

jQuery选择器

- jQuery选择器
 - ■ ■ 基本选择器
 - ■ ID选择器
 - Element选择器
 - class选择器
 - 通配符选择器
 - 多项选择器
 - 层级选择器（祖先后代选择器）
 - ■ 选中所有后代
 - 直接后代选择器
 - 下一选择器
 - 兄弟选择器
 - 属性选择器
 - ■ [attribute]
 - [attribute=value] 等于
 - [attribute!=value] 不等于
 - [attribute^=value] 以value开头
 - [attribute\$=value] 以value结尾
 - [attribute*=value] 包含value
 - [selector1][selector2][selectorN]多个属性选择
 - 过滤器
 - child（在子元素中，指定具体的位置）
 - first-child
 - last-child
 - nth-child
 - nth-last-child
 - only-child
 - type (在子元素中找到即可，不需要是具体位置)
 - first-of-type
 - last-of-type
 - nth-of-type
 - nth-last-of-type
 - only-of-type
 - 参数
 - 表单相关选择器

- 元素相关
 - 状态相关
- 查找预过滤
 - find
 - children
 - parent
 - next()/prev()/eq()/siblings()
 - filter()

基本选择器

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <!--<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> -->
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>
  </head>
  <body>
    <div id="first" class="div first">First</div>
    <div id="second" class="div second">Second</div>
    <div id="third" class="div third">Third</div>
  </body>
</html>
```

- ID选择器：根据给定的ID匹配一个元素
- Element选择器：根据给定的元素标签名匹配所有元素
- class选择器：根据给定的css类名匹配元素
- 通配符选择器：匹配所有元素

ID选择器

```
var first = $('#first');
console.log(first);

var second = $('#second');
var third = $('#third');
/*
```

打印结果:

```
jQuery.fn.init [div#first.divfirst]
0: div#first.divfirst
length: 1
__proto__: Object(0)
*/
```

Element选择器

```
var div = $('div');
console.log(div);
/*
jQuery.fn.init(3) [div#first.divfirst, div#second.divsecond, div#third.div.third, prevObject: jQuery.fn.init(1)]
0: div#first.divfirst
1: div#second.divsecond
2: div#third.div.third
length: 3
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

class选择器

```
var divClass = $('.div');
console.log(divClass)
/*
jQuery.fn.init(3) [div#first.div.first, div#second.div.second, div#third.div.third, prevObject: jQuery.fn.init(1)]
0: div#first.div.first
1: div#second.div.second
2: div#third.div.third
length: 3
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

```
var firstClass = $('.first');
var secondClass = $('.second');
var thirdClass = $('.third');
console.log(firstClass);
/*
jQuery.fn.init [div#first.div.first, prevObject: jQuery.fn.init(1)]
0: div#first.div.first
```

```
length: 1
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

通配符选择器

```
var all = $('*');
console.log(all);
/*
jQuery.fn.init(10) [html, head, meta, title, script, script, body, div#first.div.first, div#second.div.second, div#third.div.third, prevObject: jQuery.fn.init(1)]
0: html
1: head
2: meta
3: title
4: script
5: script
6: body
7: div#first.div.first
8: div#second.div.second
9: div#third.div.third
length: 10
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

多项选择器

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> -->
  >
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <div id="first" class="div first">First</div>
    <div id="second" class="div second">Second</div>
    <div id="third" class="div third">Third</div>
```

```

    <section id="first" class="div first">First</section>
    <section id="second" class="div second">Second</section>
    <section id="third" class="div third">Third</section>
  </body>
</html>

```

```

$("selector1, selector2, selectorN")

```

将每一个选择器匹配到的元素合并后一起返回。可以指定任意多个选择器，并将匹配到的元素合并到一个结果内。

```

var firstAndThird = $('#first, .third');
console.log(firstAndThird);
/*
    jQuery.fn.init(2) [div#first.div.first, div#third.div.third, prevObject: jQuery.fn.init(1)]
    0: div#first.div.first
    1: div#third.div.third
    length: 2
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

var objs = $('#first, div, .div, .third');
console.log(objs);

/*
    jQuery.fn.init(6) [div#first.div.first, div#second.div.second, div#third.div.third, section#first.div.first, section#second.div.second, section#third.div.third, prevObject: jQuery.fn.init(1)]
    0: div#first.div.first
    1: div#second.div.second
    2: div#third.div.third
    3: section#first.div.first
    4: section#second.div.second
    5: section#third.div.third
    length: 6
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

console.log(objs[0]);
/*
    <div id="first" class="div first">First</div>
*/

