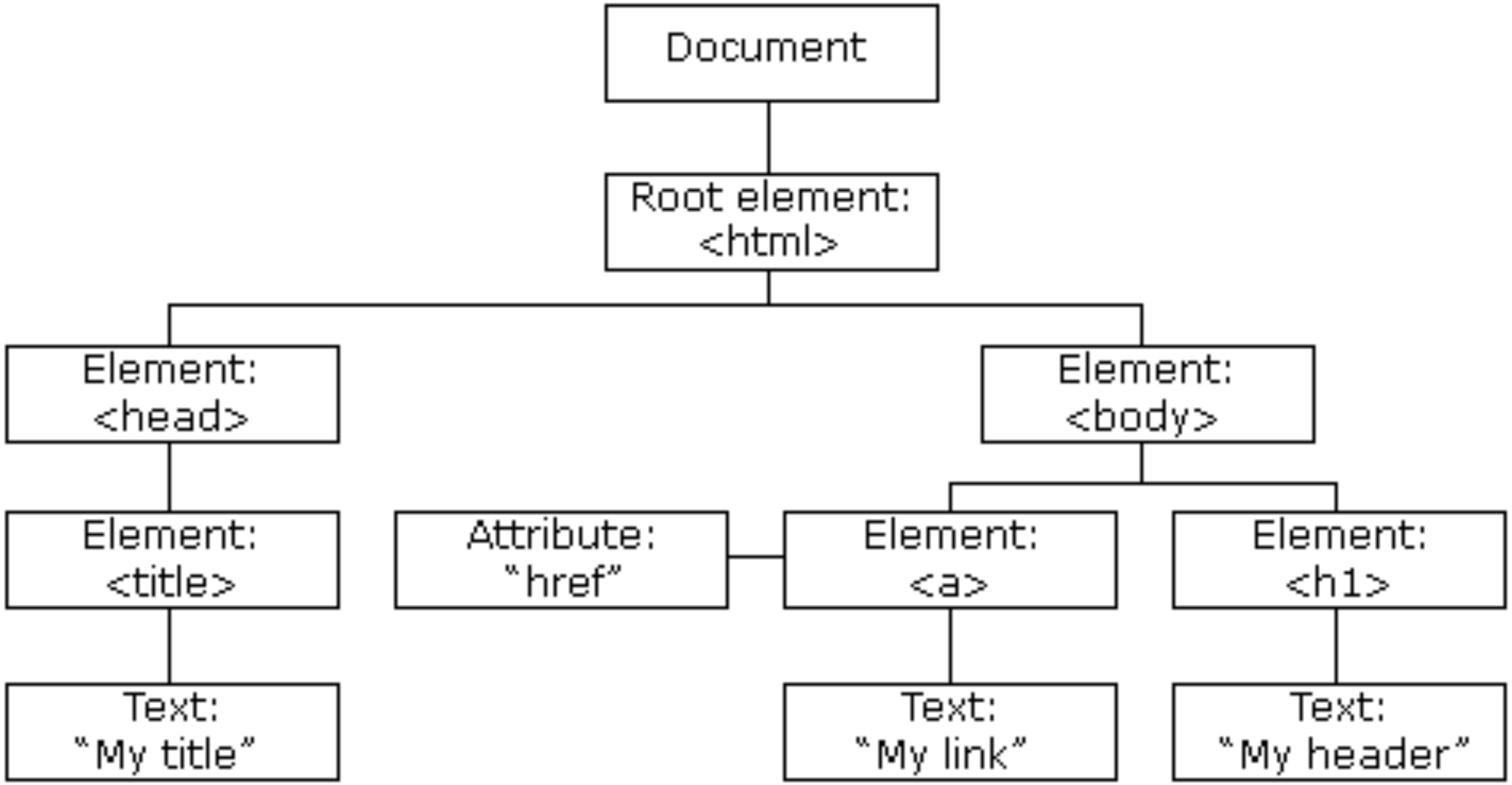
TRY jQuery

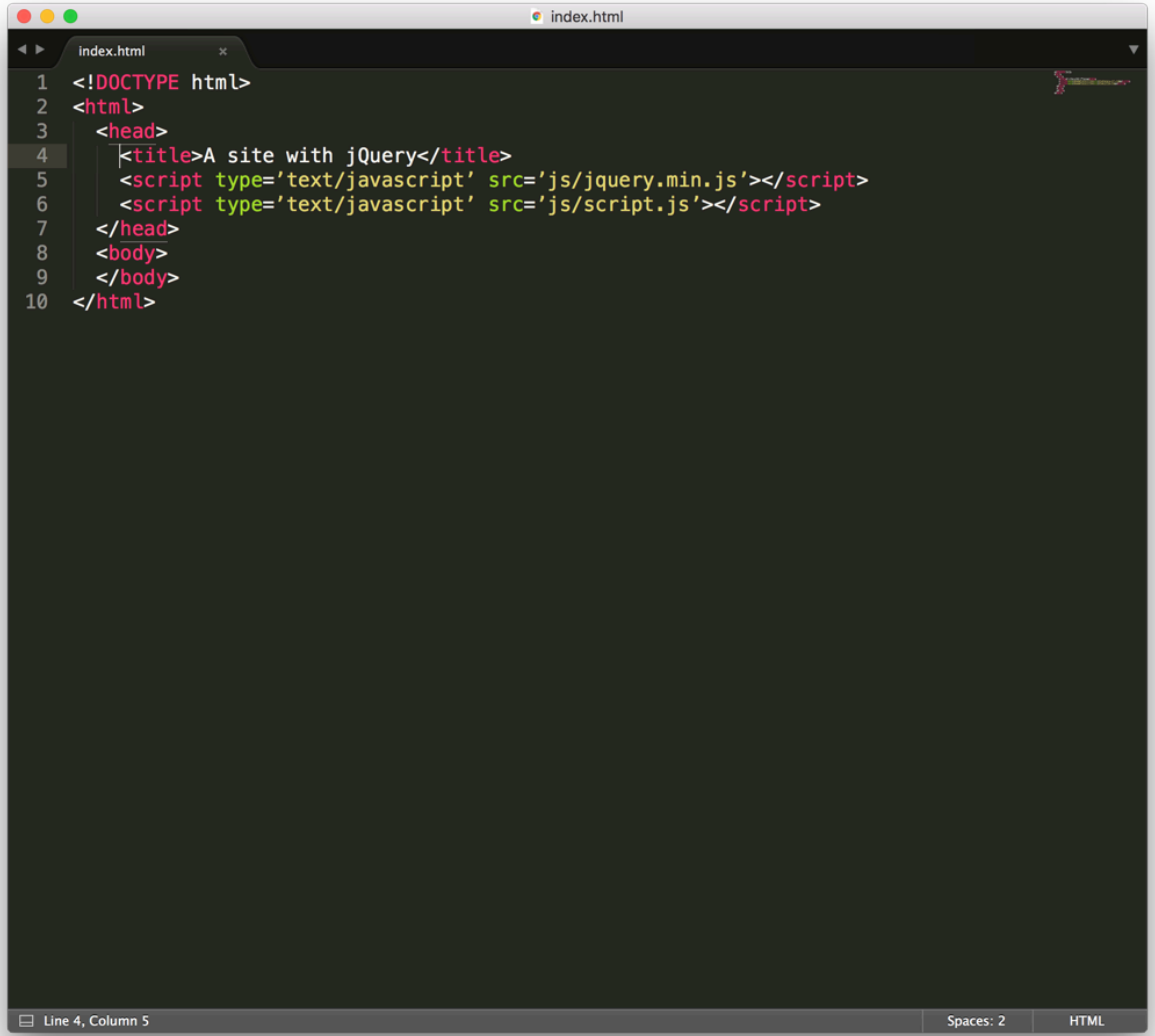
https://github.com/jquery/jquery/issues

To get the most out of jQuery, we should review how an HTML page is put together.

An HTML document is structured according to the Document Object Model, or DOM. It’s by interacting with the DOM that jQuery is able to access and modify HTML.

