jQuery

jQuery JavaScript library:

- In widespread but possibly declining use.
- Competitors exist:
  - Prototype
  - Modernizer
  - Frameworks (React, Vue, Angular, Aurelia, Ember, . . .)
- jQuery hides browser incompatibilities
- Facilitates finding & manipulating elements in document

Facilitates modifications to web page:

- Add or change specific content
- Change HTML attributes
- Change CSS properties
- Define event handlers
- Perform animation

```
<body>
 <button class="edit" onclick="change()">Change</button>
 <script src="jquery.js"></script>
 <script>
  function change(){
   $("button.edit").html("Next Step...");
 }
 </script>
</body>
```

jQuery focussed on queries
Typically uses CSS selectors

- Identify set document elements
- Return object representing these
- Object has useful methods to operate on data
- Method chaining provided where possible
- Can operate on elements as group rather than individually

```
// Returns a jQuery object containing all div elements in document.
// Observe jQuery variable naming convention: $div.
let $divs = $('div');
console.log($divs);
```

```
<body>
 <button onclick="lotsadivs()">Press</button>
 <div id= 'div-1'>
  <div id= 'div-2'>
  </div>
 </div>
 <script src="jquery-2.2.3.min.js"></script>
 <script src="jquery.js"></script>
</body>
```

```
//File: jquery.js
function lotsadivs(){
 console.log($('div').length);
}
```

```
// output
[div#div-1, div#div-2, ... selector:"div"]
```

Terminology

the jQuery function

- $() or jQuery(): single global function

a jQuery object

- is object returned by $()

