



JQuery

Part 1

AGENDA

- 1 **Jquery overview**
- 2 **Basic Selectors**
- 3 Data
- 4 Traversing
- 5 Ajax

OVERALL INFORMATION

jQuery is a JavaScript Library that focuses on simplifying DOM manipulation, AJAX calls, and Event handling. It is used by JavaScript developers frequently.

Download the jQuery library from [jQuery.com](https://jquery.com)
Include jQuery from a CDN, like Google

Code with supporting IE6 and more:

JS	jQuery

Modules jQuery (https://github.com/jquery/jquery#how-to-build-your-own-jquery)	
ajax	All AJAX functionality: \$.ajax(), \$.get(), \$.post(), \$.ajaxSetup(), .load(), transports, and ajax event shorthands such as .ajaxStart().
ajax/xhr	The XMLHttpRequest AJAX transport only.
ajax/script	The <script> AJAX transport only; used to retrieve scripts.
ajax/jsonp	The JSONP AJAX transport only; depends on the ajax/script transport.
css	The .css() method. Also removes all modules depending on css (including effects , dimensions , and offset).
css/showHide	Non-animated .show(), .hide() and .toggle(); can be excluded if you use classes or explicit .css()calls to set the display property. Also removes the effects module.
deprecated	Methods documented as deprecated but not yet removed.
dimensions	The .width() and .height() methods, including inner- and outer- variations.
effects	The .animate() method and its shorthands such as .slideUp() or .hide("slow").
event	The .on() and .off() methods and all event functionality. Also removes event/alias.
event/alias	All event attaching/triggering shorthands like .click() or .mouseover().
event/focusin	Cross-browser support for the focusin and focusout events.
event/trigger	The .trigger() and .triggerHandler() methods. Used by alias and focusin modules.
offset	The .offset(), .position(), .offsetParent(), .scrollLeft(), and .scrollTop() methods.
wrap	The .wrap(), .wrapAll(), .wrapInner(), and .unwrap() methods.
core/ready	Exclude the ready module if you place your scripts at the end of the body. Any ready callbacks bound with jQuery() will simply be called immediately. However, jQuery(document).ready() will not be a function and .on("ready", ...) or similar will not be triggered.
deferred	Exclude jQuery.Deferred. This also removes jQuery.Callbacks. <i>Note</i> that modules that depend on jQuery.Deferred(AJAX, effects, core/ready) will not be removed and will still expect jQuery.Deferred to be there. Include your own jQuery.Deferred implementation or exclude those modules as well (grunt custom:-deferred,-ajax,-effects,-core/ready).
exports/global	Exclude the attachment of global jQuery variables (\$) and jQuery) to the window.
exports/amd	Exclude the AMD definition.

How to connect

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>jQuery</title>
6  </head>
7  <body>
8    This is my page
9    <!-- jQuery CDN -->
10   <script src="//ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
11   <script>
12     window.jQuery || document.write('<script src=".js/jquery-3.3.1.min.js"></script>')
13   </script>
14   <!-- IE6-8 -->
15   <script src="//ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
16   <script>
17     var jQuery_1_6_1 = $.noConflict(true);
18   </script>
19   <script>
20     $(document).ready(function () {
21       // some operations
22       $('div').children().addBack().addClass('selected')
23       jQuery_1_6_1('div').children().andSelf().addClass('selected')
24     })
25   </script>
26 </body>
27 </html>
```

7

CONFIDENTIAL

Downloading jQuery

Production version - this is for your live website because it has been minified and compressed

Development version - this is for testing and development (uncompressed and readable code)

[jQuery.com](https://jquery.com).