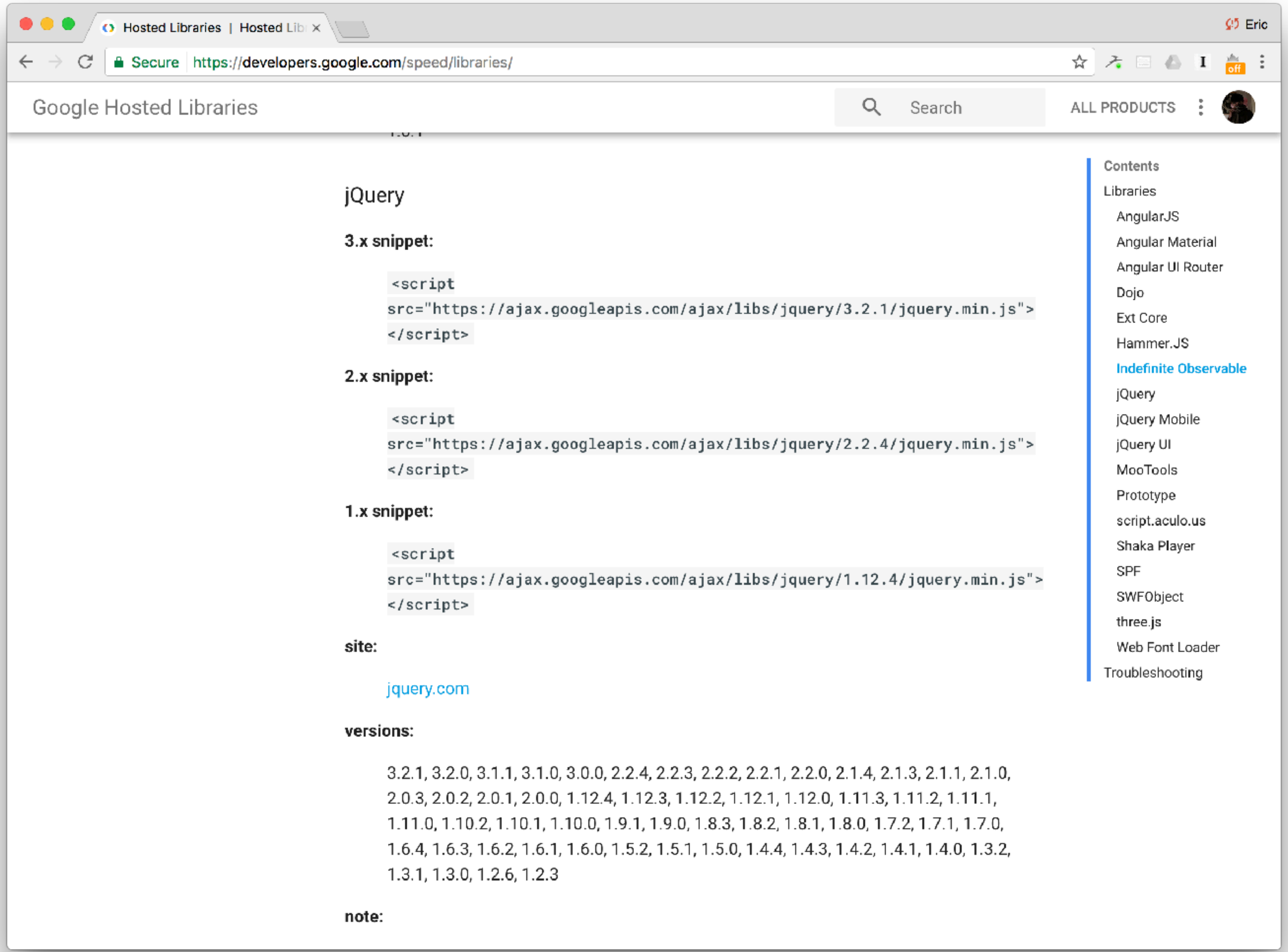
The DOM consists of every element on the page, laid out in a hierarchical way that reflects the way the HTML document is ordered.


Document
Root element: <html>
Element: <head>
Element: <body>
Element: <title>
Attribute: "href"
Element: <a>
Element: <h1>
Text: "My title"
Text: "My link"
Text: "My header"



```html
<!DOCTYPE html>
<html>
  <head>
    <title>A site with jQuery</title>
    <script type='text/javascript' src='js/jquery.min.js'></script>
    <script type='text/javascript' src='js/script.js'></script>
  </head>
  <body>
  </body>
</html>
```

You can download and add jQuery in the head of the html document.


Hosted Libraries | Hosted Lib
Eric
Secure
https://developers.google.com/speed/libraries/
Google Hosted Libraries
Search
ALL PRODUCTS

jQuery

3.x snippet:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
```

2.x snippet:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js">
</script>
```

1.x snippet:

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

site:

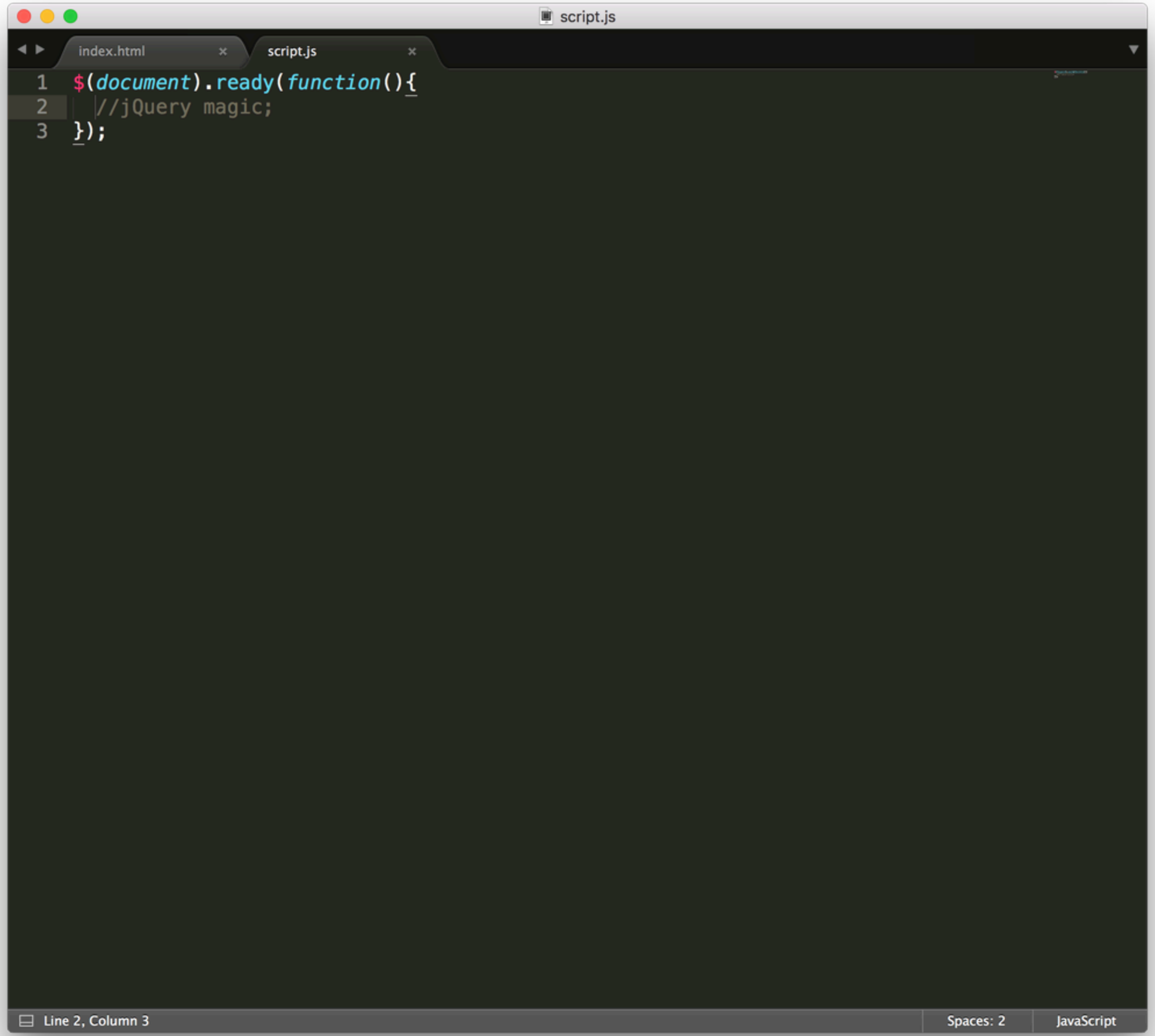
jquery.com

versions:

3.2.1, 3.2.0, 3.1.1, 3.1.0, 3.0.0, 2.2.4, 2.2.3, 2.2.2, 2.2.1, 2.2.0, 2.1.4, 2.1.3, 2.1.1, 2.1.0, 2.0.3, 2.0.2, 2.0.1, 2.0.0, 1.12.4, 1.12.3, 1.12.2, 1.12.1, 1.12.0, 1.11.3, 1.11.2, 1.11.1, 1.11.0, 1.10.2, 1.10.1, 1.10.0, 1.9.1, 1.9.0, 1.8.3, 1.8.2, 1.8.1, 1.8.0, 1.7.2, 1.7.1, 1.7.0, 1.6.4, 1.6.3, 1.6.2, 1.6.1, 1.6.0, 1.5.2, 1.5.1, 1.5.0, 1.4.4, 1.4.3, 1.4.2, 1.4.1, 1.4.0, 1.3.2, 1.3.1, 1.3.0, 1.2.6, 1.2.3

note:

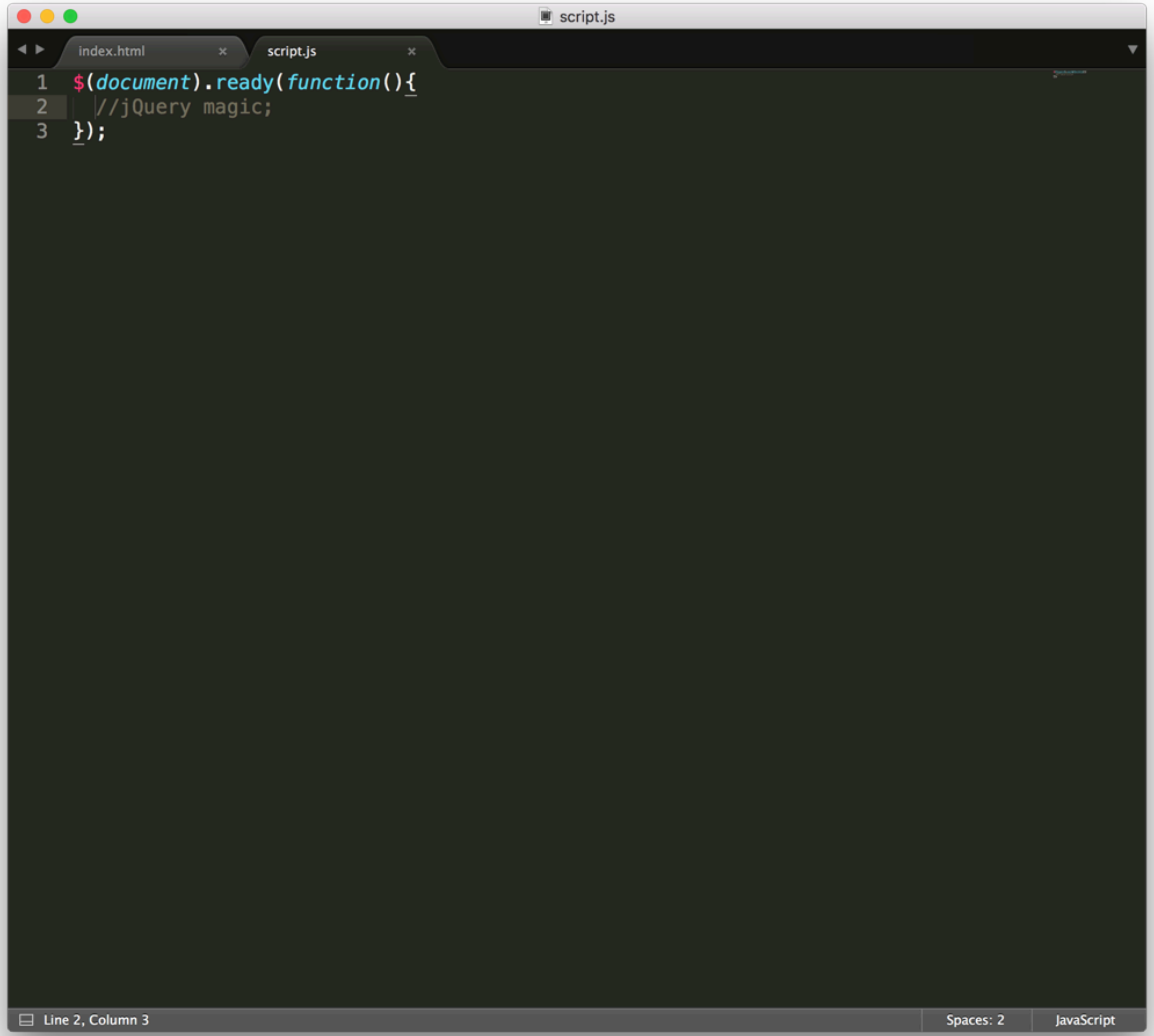
Now that we have loaded jQuery we can use it’s selectors and functions in our other .js files.


script.js

```
$(document).ready(function(){
  //jQuery magic;
});
```

Line 2, Column 3 Spaces: 2 JavaScript

Let's break this down. `$()` says: "Hey, jQuery things are about to happen."



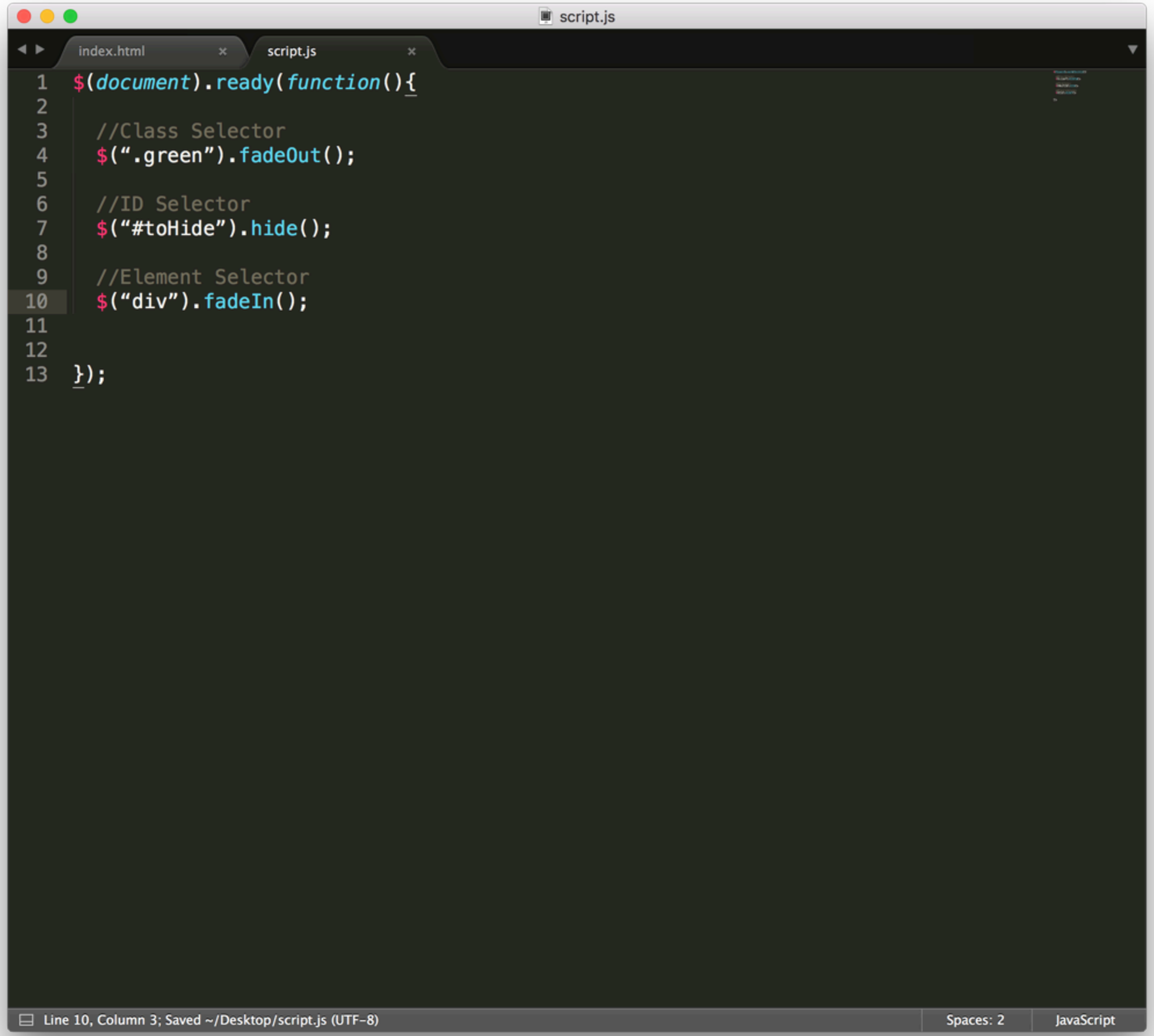
```
$(document).ready(function(){
  //jQuery magic;
});
```

Next, we'll need to start up our jQuery magic using

`$(document).ready();`

Selectors

The first thing you'll want to do is select elements from the DOM to do things with. Selecting in jQuery works just like selecting in CSS.


script.js

index.html | script.js

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

  //Class Selector
  $(".green").fadeOut();

  //ID Selector
  $("#toHide").hide();

  //Element Selector
  $("div").fadeIn();


});
```

Line 10, Column 3; Saved ~/Desktop/script.js (UTF-8) | Spaces: 2 | JavaScript

Selecting elements from the DOM in jQuery uses the below syntax:

```javascript
$("#target").someAction;
```

Next you'll want to do effects on your selections.
Here are some commonly used effects.

Hide or show the matched elements:

```
    .hide();
    .show();
```

Display or hide the matched elements by fading them:

```
.fadeIn(‘fast’);
.fadeOut(1000);
```

Add a delay before the effect:

```
    .delay(value);
```

Add a delay before the effect:

```
    .delay(value);
```

Hide or show the matched elements:

