

 免费电子书

学习

jQuery

Free unaffiliated eBook created from
Stack Overflow contributors.

#jquery

.....	1
1: jQuery	2
.....	2
.....	2
Examples.....	2
jQuery“jQuery”“\$”.....	2
.....	3
.....	3
HTML.....	4
.....	5
jQuery.....	6
jQuery.....	7
jQuery.....	7
2: CSS	9
.....	9
.....	9
Examples.....	9
CSS.....	9
CSS.....	10
/.....	10
CSS - GettersSetters.....	10
CSS Getter.....	10
CSS Setter.....	10
3: DOM	12
Examples.....	12
DOM.....	12
.....	12
API.....	14
html.....	15
.....	16
.....	17
.....	

.....	17
DOMsortList(DOM).....	17
doSort().....	17
\$('#btn-sort').....	17
.....	17
.....	18
.....	18
4: DOM	20
Examples.....	20
.....	20
jQuery.....	20
.....	20
nearest.....	21
.....	22
.....	22
.....	23
HTML.....	23
.....	23
.....	23
.....	23
.....	23
find.....	23
5: jQuery .animate	25
.....	25
.....	25
Examples.....	25
.....	25
6: jQueryPromises	27
.....	27
Examples.....	27
.....

.....	27
jQuery ajaxVS .done, .fail.....	28
.....	28
7:	29
.....	29
Examples.....	29
.....	29
.....	29
8:	31
Examples.....	31
.....	31
jQuery 2.2.3.....	31
jQuery 3.0.....	32
.....	32
.....	32
\$document.ready\$window.load.....	32
readyDOM.....	33
jQueryfn.....	33
9:	35
Examples.....	35
.....	35
.....	35
10: /	36
.....	36
Examples.....	36
2.....	36
11:	37
.....	37
Examples.....	37
HTML.....	37
HTML.....	37

.....	37
attrprop.....	38
12:	39
Examples.....	39
-	39
jQuery.fn.extend.....	40
13:	42
Examples.....	42
.....	42
jQuery.....	42
14:	43
.....	43
Examples.....	43
.....	43
.....	43
HTML.....	43
jQuery.....	43
.....	43
HTML.....	43
jQuery.....	43
jQuery.....	43
.....	44
HTML	44
.....	44
-	44
.....	44
.....	45
.load.....	45
ID.....	45
originalEvent.....	46
.....	46

jQuery	46
15:	48
Examples	48
.....	48
innerWidthinnerHeight	48
outerWidthouterHeight	48
16:	49
.....	49
.....	49
.....	49
Examples	49
.....	49
.....	51
.....	51
.....	52
“”	52
“”	52
“+”	52
“”	53
.....	53
.....	53
.....	53
.....	53
.....	53
DOM	54
HTML	54
17:	56
.....	56
.....	56
.....	56
Examples	56
\$.ajaxHTTP	56

Ajax.....	57
JSON.....	57
.....	58
Ajax.....	60
Ajax.....	60
1.....	60
2.....	61
3.FormData.....	61
4.Ajax.....	61
18:.....	63
.....	63
.....	63
.....	63
Examples.....	63
.....	63
.append.....	63
HTML.....	63
JS.....	64
.....	64
.....	64
Array*.....	65
HTMLjQuery.....	66
.....	66
.....	67
jQuery.....	67
.....	68

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jquery](#)

It is an unofficial and free jQuery ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jQuery.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: jQuery

jQuery JavaScript DOM AJAX。 DOM JavaScript。

jQuery <https://code.jquery.com/jquery/>。

1.0		2006-08-26
1.1		2007-01-14
1.2		2007-09-10
1.3	Sizzle	2009-01-14
1.4		2010-01-14
1.5	ajax	2011-01-31
1.6	attr() val()	2011-05-03
1.7	API on() off()。	2011-11-03
1.8	Sizzle \$(html, props)。	2012-08-09
1.9		2013115
1.10	1.9.2.0 beta	2013524
1.11		2014124
1.12		201618
2.0	IE 6-8	2013418
2.1		2014124
2.2		201618
3.0	jQuery	201669
3.1		201677

Examples

jQuery “jQuery” “\$”

jQuery

jQuery ◦ jQuery(...) jQuery.foo ◦

\$jQuery jQuery.noConflict(); - ◦

HTML -

```
<div id="demo_div" class="demo"></div>
```

jQuery div ◦ jQuery text() ◦ jQuery \$ ◦ -

```
jQuery("#demo_div").text("Demo Text!");
```

-

```
$("#demo_div").text("Demo Text!");
```

HTML -

```
<div id="demo_div" class="demo">Demo Text!</div>
```

\$jQuery jQuery ◦

jQuery CSS ID ◦ jQuery ◦

hello.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello, World!</title>
</head>
<body>
  <div>
    <p id="hello">Some random text</p>
  </div>
  <script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
  <script>
    $(document).ready(function() {
      $('#hello').text('Hello, World!');
    });
  </script>
</body>
</html>
```

JSBin

Web ◦ Hello, World!

1. jQuery CDN jQuery

```
<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
```

\$ global jQuery

jQuery

2. jQueryDOM

```
// When the `document` is `ready`, execute this function `...`
$(document).ready(function() { ... });

// A commonly used shorthand version (behaves the same as the above)
$(function() { ... });
```

3. DOMjQuery

1. `id#hello #hello` jQuery; [document.querySelectorAll MDN](#).
2. `text()` Hello, World!.

```
#    ↓ - Pass a `selector` to `$` jQuery, returns our element
$('#hello').text('Hello, World!');
#    ↑ - Set the Text on the element
```

[jQuery - Documentation](#)

HTML

[CDN](#) jQuery jQuery

jQuery CDN – Latest Stable Version

Powered by [MaxCDN](#)

jQuery Core

Showing the latest stable release in each major branch. [See all versions of jQuery Core.](#)

jQuery 3.x

- jQuery Core 3.1.0 - [uncompressed](#), [minified](#), [slim](#), [slim minified](#)

jQuery 2.x

- jQuery Core 2.2.4 - [uncompressed](#), [minified](#)

jQuery 1.x

- jQuery Core 1.12.4 - [uncompressed](#), [minified](#)

jQuery **jQuery 2.X**

jQuery CDN - Latest Stable Version

Powered by **MaxCDN**

Code Integration

jQuery

Showing

```
<script  
  src="https://code.jquery.com/jquery-2.2.4.min.js"  
  integrity="sha256-bbntvQ/qR/11qjAGBQqNtwq8LACKXxZKRute1T44=" "  
  crossorigin="anonymous"></script>
```

jQuery

- jQuery

jQuery

The `integrity` and `crossorigin` attributes are used for [Subresource Integrity \(SRI\) checking](#). This ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended practice, whenever libraries are loaded from a third-party source. Read more at srihash.org

- jQuery Core 2.2.4 - [uncompressed](#), [minified](#)

jQuery 1.x

- jQuery Core 1.12.4 - [uncompressed](#), [minified](#)

`html``<head><body>`

`async``head``JavaScript`

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Loading jquery-2.2.4</title>  
    <script src="https://code.jquery.com/jquery-2.2.4.min.js" async></script>  
  </head>  
  <body>  
    <p>This page is loaded with jquery.</p>  
  </body>  
</html>
```

`async``javascript`

`jQuery``$``jQuery`

`$`

```
jQuery.noConflict();
```

`$jQuery``jQuery``jQuery`

```
jQuery('#hello').text('Hello, World!');
```

jQuery

```
var jqy = jQuery.noConflict();
jqy('#hello').text('Hello, World!');
```

jQuery IIFE

```
(function($) {
    $(document).ready(function() {
        $('#hello').text('Hello, World!');
    });
})(jQuery);
```

IIFE \$jQuery

jQuery \$DOM

```
jQuery(function( $ ) { // DOM is ready
    // You're now free to use $ alias
    $('#hello').text('Hello, World!');
});
```

- `jQuery.noConflict() $jQuery jQuery`
- `var jQuery2 = jQuery.noConflict() - $jQuery jQuery2`

- jQuery jQuery2 jQuery2jQuery2

```
var jQuery2 = jQuery.noConflict(true)
```

`$jQuery jQuery`

jQuery

```
<script src='https://code.jquery.com/jquery-1.12.4.min.js'></script>
<script>
    var jQuery1 = jQuery.noConflict(true);
</script>
<script src='https://code.jquery.com/jquery-3.1.0.min.js'></script>
<script>
    // Here, jQuery1 refers to jQuery 1.12.4 while, $ and jQuery refers to jQuery 3.1.0.
</script>
```

