JavaScript Introduction

Topic discussed

- jQuery

jQuery

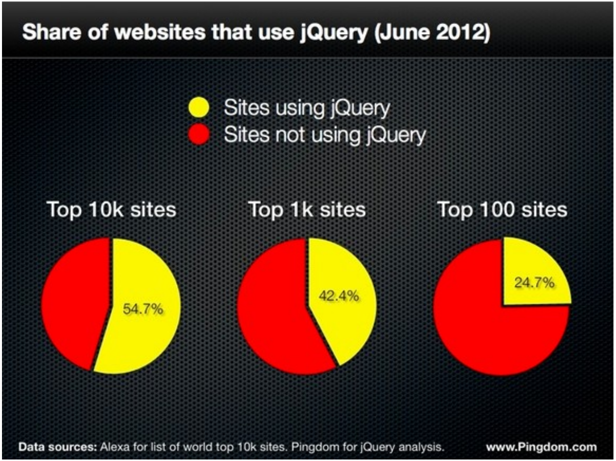
Introduction

jQuery JavaScript library:

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

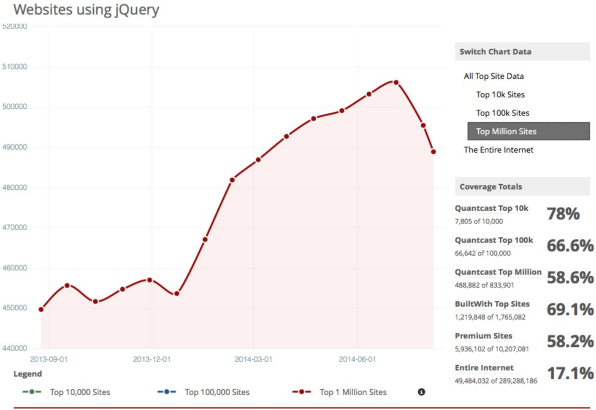
jQuery

Statistics



jQuery

Statistics



Websites using jQuery

Coverage Totals

Quantcast Top 10k 78%
7,805 of 10,000

Quantcast Top 100k 66.6%
66,642 of 100,000

Quantcast Top Million 58.6%
488,882 of 833,901

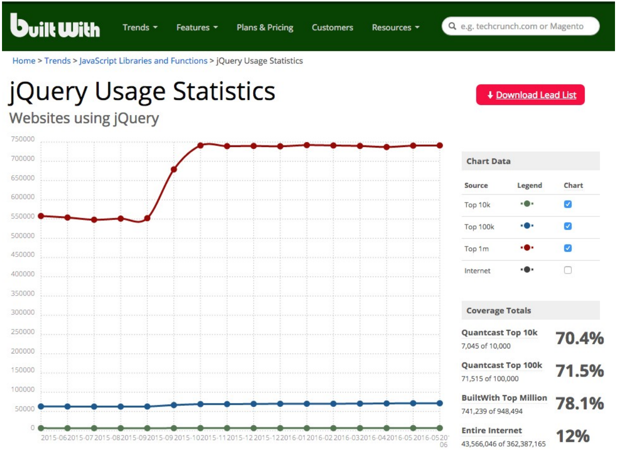
BuiltWith Top Sites 69.1%
1,219,848 of 1,765,082

Premium Sites 58.2%
5,936,102 of 10,207,081

Entire Internet 17.1%
49,484,032 of 289,288,186

jQuery

Statistics



BuiltWith Trends ▾ Features ▾ Plans & Pricing Customers Resources ▾ e.g. techcrunch.com or Magento

Home > Trends > JavaScript Libraries and Functions > jQuery Usage Statistics

jQuery Usage Statistics

⬇ Download Lead List

Websites using jQuery

Chart Data

Source	Legend	Chart
Top 10k	●	☑
Top 100k	●	☑
Top 1m	●	☑
Internet	●	□

Coverage Totals

Quantcast Top 10k 7,045 of 10,000	70.4%
Quantcast Top 100k 71,515 of 100,000	71.5%
BuiltWith Top Million 741,239 of 948,494	78.1%
Entire Internet 43,566,046 of 362,387,165	12%

jQuery

Features

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

Focus

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);
```

jQuery

Returned jQuery object

```
<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>
```

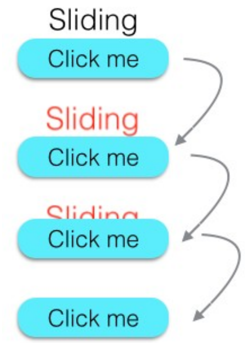
```
//File: jquery.js
function lotsadivs(){
  console.log($('div').length);
}
```

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

jQuery

Method chaining