```
.toggle();
```

Display or hide the matched elements by fading them:

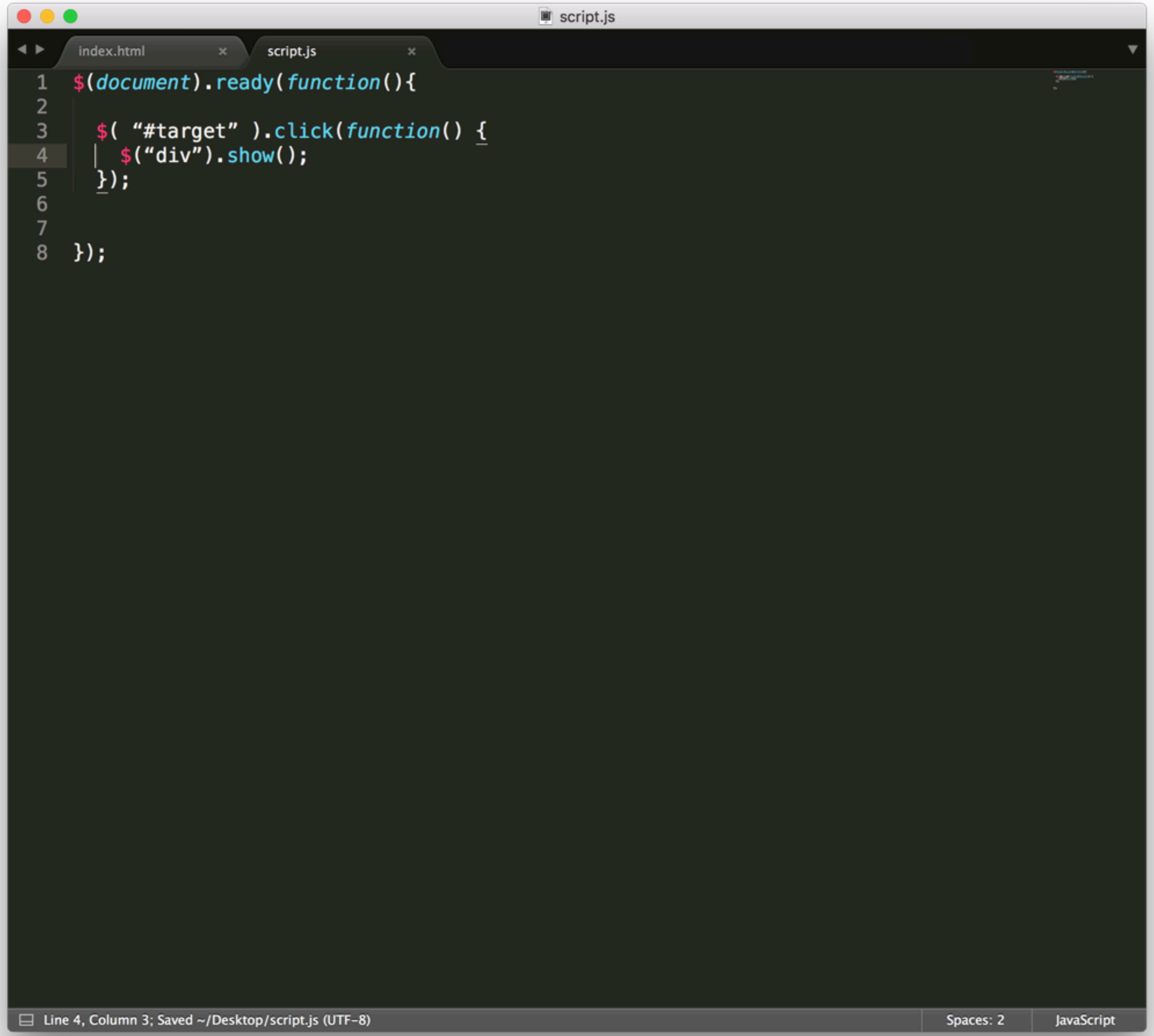
```
.fadeToggle();
```

js Fiddle Examples

You can also tie changes to events the user performs such as clicks, scrolls, etc.

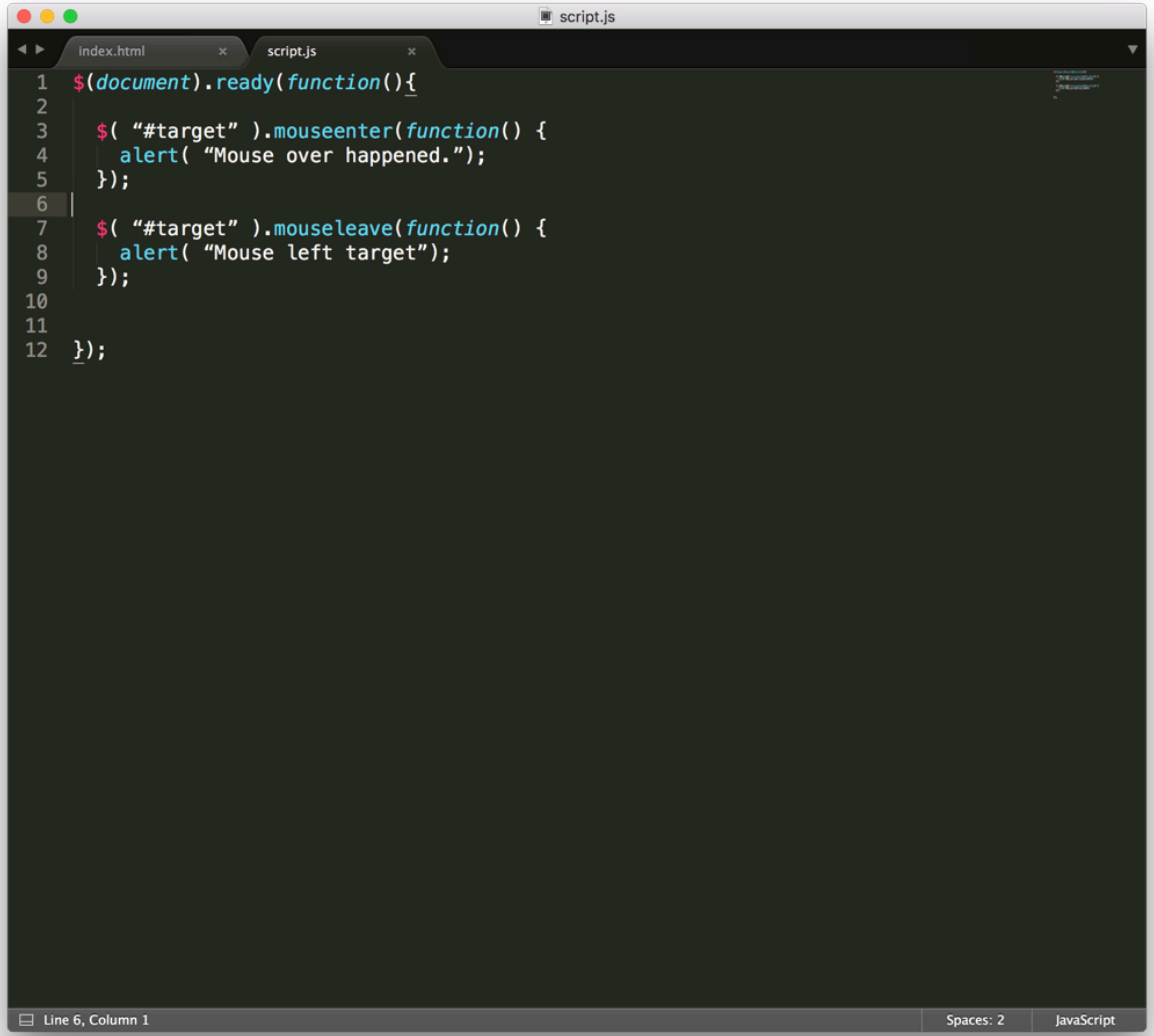
```
$('thingToTouch').event(function() {

        $('thingToAffect').effect();

 });
```

“Thing To Touch” is the HTML element you’ll click on, hover over, or otherwise interact with, and “thing to affect” is the HTML element that fades away, changes size, or undergoes some other transformation.



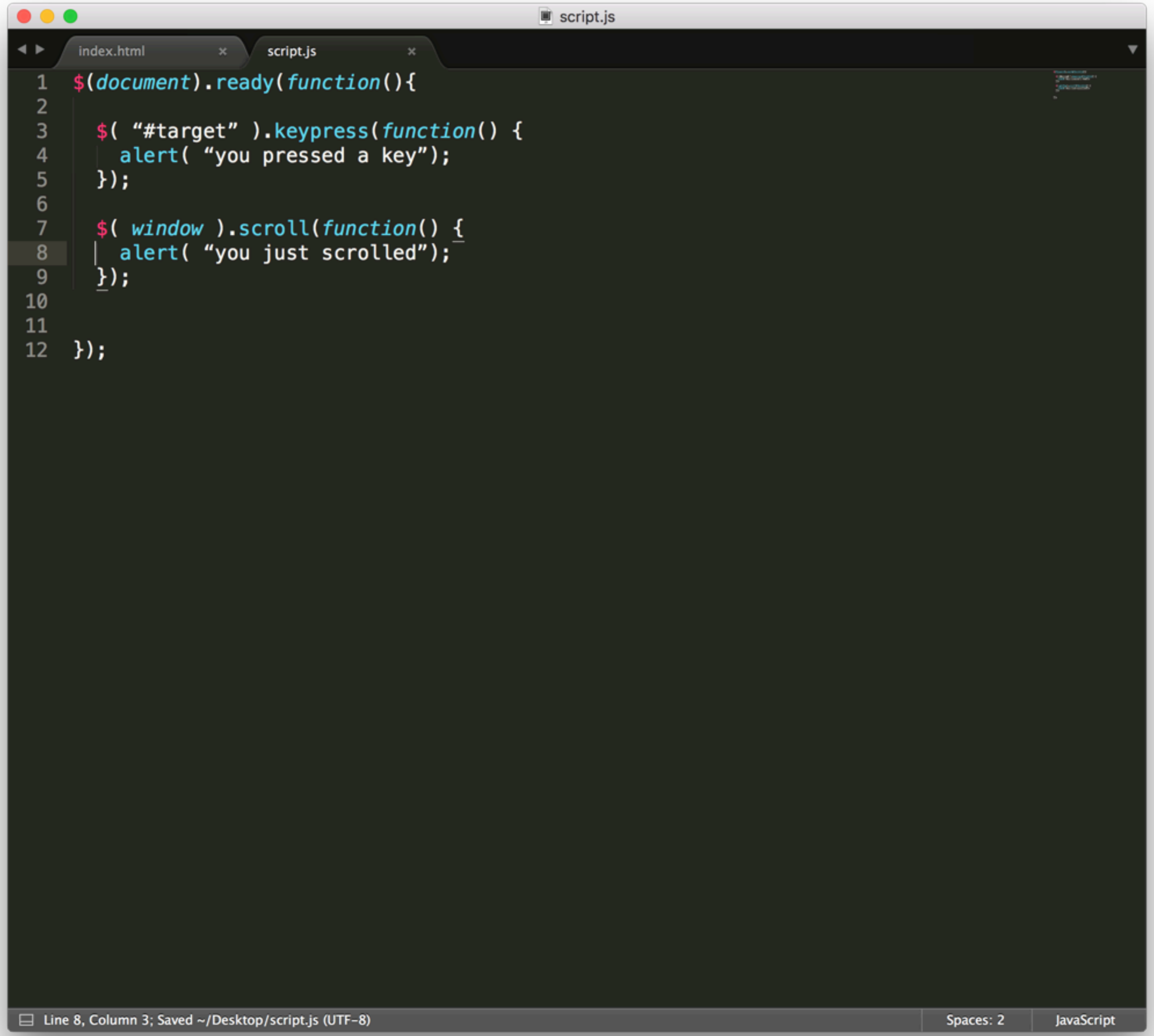
```
$(document).ready(function(){