<https://learn.jquery.com/using-jquery-core/avoid-conflicts-other-libraries/>

jQuery

`jQuery jQuery`

Chrome Developer Tools F12 jQuery

```
var j = document.createElement('script');
```

```
j.onload = function(){ jQuery.noConflict(); };
j.src = "https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js";
document.getElementsByTagName('head')[0].appendChild(j);
```

1.12.4 ◦

jQuery

jQuery\$() jQuery() newjQuery◦

```
// Define a local copy of jQuery
jQuery = function( selector, context ) {

    // The jQuery object is actually just the init constructor 'enhanced'
    // Need init if jQuery is called (just allow error to be thrown if not included)
    return new jQuery.fn.init( selector, context );
}
```

jQuery.fn jQuerynew ◦

jQuery API.each children filterjQuery undefined null◦ ◦

idjQueryDOM◦

```
var $div = $("#myDiv");//populate the jQuery object with the result of the id selector
var div = $div[0];//access array-like structure of jQuery object to get the DOM Element
```

jQuery

jQuery ◦

```
<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script src="some-plugin.min.js"></script>
```

jQueryjQueryjQuery.noConflict(true) ;jQuery

```
<script src="https://code.jquery.com/jquery-1.7.0.min.js"></script>
<script src="plugin-needs-1.7.min.js"></script>
<script>
// save reference to jQuery v1.7.0
var $oldjq = jQuery.noConflict(true);
</script>
<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script src="newer-plugin.min.js"></script>
```

jQuery

```
<script>
// newer jQuery document ready
jQuery(function($){
    // "$" refers to the newer version of jQuery
```

```
// inside of this function

// initialize newer plugin
$('#new').newerPlugin();
});

// older jQuery document ready
$(function(){
  // "$" refers to the older version of jQuery
  // inside of this function

  // initialize plugin needing older jQuery
  $('#old').olderPlugin();
});
</script>
```

jQuery。

jQuery <https://riptutorial.com/zh-CN/jquery/topic/211/jquery>

2: CSS

- `.csscssProperty`//CSS
- `.css[cssProperty...]`//Array of cssProperties
- `.csscssPropertyvalue`//
- `.css{cssPropertyvalue...}`//
- `.csscssPropertyfunction`//cssProperty

```
"auto" "%" "vw" "vw" px .css()px
```

```
.myElement{ width: 20%; }
```

```
var width = $(".myElement").css("width"); // "123px"
```

CSSStringcamelCase

```
"margin-bottom"  
marginBottom
```

String ◦ jQuery px

```
.css(fontSize: "1em")  
.css(fontSize: "16px")  
.css(fontSize: 16) // px will be used
```

jQuery 3 `.show()` `.show()` `.hide()`

jQuery `.show()` `.show()` `.hide()` ◦

```
display: none .show() ◦ jQuery 3.0 display: none .show() - .slideDown().fadeIn() - ◦  
"hidden" display: none display: none ◦ jQuery.addClass().removeClass() ◦ .ready()  
.hide() ◦ .css("display", "block") ◦
```

Examples

CSS

```
$('#target-element').css('color', '#000000');
```

```
$('#target-element').css({  
  'color': '#000000',  
  'font-size': '12pt',  
  'float': 'left',  
});
```


CSS

CSS.css (propertyName)

```
var color = $('#element').css('color');  
var fontSize = $('#element').css('font-size');
```

/

.css() += -= CSS

```
// Increment using the += syntax  
$("#target-element").css("font-size", "+=10");  
  
// You can also specify the unit to increment by  
$("#target-element").css("width", "+=100pt");  
$("#target-element").css("top", "+=30px");  
$("#target-element").css("left", "+=3em");  
  
// Decrementing is done by using the -= syntax  
$("#target-element").css("height", "-=50pt");
```

CSS - GettersSetters

CSS Getter

.css() **getter**DOM

```
// Rendered width in px as a string. ex: `150px`  
// Notice the `as a string` designation - if you require a true integer,  
// refer to `$.width()` method  
$("#body").css("width");
```

CSS\$("selector") DOMCSSundefinedundefined

CSS getter

```
$("#body").css(["animation", "width"]);
```

```
Object {animation: "none 0s ease 0s 1 normal none running", width: "529px"}
```

CSS Setter

.css() **setter**DOM

```
$("#selector").css("width", 500);
```

\$("#selector").width(500px)jQuery

.css() **setterCSS**

```
$("#body").css({"height": "100px", width:100, "padding-top":40, paddingBottom:"2em"});
```

setterDOMstyle!important◦

CSS <https://riptutorial.com/zh-CN/jquery/topic/2732/css>

3: DOM

Examples

DOM

jQuery\$ ◦

```
var myLink = $('<a href="http://stackexchange.com"></a>');
```

```
var myLink = $('<a>', { 'href': 'http://stackexchange.com' });
```

'<a>' ->DOM ◦ ◦ a ◦

{ 'href': 'http://stackexchange.com' } ->/JavaScript ◦

'name"value'< ><a name:value>

HTML

```
<p class="small-paragraph">
  This is a small <a href="https://en.wikipedia.org/wiki/Paragraph">paragraph</a>
  with a <a class="trusted" href="http://stackexchange.com">link</a> inside.
</p>
```

jQueryDOMhasClass() addClass() removeClass()toggleClass() ◦ class◦

```
$('.p').hasClass('small-paragraph'); // true
$('.p').hasClass('large-paragraph'); // false

// Add a class to all links within paragraphs
$('.p a').addClass('untrusted-link-in-paragraph');

// Remove the class from a.trusted
$('.a.trusted.untrusted-link-in-paragraph')
  .removeClass('untrusted-link-in-paragraph')
  .addClass('trusted-link-in-paragraph');
```

.toggleClass()

```
$(".small-paragraph").toggleClass("pretty");
```

true \$(".small-paragraph").hasClass("pretty")

toggleClass

```
if($(".small-paragraph").hasClass("pretty")){
  $(".small-paragraph").removeClass("pretty");}
```

```
else {
    $(".small-paragraph").addClass("pretty"); }

```

```
$(".small-paragraph").toggleClass("pretty cool");
```

```
/
```

```
$(".small-paragraph").toggleClass("pretty",true); //cannot be truthy/falsey
```

```
$(".small-paragraph").toggleClass("pretty",false);
```

```
$( "div.surface" ).toggleClass(function() {
    if ( $( this ).parent().is( ".water" ) ) {
        return "wet";
    } else {
        return "dry";
    }
});
```

```
// functions to use in examples
function stringContains(myString, mySubString) {
    return myString.indexOf(mySubString) !== -1;
}
function isOdd(num) { return num % 2;}
var showClass = true; //we want to add the class
```

```
/
```

```
$( "div.gridrow" ).toggleClass(function(index,oldClasses, false), showClass ) {
    showClass
    if ( isOdd(index) ) {
        return "wet";
    } else {
        return "dry";
    }
});
```

toggleClass

```
<div class="grid">
  <div class="gridrow">row</div>
  <div class="gridrow">row</div>
  <div class="gridrow">row</div>
  <div class="gridrow">row</div>
  <div class="gridrow">row</div>
  <div class="gridrow gridfooter">row but I am footer!</div>
</div>
```

```
function isOdd(num) {
    return num % 2;
}

function stringContains(myString, mySubString) {
    return myString.indexOf(mySubString) !== -1;
}
```

```
}  
var showClass = true; //we want to add the class
```

gridrow/

```
$("#div.gridrow").toggleClass(function(index, oldClasses, showThisClass) {  
  if (isOdd(index)) {  
    return "odd";  
  } else {  
    return "even";  
  }  
  return oldClasses;  
}, showClass);
```

gridfooter/

```
$("#div.gridrow").toggleClass(function(index, oldClasses, showThisClass) {  
  var isFooter = stringContains(oldClasses, "gridfooter");  
  if (isFooter) {  
    oldClasses = oldClasses.replace('even', ' ').replace('odd', ' ');  
    $(this).toggleClass("even odd", false);  
  }  
  return oldClasses;  
}, showClass);
```

◦ gridfooter **even / odd** ◦ **OLD** ◦ else return oldClasses;gridfooter - ◦

```
$("#div.gridrow").toggleClass(function(index, oldClasses, showThisClass) {  
  var isFooter = stringContains(oldClasses, "gridfooter");  
  if (!isFooter) {  
    if (isOdd(index)) {  
      return "oddLight";  
    } else {  
      return "evenLight";  
    }  
  } else return oldClasses;  
}, showClass);
```

API

jQueryDOM.

[.empty](#).