the selected elements

- determined by CSS selector parameter in $

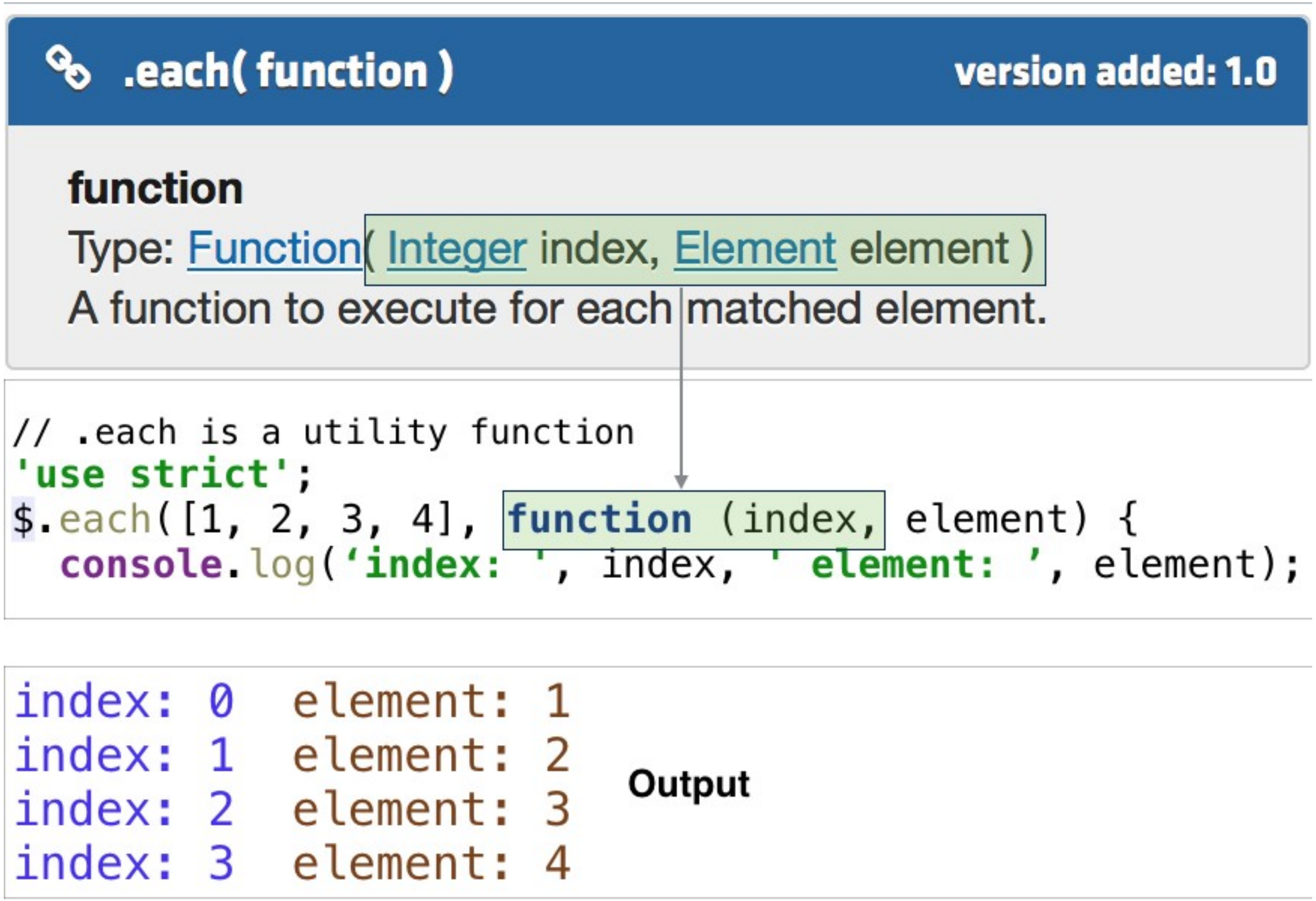
a jQuery function

- a function defined within $()

a jQuery method

- bound to jQuery object

```
//these 2 methods exactly the same
function change(){
  $('button.edit').html('Next');
}
function change(){
  jQuery('button.edit').html('Next');
}
//jQuery function: invoke func for each element of array
$.each(array, func);

//jQuery method: invoke func2 once for each selected element
$('a').each(func2);
```



.each(function) version added: 1.0

function

Type: Function(Integer index, Element element)

A function to execute for each matched element.

```
// .each is a utility function
'use strict';
$.each([1, 2, 3, 4], function (index, element) {
  console.log(‘index: ', index, ' element: ’, element);
```

```
index: 0  element: 1
index: 1  element: 2
index: 2  element: 3
index: 3  element: 4
```

Output

jQuery: How to Obtain

Method 1

- Download from *jquery.com*

Method 2

- Use a content distribution network (CDN)
  - code.jquery.com/jquery-3.0.0.min.js
  - ajax.microsoft.com/ajax/jquery/jquery-3.0.0.min.js
  - ajax.googleapis.com/ajax/libs/jquery/3.0.0/jquery.min.js

```
//include jquery script tag before other script calls at end document    body
//ensure use latest versions jquery (not shown here)
<script src=“//ajax.googleapis.com/ajax/libs/jquery.3.0.0/jquery.min.js”>
```

JavaScript

JQuery source - development usage

```
// extract from jquery−2.2.4.js
jQuery = function( selector, context ) {
 // The jQuery object is actually just the init constructor 'enhanced' return
 new jQuery.fn.init( selector, context, rootjQuery );
},
// Match a standalone tag
rsingleTag = /^<(\w+)\s*\/?>(?:<\/\1>|)$/,

// Matches dashed string for camelizing
rmsPrefix = /^-ms-/,
rdashAlpha = /-([\da-z])/gi,

// Used by jQuery.camelCase as callback to replace()
fcamelCase = function( all, letter )
  { return letter.toUpperCase();
},
```