```
<head>  
  <script src="jquery-3.2.1.min.js"></script>  
</head>
```


jQuery CDN

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network). Both Google and Microsoft host jQuery.

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

What is \$???

\$ is just an alias for jQuery, so all functionality is available without using \$

(\$(); === jQuery();)

```
1 $(document).ready(function () {  
2     // some operations  
3     $('div'). addClss('selected')  
4 }  
5 }
```

```
1  
2 var j = jQuery.noConflict();  
3 // Do something with jQuery  
4 j( "div p" ).hide();  
5 // Do something with another library's $()  
6 $( "content" ).style.display = "none";  
7
```

```
1 var dom = {};  
2 dom.query = jQuery.noConflict( true );  
3  
4 // Do something with the new jQuery  
5 dom.query( "div p" ).hide();  
6 // Do something with another library's $()  
7 $( "content" ).style.display = "none";  
8 // Do something with another version  
9 of jQuery  
10 jQuery( "div > p" ).hide();
```

The Document Ready Event

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

```
    // jQuery methods go here...
```

```
});
```

```
$(function(){
```

```
    // jQuery methods go here...
```

```
});
```

Selectors

Attribute

Basic

Basic Filter

Child Filter

Content Filter

Form

Hierarchy

jQuery Extensions

Visibility Filter

selectors

Filters

`:even`, `:odd`, `:first`, `:last`, `:not()`, `:empty`

`:gt()` /* all elements at an index greater then specified */

`:lt()` /* all elements at an index less then specified */

`:hidden` /* display: none, type="hidden", width & height = 0, ancestor is hidden */

`:visible` /* are visible */

`:parent` /* are parents to other elements, including text node */

`:contains()` /* contain the specified text */

`:has()` /* contain at least one element that matches the specified selector */

Attribute

`[name]`, `[name|='value']`, `[name*='value']`, `[name~='value']`,
`[name$='value']`, `[name^='value']`
`[name!='value']`

Child filters

`:first-child`, `:last-child`, `:only-child`, `:first-of-type`, `:last-of-type`,
`:only-of-type`, `:nth-child()`, `:nth-last-child()`, `:nth-last-of-type()`,
`:nth-of-type()`

example

```
var $products = $("div.products");  
var $products = $(".products");
```

```
$("div.data .gonzalez");
```

```
$(".data td.gonzalez");
```

```
$('#outer #inner');  
$('#div#inner');  
$('.outer-container #inner');  
$('#inner');
```

Data

These methods allow us to associate arbitrary data with specific DOM elements.

`.clearQueue()`

`.data()`

`.dequeue()`

`jQuery.dequeue()`

`jQuery.hasData()`

`jQuery.removeData()`

`.queue()`

`.removeData()`

```
<div id="book-1" data-name="HTML for kids" data-price="42"></div>
```

```
<script>
```

```
$('#book-1').data('price');
```

```
$('#book-1').attr('data-price', '51');
```

```
$('#book-1').data('price'); //42
```

```
$('#book-1').removeData('price');
```

```
$('#book-1').data('price'); //51
```

```
</script>
```

data()

The data() method attaches data to, or gets data from, selected elements.

```
$( "body" ).data( "foo", 52 );  
$( "body" ).data( "bar", { myType: "test", count: 40 } );  
$( "body" ).data( { baz: [ 1, 2, 3 ] } );  
$( "body" ).data( "foo" );  
$( "body" ).data();
```


attributes

These methods get and set DOM attributes of elements.

`.addClass()`

// Get the value from a dropdown select

`.attr()`

`$("select.foo option:selected").val();`

`.hasClass()`

// Get the value from a dropdown select even easier

`.html()`

`$("select.foo").val();`

`.prop()`

// Get the value from a checked checkbox

`.removeAttr()`

`$("input.checkbox:checked").val();`

`.removeClass()`

// Get the value from a set of radio buttons

`.removeProp()`

`$("input.radio[name=bar]:checked").val();`

jQuery HTML / CSS Methods

html()

```
$("#button").click(function(){  
    $("#p").html("Hello <b>world</b>!");  
});
```

val()

```
$("#button").click(function(){  
    $("#input:text").val("Glenn Quagmire");  
});
```

text()

```
$("#button").click(function(){  
    $("#p").text("Hello world!");  
});
```

empty()

```
<div class="vote">Content...</div>  
...  
<script>  
    $('<div class="vote">Content...</div>').empty();  
</script>
```

Traversing