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

`$('#content').empty();` **div** ◦ `$('#content').html('');` `$('#content').html('');` ◦

[.closest](#)

```
<tr id="row_1">
  <td><button type="button" class="delete">Delete</button>
</tr>
```

```
$('.delete').click(function() {
  $(this).closest('tr');
});
```

delete.[click](#)\$(this)。

Deleteid

```
$('.delete').click(function() {
  var $row = $(this).closest('tr');
  var id = $row.attr('id');
});
```

\$ [jQuery](#)。

.closest() [.parents](#)

```
$('.delete').click(function() {
  var $row = $(this).parents('tr');
  var id = $row.attr('id');
});
```

[.parent](#)

```
$('.delete').click(function() {
  var $row = $(this).parent().parent();
  var id = $row.attr('id');
});
```

.parent() [DOM](#) [span](#) [jQuery](#)。

html

HTML。html

```
<div class="row">
  <div class="col-md-12">
    <div id="information">
      <p>Old message</p>
    </div>
  </div>
</div>
```

.html() 。

```
$("#information").html("<p>This is the new alert!</p>");
```

DOM

- 1.
- 2.
3. DOM

```
<ul id='my-color-list'>
  <li class="disabled">Red</li>
  <li>Green</li>
  <li class="disabled">Purple</li>
  <li>Orange</li>
</ul>
```

1. - `.children().find()`

◦

```
var $myColorList = $('#my-color-list');

// Elements one layer deep get .children(), any deeper go with .find()
var $colors = $myColorList.children('li');
```

2. - `Array.prototype.sort()`

HTML◦

```
/**
 * Bind $colors to the sort method so we don't have to travel up
 * all these properties more than once.
 */
var sortList = Array.prototype.sort.bind($colors);

sortList(function ( a, b ) {

  // Cache inner content from the first element (a) and the next sibling (b)
  var aText = a.innerHTML;
  var bText = b.innerHTML;

  // Returning -1 will place element `a` before element `b`
  if ( aText < bText ) {
    return -1;
  }

  // Returning 1 will do the opposite
  if ( aText > bText ) {
    return 1;
  }

  // Returning 0 leaves them as-is
  return 0;
});
```

3. - `.append()`

- `append()` **DOM**

```
// Put it right back where we got it
$myColorList.append($colors);
```

```
<!-- previous HTML above -->
<button type='button' id='btn-sort'>
  Sort
</button>
```

```
var ascending = true;
```

DOMsortList()DOM

```
var $myColorList = $('#my-color-list');
var $colors = $myColorList.children('li');
var sortList = Array.prototype.sort.bind($colors);
```

doSort ()

```
// Put the sortList() and detach/append calls in this portable little thing
var doSort = function ( ascending ) {

  sortList(function ( a, b ) {
    // ...
  });

  $myColorList.append($colors);
};
```

\$('#btn-sort')

```
$('#btn-sort').on('click', function () {
  // Run the sort and pass in the direction value
  doSort(ascending);

  // Toggle direction and save
  ascending = !ascending;
});
```

```
var ascending = true;

var $myColorList = $('#my-color-list');
var $colors = $myColorList.children('li');
var sortList = Array.prototype.sort.bind($colors);

var doSort = function ( ascending ) {

  sortList(function ( a, b ) {
```



```
var aText = a.innerHTML;
var bText = b.innerHTML;

if ( aText < bText ) {
    return ascending ? -1 : 1;
}

if ( aText > bText ) {
    return ascending ? 1 : -1;
}

return 0;
});

$myColorList.append($colors);

});

$('#btn-sort').on('click', function () {
    doSort(ascending);
    ascending = !ascending;
});
```

```
// ...

var doSort = function ( ascending ) {

    sortList(function ( a, b ) {

        // ...initial sorting...

    }).sort(function ( a, b ) {

        // We take the returned items and sort them once more
        var aClass = a.className;
        var bClass = b.className;

        // Let's group the disabled items together and put them at the end

        /**
         * If the two elements being compared have the same class
         * then there's no need to move anything.
         */
        if ( aClass !== bClass ) {
            return aClass === 'disabled' ? 1 : -1;
        }
        return 0;
    });

    // And finalize with re-insert
    $myColorList.append($colors);
};

// ...
```


4: DOM

Examples

`children()` ◦

```
<div class="parent">
  <h2>A headline</h2>
  <p>Lorem ipsum dolor sit amet...</p>
  <p>Praesent quis dolor turpis...</p>
</div>
```

`.parent.parent`

```
$('.parent').children().css("color", "green");
```

`selector` ◦

```
// Only get "p" children
$('.parent').children("p").css("color", "green");
```

jQuery

jQuery ◦

DOM

```
<div class="container">
  <div class="red one">RED 1 Info</div>
  <div class="red two">RED 2 Info</div>
  <div class="red three">RED 3 Info</div>
</div>
```

`reddiv`

```
$(".red").each(function(key, ele) {
  var text = $(ele).text();
  console.log(text);
});
```

`key` `div` `red` ◦ `ele` `HTML` `$()` `jQuery()` `jQuery` `$(ele)` ◦ `jQuery` `css()` `hide()` ◦ ◦

`.siblings()` `.siblings()` ◦

```
<ul class="menu">
  <li class="selected">Home</li>
  <li>Blog</li>
  <li>About</li>
</ul>
```

selected

```
$(".menu").on("click", "li", function () {  
    $(this).addClass("selected");  
    $(this).siblings().removeClass("selected");  
});
```

selector

```
$(this).siblings("li").removeClass("selected");
```

nearest

DOM。

HTML

```
<div id="abc" class="row">  
  <div id="xyz" class="row">  
  </div>  
  <p id="origin">  
    Hello  
  </p>  
</div>
```

jQuery

```
var target = $('#origin').closest('.row');  
console.log("Closest row:", target.attr('id') );  
  
var target2 = $('#origin').closest('p');  
console.log("Closest p:", target2.attr('id') );
```

OUTPUT

```
"Closest row: abc"  
"Closest p: origin"
```

first。

HTML

```
<div class='.firstExample'>  
  <p>This is first paragraph in a div.</p>  
  <p>This is second paragraph in a div.</p>  
  <p>This is third paragraph in a div.</p>  
  <p>This is fourth paragraph in a div.</p>  
  <p>This is fifth paragraph in a div.</p>  
</div>
```

JQuery

```
var firstParagraph = $("div p").first();
console.log("First paragraph:", firstParagraph.text());
```

First paragraph: This is first paragraph in a div.

`.next()` ◦

```
<ul>
  <li>Mark</li>
  <li class="anna">Anna</li>
  <li>Paul</li>
</ul>
```

“Anna”“Paul”.next() ◦

```
// "Paul" now has green text
$(".anna").next().css("color", "green");
```

selector ◦

```
// Next element is a "li", "Paul" now has green text
$(".anna").next("li").css("color", "green");
```

selector ◦

```
// Next element is not a ".mark", nothing will be done in this case
$(".anna").next(".mark").css("color", "green");
```

`.prev()` ◦

```
<ul>
  <li>Mark</li>
  <li class="anna">Anna</li>
  <li>Paul</li>
</ul>
```

“Anna”“Mark”.prev() ◦

```
// "Mark" now has green text
$(".anna").prev().css("color", "green");
```

selector ◦

```
// Previous element is a "li", "Mark" now has green text
$(".anna").prev("li").css("color", "green");
```

selector ◦

```
// Previous element is not a ".paul", nothing will be done in this case
$(".anna").prev(".paul").css("color", "green");
```

`.filter()`

◦ ◦ ◦

HTML

HTML

```
<ul>
  <li class="zero">Zero</li>
  <li class="one">One</li>
  <li class="two">Two</li>
  <li class="three">Three</li>
</ul>
```

◦

```
$(".li").filter(":even").css("color", "green"); // Color even elements green
$(".li").filter(".one").css("font-weight", "bold"); // Make ".one" bold
```

◦

◦ true

```
var selection = $(".li").filter(function (index, element) {
  // "index" is the position of the element
  // "element" is the same as "this"
  return $(this).hasClass("two");
});
selection.css("color", "green"); // ".two" will be colored green
```

DOM

```
var three = document.getElementsByClassName("three");
$(".li").filter(three).css("color", "green");
```

◦ ◦

```
var elems = $(".one, .three");
$(".li").filter(elems).css("color", "green");
```

find

`.findDOMjQuery`

HTML

```
<div class="parent">
  <div class="children" name="first">
    <ul>
      <li>A1</li>
      <li>A2</li>
      <li>A3</li>
    </ul>
  </div>
  <div class="children" name="second">
    <ul>
      <li>B1</li>
      <li>B2</li>
      <li>B3</li>
    </ul>
  </div>
</div>
```

jQuery

```
$('.parent').find('.children[name="second"] ul li').css('font-weight','bold');
```

