



jQuery

Marco Torchiano

Version 1.2.0 - May 2020

License

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

- You are free to:
 - Share - copy and redistribute the material in any medium or format
 - Adapt - remix, transform, and build upon the materialfor any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.
- Under the following terms:
 - **Attribution** - You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **ShareAlike** - If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

jQuery

write less, do more.

- lightweight Javascript library
- works uniformly cross-browser
- simplifies most common operations
- history
 - first released in 2006
 - currently maintained by jQuery foundation

Official web site: <https://jquery.com>

3

Include library

The library can be downloaded and included locally or it can be included from a Content Delivery Network (CDN)

- recommended version is the compressed (*minified*) version

```
<script  
  src="https://code.jquery.com/jquery-3.5.1.min.js">  
</script>
```

```
script src="https://code.jquery.com/jquery-3.5.1.min.js">
```

4

Basic syntax

`$(selector).action()`

- `$` is used to access jQuery
- `selector` identifies DOM elements
- `action` operation to perform on selected element(s)

5

Selectors

- CSS selectors for
 - element, e.g. `"p"`
 - id, e.g. `"#basic-syntax"`
 - class, e.g. `".highlight"`
 - etc.
- special jQuery selectors
 - `$(document)` document element
 - `$(this)` current element
 - used inside the actions

6

Actions

Different methods groups:

- Events: set the function to be called when an event occurs on the selected element
- Visual effects: make the element appear/disappear
- HTML content: get or set content of element
- DOM traversing: navigate the DOM
- Selection: select a subset of the elements
- AJAX: load content through an AJAX request

7

HTML Content

- `text()`, `html()`, `attr()`:
 - get or set internal text or html, or attribute
- `append()`, `prepend()`:
 - add an element inside the selected one
- `before()`, `after()`:
 - add an element before/after the selected one
- `css()`:
 - get/set a css style property
- `addClass()`, `removeClass()`:
 - modify the class list of the element

8

Content generation (text)

```
var list = $('#content-generation-text ul');
['first', 'second', 'third'].forEach(
  item => list.append("<li>" + item + "</li>"));
```

```
<ul><li>item</li></ul>
```

- item
- first
- second
- third

9

Content generation (jquery)

```
var ol = $("<ol>");
['first', 'second', 'third'].forEach(
  item => ol.append($("<li>").text(item)) );
$('#content-generation-jquery li').append(ol);
```

`$("")` creates a new (detached) `` element

```
<ul><li>item</li></ul>
```

- item
 1. first
 2. second
 3. third

10

Visual effects

- `hide()` and `show()`: hides and makes the elements displayed again
- `toggle()` changes the display property of the elements
- `fadeIn()`, `fadeOut()` make element progressive appear/disappear
 - take optional parameter with speed
- `slideDown()`, `slideUp()` make element slide down/up

11

Event methods

Set the (callback) function to be called when an event occurs.

The main methods are:

- `ready()` element (document) loading is complete
- `click()` element is clicked
- `dblclick()` element is double clicked
- `mouseenter()` mouse enters the element area
- `mouseleave()` mouse leaves the element area

12

Event methods example

- `ready()` is used for the document

```
$(document).ready(function(){  
    console.log("Document finished loading...");  
});
```

- `click()` set the `onclick` event handler

```
$("#h2").click(function(){  
    $(this).css('color', 'purple');  
});
```

13

Traversal methods

- `parent()`, `parents()`
 - return immediate or all parent elements
- `children()`, `siblings()`, `next()`, `prev()`
 - return immediate children, siblings, next or previous elements
- `find()`
 - look for a element matching a selector

The methods have an optional selector argument to restrict the returned elements.

14

Traversal methods example

```
var thisCode = $('#traversal-methods-example').  
    find('pre');  
var originalColor = thisCode.css("background-color");  
thisCode.click(  
    function(){  
        if($(this).css("background-color")==originalColor){  
            $(this).css("background-color","seashell");  
        }else{  
            $(this).css("background-color", originalColor);  
        }  
    });
```

15

Selection methods

- `first()`, `last()`, `eq(n)`
 - select the first, last or n-th element
- `filter()`
 - apply a filtering selector to the set of elements

Example:

```
$('#selection-methods code').eq(2).click(  
    function(){ $(this).css('color','magenta'); }  
);
```

16

AJAX

Method `get ()`

`$.get (URL, outData, function (data, status, xhr))`

- `URL` is the URL of the resource
- `outData` (optional) is the data to be sent to the server
- `function` (optional) is the callback function
 - `data` contains the retrieved data
 - `status` contains the status of the request
 - `xhr` contains the `XMLHttpRequest` object

get () txt sample

```
$.get(DS+"GoT.txt",function(data,result){
    $('#get-txt-sample').append(
        $("<p>").addClass("jsresult").append("Result: ").
            append($("<code>").text(result)).
            append($("<pre>").addClass("jsresult").text(data))
    });
```

Result: success

Jon Snow
Tyrion Lannister
Daenerys Targaryen
Arya Stark

19

Method load ()

selector.load(URL,function(data,status,xhr))

- URL is the URL of the resource
- data (optional) is the data to be sent to the server
- function (optional) is the callback function
 - data contains the data retrieved
 - status contains the status of the request
 - xhr contains the *XMLHttpRequest* object

20

load() sample

```
$( '#load-sample' ).append(  
    $( "<pre>" ).addClass( 'jsresult' ).  
        load( DS+"GoT.txt" )  
);
```

```
Jon Snow  
Tyrion Lannister  
Daenerys Targaryen  
Arya Stark
```

21

Method getJSON()

```
$.getJSON( URL, function( data, status, xhr ) )
```

- `URL` is the URL of the resource
- `data` (optional) is the data to be sent to the server
- `function` (optional) is the callback function
 - `data` contains the data retrieved
 - `status` contains the status of the request
 - `xhr` contains the *XMLHttpRequest* object

22

getJSON() sample

```
$.getJSON(DS+"GoT.json",function(json){  
    var tab = tableToHtmlElement(json)  
    tab.classList.add('jsresult')  
    $('#getjson-sample').append(tab)  
});
```

ID	Surname	Name
4321	Snow	Jon
5765	Lannister	Tyrion
4663	Targaryen	Daenerys
9896	Stark	Arya

23

References

- W3Schools. JSON Tutorial. <http://www.w3schools.com/json/default.asp>
- jQuery API Documentation. <http://api.jquery.com>

24