JavaScript

Minified jQuery

- // extract from jquery-1.6.4.min.js
- /*! jQuery v1.6.4 http://jquery.com/ | http://jquery.org/license */ (function(a,b)
  {function cu(a){return f.isWindow(a)?a:a.nodeType===9?a.
  - defaultView||a.parentWindow:!1}function cr(a){if(!cg[a]){var b=c.body,d=f
  - ("<"+a+">").appendTo(b),e=d.css("display");d.remove();if(e==="none"||e
  - ===""){ch||(ch=c.createElement("iframe"),ch.frameBorder=ch.width=ch.
    height=0),b.appendChild(ch);if(!ci||!ch.createElement)ci=(ch. contentWindow||
    ch.contentDocument).document,ci.write((c.compatMode=== "CSS1Compat"?"<!doctype
    html>":"")+"<html><body>"),ci.close();d=ci.
    createElement(a),ci.body.appendChild(d),e=f.css(d,"display"),b. removeChild(ch)}cg[a]=e}
    return cg[a]}function cq(a,b){var c={};f.each(cm. concat.apply([],cm.slice(0,b)),function()
    {c[this]=a});return c}function cp()
  - {cn=b}function co(){setTimeout(cp,0);return cn=f.now()}function cf(){try
  - {return new a.ActiveXObject("Microsoft.XMLHTTP")}catch(b){}}function
    ce(){try{return new a.XMLHttpRequest}catch(b){}}

JavaScript

Window object

- Window object
  - The global object
  - Represents open window in browser
- Entry point client-side JavaScript
- Defines properties such as:
  - location: navigates to new page
  - document: returns DOM object

```
<!DOCTYPE html>
<html>
 <head>
  <title>Window</title>
 </head>
 <body>
  <script>
   window.location = "http://www.wit.ie";
  </script>
 </body>
</html>
```

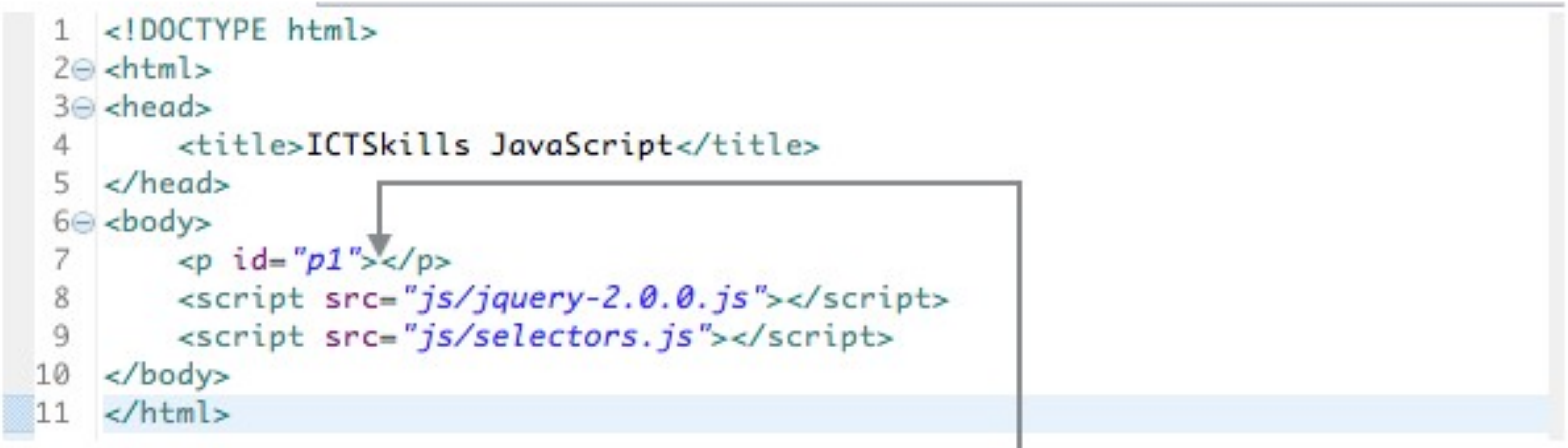
Invoking $() function

jQuery(), a.k.a **$()**, invokable with 3 different parameter types:

- CSS selector
- Element, Document or Window object
- String of HTML

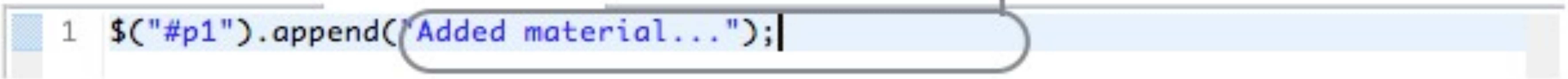
```
//CSS selector: <p id="p1"></p>
$('#p1').append('Added material...');

//Element: <p class='p2'></p>
$('p.p2').append('Second para');

//String HTML: dynamically add node
$('<p>Third para</p>').appendTo('body');
```

example.html



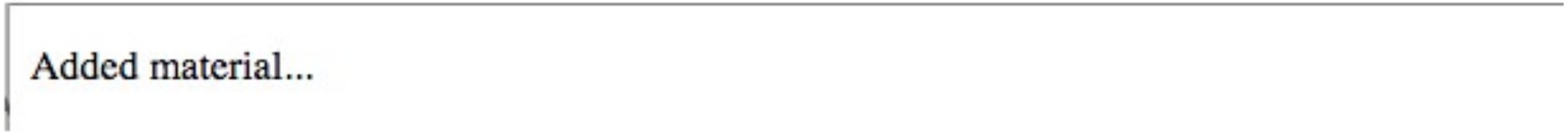
```html
<!DOCTYPE html>
<html>
<head>
    <title>ICTSkills JavaScript</title>
</head>
<body>
    <p id="p1"></p>
    <script src="js/jquery-2.0.0.js"></script>
    <script src="js/selectors.js"></script>
</body>
</html>
```

selectors.js



```javascript
$("#p1").append("Added material...");
```

example.html : output in browser



```
Added material...
```

jQuery

Immediately-Invoked Function Expression (IIFE)

```
// Click button to trigger alert
<button onclick="clickbutton()">Click</button>
<script src="jquery.js"></script>
<script>
    function clickbutton(){
      alert('Button clicked');
 }
</script>
```

```
// Example of self−invoking function
// Alert triggered on refreshing page
<script src="jquery.js"></script>
<script>
    (function clickbutton(){
      alert('Button clicked');
    }());

</script>
```