- A1
- A2
- A3

- **B1**
- **B2**
- **B3**

DOM <https://riptutorial.com/zh-CN/jquery/topic/1189/dom>

5: jQuery .animate

1. .animate({})

	CSS
	400
	swing
	◦
	◦
	◦
	◦
	◦
DONE	◦
	◦
specialEasing	stylesCSS◦
	◦

Examples

◦ **jQuery** `fadeIn()`, `hide()`, `slideDown()`◦

jQuery `.animate()`◦ **CSS**◦ **CSS** `width:200px;DOM50animatecss150.`◦

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
    $("#btn1").click(function() {
        $("#box").animate({width: "200px"});
    });
</script>
```

```
<button id="btn1">Animate Width</button>
<div id="box" style="background:#98bf21;height:100px;width:100px;margin:6px;"></div>
```

.animate() CSS◦

`backgroundPositionX`, `backgroundPositionY`, `borderWidth`, `borderBottomWidth`, `borderLeftWidth`,

borderRightWidth, borderTopWidth, borderSpacing, margin, marginBottom, marginLeft, marginRight, marginTop, outlineWidth, padding, paddingBottom, paddingLeft, paddingRight, paddingTop, height, width, maxHeight, maxWidth, minHeight, minWidth, fontSize, bottom, left, right, top, letterSpacing, wordSpacing, lineHeight, textIndent,

.animate() ◦

```
milliseconds (Ex: 100, 1000, 5000, etc.),  
"slow",  
"fast"
```

.animate().animate() ◦



◦

1

```
$( "#book" ).animate({  
  width: [ "toggle", "swing" ],  
  height: [ "toggle", "swing" ],  
  opacity: "toggle"  
}, 5000, "linear", function() {  
  $( this ).after( "<div>Animation complete.</div>" );  
});
```

2

```
$("#box").animate({  
  height: "300px",  
  width: "300px"  
}, {  
  duration: 5000,  
  easing: "linear",  
  complete: function(){  
    $(this).after("<p>Animation is complete!</p>");  
  }  
});
```

jQuery .animate <https://riptutorial.com/zh-CN/jquery/topic/5064/jquery--animate-->

6: jQueryPromises

jQuery promises。

Examples

“”。

DeferredDeferred

```
function waitPromise(milliseconds) {  
  
    // Create a new Deferred object using the jQuery static method  
    var def = $.Deferred();  
  
    // Do some asynchronous work - in this case a simple timer  
    setTimeout(function() {  
  
        // Work completed... resolve the deferred, so it's promise will proceed  
        def.resolve();  
    }, milliseconds);  
  
    // Immediately return a "promise to proceed when the wait time ends"  
    return def.promise();  
}
```

```
waitPromise(2000).then(function() {  
    console.log("I have waited long enough");  
});
```

promise。

```
function First() {  
    console.log("Calling Function First");  
    return $.get("/ajax/GetFunction/First");  
}  
  
function Second() {  
    console.log("Calling Function Second");  
    return $.get("/ajax/GetFunction/Second");  
}  
  
function Third() {  
    console.log("Calling Function Third");  
    return $.get("/ajax/GetFunction/Third");  
}  
  
function log(results) {  
    console.log("Result from previous AJAX call: " + results.data);  
}  
  
First().done(log)  
    .then(Second).done(log)  
    .then(Third).done(log);
```

jQuery ajax VS .done, .fail

Ajax

◦

```
$.ajax({
  url: 'URL',
  type: 'POST',
  data: yourData,
  datatype: 'json',
  success: function (data) { successFunction(data); },
  error: function (jqXHR, textStatus, errorThrown) { errorFunction(); }
});
```

.done.fail

.ajax ◦ done(function(data, textStatus, jqXHR){}; jQuery 1.8.success ◦ ◦

.ajax ◦ fail(function(jqXHR, textStatus, errorThrown){}; jQuery 1.8.error ◦ ◦

```
$.ajax({
  url: 'URL',
  type: 'POST',
  data: yourData,
  datatype: 'json'
})
.done(function (data) { successFunction(data); })
.fail(function (jqXHR, textStatus, errorThrown) { errorFunction(); });
```

◦ promise/promise

```
var deferred = new $.Deferred();
var d1= deferred.promise({
  prop: "value"
});
var d2= $("div").promise();
var d3= $("div").hide(1000).promise();

console.log(d1.state()); // "pending"
console.log(d2.state()); // "resolved"
console.log(d3.state()); // "pending"
```

jQuery Promises <https://riptutorial.com/zh-CN/jquery/topic/8308/jquerypromises>

7:

```
.hide().show().toggle();°
```

Examples

```
$(element).hide()           // sets display: none
$(element).show()          // sets display to original value
$(element).toggle()        // toggles between the two
$(element).is(':visible')   // returns true or false
$('element:visible')       // matches all elements that are visible
$('element:hidden')        // matches all elements that are hidden

$('element').fadeIn();      // display the element
$('element').fadeOut();    // hide the element

$('element').fadeIn(1000);  // display the element using timer
$('element').fadeOut(1000); // hide the element using timer

// display the element using timer and a callback function
$('element').fadeIn(1000, function(){
  // code to execute
});

// hide the element using timer and a callback function
$('element').fadeOut(1000, function(){
  // code to execute
});
```

toggle()

```
function toggleBasic() {
  $(".target1").toggle();
}
```

```
function toggleDuration() {
  $(".target2").toggle("slow"); // A millisecond duration value is also acceptable
}
```

.....

```
function toggleCallback() {
  $(".target3").toggle("slow",function(){alert('now do something');});
}
```

.....°

```
function toggleEasingAndCallback() {
  // You may use jQueryUI as the core only supports linear and swing easings
  $(".target4").toggle("slow","linear",function(){alert('now do something');});
}
```

```
}
```

..... °

```
function toggleWithOptions() {
  $(".target5").toggle(
    { // See all possible options in: api.jquery.com/toggle/#toggle-options
      duration:1000, // milliseconds
      easing:"linear",
      done:function(){
        alert('now do something');
      }
    }
  );
}
```

slideToggle()

```
function toggleSlide() {
  $(".target6").slideToggle(); // Animates from top to bottom, instead of top corner
}
```

...fadeToggle()/

```
function toggleFading() {
  $( ".target7" ).fadeToggle("slow")
}
```

...toggleClass()

```
function toggleClass() {
  $(".target8").toggleClass('active');
}
```

toggle()

```
function toggleX() {
  $(".targetX").toggle("slow");
}
```

<https://riptutorial.com/zh-CN/jquery/topic/1298/>

8:

Examples

jQuery jQuery(function(\$){...});DOM

```
<script type="text/javascript">
  jQuery(function($){
    // this will set the div's text to "Hello".
    $("#myDiv").text("Hello");
  });
</script>

<div id="myDiv">Text</div>
```

jQueryJavaScriptDOM

```
<script type="text/javascript">
  // no element with id="myDiv" exists at this point, so $("#myDiv") is an
  // empty selection, and this will have no effect
  $("#myDiv").text("Hello");
</script>

<div id="myDiv">Text</div>
```

.ready()jQuery JSjQuery\$.noConflict(); - jQuery\$.ready()jQuery

```
$.noConflict();

jQuery( document ).ready(function( $ ) {
  // Here we can use '$' as jQuery alias without it conflicting with other
  // libraries that use the same namespace
  $('body').append('<div>Hello</div>')
});

jQuery( document ).ready(function( jq ) {
  // Here we use a custom jQuery alias 'jq'
  jq('body').append('<div>Hello</div>')
});
```

jQuery\$(document).readyHTMLDOMJavaScript

jQuery 2.2.3

```
$(function() {
  // code
});

$.ready(function() {
  // code
});
```

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

jQuery\$

```
jQuery(function() {  
  // code  
});
```

jQuery 3.0

jQuery 3.0

```
jQuery(function($) {  
  // Run when document is ready  
  // $ (first argument) will be internal reference to jQuery  
  // Never rely on $ being a reference to jQuery in the global namespace  
});
```

jQuery 3.0。

jQuery 3.0 ready。 ""。

```
$(function() {  
  console.log("inside handler");  
});  
console.log("outside handler");
```

>
>

\$document.ready\$window.load

\$(window).load() [jQuery 1.8](#) [jQuery 3.0](#) ◦ [jQuery](#)

.load() ◦ ◦

-
- srcsrc WebKit
- DOM
-

load()

\$(document).ready() [DOM - HTML](#) ◦ ◦ \$(window).load()

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