```
$("#ul").click(function(event) {  
  var target = $(event.target);  
  if (target.is("li")) {  
    target.css("background-color", "red");  
  }  
});
```

is(),
not()
has()
eq()
first()
last()

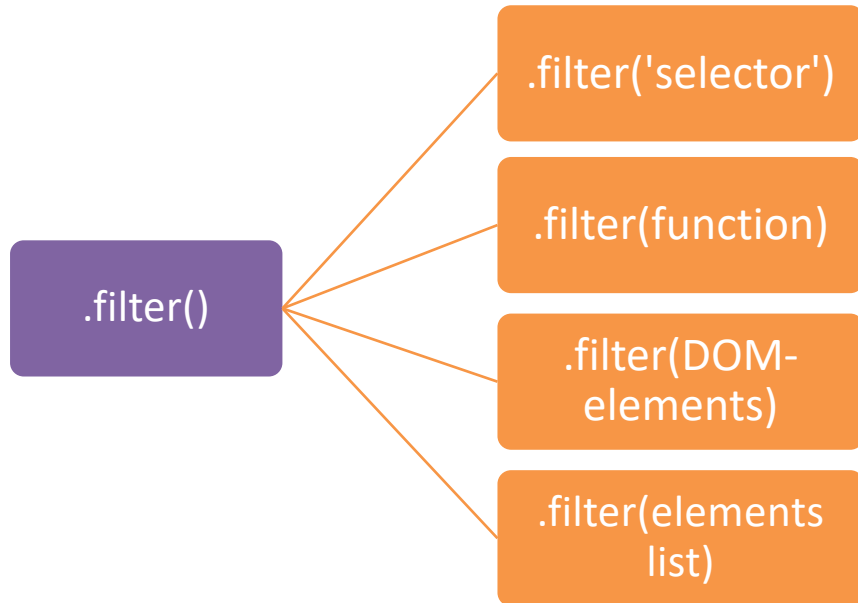
find()
parent()
parents()
closest()
children()
prev()

Traversing

Given a jQuery object that represents a set of DOM elements, the `.filter()` method constructs a new jQuery object from a subset of the matching elements. The supplied selector is tested against each element; all elements matching the selector will be included in the result.

`.filter()`

Reduce the set of matched elements to those that match the selector or pass the function's test.



filter()

```
<ul id=list>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
$('li').filter(function(index) {
    return index % 2 === 0;
});
```

```
$("li").filter($("#list>li"));
```

```
$( "li" ).filter( ":odd" ).css( "background-color", "red" );
```

```
$("li").filter(document.querySelectorAll("li:first-child"));
```

Traversing

.find()

Get the descendants of each element in the current set of matched elements, filtered by a selector, jQuery object, or element

```
<li class="item-ii">II
  <ul class="level-2">
    <li class="item-a">A</li>
    <li class="item-b">B
      <ul class="level-3">
        <li class="item-1">1</li>
        <li class="item-2">2</li>
        <li class="item-3">3</li>
      </ul>
    </li>
    <li class="item-c">C</li>
  </ul>
</li>
```

```
$( "li.item-ii" ).find( "li" ).css( "background-color", "red" );
```

Find element in DOM

```
var $productIds = $("#products div.id");
```

```
var $productIds = $("#products").find("div.id");
```

DOM manipulation

<u>.after()</u>	Insert content, specified by the parameter, after each element in the set of matched elements.
<u>.append()</u>	Insert content, specified by the parameter, to the end of each element in the set of matched elements.
<u>.before()</u>	Insert content, specified by the parameter, before each element in the set of matched elements.
<u>.clone()</u>	Create a deep copy of the set of matched elements.
<u>.detach()</u>	Remove the set of matched elements from the DOM.
<u>.empty()</u>	Remove all child nodes of the set of matched elements from the DOM.
<u>.position()</u>	Get the current coordinates of the first element in the set of matched elements, relative to the offset parent.
<u>.remove()</u>	Remove the set of matched elements from the DOM.
<u>.removeAttr()</u>	Remove an attribute from each element in the set of matched elements.
<u>.val()</u>	Get the current value of the first element in the set of matched elements or set the value of every matched element.

example

```
<div class="container">  
  <div class="hello">Hello</div>  
  <div class="goodbye">Goodbye</div>  
</div>
```

```
$( ".hello" ).remove();
```

This will result in a DOM structure with the `<div>` element deleted:

```
<div class="container">  
  <div class="goodbye">Goodbye</div>  
</div>
```

Document Loading

`jQuery.holdReady()`

Holds or releases the execution of jQuery's ready event.

`jQuery.ready`

A Promise-like object (or “thenable”) that resolves when the document is ready.

`.load()`

Bind an event handler to the “load” JavaScript event.

`.ready()`

Specify a function to execute when the DOM is fully loaded.

