

## jquery节点操作

查找节点（子节点、父节点、筛选某些条件下的节点）

方法名	解释
元素.children()   元素.children("元素")	当前元素的直接子节点，不考虑其它间接后代节点
元素.siblings()   元素.siblings("元素")	当前元素的兄弟节点,不包含当前自身节点
元素.parent()	当前元素的直接父节点
元素.parents()	当前元素的所有父节点
父元素.find("元素")	先找到父元素，再查找父元素下面的子元素，find 查找子孙节点
元素.filter("元素")	先找到外层的元素，再过滤找到filter中的元素
元素.not("元素")	同 filter 相反，排除not中的元素

## jQuery 遍历 - 后代

### children(selector) 方法

`children()` 方法返回被选元素的所有直接子元素。

该方法只会向下一级对 DOM 树进行遍历。

```
1      <div class="nav">
2          <p>nav的亲儿子</p>
3          <div>
4              <p>nav的孙子</p>
5          </div>
6          <p class="p2">nav的亲儿子</p>
7      </div>
8
9      <script>
10         var a = $('.nav').children('.p2').css('color','red');
11         console.log(a);
12     </script>
```

### find(selector) 方法

`find()` 方法返回被选元素的**后代元素**，一路向下直到最后一个后代。

下面的例子返回属于 `<div>` 后代的所有 `<span>` 元素

```

1      <div class="nav">
2          <p>nav的亲儿子</p>
3          <div>
4              <p>nav的孙子</p>
5              <span>nav的孙子，span标签</span>
6          </div>
7          <p class="p2">nav的亲儿子</p>
8      </div>
9
10     <script>
11         var a = $('nav').find('span').css('color','red');
12         console.log(a);
13     </script>

```

## jQuery 遍历 - 祖先

### parent() 方法 返回被选元素的直接父元素

`parent()` 方法返回被选元素的直接父元素。

该方法只会向上一级对 DOM 树进行遍历。

```

1      <button id="btn">集成按钮，可以显示也可以隐藏</button>
2      <ul>
3          <li>123</li>
4          <li>456</li>
5          <li class="a">789</li>
6      </ul>
7      <script>
8          $('#btn').click(function(){
9              $('.a').parent().hide();
10          })
11      </script>

```

### parents() 方法

`parents()` 方法返回被选元素的所有祖先元素，它一路向上直到文档的根元素 (`<html>`)。

```

1      <button class="btn">按钮</button>
2      <div class="yeye">
3          <div class="father">
4              <p class="son">文本内容工</p>
5          </div>
6      </div>
7      <script>
8          $('.btn').click(function(){
9              var ps = $('.son').parents();
10             console.log(ps);
11          })
12      </script>

```

## jQuery 遍历 - 同胞(siblings)

---

### siblings(selector) 被选元素的所有同胞元素

```
1      <ul>
2          <li>123</li>
3          <li>456</li>
4          <li class="a">789</li>
5          <li>012</li>
6      </ul>
7      <script>
8          $('a').siblings().css('color','red');
9      </script>
```

### next() 返回被选元素的下一个同胞元素

```
1      <ul>
2          <li>123</li>
3          <li>456</li>
4          <li class="a">789</li>
5          <li>012</li>
6      </ul>
7      <script>
8          $('a').next().css('color','red');
9      </script>
```

## filter( ) 过滤

---

先找到外层的元素，再过滤找到 filter 中的元素

```
1      <div id="box">
2          <span>1</span>
3          <span>2</span>
4          <span class="sp">sp3</span>
5          <span>4</span>
6          <span class="sp">5</span>
7          <span class="sp">6</span>
8      </div>
9      <script>
10         //找span中class名称叫sp的
11         //filter:过滤
12         var a = $("#box>span").filter(".sp").css('color','red')
13         console.log(a);
14     </script>
```

## not( ) 排除not中的元素

---

```
1      <div id="box">
2          <span>1</span>
3          <span>2</span>
4          <span class="sp">sp3</span>
5          <span>4</span>
6          <span class="sp">5</span>
7          <span class="sp">6</span>
8      </div>
9      <script>
10         //找到span中class名称不带sp的
11         $("#box>span").not('.sp').css('color', 'red');
12         console.log($("#box>span").not('.sp'))
13     </script>
```