```
console.log($("#my_large_image").height()); // may be 0 because the image isn't available
});

$(window).load(function() {
  console.log($("#my_large_image").height()); // will be correct
});
```

readyDOM

```
$(document).ready()
```

1. jQuery

```
$(document).ready(function() {
  $("button").click(function() {
    // Code for the click function
  });
});
```

2. jQuery

```
jQuery(function($) {
  // set the value of an element.
  $("#myElement").val("Hello");
});
```

3. DOM

```
div div
```

```
$(document).ready(function() {
  $("#toggleDiv").hide();
  $("button").click(function() {
    $("#toggleDiv").show();
  });
});
```

jQueryfn

```
300◦ </body>
```

```
<body>
  <span id="greeting"></span> world!
  <script>
    $("#greeting").text("Hello");
  </script>
</body>
```

```
<head>
  <script>
    jQuery(function($) {
      $("#greeting").text("Hello");
    });
  </script>
</head>
```



```
<body>
  <span id="greeting"></span> world!
</body>
```

</body>span°

<https://riptutorial.com/zh-CN/jquery/topic/500/>

9:

Examples

1

```
$('#parent').prepend($('#child'));
```

2

```
$('#child').prependTo($('#parent'));
```

#child #parent ◦

```
<div id="parent">
  <span>other content</span>
</div>
<div id="child">

</div>
```

```
<div id="parent">
  <div id="child">

  </div>
  <span>other content</span>
</div>
```

[prepend\(\)](#) - ◦

1. `prepend(content [, content])`

```
// with html string
jQuery('#parent').prepend('<span>child</span>');
// or you can use jQuery object
jQuery('#parent').prepend($('#child'));
// or you can use comma separated multiple elements to prepend
jQuery('#parent').prepend($('#child1'), $('#child2'));
```

2. `prepend(function)`

jQuery *version: 1.4* ◦ HTML ◦ this ◦

```
jQuery('#parent').prepend(function(i, oldHTML) {
  // return the value to be prepend
  return '<span>child</span>';
});
```

<https://riptutorial.com/zh-CN/jquery/topic/1909/>

10: /

Stackoverflow“”/。 “”ID。 IDcbGroup1。 cbGroup1

if。

Examples

2

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

<p>
<input id="cbGroup1" type="checkbox">Select all
<input name="cbGroup1" type="checkbox" value="value1_1">Group1 value 1
<input name="cbGroup1" type="checkbox" value="value1_2">Group1 value 2
<input name="cbGroup1" type="checkbox" value="value1_3">Group1 value 3
</p>
<p>
<input id="cbGroup2" type="checkbox">Select all
<input name="cbGroup2" type="checkbox" value="value2_1">Group2 value 1
<input name="cbGroup2" type="checkbox" value="value2_2">Group2 value 2
<input name="cbGroup2" type="checkbox" value="value2_3">Group2 value 3
</p>

<script type="text/javascript" language="javascript">
    $("input").change(function() {
        $('input[name=\''+this.id+\'']').not(this).prop('checked', this.checked);
        $('#'+this.name).prop('checked', $('input[name=\''+this.name+\'']').length ===
        $('input[name=\''+this.name+\'']').filter(':checked').length);
    });
</script>
```

<https://riptutorial.com/zh-CN/jquery/topic/10076/>

11:

jQuery.attr()°

```
$("#input").attr("type");
```

°

Examples

HTML

.attr()°

```
$([selector]).attr([attribute name]);
```

HTML

```
<a href="/home">Home</a>
```

jQuery

```
$('#a').attr('href');
```

data

jQuery.data()° .data**data**°

```
$([selector]).data([attribute name]);
```

HTML

```
<article data-column="3"></article>
```

jQuery

```
$("#article").data("column")
```

jQuery.data("data- *").

HTML

attr(attributeName, attributeValue)°

```
$('#a').attr('title', 'Click me');
```

"Click me"°

°

.removeAttr(attributeName)

◦

```
$('#home').removeAttr('title');
```

IDhometitle◦

attrprop

`attr()` DOM `getAttribute()` `setAttribute()` /HTML◦ `prop()` DOM◦ ◦

```
$('#tosAccept').prop('checked', true); // using attr() won't work properly here
```

`removeProp()`◦ `removeAttr()`◦

<https://riptutorial.com/zh-CN/jquery/topic/4429/>

12:

Examples

jQuery API ◦ API.hide() .fadeIn() .hasClass() ◦

jQuery\$.fn

```
jQuery.fn = jQuery.prototype
```

jQueryjQuery\$◦

jQuery◦ \$(' .active')jQuery◦

thisjQuery◦ this◦

```
$.fn.highlight = function() {  
    this.css({ background: "yellow" });  
};  
  
// Use example:  
$("span").highlight();
```

[jsFiddle](#)

jQuery ◦

```
$(".warn").append("WARNING! ").css({color:"red"}).css().append().css().warn
```

/

```
(function($) {  
    $.fn.highlight = function( custom ) {  
  
        // Default settings  
        var settings = $.extend({  
            color : "", // Default to current text color  
            background : "yellow" // Default to yellow background  
        }, custom);  
  
        return this.css({ // `return this` maintains method chainability  
            color : settings.color,  
            backgroundColor : settings.background  
        });  
  
    };  
})( jQuery );
```

```
// Use Default settings
$("span").highlight(); // you can chain other methods

// Use Custom settings
$("span").highlight({
  background: "#f00",
  color: "white"
});
```

[jsFiddle](#)

◦ ◦

.highlight()

```
//...
// Default settings
var settings = $.extend({
  text : "", // text to highlight
  class : "highlight" // reference to CSS class
}, custom);

return this.each(function() {
  // your word highlighting logic here
});
//...
```

CSS

```
$("#content").highlight({
  text : "hello",
  class : "makeYellowBig"
});
```

[jsFiddle](#)

jQuery.fn.extend

jQuery\$.fnjQuery◦

```
<div>Hello</div>
<div>World!</div>

<script>
jQuery.fn.extend({
  // returns combination of div texts as a result
  getMessage: function() {
    var result;
    // this keyword corresponds result array of jquery selection
    this.each(function() {
      // $(this) corresponds each div element in this loop
      result = result + " " + $(this).val();
    });
    return result;
  }
});
```

```
});  
  
// newly created .getMessage() method  
var message = $("div").getMessage();  
  
// message = Hello World!  
console.log(message);  
</script>
```

<https://riptutorial.com/zh-CN/jquery/topic/1805/>

13:

Examples

```
// array
var arr = [
  'one',
  'two',
  'three',
  'four'
];
$.each(arr, function (index, value) {
  console.log(value);

  // Will stop running after "three"
  return (value !== 'three');
});
// Outputs: one two three
```

jQuery

HTML

```
<ul>
  <li>Mango</li>
  <li>Book</li>
</ul>
```

```
$( "li" ).each(function( index ) {
  console.log( index + ": " + $( this ).text() );
});
```

0

1

<https://riptutorial.com/zh-CN/jquery/topic/10853/>

14:

jQuery [addEventListener](#) ◦ DOM ◦

Examples

1.7 jQueryAPI [.on\(\)](#) ◦ [javascriptjQuery](#) ◦ [.click\(\).on\(\)](#) ◦

HTML

```
<button id="foo">bar</button>
```

jQuery

```
$( "#foo" ).on( "click", function() {  
    console.log( $( this ).text() ); //bar  
});
```

jQuery ◦ [.off\(events \[, selector \] \[, handler \] \)](#) ◦

HTML

```
<button id="hello">hello</button>
```

jQuery

```
$('#hello').on('click', function(){  
    console.log('hello world!');  
    $(this).off();  
});
```

[\\$\(this\)jQuery](#) ◦ ◦

jQuery

```
$('#hello').on('click', function(){  
    console.log('hello world!');  
    $(this).off('click');
```

```
});  
  
$('#hello').on('mouseenter', function(){  
    console.log('you are about to click');  
});
```

mouseenter°

◦ HTML◦

HTML

```
<html>  
  <head>  
  </head>  
  <body>  
    <ul>  
      <li>  
        <a href="some_url/">Link 1</a>  
      </li>  
      <li>  
        <a href="some_url/">Link 2</a>  
      </li>  
      <li>  
        <a href="some_url/">Link 3</a>  
      </li>  
    </ul>  
  </body>  
</html>
```

<a>◦ ◦ ◦ \$('a').click()◦

<a>◦

◦ ◦ “”◦

<a>“click”

-
-
- UL
-
- HTML
-

HTML


```
$('#ul').on('click', 'a', function () {
  console.log(this.href); // jQuery binds the event function to the targeted DOM element
                          // this way `this` refers to the anchor and not to the list
  // Whatever you want to do when link is clicked
});
```

- 'ul'
- 'click'
- 'a'
- .