```

层级选择器（祖先后代选择器）

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> --
  >
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charse
t="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <aside>
      <details>
        <summary>网页设计</summary>
        <ul>
          <li>Photoshop</li>
          <li>Illustrator</li>
          <li>Sketch</li>
        </ul>
      </details>
      <details>
        <summary>网站前端</summary>
        <details>
          <summary>结构层</summary>
          <ul>
            <li>HTML</li>
            <li>Jade</li>
          </ul>
        </details>
        <details>
          <summary>表示层</summary>
          <ul>
            <li>CSS</li>
            <li>LESS</li>
            <li>SASS</li>
          </ul>
        </details>
      </details>
    </aside>
  </body>
</html>
```

选中所有后代

```
$('#ancestor descendant');
```

在指定的祖先元素下匹配所有的后代元素

```
// 祖先aside 后代summary
var summaries = $('#aside summary');
console.log(summaries);
/*
  jQuery.fn.init(3) [summary, summary, summary, prevObject: jQuery.fn.init(1)]
  0: summary
  1: summary
  2: summary
  length: 3
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/

var details = $('#aside details');
console.log(details);
/*
  jQuery.fn.init(4) [details, details, details, details, prevObject: jQuery.fn.i
  0: details
  1: details
  2: details
  3: details
  length: 4
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/
```

直接后代选择器

```
$('#parent>child');
```

在给定的父元素下匹配所有的子元素

```
var details = $('#aside > details');
console.log(details);
/*
  jQuery.fn.init(2) [details, details, prevObject: jQuery.fn.init(1)]
  0: details
  1: details
  length: 2
  prevObject: jQuery.fn.init [document]
*/
```

```
__proto__: Object(0)
*/
```

下一选择器

```
$('#prev + next')
```

屁屁额所有进阶在prev元素后的next元素

```
var uls = $('#summary + ul');
console.log(uls);
/*
    jQuery.fn.init(3) [ul, ul, ul, prevObject: jQuery.fn.init(1)]
    0: ul
    1: ul
    2: ul
    length: 3
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

var uls = $('#summary + details');
console.log(uls);
/*
    jQuery.fn.init [details, prevObject: jQuery.fn.init(1)]
    0: details
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/
```

兄弟选择器

```
$('#pre ~ sublings')
```

匹配prev元素之后的所有siblings元素

```
var uls = $('#summary ~ ul');
console.log(uls);
/*
    jQuery.fn.init(3) [ul, ul, ul, prevObject: jQuery.fn.init(1)]
    0: ul
    1: ul
    2: ul
    length: 3
    prevObject: jQuery.fn.init [document]
```



```

    __proto__: Object(0)
  */
  var details = $('summary ~ details');
  console.log(details);
  /*
    jQuery.fn.init(2) [details, details, prevObject: jQuery.fn.init(1)]
    0: details
    1: details
    length: 2
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
  */

```

属性选择器

- [attribute]
- [attribute=value]
- [attribute!=value]
- [attribute=value]
- [attribute\$=value]
- [attribute*=value]
- [selector1][selector2][selectorN]

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> -->
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <details class="tool">
      <summary>开发工具</summary>
      <p class="tool_sublime">Sublime Text</p>
      <p class="tool_webstorm">WebStorm Text</p>
      <p class="tool_vs">Visual Studio</p>
      <p class="tool_vscode">Visual Studio Code</p>
    </details>
    <details class="lang">
      <summary>开发语言</summary>

```

```

    <p class="lang_html">html</p>
    <p class="lang_css">css</p>
    <p class="lang_js">javascript</p>
    <p class="lang_jquery">jquery</p>
  </details>
</body>
</html>

```

[attribute]

```

var all = $('[class]');
console.log(all);
/*
   jQuery.fn.init(10) [details.tool, p.tool_sublime, p.tool_webstorm, p.tool_vs,
   0: details.tool
   1: p.tool_sublime
   2: p.tool_webstorm
   3: p.tool_vs
   4: p.tool_vscode
   5: details.lang
   6: p.lang_html
   7: p.lang_css
   8: p.lang_js
   9: p.lang_jquery
   length: 10
   prevObject: jQuery.fn.init [document]
   __proto__: Object(0)
*/

var all = $('[lang]');
console.log(all);
/*
   jQuery.fn.init [html, prevObject: jQuery.fn.init(1)]
   0: html
   length: 1
   prevObject: jQuery.fn.init [document]
   __proto__: Object(0)
*/