  $( “#target” ).click(function() {
    $(“div”).show();
  });


});
```

In this example when #target is clicked, all the divs on the page will fadeout.



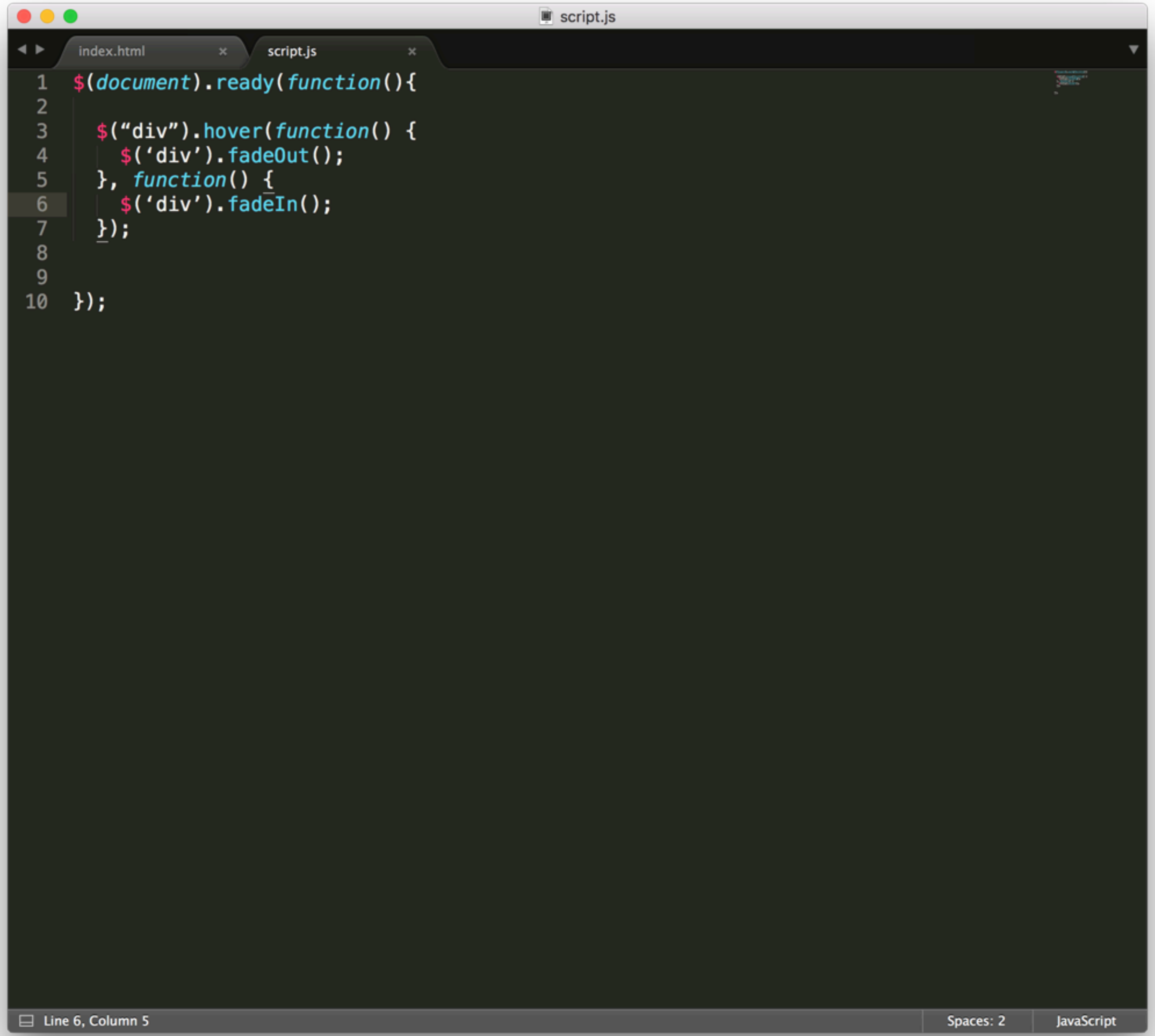
```
$(document).ready(function(){

  $( “#target” ).mouseenter(function() {
    alert( “Mouse over happened.”);
  });

  $( “#target” ).mouseleave(function() {
    alert( “Mouse left target”);
  });


});
```

In this example when the mouse enters the element with an id of #target and alert is triggered. Then when the mouse leaves another alert is triggered.



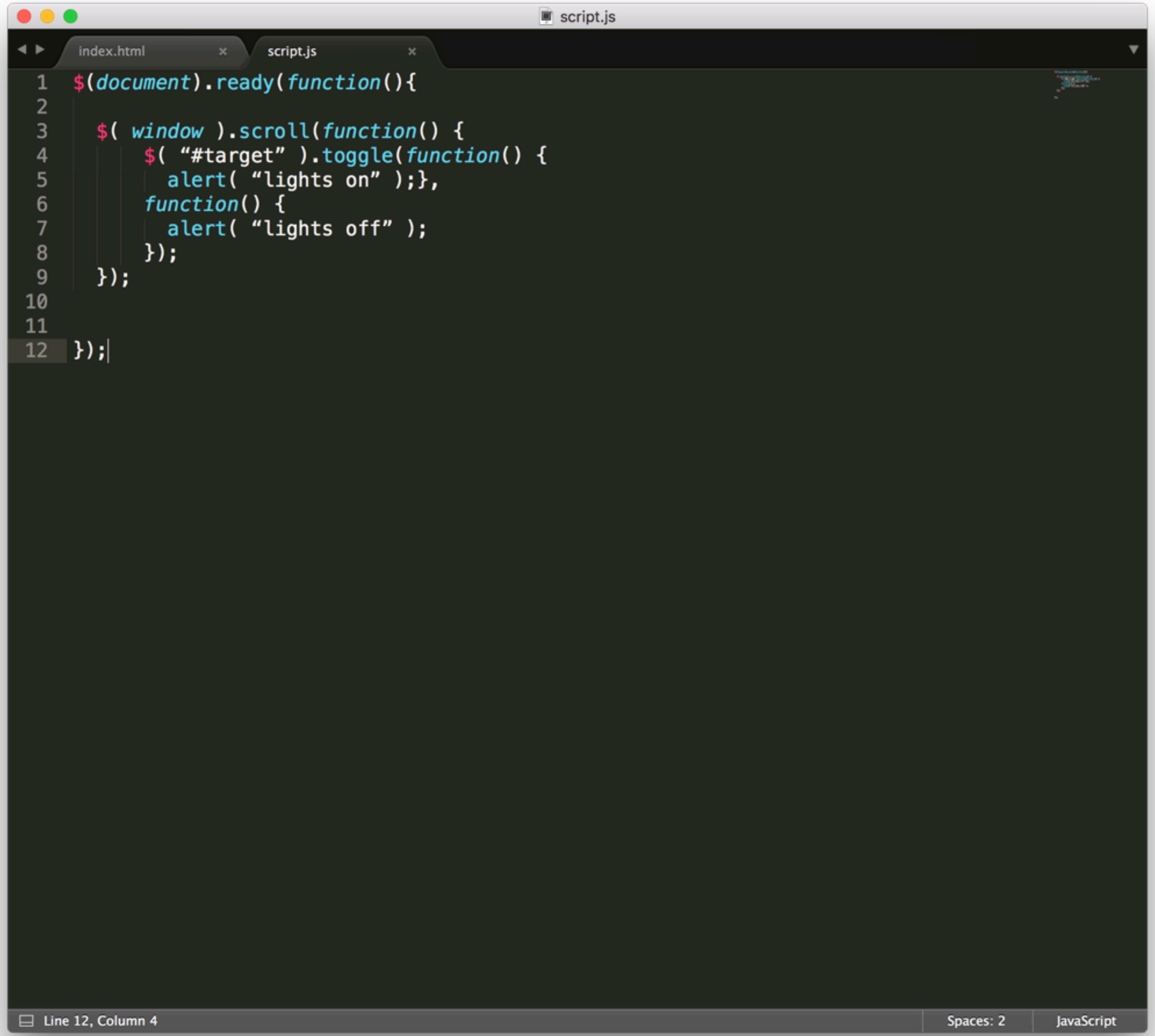
```
$(document).ready(function(){

  $( “#target” ).keypress(function() {
    alert( “you pressed a key”);
  });

  $( window ).scroll(function() {
    alert( “you just scrolled”);
  });


});
```

Here we see examples of the keypress event and the scroll event.



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

  $("div").hover(function() {
    $('div').fadeOut();
  }, function() {
    $('div').fadeIn();
  });


});
```

Here we see an example of the hover event which takes two functions. The first is for when you enter the target and the second is executed when your mouse leaves.


script.js

index.html × script.js ×

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

  $( window ).scroll(function() {
      $( “#target” ).toggle(function() {
        alert( “lights on” );},
      function() {
        alert( “lights off” );
      });
  });