-
1. <a>
 2. <a>click
 3. .
 4.
 5.
 6. . " " .
 7. 'a' . 'a' .
 8. this . stopPropagation() document .

/document . 'body'

- bodybug . 0 . document .
- document *DOM-ready* document .

.load

PDF .load() .load() .on("load", handler) .

HTML

```

```

jQuery

```
$( "#image" ).load(function() {
  // run script
});
```

ID

.

-

- ◦ this
- this
- find()

HTML

```
<div class="item-wrapper" data-item_id="346">
  <div class="item"><span class="person">Fred</span></div>
  <div class="item-toolbar">
    <button class="delete">Delete</button>
  </div>
</div>
<div class="item-wrapper" data-item_id="393">
  <div class="item"><span class="person">Wilma</span></div>
  <div class="item-toolbar">
    <button class="delete">Delete</button>
  </div>
</div>
```

jQuery

```
$(function() {
  $('.delete').on('click', function() {
    // "this" is element event occurred on
    var $btn = $(this);
    // traverse to wrapper container
    var $itemWrap = $btn.closest('.item-wrapper');
    // look within wrapper to get person for this button instance
    var person = $itemWrap.find('.person').text();
    // send delete to server and remove from page on success of ajax
    $.post('url/string', { id: $itemWrap.data('item_id')}).done(function(response) {
      $itemWrap.remove()
    }).fail(function() {
      alert('Oops, not deleted at server');
    });
  });
});
```

originalEvent

jQuery ◦ Event.originalEvent

```
$(document).on("wheel",function(e) {
  console.log(e.originalEvent.deltaY)
  // Returns a value between -100 and 100 depending on the direction you are scrolling
})
```

jQuery ◦

◦

```
//Adding a normal click handler
```

```
$(document).on("click",function(){
    console.log("Document Clicked 1")
});
//Adding another click handler
$(document).on("click",function(){
    console.log("Document Clicked 2")
});
//Removing all registered handlers.
$(document).off("click")
```

document◦

◦

```
//Add named event listener.
$(document).on("click.mymodule",function(){
    console.log("Document Clicked 1")
});
$(document).on("click.mymodule",function(){
    console.log("Document Clicked 2")
});

//Remove named event listener.
$(document).off("click.mymodule");
```

◦

<https://riptutorial.com/zh-CN/jquery/topic/1321/>

15:

Examples

```
var width = $('#target-element').width();  
var height = $('#target-element').height();
```

```
$('#target-element').width(50);  
$('#target-element').height(100);
```

innerWidthinnerHeight

```
var width = $('#target-element').innerWidth();  
var height = $('#target-element').innerHeight();
```

```
$('#target-element').innerWidth(50);  
$('#target-element').innerHeight(100);
```

outerWidthouterHeight

```
var width = $('#target-element').outerWidth();  
var height = $('#target-element').outerHeight();
```

```
var width = $('#target-element').outerWidth(true);  
var height = $('#target-element').outerHeight(true);
```

```
$('#target-element').outerWidth(50);  
$('#target-element').outerHeight(100);
```

<https://riptutorial.com/zh-CN/jquery/topic/2167/>

16:

jQueryHTMLDOM。idHTML。CSS。

-
- **Id** #id
- .className
- [attributeName]
- [attributeName ='value']
- ^= [attributeName ^= 'value']
- \$= [attributeName \$='value']
- *= [attributeName *= 'value']
- :pseudo-selector
-
- >
- +
- ~
- ,

classidattributeselectors

! " # \$ % & ' () * + , . / : ; < = > ? @ [\] ^ { | } ~

\\。

。

```
<span id="temp.foo" ></span>
```

```
$('#temp\\.foo')
```

Examples

jQuery

-
-
- ID
-
-
-
-

CSSjQuery。

HTML

```
<a href="index.html"></a>
```

```
<!-- 1 -->
```



```
<a id="second-link"></a> <!-- 2 -->
<a class="example"></a> <!-- 3 -->
<a class="example" href="about.html"></a> <!-- 4 -->
<span class="example"></span> <!-- 5 -->
```

jQuery[1,2,3,4](#)。

```
$("#a")
```

jQuery[example3,4,5](#)。

```
$(".example")
```

ID

jQuery[ID2](#)。

```
$("#second-link")
```

jQuery[href14](#)。

```
$("#[href]")
```

jQuery[hrefindex.html 1](#)。

```
$("#[href='index.html']")
```

jQuery[1](#)[<a> ie](#)。 [second-link](#)[leq\(1\) 0](#)。

```
$("#a:eq(1)")
```

```
:not(:eq())
```

[<a>example](#)[1](#)

```
$("#a").not(":eq(0)")
```

```
:not()
```

[<a>example](#)[12](#)。

```
$("#a:not(.example)")
```

jQuery[:first-child](#) [:last-child](#) [:first-of-type](#) [:last-of-type](#)。

jQuery[<a>1](#)。

```
$("#a:first-of-type")
```

jQuery

jQuery; ° °

```
$("#a.class1.class2.class3#someID[attr1][attr2='something'][attr3='something']:first-of-type:first-child")
```

<a>

- class1, class2, and class3
- **ID** someID
- attr1
- attr2something attr3something
- first-childfirst-of-type first-child

```
$("#a, .class1, #someID")
```

- <a>
- **class** class1
- **id**#someID

jQueryCSS. °

-
- >
- +
- ~

° *

```
$('#wrapper *') // Select all elements inside #wrapper element
```

DOM

```
<ul class="parentUl">
  <li> Level 1
    <ul class="childUl">
      <li>Level 1-1 <span> Item - 1 </span></li>
      <li>Level 1-1 <span> Item - 2 </span></li>
    </ul>
  </li>
  <li> Level 2
    <ul class="childUl">
      <li>Level 2-1 <span> Item - 1 </span></li>
      <li>Level 2-1 <span> Item - 1 </span></li>
    </ul>
  </li>
</ul>
```

```
<ul> - parentUl <li>
```

```
1. $('parent child')
```

```
>> $('ul.parentUl li')
```

```
◦
```

```
2. > - $('parent > child')
```

```
>> $('ul.parentUl > li')
```

```
◦
```

```
3. - $('child', 'parent')
```

```
>> $('li', 'ul.parentUl')
```

```
1.◦
```

```
4. find() - $('parent').find('child')
```

```
>> $('ul.parentUl').find('li')
```

```
1.◦
```

```
5. children() - $('parent').find('child')
```

```
>> $('ul.parentUl').children('li')
```

```
2.◦
```

```
“”
```

```
<ul><li><span>
```

```
$('ul, li, span')
```

```
“”
```

```
class parentUl<ul>
```

```
$('ul.parentUl')
```

```
“+”
```

```
<li><li>
```

```
$('#li + li')
```

“”

```
<li><li>
```

```
$('#li ~ li')
```

jQueryjQuery。。

```
$("#*") // All elements
$("#div") // All <div> elements
$(".blue") // All elements with class=blue
$(".blue.red") // All elements with class=blue AND class=red
$(".blue,.red") // All elements with class=blue OR class=red
$("#headline") // The (first) element with id=headline
$("#[href]") // All elements with an href attribute
$("#[href='example.com']") // All elements with href=example.com
```

```
$("#div span") // All <span>s that are descendants of a <div>
$("#div > span") // All <span>s that are a direct child of a <div>
$("#a ~ span") // All <span>s that are siblings following an <a>
$("#a + span") // All <span>s that are immediately after an <a>
```

jQueryDOM。。

```
var nav = $('#navigation');
nav.show();
```

```
$('#navigation').show();
```

/。

```
<div class="parent">
  <div class="child">Child 1</div>
  <div class="child">Child 2</div>
</div>

<script>
  var children = $('.child');
  var firstChildText = children[0].text();
  console.log(firstChildText);

  // output: "Child 1"
</script>
```

DOM。 DOMchild。

```
<div class="parent"></div>

<script>
```

```

var parent    = $('.parent');
var children = $('.child');
console.log(children);

// output: []

parent.append('<div class="child">Child 1</div>');
children = $('.child');
console.log(children[0].text());

// output: "Child 1"
</script>

```

/DOM。

jQuery

```

var $nav = $('#navigation');
$nav.show();