```

[attribute=value] 等于

```

var all = $('[class=tool_sublime]');
console.log(all);
/*
   jQuery.fn.init [p.tool_sublime, prevObject: jQuery.fn.init(1)]
   0: p.tool_sublime

```

```
length: 1
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

[attribute!=value] 不等于

```
var all = $('[class!=tool]');
console.log(all);
/*
    jQuery.fn.init(18) [html, head, meta, title, script, script, body, summary,
    p.tool_sublime, p.tool_webstorm, p.tool_vs, p.tool_vscode, details.lang, summary, p.1
    1)]
    0: html
    1: head
    2: meta
    3: title
    4: script
    5: script
    6: body
    7: summary
    8: p.tool_sublime
    9: p.tool_webstorm
    10: p.tool_vs
    11: p.tool_vscode
    12: details.lang
    13: summary
    14: p.lang_html
    15: p.lang_css
    16: p.lang_js
    17: p.lang_jquery
    length: 18
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/
```

[attribute^=value] 以value开头

```
var all = $('[class^=tool]');
console.log(all);
/*
    jQuery.fn.init(5) [details.tool, p.tool_sublime, p.tool_webstorm, p.tool_vs, p
    0: details.tool
    1: p.tool_sublime
    2: p.tool_webstorm
    3: p.tool_vs
    4: p.tool_vscode
    length: 5
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/
```

```

    4: p.tool_vscode
    length: 5
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
  */

```

[attribute\$=value] 以value结尾

```

var all = $('[class$=sublime]');
console.log(all);
/*
    jQuery.fn.init [p.tool_sublime, prevObject: jQuery.fn.init(1)]
    0: p.tool_sublime
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
  */

```

[attribute*=value] 包含value

```

var all = $('[class*=_v]');
console.log(all);
/*
    jQuery.fn.init(2) [p.tool_vs, p.tool_vscode, prevObject: jQuery.fn.init(1)]
    0: p.tool_vs
    1: p.tool_vscode
    length: 2
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
  */

```

[selector1][selector2][selectorN]多个属性选择

\$('[class][class*=lang][class\$=jquery]')

```

var all = $('[class][class*=lang][class$=jquery]');
console.log(all);
/*
    jQuery.fn.init [p.lang_jquery, prevObject: jQuery.fn.init(1)]
    0: p.lang_jquery
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
  */

```

过滤器

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> --
  >
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charse
t="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <details class="tool" open="open">
      <p class="only-child">Sublime Text</p>
    </details>
    <details class="tool" open="open">
      <summary>编辑器</summary>
      <p class="tool_sublime">Sublime Text</p>
    </details>
    <details class="tool">
      <summary>开发工具</summary>
      <p class="tool_webstorm">WebStorm Text</p>
      <p class="tool_vs">Visual Studio</p>
      <p class="tool_vscode">Visual Studio Code</p>
    </details>
    <details class="lang">
      <summary>开发语言</summary>
      <p class="lang_html">html</p>
      <p class="lang_css">css</p>
      <p class="lang_js">javascript</p>
      <p class="lang_jquery">jquery</p>
    </details>
  </body>
</html>
```

child（在子元素中，指定具体的位置）

- :first-child
- :last-child
- :nth-child(n|even|odd|formula)
- :nth-last-child(n|even|odd|formula)
- :only-child

```

var first = $('details');
console.log(first);
/*
  jQuery.fn.init(3) [details.tool, details.ide, details.lang, prevObject: jQuery
  0: details.tool
  1: details.ide
  2: details.lang
  length: 3
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/

```

first-child

```

var first = $('details:first-child');
console.log(first);
/*
  jQuery.fn.init [details.tool, prevObject: jQuery.fn.init(1)]
  0: details.tool
  length: 1
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/

```

```

// details > p:first-child details下的第一个标签是p标签
var first = $('details > p:first-child');
console.log(first);
/*
  jQuery.fn.init [prevObject: jQuery.fn.init(1)]
  length: 0
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/

```

last-child

```

// details > p:last-child details下的最后一个标签是p标签
var last = $('details > p:last-child');
console.log(last);
/*
  jQuery.fn.init(3) [p.tool_sublime, p.tool_vscode, p.lang_jquery, prevObject: j
  0: p.tool_sublime
  1: p.tool_vscode
  2: p.lang_jquery