});
```

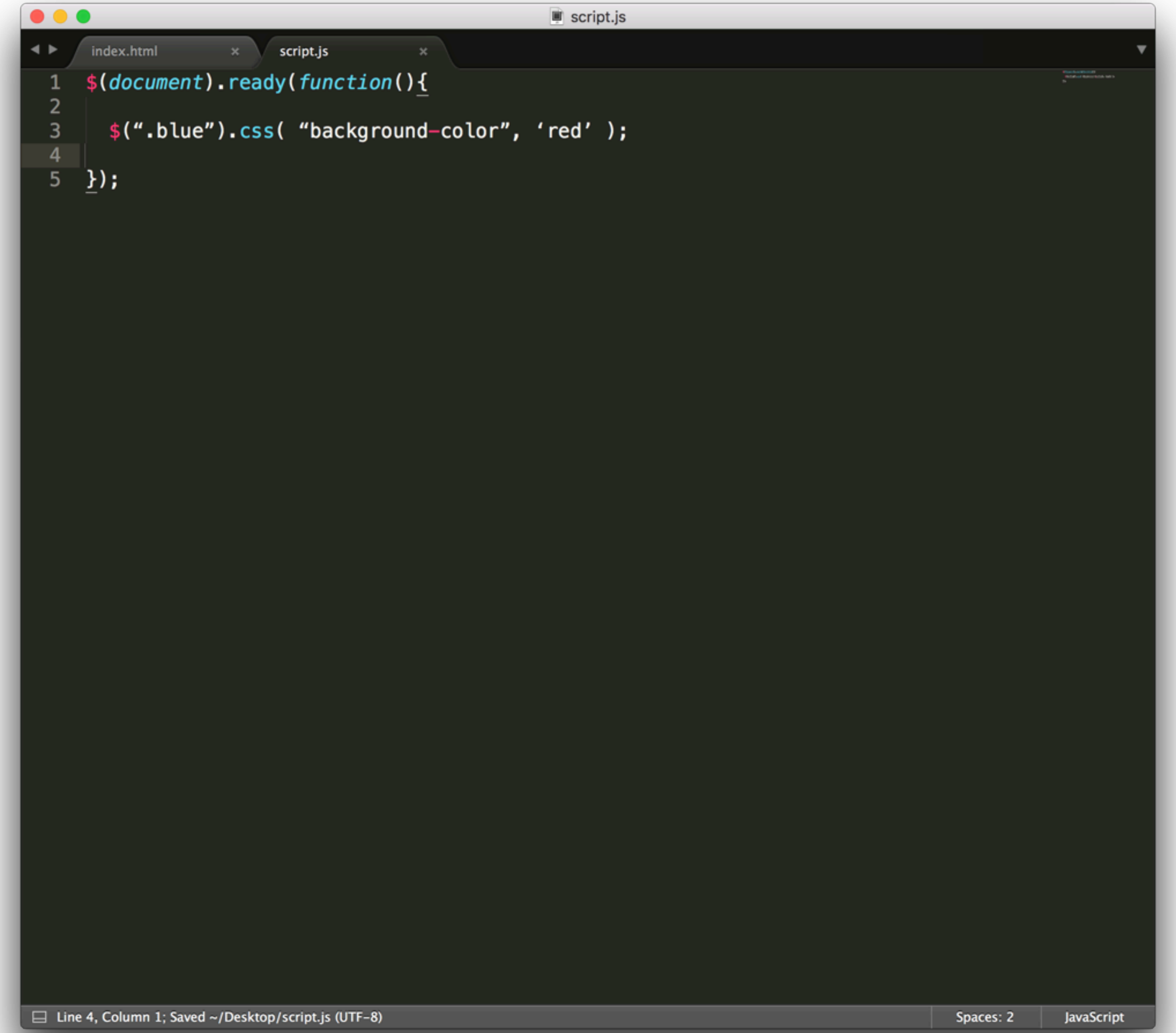
Line 12, Column 4 Spaces: 2 JavaScript

This example combines the scroll event with a toggle event which combines two functions. The function will toggle between the two states.

js Fiddle Examples

You can also use jQuery to effect the css of your page. Here is the syntax for making css changes via jQuery:

```
$('#target').css( propertyName, value );
```



script.js

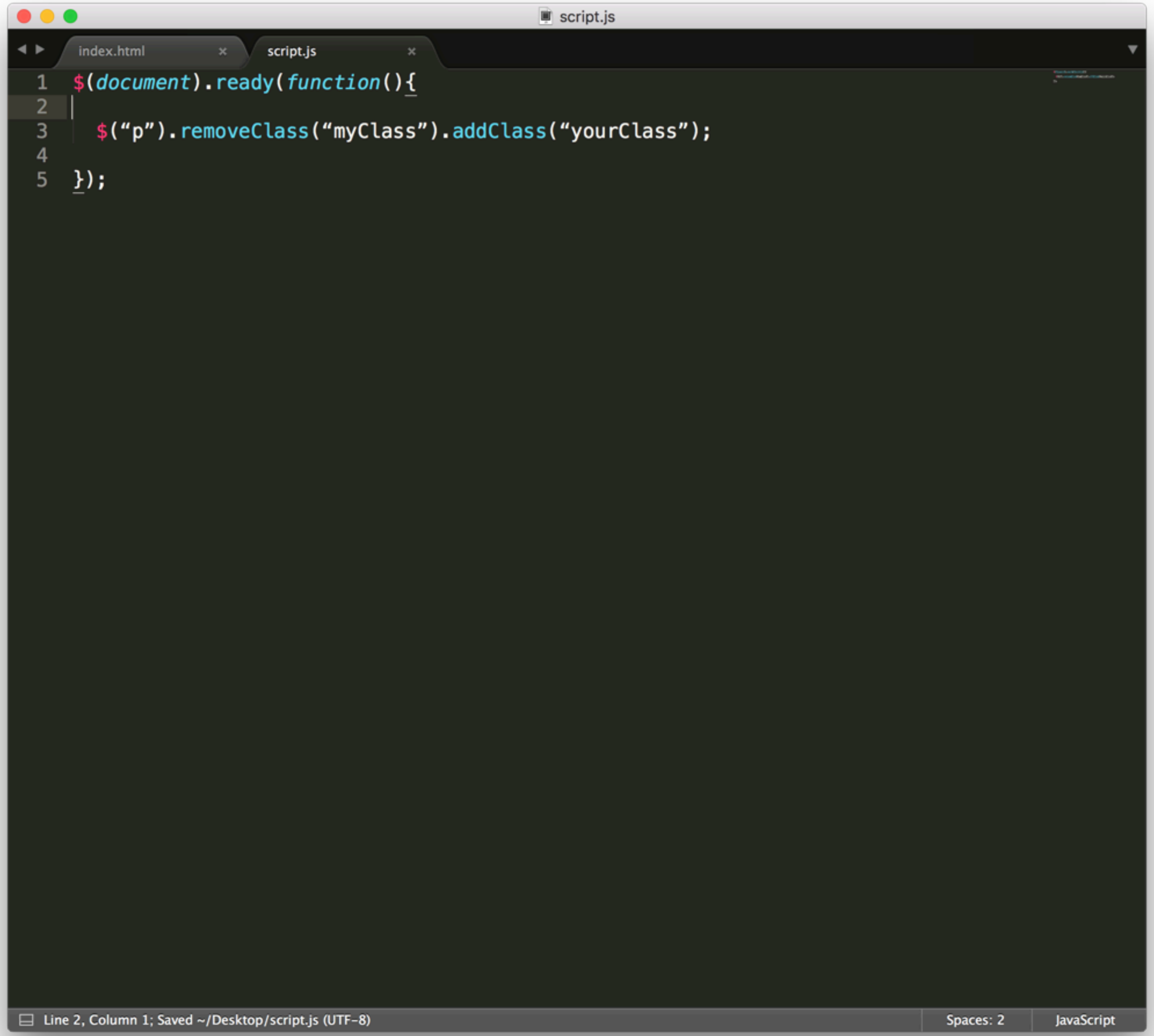
index.html × script.js ×

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

  $(".blue").css( "background-color", 'red' );

});
```

Line 4, Column 1; Saved ~/Desktop/script.js (UTF-8)   Spaces: 2   JavaScript

Here is an example. Note however that you'll likely want to make this changes in relation to events on the page.

You can also use jQuery to change between CSS classes. This is a very common pattern and one of the best ways to make visual changes.