jQuery
Execute function when document loaded

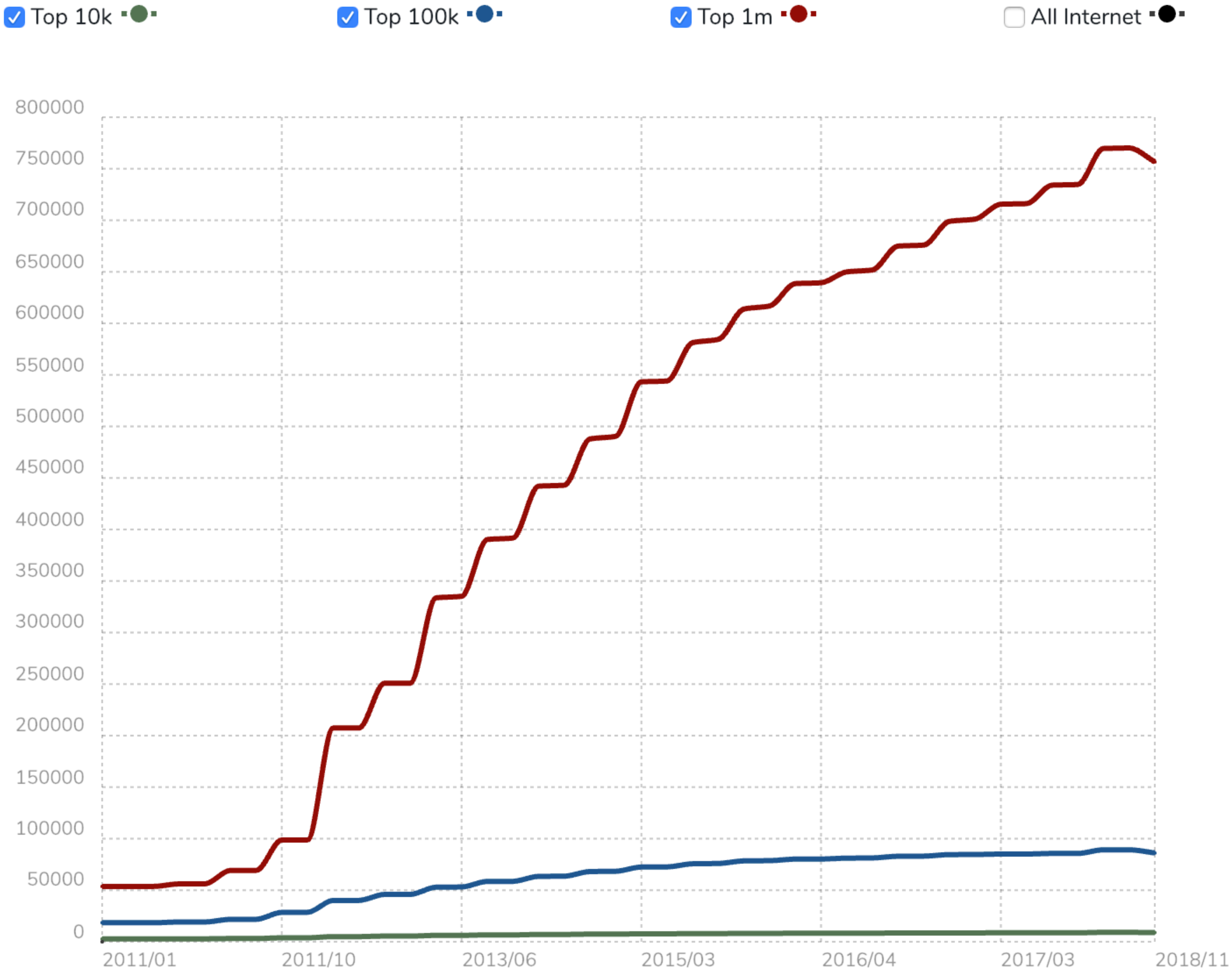
onload event

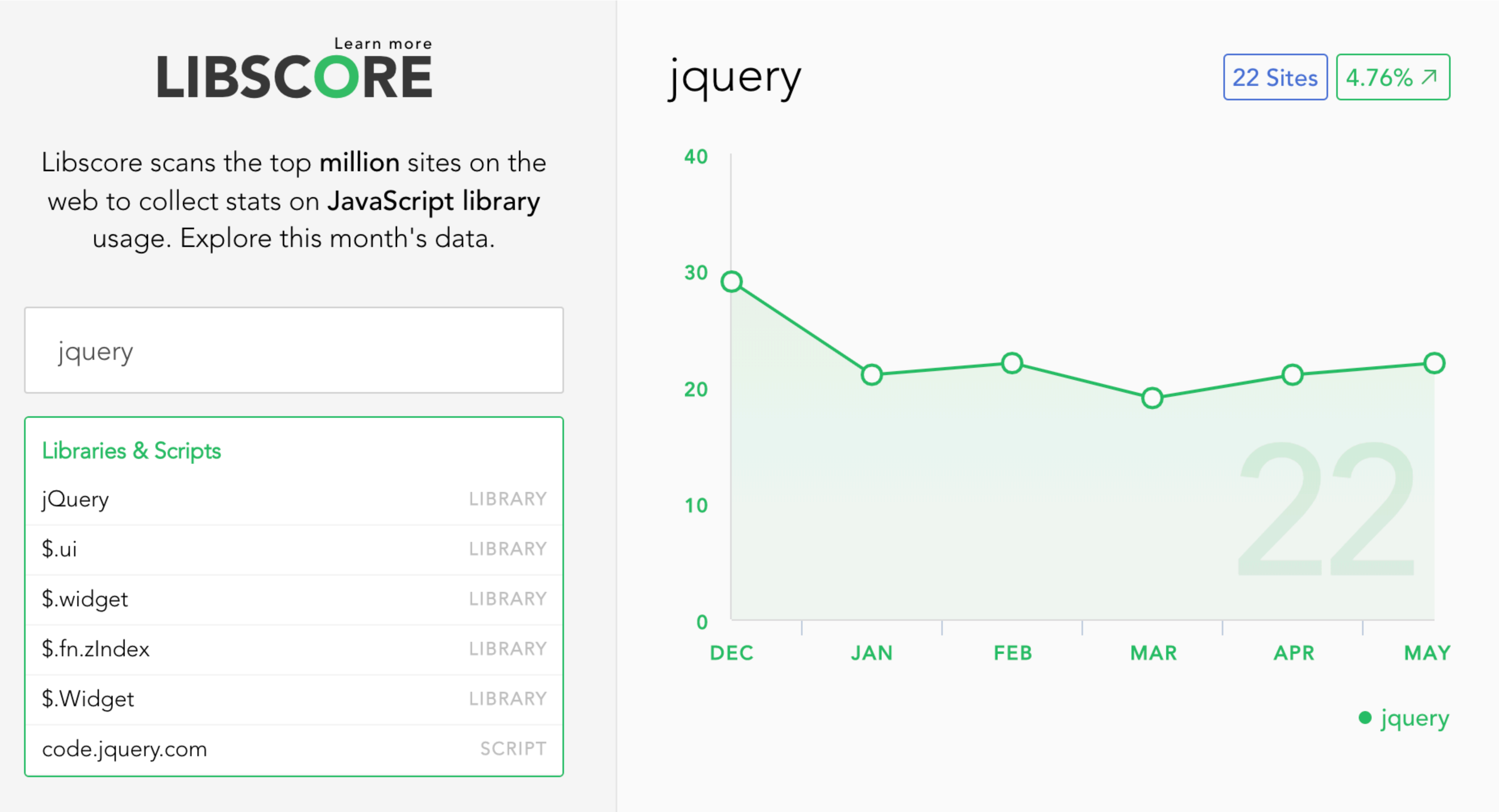
- Ensure page load complete before accessing script code.

```
$(function () {
 alert('Page has loaded');
});
```

```
// Older, verbose equivalent
$(document).ready(function () {
  alert('Page has loaded');
});
```

jQuery Usage Statistics





Learn more

LIBSCORE

Libscore scans the top **million** sites on the web to collect stats on **JavaScript library** usage. Explore this month's data.