```

```
length: 3
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/
```

nth-child

```
// details下第二个标签是p会返回p标签
var nthFirst = $('details > p:nth-child(2)');
console.log(nthFirst);
/*
  jQuery.fn.init(3) [p.tool_sublime, p.tool_webstorm, p.lang_html, prevObject: j
  0: p.tool_sublime
  1: p.tool_webstorm
  2: p.lang_html
  length: 3
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/
```

```
// details下第一个标签是p会返回p标签
var nthFirst = $('details > p:nth-child(1)');
console.log(nthFirst);
/*
  jQuery.fn.init [prevObject: jQuery.fn.init(1)]
  length: 0
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/
```

nth-last-child

```
var nthLast = $('details > p:nth-last-child(2)');
console.log(nthLast);
/*
  jQuery.fn.init(2) [p.tool_vs, p.lang_js, prevObject: jQuery.fn.init(1)]
  0: p.tool_vs
  1: p.lang_js
  length: 2
  prevObject: jQuery.fn.init [document]
  __proto__: Object(0)
*/
```

only-child

```

var onlyChild = $('details > p:only-child');
console.log(onlyChild);
/*
    jQuery.fn.init [p.only-child, prevObject: jQuery.fn.init(1)]
    0: p.only-child
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

```

type (在子元素中找到即可，不需要是具体位置)

- :first-of-type
- :last-of-type
- :nth-of-type(n|even|odd|formula)
- :nth-last-of-type(n|even|odd|formula)
- :only-of-type

first-of-type

```

// details > p:first-of-type 找到details下的第一次出现的p标签
var first = $('details > p:first-of-type');
console.log(first);
/*
    jQuery.fn.init(4) [p.only-child, p.tool_sublime, p.tool_webstorm, p.lang_html, prevObject: jQuery.fn.init(1)]
    0: p.only-child
    1: p.tool_sublime
    2: p.tool_webstorm
    3: p.lang_html
    length: 4
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

```

last-of-type

```

var last = $('details > p:last-of-type');
console.log(last);
/*
    jQuery.fn.init(4) [p.only-child, p.tool_sublime, p.tool_vscode, p.lang_jquery, prevObject: jQuery.fn.init(1)]
    0: p.only-child
    1: p.tool_sublime

```



```

2: p.tool_vscode
3: p.lang_jquery
length: 4
prevObject: jQuery.fn.init [document]
__proto__: Object(0)
*/

```

nth-of-type

```

var nthF = $('details > p:nth-of-type(2)');
console.log(nthF);
/*
    jQuery.fn.init(2) [p.tool_vs, p.lang_css, prevObject: jQuery.fn.init(1)]
    0: p.tool_vs
    1: p.lang_css
    length: 2
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

```

nth-last-of-type

```

var nthL = $('details > p:nth-last-of-type(2)');
console.log(nthL);
/*
    jQuery.fn.init(2) [p.tool_vs, p.lang_js, prevObject: jQuery.fn.init(1)]
    0: p.tool_vs
    1: p.lang_js
    length: 2
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

```

only-of-type

```

var only = $('details > p:only-of-type');
console.log(only);
/*
    jQuery.fn.init(2) [p.only-child, p.tool_sublime, prevObject: jQuery.fn.init
(1)]
    0: p.only-child
    1: p.tool_sublime
    length: 2
    prevObject: jQuery.fn.init [document]
*/

```

```
__proto__: Object(0)
*/
```

参数

- n 匹配子元素序号。必须为证书，从1开始。
- even 匹配所有偶数元素。
- odd 匹配所有奇数元素。
- formula 使用特殊公式如 (an+b) 进行选择。(2n)、(2n+1)

表单相关选择器

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> --
  >
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charse
t="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <div>
      <input type="hidden" value="id">
    </div>
    <div>
      <label>用户名 : </label>
      <input type="text" placeholder="用户名">
    </div>
    <div>
      <label>密码 : </label>
      <input type="password" placeholder="密码">
    </div>
    <div>
      <label>姓名: </label>
      <input type="password" placeholder="密码" disabled="disabled">
    </div>
    <div>
      <input type="button" value="按钮">
      <input type="submit" value="提交">
      <button>按钮</button>
    </div>
  </body>
```

```
</html>
```

- :input 可以选择<input>,<textarea>,<select>和<button>
- :text 匹配所有的单行文本框, 和input[type='text']一样
- 其它input的type :password/:radio/:checkbox/:image/:reset/button/:file

元素相关