```
//html
  <body>
    <p id="p1">Sliding</p>
    <button>Click me</button>

    <script src="jquery.js"></script>
    <script>
    $(function()
      {
      $('button').click(function(){
        $('#p1').css('color','red').slideUp(2000);
      });
    });
    </script>
  </body>
```



jQuery

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 $()

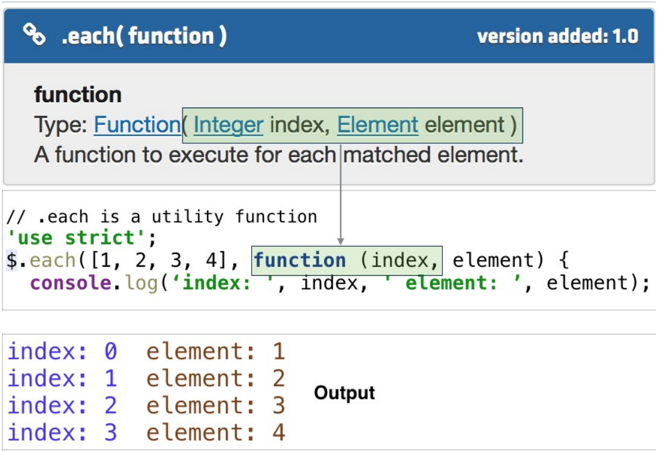
a jQuery method

- 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);
```

jQuery

$array.each()



.each(function) version added: 1.0

function

Type: Function(Integer index, Element element)

A function to execute for each matched element.

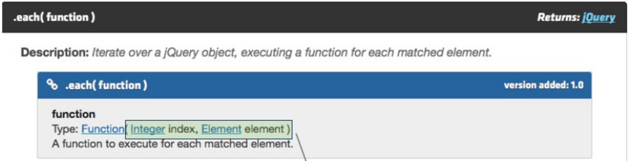
```
// .each is a utility function
'use strict';
$.each([1, 2, 3, 4], function (index, element) {
  console.log(‘index: ', index, ' element: ’, element);
```

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

Output

jQuery

$(array).each()

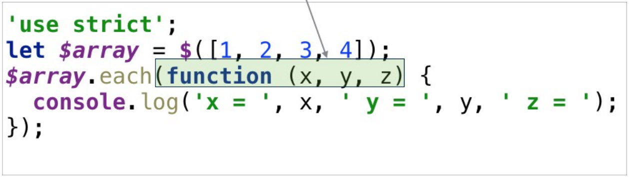


.each(function) *Returns:* *jQuery*

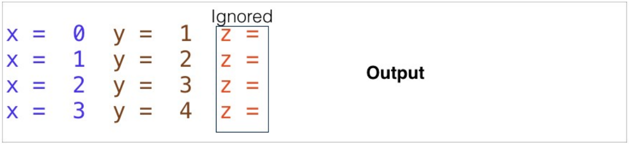
Description: *Iterate over a jQuery object, executing a function for each matched element.*

.each(function) version added: 1.0

function
Type: Function(Integer index, Element element)
A function to execute for each matched element.



```
'use strict';
let $array = $([1, 2, 3, 4]);
$array.each(function (x, y, z) {
  console.log('x = ', x, ' y = ', y, ' z = ');
});
```

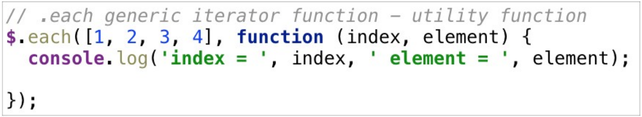


```
                  Ignored
x = 0   y = 1    z =
x = 1   y = 2    z =
x = 2   y = 3    z =
x = 3   y = 4    z =
```

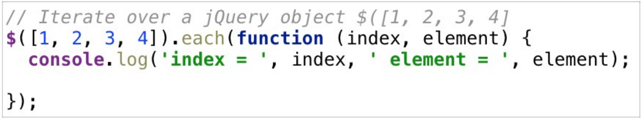
Output

jQuery

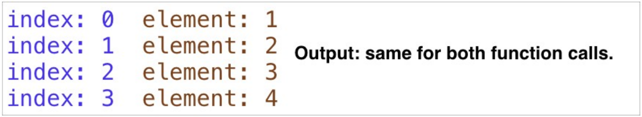
$.each() methods comparison



```
// .each generic iterator function – utility function
$.each([1, 2, 3, 4], function (index, element) {
  console.log('index = ', index, ' element = ', element);

});
```



```
// Iterate over a jQuery object $([1, 2, 3, 4]
$([1, 2, 3, 4]).each(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: same for both function calls.

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/1.11.1/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

Plaintext