```

DOM

jQueryDOM。DOMjQueryjQuery。

jQuerynodeTypeDOM。

DOMjQueryjQuery API。

each。

```

$(".elements").each(function(){
    //the current element is bound to `this` internally by jQuery when using each
    var currentElement = this;

    //at this point, currentElement (or this) has access to the Native API

    //construct a jQuery object with the currentElement(this)
    var $currentElement = $(this);

    //now $currentElement has access to the jQuery API
});

```

HTML

jQueryHTML。HTMLjQueryjQueryHTML。

jQueryHTMLstring<。quickExpr = /^(?:\s*(<[\w\W]+>) [\^>]*|#([\w-]*))\$/ regex101.com。

HTMLDOMModal。

div

```
function template(href,text){
    return $("<div><a href='" + href + "'>" + text + "</a></div>");
}
```

jQuery

```
<div>
  <a href="google.com">Google</a>
</div>
```

template("google.com","Google") ◦

<https://riptutorial.com/zh-CN/jquery/topic/389/>

17:

- `var jqXHR = $.ajaxurl [settings]`
- `var jqXHR = $.ajax[settings]`
- `jqXHR.done(function(data, textStatus, jqXHR){};`
- `jqXHR.fail(function(jqXHR, textStatus, errorThrown){};`
- `jqXHR.always(function(jqXHR){};`

	URL
	HTTP
StatusCode	HTTP
	◦ "application / x-www-form-urlencoded; charset = UTF-8"
	this◦

AJAX JavaScript XML。AJAX HTTP AJAX。

Examples

\$.ajaxHTTP

`.done .fail.always promiseHTTP◦ statusCode◦`

```
$.ajax({
  type: {POST or GET or PUT etc.},
  url: {server.url},
  data: {someData: true},
  statusCode: {
    404: function(responseObject, textStatus, jqXHR) {
      // No content found (404)
      // This code will be executed if the server returns a 404 response
    },
    503: function(responseObject, textStatus, errorThrown) {
      // Service Unavailable (503)
      // This code will be executed if the server returns a 503 response
    }
  }
})
.done(function(data) {
  alert(data);
})
```

```
.fail(function(jqXHR, textStatus){
    alert('Something went wrong: ' + textStatus);
})
.always(function(jqXHR, textStatus) {
    alert('Ajax request was finished')
});
```

jQuery

```
;3xxerror°
```

Ajax

ajax°

-

```
<form id="ajax_form" action="form_action.php">
  <label for="name">Name :</label>
  <input name="name" id="name" type="text" />
  <label for="name">Email :</label>
  <input name="email" id="email" type="text" />
  <input type="submit" value="Submit" />
</form>
```

jQuery\$(document).ready -

```
$('#ajax_form').submit(function(event) {
    event.preventDefault();
    var $form = $(this);

    $.ajax({
        type: 'POST',
        url: $form.attr('action'),
        data: $form.serialize(),
        success: function(data) {
            // Do something with the response
        },
        error: function(error) {
            // Do something with the error
        }
    });
});
```

- var \$form = \$(this) -
- \$('#ajax_form').submit(function(event) { - ID“ajax_form”°
- event.preventDefault(); - ajax({});return false
- url: \$form.attr('action'), - “action”“url”°
- data: \$form.serialize(), - ° “name=Bob&email=bob@bobsemailaddress.com”

JSON

jQueryjSON JSON


```
$.ajax("/json-consuming-route", {
  data: JSON.stringify({author: {name: "Bullwinkle J. Moose",
                                email: "bullwinkle@example.com"} }),
  method: "POST",
  contentType: "application/json"
});
```

`contentType`; API - jQuery%20+contentTypeapplication/x-www-form-urlencoded。 `contentType`
`processData``false`。

`JSON.stringify` JavaScript JSON。

Ajax

1

```
$.get('url.html', function(data){
  $('#update-box').html(data);
});
```

2

```
$.ajax({
  type: 'GET',
  url: 'url.php',
}).done(function(data){
  $('#update-box').html(data);
}).fail(function(jqXHR, textStatus){
  alert('Error occured: ' + textStatus);
});
```

Ajax Load ajax get

```
$('#update-box').load('url.html');
```

`.load`。

```
$('#update-box').load('url.php', {data: "something"});
$('#update-box').load('url.php', "data=something");
```

`.load`

```
$('#update-box').load('url.php', {data: "something"}, function(resolve){
  //do something
});
```

Ajax Post

1

```
$.post('url.php',
```

```

    {date1Name: data1Value, date2Name: data2Value}, //data to be posted
    function(data){
        $('#update-box').html(data);
    }
    );

```

2

```

$.ajax({
    type: 'Post',
    url: 'url.php',
    data: {date1Name: data1Value, date2Name: data2Value} //data to be posted
}).done(function(data){
    $('#update-box').html(data);
}).fail(function(jqXHR, textStatus){
    alert('Error occured: ' + textStatus);
});

```

Ajax Post JSON

```

var postData = {
    Name: name,
    Address: address,
    Phone: phone
};

$.ajax({
    type: "POST",
    url: "url.php",
    dataType: "json",
    data: JSON.stringify(postData),
    success: function (data) {
        //here variable data is in JSON format
    }
});

```

AjaxJSON

1

```

$.getJSON('url.php', function(data){
    //here variable data is in JSON format
});

```

2

```

$.ajax({
    type: "Get",
    url: "url.php",
    dataType: "json",
    data: JSON.stringify(postData),
    success: function (data) {
        //here variable data is in JSON format
    },
    error: function(jqXHR, textStatus){
        alert('Error occured: ' + textStatus);
    }
});

```

```
    }  
  });
```

Ajax

```
var xhr = $.ajax({  
  type: "POST",  
  url: "some.php",  
  data: "name=John&location=Boston",  
  success: function(msg) {  
    alert( "Data Saved: " + msg );  
  }  
});
```

```
//
```

```
xhr.abort()
```

Ajax

1.

o

```
<input type="file" id="file-input" multiple>
```

```
var files;  
var fdata = new FormData();  
$("#file-input").on("change", function (e) {  
  files = this.files;  
  
  $.each(files, function (i, file) {  
    fdata.append("file" + i, file);  
  });  
  
  fdata.append("FullName", "John Doe");  
  fdata.append("Gender", "Male");  
  fdata.append("Age", "24");  
  
  $.ajax({  
    url: "/Test/Url",  
    type: "post",  
    data: fdata, //add the FormData object to the data parameter  
    processData: false, //tell jquery not to process data  
    contentType: false, //tell jquery not to set content-type  
    success: function (response, status, jqxhr) {  
      //handle success  
    },  
    error: function (jqxhr, status, errorMessage) {  
      //handle error  
    }  
  });
```

```
});
```

◦

2.

[MDNWeb](#) ◦ ◦

◦ file input ◦ multiple ◦

```
<input type="file" id="file-input" multiple>
```

change event ◦

```
var files;
$("#file-input").on("change", function(e) {
    files = this.files;
});
```

files ◦ [FileList](#) object ◦

3. FormData

[AjaxFormData](#) ◦

```
var fdata = new FormData();
```

[FileList](#) for [jQuery.each](#) ◦ [jQuery](#) ◦

```
$.each(files, function(i, file) {
    //...
});
```

[FormData](#) [append](#) [formdata](#) ◦

```
$.each(files, function(i, file) {
    fdata.append("file" + i, file);
});
```

◦ ◦ [formdata](#) ◦

```
fdata.append("FullName", "John Doe");
fdata.append("Gender", "Male");
fdata.append("Age", "24");
//...
```

4.Ajax

```
$.ajax({
  url: "/Test/Url",
  type: "post",
  data: fdata, //add the FormData object to the data parameter
  processData: false, //tell jquery not to process data
  contentType: false, //tell jquery not to set content-type
  success: function (response, status, jqxhr) {
    //handle success
  },
  error: function (jqxhr, status, errorMessage) {
    //handle error
  }
});
```

processDatacontentTypefalse ° °

<https://riptutorial.com/zh-CN/jquery/topic/316/>

18:

1. \$.append
2. \$.appendTo

HTML。

- .append。 after。 HTML。 URLCookie。 XSS。 ◦
- jQuerySVG。 SVGjQuery
 - SVG
- jQuery 3.0addClassremoveClass。

Examples

1

```
$('#parent').append($('#child'));
```

2

```
$('#child').appendTo($('#parent'));
```

#child #parent ◦

```
<div id="parent">
  <span>other content</span>
</div>
<div id="child">
</div>
```

```
<div id="parent">
  <span>other content</span>
  <div id="child">
</div>
</div>
```

◦ .append() .appendTo() ◦ .clone() -> [<http://api.jquery.com/clone/>][1]

.append

HTML

```
<table id='my-table' width='960' height='500'></table>
```

JS

```
var data = [  
  { type: "Name", content: "John Doe" },  
  { type: "Birthdate", content: "01/01/1970" },  
  { type: "Salary", content: "$40,000,000" },  
  // ...300 more rows...  
  { type: "Favorite Flavour", content: "Sour" }  
];
```

◦ ◦

```
var i; // <- the current item number  
var count = data.length; // <- the total  
var row; // <- for holding a reference to our row object  
  
// Loop over the array  
for ( i = 0; i < count; ++i ) {  
  row = data[ i ];  
  
  // Put the whole row into your table  
  $('#my-table').append(  
    $('<tr></tr>').append(  
      $('<td></td>').html(row.type),  
      $('<td></td>').html(row.content)  
    )  
  );  
}
```

.....