```
var inputs = $(':input');
console.log(inputs);
/*
    jQuery.fn.init(6) [input, input, input, input, input, button, prevObject: jQuery
    0: input
    1: input
    2: input
    3: input
    4: input
    5: button
    length: 6
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

var texts = $(':text');
console.log(texts);
/*
    jQuery.fn.init [input, prevObject: jQuery.fn.init(1)]
    0: input
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

var texts = $(':password');
console.log(texts);
/*
    jQuery.fn.init [input, prevObject: jQuery.fn.init(1)]
    0: input
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/
```

状态相关

```
var disabled = $(':disabled');
console.log(disabled);
/*
    jQuery.fn.init [input, prevObject: jQuery.fn.init(1)]
    0: input
    length: 1
    prevObject: jQuery.fn.init [document]
    __proto__: Object(0)
*/

var enabled = $(':enabled');
console.log(enabled);

var checked = $(':checked');
console.log(checked);

var checked = $(':selected');
console.log(checked);
```

查找预过滤

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script> --
  >
    <script src="../jquery-3.5.1/jquery-3.5.1.js" type="text/javascript" charse
t="utf-8"></script>
    <script type="text/javascript" src="index.js"></script>

  </head>
  <body>
    <aside>
      <details>
        <summary>网页设计</summary>
        <ul>
          <li>Photoshop</li>
          <li>Illustrator</li>
          <li>Sketch</li>
        </ul>
      </details>
    </details>
```

```

<summary>网站前端</summary>
<details>
  summary>结构层</summary>
  <ul>
    <li>HTML</li>
    <li class="javascript">javascript</li>
    <li class="html">html</li>
  </ul>
</details>
<details>
  <summary>表示层</summary>
  <ul>
    <li>CSS</li>
    <li>LESS</li>
    <li>SASS</li>
  </ul>
</details>
</details>
</aside>
</body>
</html>

```

- find
- children
- parent
- next
- prev
- eq
- siblings
- filter

find

```

var js = $('aside').find('.javascript');
console.log(js);
/*
  jQuery.fn.init [li.javascript, prevObject: jQuery.fn.init(1)]
  0: li.javascript
  length: 1
  prevObject: jQuery.fn.init [aside, prevObject: jQuery.fn.init(1)]
  __proto__: Object(0)
*/

```

children

```

var list = $('aside').children('details');
console.log(list);
/*
    jQuery.fn.init(2) [details, details, prevObject: jQuery.fn.init(1)]
    0: details
    1: details
    length: 2
    prevObject: jQuery.fn.init [aside, prevObject: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

```

parent

```

var js = $('aside').find('.javascript');
var js_parent = js.parent();
console.log(js_parent);
/*
    jQuery.fn.init [ul, prevObject: jQuery.fn.init(1)]
    0: ul
    length: 1
    prevObject: jQuery.fn.init [li.javascript, prevObject: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

```

next()/prev()/eq()/siblings()

```

var js = $('aside').find('.javascript');
var html = js.next();
console.log(html);
/*
    jQuery.fn.init [li.html, prevObject: jQuery.fn.init(1)]
    0: li.html
    length: 1
    prevObject: jQuery.fn.init [li.javascript, prevObject: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

console.log(html.prev());
/*
    jQuery.fn.init [li.javascript, prevObject: jQuery.fn.init(1)]
    0: li.javascript
    length: 1
    prevObject: jQuery.fn.init [li.html, prevObject: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

```

```

var lis = $('li');
var html = lis.eq(5);
console.log(html);
/*
    jQuery.fn.init [li.html, prevObject: jQuery.fn.init(9)]
    0: li.html
    length: 1
    prevObject: jQuery.fn.init(9) [li, li, li, li, li.javascript, li.html, li,
li, li, prevObject: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

var details = $('details');
var detailsSiblings = details.siblings();
console.log(detailsSiblings);

/*
    jQuery.fn.init(5) [details, details, summary, details, details, prevObject:
jQuery.fn.init(4)]
    0: details
    1: details
    2: summary
    3: details
    4: details
    length: 5
    prevObject: jQuery.fn.init(4) [details, details, details, details, prevObje
ct: jQuery.fn.init(1)]
    __proto__: Object(0)
*/

```

filter()

```

var details = $('details');
var detailsSiblings = details.siblings();
var details = detailsSiblings.filter(".details");
console.log(details);
/*
    jQuery.fn.init [details.details, prevObject: jQuery.fn.init(5)]
    0: details.details
    length: 1
    prevObject: jQuery.fn.init(5) [details, details.details, summary, details, det
    __proto__: Object(0)
*/

```