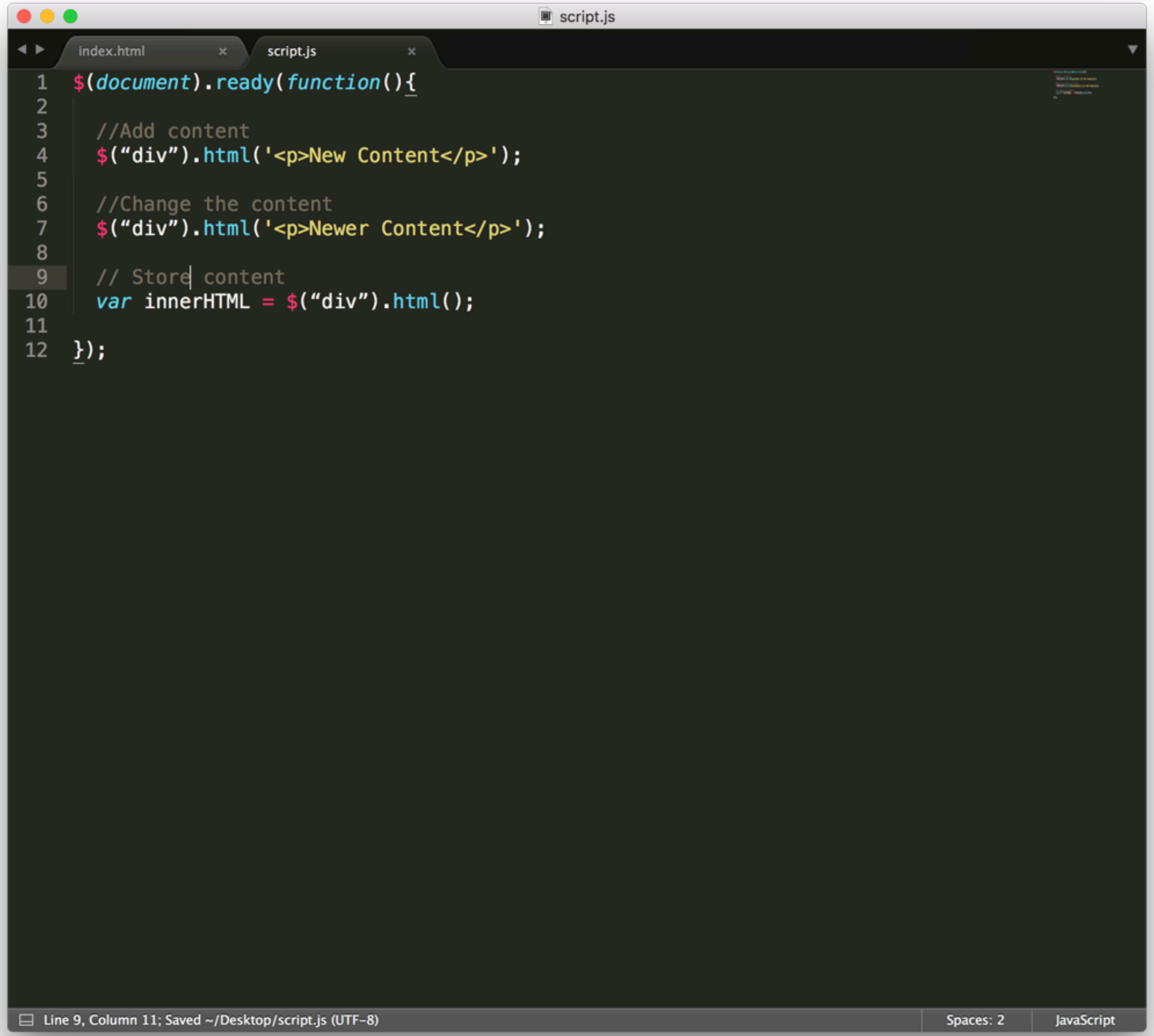
```
$(document).ready(function(){

  $(“p”).removeClass(“myClass”).addClass(“yourClass”);

});
```

Line 2, Column 1; Saved ~/Desktop/script.js (UTF-8)   Spaces: 2   JavaScript

Note that you do not use a **.** to select the class. This example also shows that you can chain jQuery functions.

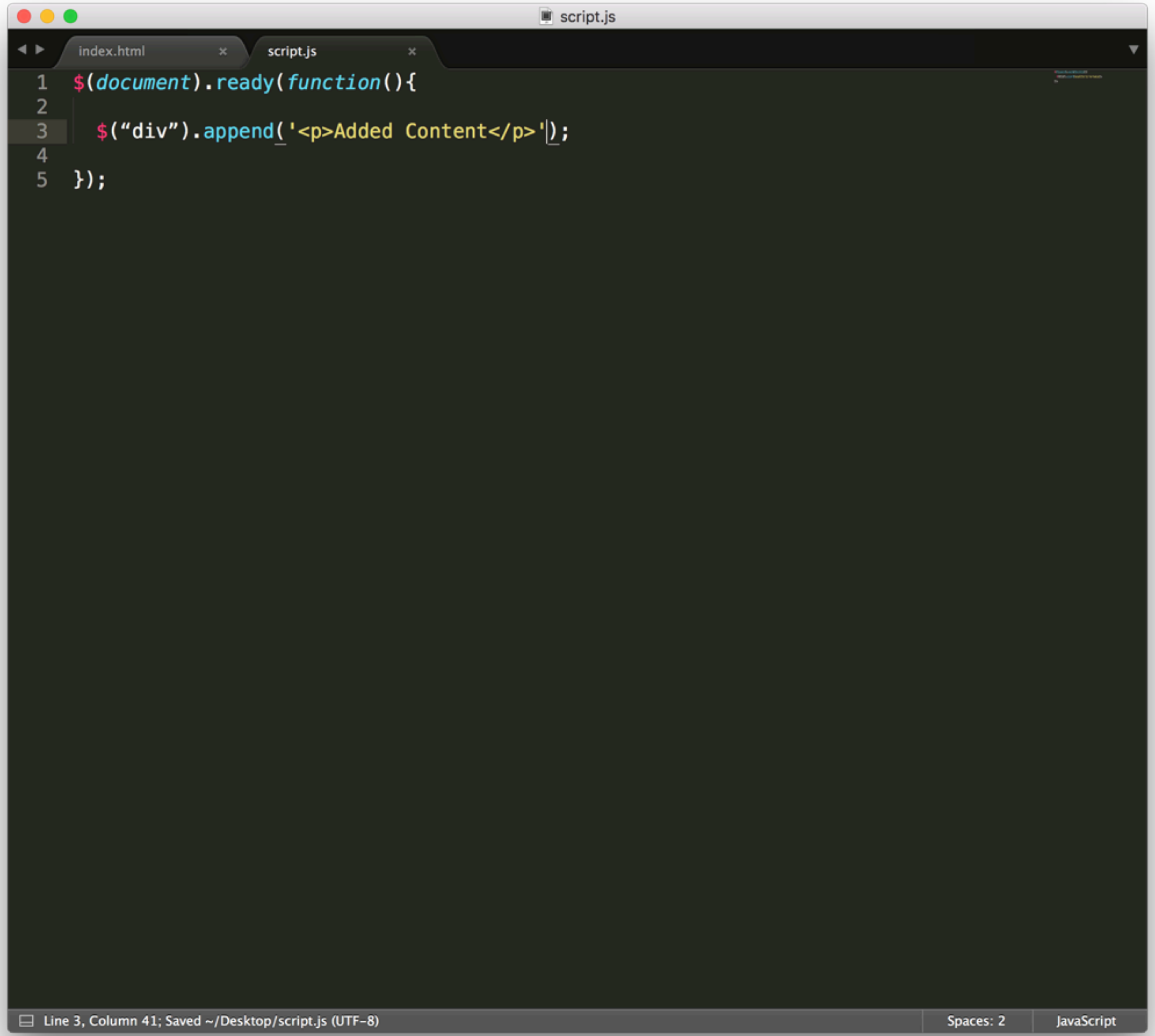
js Fiddle Examples

You can also use jQuery to edit the HTML of a page directly. In these examples you are adding or removing elements from the DOM.



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

  //Add content
  $("div").html('<p>New Content</p>');

  //Change the content
  $("div").html('<p>Newer Content</p>');

  // Store content
  var innerHTML = $("div").html();

});
```

The **.html** function can be used to both set content or to get content from the DOM.

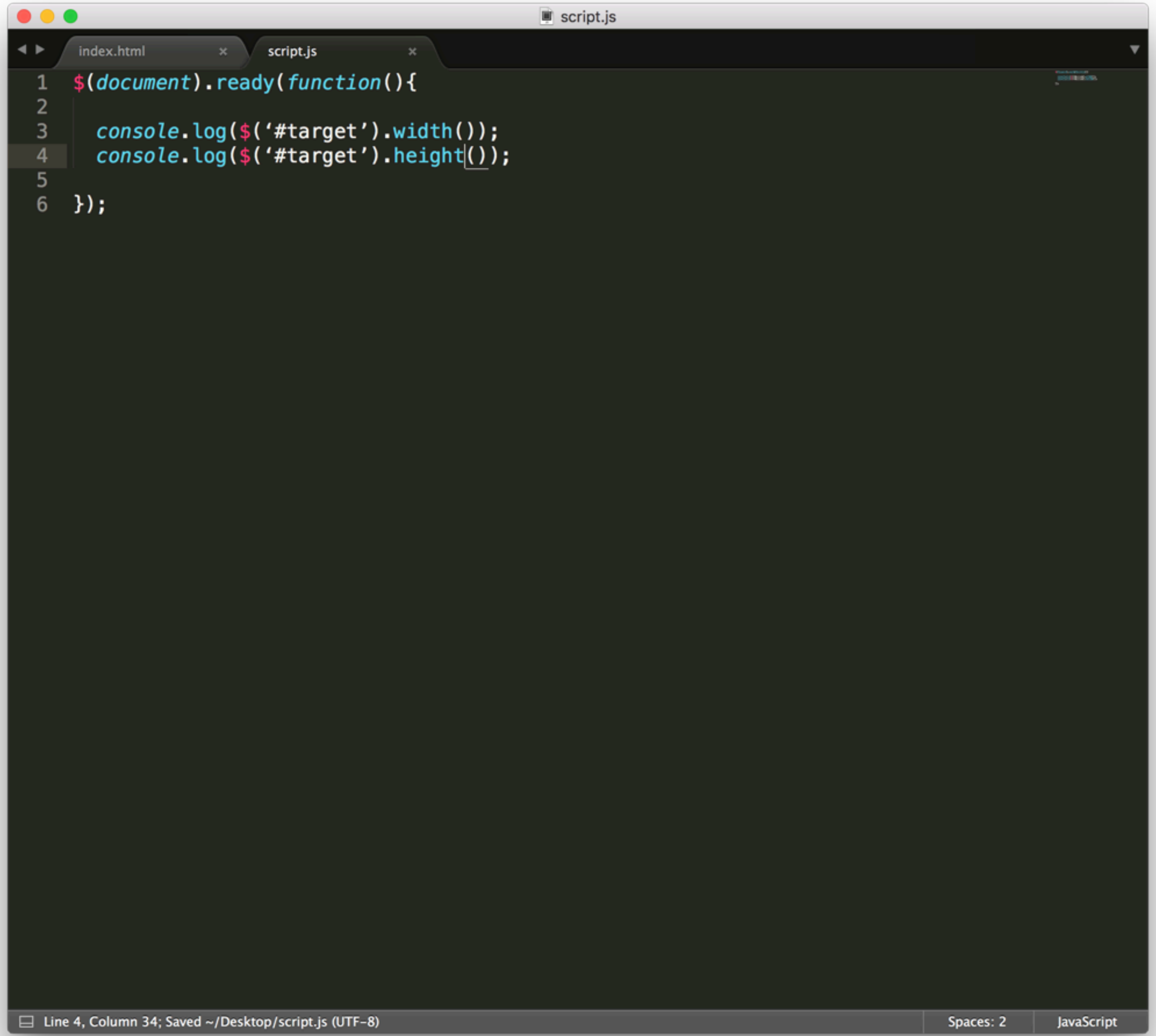