300

- <table> ◦

◦ ◦

1.

```
/**
```

```

* Repeated DOM traversal (following the tree of elements down until you reach
* what you're looking for - like our <table>) should also be avoided wherever possible.
*/

// Keep the table cached in a variable then use it until you think it's been removed
var $myTable = $('#my-table');

// To hold our new <tr> jQuery objects
var rowElements = [];

var count = data.length;
var i;
var row;

// Loop over the array
for ( i = 0; i < count; ++i ) {
    rowElements.push(
        $('<tr></tr>').append(
            $('<td></td>').html(row.type),
            $('<td></td>').html(row.content)
        )
    );
}

// Finally, insert ALL rows at once
$myTable.append(rowElements);

```

jQuery.

2. Array ^{*}

```

var $myTable = $('#my-table');

// Looping with the .map() method
// - This will give us a brand new array based on the result of our callback function
var rowElements = data.map(function ( row ) {

    // Create a row
    var $row = $('<tr></tr>');

    // Create the columns
    var $type = $('<td></td>').html(row.type);
    var $content = $('<td></td>').html(row.content);

    // Add the columns to the row
    $row.append($type, $content);

    // Add to the newly-generated array
    return $row;
});

// Finally, put ALL of the rows into your table
$myTable.append(rowElements);

```

◦

3. HTMLjQuery

```
// ...
var rowElements = data.map(function ( row ) {
    var rowHTML = '<tr><td>';
    rowHTML += row.type;
    rowHTML += '</td><td>';
    rowHTML += row.content;
    rowHTML += '</td></tr>';
    return rowHTML;
});

// Using .join('') here combines all the separate strings into one
$myTable.append(rowElements.join(''));
```

- jQuery ◦ jQuery ◦

4.

```
var $myTable = $(document.getElementById('my-table'));

/**
 * Create a document fragment to hold our columns
 * - after appending this to each row, it empties itself
 *   so we can re-use it in the next iteration.
 */
var colFragment = document.createDocumentFragment();

/**
 * Loop over the array using .reduce() this time.
 * We get a nice, tidy output without any side-effects.
 * - In this example, the result will be a
 *   document fragment holding all the <tr> elements.
 */
var rowFragment = data.reduce(function ( fragment, row ) {

    // Create a row
    var rowEl = document.createElement('tr');

    // Create the columns and the inner text nodes
    var typeEl = document.createElement('td');
    var typeText = document.createTextNode(row.type);
    typeEl.appendChild(typeText);

    var contentEl = document.createElement('td');
    var contentText = document.createTextNode(row.content);
    contentEl.appendChild(contentText);

    // Add the columns to the column fragment
    // - this would be useful if columns were iterated over separately
    //   but in this example it's just for show and tell.
    colFragment.appendChild(typeEl);
    colFragment.appendChild(contentEl);

    rowEl.appendChild(colFragment);
```

```
// Add rowEl to fragment - this acts as a temporary buffer to
// accumulate multiple DOM nodes before bulk insertion
fragment.appendChild(rowEl);

return fragment;
}, document.createDocumentFragment());

// Now dump the whole fragment into your table
$myTable.append(rowFragment);
```

◦ jQuery◦

- [jQuery](#)
- [Array.prototype.join](#)
- [Array.prototype.map](#)
- [Array.prototype.reduce](#)
- [document.createDocumentFragment](#)
- [document.createTextNode](#)
- [Google Web Fundamentals -](#)

jQuery

HTML

```
<p>This is a nice </p>
<p>I like </p>

<ul>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ul>

<button id="btn-1">Append text</button>
<button id="btn-2">Append list item</button>
```

```
$("#btn-1").click(function() {
  $("p").append(" <b>Book</b>.");
});
$("#btn-2").click(function() {
  $("ul").append("<li>Appended list item</li>");
});
});
```

<https://riptutorial.com/zh-CN/jquery/topic/1910/>

S. No		Contributors
1	jQuery	A.J, acdcjunior, amflare, Anil, bwegs, Community, DGS, empiric, Fueled By Coffee, hairboat, Hirshy, Iceman, Igor Raush, J F, jkdev, John C, Kevin B, Kevin Katzke, Kevin Montrose, Luca Putzu, Matas Vaitkevicius, mico, Mottie, Neal, ni8mr, Prateek, RamenChef, Rion Williams, Roko C. Buljan, secelite, Shaunak D, Stephen Leppik, Suganya, Sverri M. Olsen, the12, Travis J, user2314737, Velocibadgery, Yosvel Quintero
2	CSS	abaracedo, Ashkan Mobayen Khiabani, Brandt Solovij, J F, j08691, Jonathan Michalik, Kevin B, Petroff, Roko C. Buljan, ScientiaEtVeritas, Shlomi Haver, Sorangwala Abbasali, Stephen Leppik, Sverri M. Olsen
3	DOM	Angelos Chalaris, Assimilater, Brock Davis, DawnPaladin, DefyGravity, Deryck, Marimba, Mark Schultheiss, martincarin87, Neal, Paul Roub, Shaunak D, still_learning
4	DOM	A.J, charlietfl, Community, dlsso, mark.hch, rmondesilva, SGS Venkatesh, sucil, Sverri M. Olsen, The_Outsider, Zaz
5	jQuery .animate	RamenChef, Rust in Peace, Simplans, VJS
6	jQueryPromises	Alex, Ashiquzzaman, Gone Coding
7		Alex Char, Paul Roub, Rupali Pemare, The_Outsider, Theodore K., user2314737, Zaz
8		Adjit, Alon Eitan, amflare, charlietfl, Emanuel Vintilă, Igor Raush, J F, jkdev, Joram van den Boezem, Liam, Mark Schultheiss, Melanie, Nhan, Nico Westerdale, Scimonster, secelite, TheDeadMedic, the-noob, URoy
9		Ashkan Mobayen Khiabani, empiric, J F, Pranav C Balan
10	/	user1851673
11		A.J, acdcjunior, ban17, ochi, Scimonster
12		hasan, Roko C. Buljan, Travis J
13		bipon, Renier
14		Adjit, charlietfl, DelightedD0D, doydoy44, empiric, Gone Coding, Horst Jahns, Jatniel Prinsloo, Kevin B, Louis, Luca Putzu, Marimba, NotJustin, SGS Venkatesh, Stephen Leppik, Sunny R

	Gupta, Washington Guedes, WMios, Zakaria Acharki
15	Ashkan Mobayen Khiabani
16	alepeino, Alon Eitan, Brock Davis, Castro Roy, David, DelightedD0D, devlin carnate, dlso, hasan, Iceman, James Donnelly, JLF, John Slegers, kapantzak, Kevin B, Keyslinger, Matas Vaitkevicius, Melanie, MikeC, Nux, Petr R., rbashish, Scimonster, Shaunak D, Shekhar Pankaj, Sorangwala Abbasali, ssb, Sverri M. Olsen, Travis J, whales, WOUNDEDStevenJones, Zaz
17	Alon Eitan, amflare, Andrew Brooke, Ashkan Mobayen Khiabani, Athafoud, atilacamura, Ben H, Cass, Community, csbarnes, Dr. J. Testington, Edathadan Chief aka Arun, empiric, hasan, joe_young, John C, kapantzak, Kiren Siva, Lacroque, Marimba, Nirav Joshi, Ozan, shaN, Shaunak D, Teo Dragovic, Yosvel Quintero
18	Ashkan Mobayen Khiabani, bapon, Community, Darshak, Deryck, empiric, Flyer53, J F, JF it, Paul Roub, Pranav C Balan, Proto, Shaunak D