```
// sample javascript code
function OBJNAME(n){ return n;}
function dw(str){ document.write(str);}
function dumptbl(obj, row_callback_str,cell_callback_str)
{
  dw('<table border=1>');
  for (var i = 0; i < obj.length; ++i)
  {
    var tr = '<tr>';
    eval(row_callback_str);
    dw(tr);
    for (var j = 0; j < obj[i].length; ++j)
    {
      var td_s = '<td>', td_e = '</td>';
      eval(cell_callback_str);
      dw(td_s); dw(obj[i][j]); dw(td_e);
    }
  }
  dw('</table>');
}
```

JavaScript

Obfuscated

```
// sample javascript code minified and obfuscated
function z8c231aa888(z14851c4b0f){return z14851c4b0f;}function
z0ab1f0a49e(z0721975593){document.write(z0721975593);}function
zcd8c17c79d(z4716861143,z500f443098,z9bc82e0042){z0ab1f0a49e("\
x3c\x74\x61\x62\x6c\x65\x20\x62\x6f\x72\x64\x65\x72\x3d\x31\x3e
");for(varzd1ea46315e=(0x8e9+2039-0x10e0);zd1ea46315e<z4716861
143.length;++zd1ea46315e){var z708eb69ac7="\x3c\x74\x72\x3e";
eval(z500f443098);z0ab1f0a49z708eb69ac7);for(var z2d29194d43=
(0x139b+2094-0x1bc9);z2d29194d43<z4716861143[zd1ea46315e].length;
++z2d29194d43){var z23b8891aeb="\x3c\x74\x64\x3e",z7f5411ee29=
"\x3c\x2f\x74\x64\x3e";eval(z9bc82e0042);z0ab1f0a49e(z23b8891aeb)
;z0ab1f0a49e(z4716861143[zd1ea46315e][z2d29194d43]);z0ab1f0a49e
(z7f5411ee29);}}z0ab1f0a49e("\x3c\x2f\x74\x61\x62\x6c\x65\x3e")
;}zcd8c17c79d([[(0x2d7+5314-0x1798),(0xf7c+295-0x10a1),(0x900+
1599-0xf3c)],[(0x1e8+1063-0x60b),(0xfc1+580-0x1200),(0x1cf5+18
43-0x2422)],[(0x9f9+4410-0x1b2c),(0x1c6+8452-0x22c2),
```

JavaScript

Scope - 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>
```

jQuery

Invoking $() function

jQuery(), a.k.a **$()**, invokable with 4 different parameters:

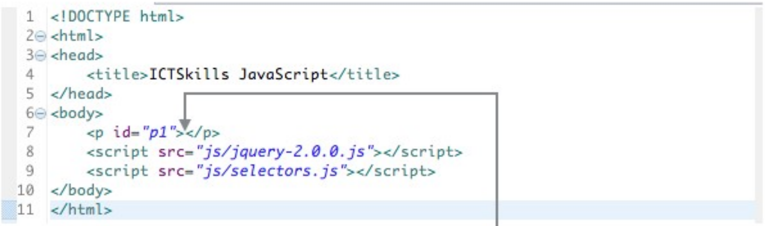
- CSS selector
- Element, Document or Window object
- String of HTML
- Function

```
//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');

//Function: function clickbutton param
$(function clickbutton(){
  alert('Button clicked');
});
```

jQuery

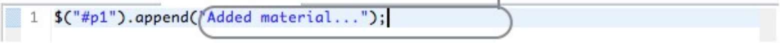
Invoking $() function

example.html



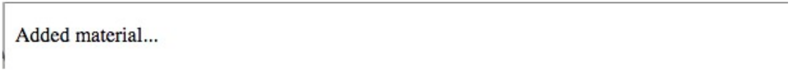
```
<!DOCTYPE html>
<html>
<head>
    <title>ICTSkills JavaScript</title>
</head>
<body>
    <p id="p1"></p>
    <script src="js/jquery-2.0.0.js"></script>
    <script src="js/selectors.js"></script>
</body>
</html>
```

selectors.js



```
$("#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');
});
```

Summary

- jQuery
  - A popular JavaScript library.
  - May be declining in popularity.
  - Losing market share to frameworks.
  - Abstracts browser inconsistencies.
  - Community support.
  - jquery-3.0.0 (June 2016) contains more than 10,000 lines.