```

$( "#book" ).load(function() {
    // Handler for .load() called.
});
```

Basics effects

```
<script>
$( "#showr" ).click(function() {
    $( "div" ).first().show( "fast", function showNext() {
        $( this ).next( "div" ).show( "fast", showNext );
    });
});
```

```
$( "#hidr" ).click(function() {
    $( "div" ).hide( 1000 );
});
```

```
</script>
```

.hide() Hide the matched elements.

.show() Display the matched elements.

.toggle() Display or hide the matched elements.



With no parameters, the `.show()` method is the simplest way to display an element:

```
$("#myLink").on("click", function(){...});
```

```
function myLinkClickHandler(){...}  
$("#myLink").on("click", myLinkClickHandler);
```

```
$("#list a").on("click", myClickHandler);
```

```
$("#list").on("click", "a", myClickHandler);
```

Ajax

These methods register handlers to be called when certain events, such as initialization or completion, take place for any Ajax request on the page. The global events are fired on each Ajax request if the global property in `jQuery.ajaxSetup()` is true, which it is by default.

```
$.ajax({  
  type: 'POST',  
  url: 'authorization.php',  
  data: { login: 'Max',  
          password: '123654'  
        },  
  success: onSuccess,  
  error: onError,  
  complete: onComplete  
});
```

load

```
.load(url [,data] [,complete])
```

```
<div id="ajaxcontent">
<button id="ajaxclick"> Get content
demo.html</button>
</div>
...
<script>
$('#ajaxclick').click(function(){
$(this).parent().load('demo.html
#content1');
});
</script>
```

```
Function(responseText, textStatus, xhr))
```

```
<!DOCTYPE html>
<html lang="uk">
  <head>
    <meta charset="utf-8">
    <title>Файл demo.html</title>
  </head>
  <body>
    <div id="content1">...</div>
    <div id="content2">...</div>
  </body>
</html>
```

get()

```
$.get(url [,data] [,success] [,dataType]);  
$.get([settings]);
```

```
<div id="content"></div>
```

```
...
```

```
<script>
```

```
$(function(){
```

```
    $.get({
```

```
        url: 'content.tpl',
```

```
        success: function(data) {
```

```
            $('#content').empty().append(data);
```

```
        }
```

```
    });
```

```
});
```

```
</script>
```

getJSON

```
$.getJSON(url [,data] [,success]);
```

```
<ul id="pages"></ul>
```

```
...
```

```
<script>
```

```
$(function () {
```

```
    $.getJSON({
```

```
        url: 'pages.json',
```

```
        success: function (data) {
```

```
            var output = '';
```

```
            for (var i in data.pages) {
```

```
                output += '<li><a href="' + data.pages[i].url + '>' +  
                    data.pages[i].title + "</a></li>";
```

```
            }
```

```
$('#pages').empty().append(output);
```

```
    } }); });
```

```
</script>
```


Sample Ajax Template

```
var jqxhr = $.ajax({  
  url: url, type: "GET",  
  cache: true,  
  data: {},  
  dataType: "json",  
  jsonp: "callback",  
  statusCode: {  
    404: handler404,  
    500: handler500  
  } });  
jqxhr.done(successHandler);  
jqxhr.fail(failureHandler);
```

FE Online UA Training Course Feedback

I hope that you will find this material useful.

If you find errors or inaccuracies in this material or know how to improve it, please report on to the electronic address:

serhii_shcherbak@epam.com

With the note [FE Online UA Training Course Feedback]

Thank you.

Q&A



DRIVEN



CANDID



CREATIVE



ORIGINAL



INTELLIGENT



EXPERT

UA Frontend Online LAB