```
$(document).ready(function(){

  $(“div”).append('<p>Added Content</p>');

});
```

Line 3, Column 41; Saved ~/Desktop/script.js (UTF-8) Spaces: 2 JavaScript

The .append function will add content at the end of the selected element.

Finally you can use jQuery to return information about the DOM.



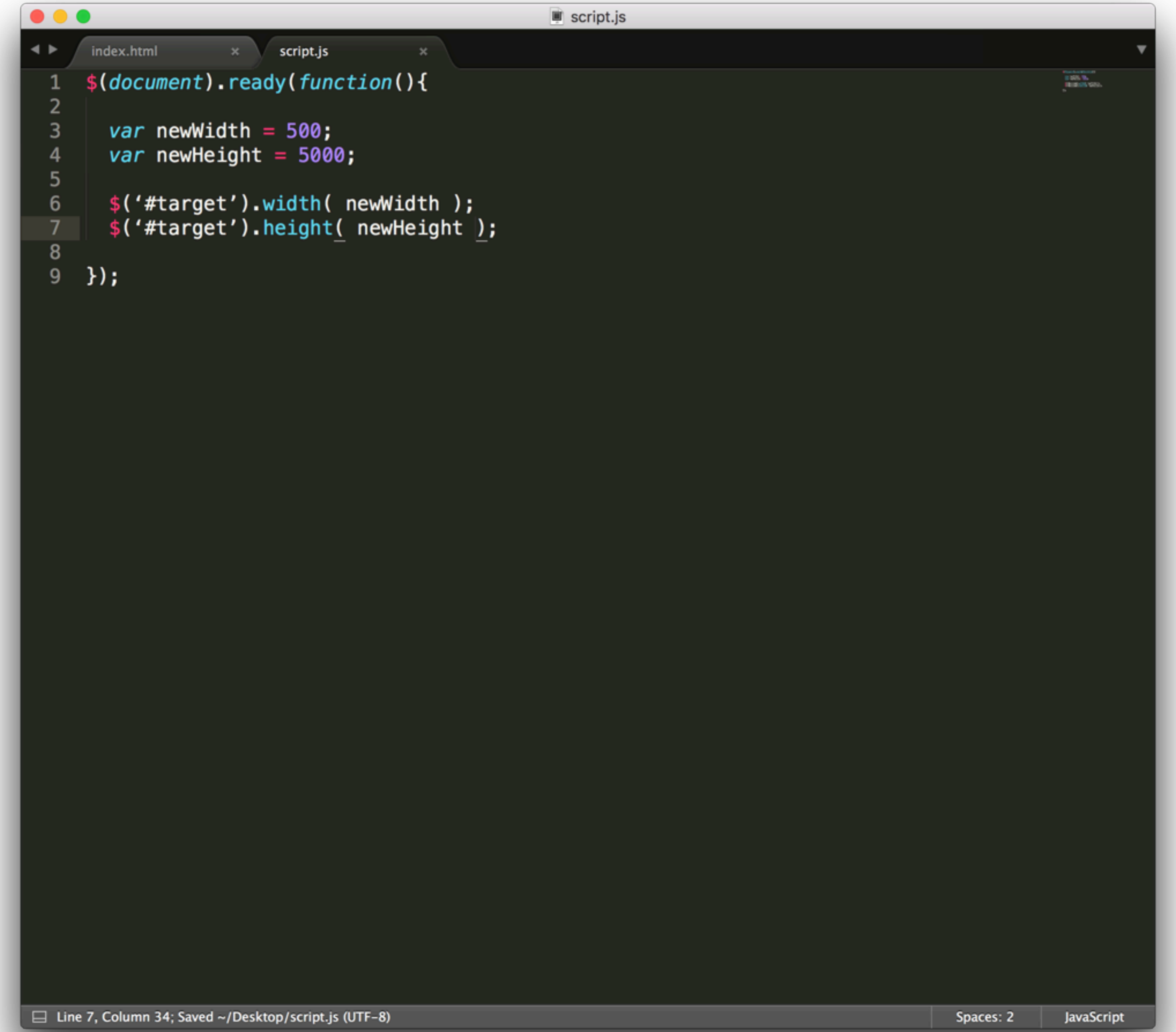
script.js

index.html × script.js ×

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

  console.log($('#target').width());
  console.log($('#target').height());

});
```

Line 4, Column 34; Saved ~/Desktop/script.js (UTF-8) Spaces: 2 JavaScript

In this example we're getting the width and height of the element with an id of target using `.width` and `.height`. We then print the values to the console.



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

  var newWidth = 500;
  var newHeight = 5000;

  $('#target').width( newWidth );
  $('#target').height( newHeight );

});
```

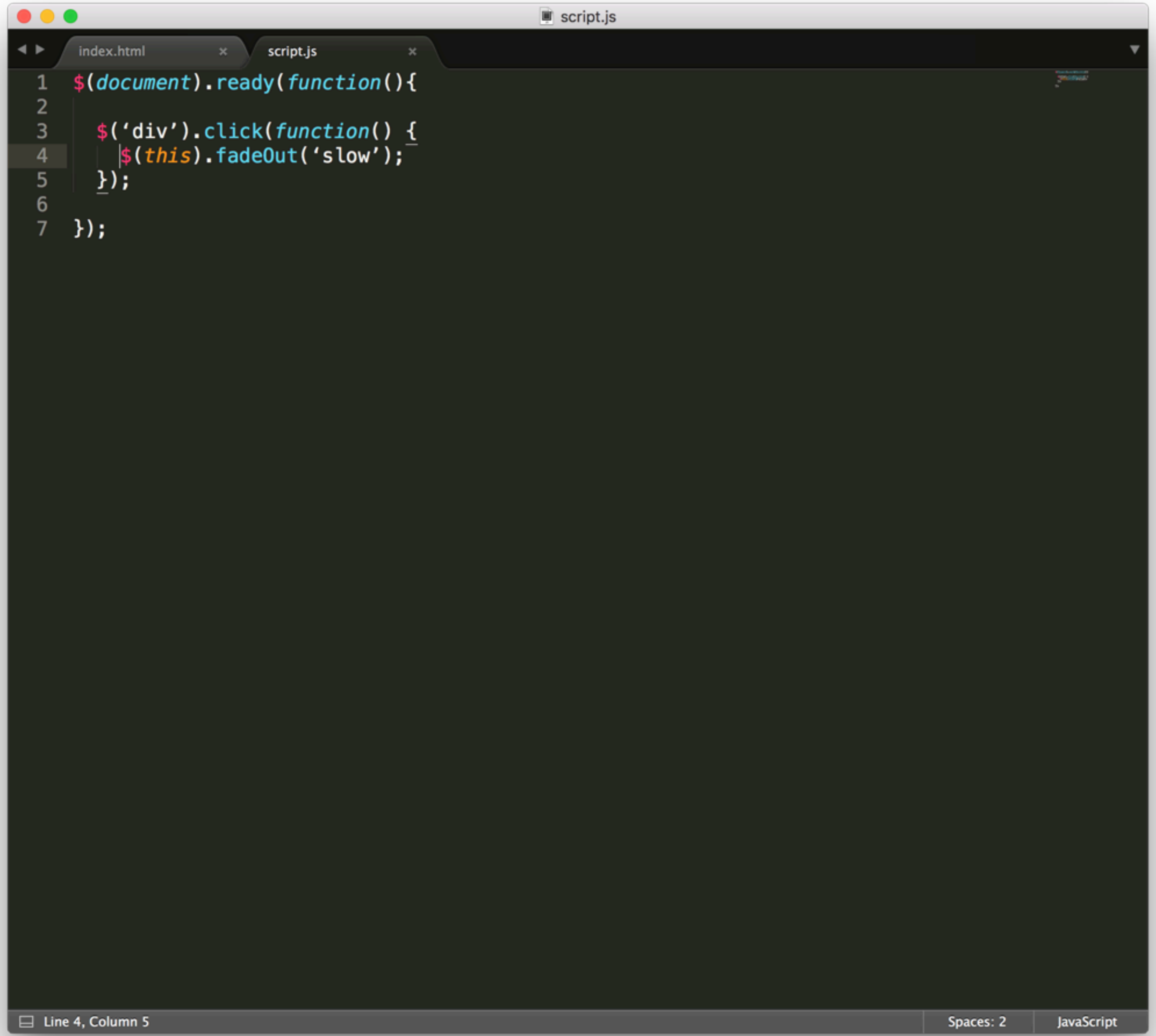
You can also use .width and .height to set these attributes on an element or group of elements.

js Fiddle Examples

‘this’ is Important!

The this keyword refers to the jQuery object you’re currently doing something with.

`$(this)`, and the event will only affect the element you’re currently doing something with (for example, clicking on or mousing over).



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

  $('div').click(function() {
    $(this).fadeOut('slow');
  });

});
```