jquery

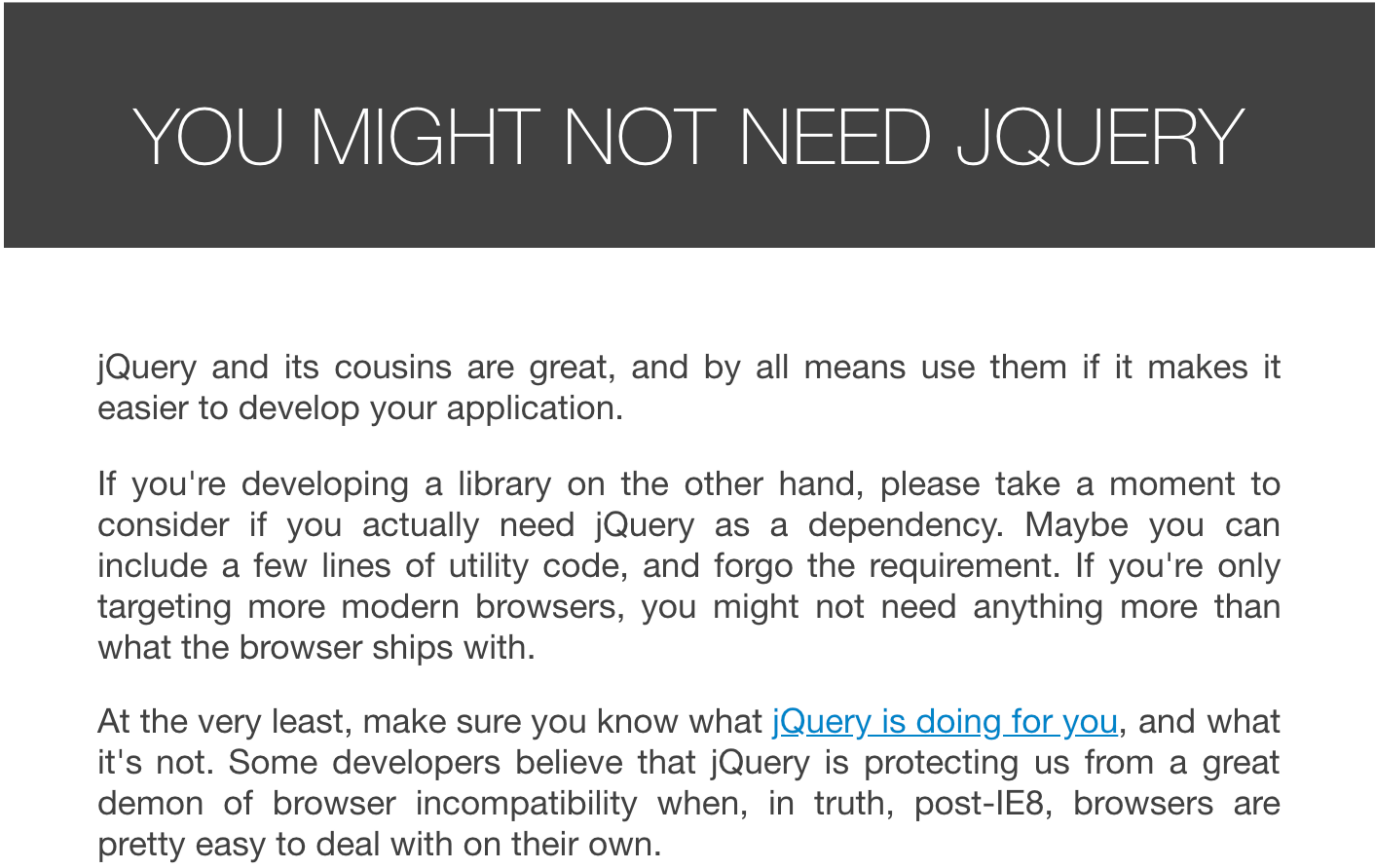
Libraries & Scripts

jQuery	LIBRARY
$.ui	LIBRARY
$.widget	LIBRARY
$.fn.zIndex	LIBRARY
$.Widget	LIBRARY
code.jquery.com	SCRIPT

jquery

22 Sites

4.76% ↗



YOU MIGHT NOT NEED JQUERY

jQuery and its cousins are great, and by all means use them if it makes it easier to develop your application.

If you're developing a library on the other hand, please take a moment to consider if you actually need jQuery as a dependency. Maybe you can include a few lines of utility code, and forgo the requirement. If you're only targeting more modern browsers, you might not need anything more than what the browser ships with.

At the very least, make sure you know what jQuery is doing for you, and what it's not. Some developers believe that jQuery is protecting us from a great demon of browser incompatibility when, in truth, post-IE8, browsers are pretty easy to deal with on their own.

http://youmightnotneedjquery.com/

JQUERY

```
$.getJSON('/my/url', function(data) {

});
```

IE9+

```
var request = new XMLHttpRequest();
request.open('GET', '/my/url', true);

request.onload = function() {
  if (request.status >= 200 && request.status < 400) {
    // Success!
    var data = JSON.parse(request.responseText);
  } else {
    // We reached our target server, but it returned an er

  }
};

request.onerror = function() {
  // There was a connection error of some sort
};

request.send();
```

Post

JQUERY

```
$.ajax({
  type: 'POST',
  url: '/my/url',
  data: data
});
```

IE8+

```
var request = new XMLHttpRequest();
request.open('POST', '/my/url', true);
request.setRequestHeader('Content-Type', 'application/x-ww
request.send(data);
```

Summary

- jQuery
  - A popular JavaScript library.
  - May be declining in popularity.
  - Losing market share to frameworks.
  - Abstracts browser inconsistencies.
  - Community support.