

1.说明

- 选择器本身只是**一个有特定语法规则的字符串**, 没有实质用处
- 它的基本语法规则使用的就是CSS的选择器语法, 并对基进行了扩展
- 只有调用**\$()**, 并将选择器作为参数传入才能起作用
- **\$(selector)**作用 :根据选择器规则在整个文档中查找所有匹配的标签的数组, 并封装成jQuery对象返回

2.分类

2.1基本选择器：最基本最常用的选择器

2.1.1 **#id** :id选择器

2.1.2 **element**:元素选择器

2.1.3 **.class**:属性选择器

2.1.4 ***** : 任意标签

2.1.5 **selector1,selector2,selectorN**:取多个选择器的并集(组合选择器)

2.1.6 **selector1selector2selectorN**:取多个选择器的交集(相交选择器)

- 交集选择器与并集选择器

交集选择器就是在两个标签相交的部分, 也就是**交集**进行修改格式。

并集选择器是对两个标签**同时**进行修改样式。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
```

```
6 <title>05_基本选择器</title>
7 </head>
8
9 <body>
10 <div id="div1" class="box">div1(class="box")</div>
11 <div id="div2" class="box">div2(class="box")</div>
12 <div id="div3">div3</div>
13 <span class="box">span(class="box")</span>
14
15 <br>
16 <ul>
17 <li>AAAAA</li>
18 <li title="hello">BBBBB(title="hello")</li>
19 <li class="box">CCCCC(class="box")</li>
20 <li title="hello">DDDDD(title="hello")</li>
21 </ul>
22
23 <!--
24 1. 是什么?
25 - 有特定格式的字符串
26 2. 作用
27 - 用来查找特定页面元素
28 3. 基本选择器
29 - #id : id选择器
30 - element : 元素选择器
31 - .class : 属性选择器
32 - * : 任意标签
33 - selector1,selector2,selectorN : 取多个选择器的并集(组合选择器)
34 - selector1selector2selectorN : 取多个选择器的交集(相交选择器)
35 -->
36 <script src="../../课件/jQuery库/jquery-1.10.1.js" type="text/javascript">
37 <script type="text/javascript">
38 /*
39 需求:
40 1. 选择id为div1的元素
41 2. 选择所有的div元素
42 3. 选择所有class属性为box的元素
43 4. 选择所有的div和span元素
44 5. 选择所有class属性为box的div元素
45 */
46
47 // 1. 选择id为div1的元素
48 $('#div1').css('background','pink')
```

```

49 // 2. 选择所有的div元素
50 $('div').css('background','yellow')
51 // 3. 选择所有class属性为box的元素
52 $('.box').css('background','red')
53 // 4. 选择所有的div和span元素!!!!!!
54 $('div,span').css('background','blue')
55 // 5. 选择所有class属性为box的div元素!!!!!!
56 $('div.box').css('background','green')
57 // 6. *
58 $('*').css('background','white')
59 </script>
60 </body>
61
62 </html>

```

2.2层次选择器：查找子元素, 后代元素, 兄弟元素的选择器

2.2.1 **ancestor descendant** :在给定的祖先元素下的后代元素中匹配元素

2.2.2 **parent > child** :在给定的父元素下的子元素中匹配元素

2.2.3 **prev + next** :匹配所有紧接在prev元素后的next元素 (用的少)

2.2.4 **prev ~ siblings** :匹配prev元素之后的所有siblings元素 (用的少)

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>06_层次选择器</title>
7 </head>
8
9 <body>
10 <ul>
11   <li>AAAAA</li>
12   <li class="box">CCCCC</li>

```

```
13 <li title="hello"><span>BBBBB</span></li>
14 <li title="hello"><span class="box">DDDD</span></li>
15 <span>EEEE</span>
16 </ul>
17
18 <!--
19 层次选择器：查找子元素，后代元素，兄弟元素的选择器
20 1. ancestor descendant
21 在给定的祖先元素下匹配所有的后代元素
22 2. parent>child
23 在给定的父元素下匹配所有的子元素
24 3. prev+next
25 匹配所有紧接在 prev 元素后的 next 元素
26 4. prev~siblings
27 匹配 prev 元素之后的所有 siblings 元素
28 -->
29
30 <script src="../../../../课件/jQuery库/jquery-1.10.1.js" type="text/javascript">
31 <script type="text/javascript">
32  /*
33  需求：
34  1. 选中ul下所有的span
35  2. 选中ul下所有的子元素span
36  3. 选中class为box的下一个li
37  4. 选中ul下的class为box的元素后面的所有兄弟元素
38  */
39  // 1. 选中ul下所有的span
40  $('ul span').css('background','green')
41  // 2. 选中ul下所有的子元素span
42  $('ul>span').css('background','red')
43  // 3. 选中class为box的下一个li
44  //补充：<li>是块元素
45  $('.box+li').css('background','pink')
46  // 4. 选中ul下的class为box的元素后面的所有兄弟元素
47  $('ul .box~*').css('background','blue')
48
49 </script>
50 </body>
51 </html>
```

2.3 过滤选择器:在原有选择器匹配的元素中进一步进行过滤的选择器

2.3.1 基本

2.3.2 内容

2.3.3 可见性

2.3.4 属性

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>07_过滤选择器</title>
7 </head>
8
9 <body>
10 <div id="div1" class="box">class为box的div1</div>
11 <div id="div2" class="box">class为box的div2</div>
12 <div id="div3">div3</div>
13 <span class="box">class为box的span</span>
14 <br/>
15 <ul>
16   <li>AAAAA</li>
17   <li title="hello">BBBBB</li>
18   <li class="box">CCCCC</li>
19   <li title="hello">DDDDD</li>
20   <li title="two">BBBBB</li>
21   <li style="display:none">我本来是隐藏的</li>
22 </ul>
23 <!--
24 在原有选择器匹配的元素中进一步进行过滤的选择器
25   * 基本
26   * 内容
27   * 可见性
28   * 属性
29 -->
```

```

30 <script src="../../课件/jQuery库/jquery-1.10.1.js" type="text/javascript">
31 <script type="text/javascript">
32
33 /*
34 需求：
35 1. 选择第一个div
36 2. 选择最后一个class为box的元素
37 3. 选择所有class属性不为box的div
38 4. 选择第二个和第三个li元素
39 5. 选择内容为BBBBB的li
40 6. 选择隐藏的li
41 7. 选择有title属性的li元素
42 8. 选择所有属性title为hello的li元素
43 */
44
45 // 1. 选择第一个div :first
46 $('div:first').css('background','red')
47 // 2. 选择最后一个class为box的元素
48 $('.box:last').css('background','green')
49 // 3. 选择所有class属性不为box的div
50 $('div:not(.box)').css('background','yellow') //没有class属性也是（选择所有
51
52 //ZG --start
53 // $('div:not(.box), xxxx')
54 // 3个选择器
55 // 第一个选择器：div 标签选择器（选择所有指定的标签）
56 // 第二个选择器：:not(arg0) 伪类选择器（代表非....，也就是说不是....(arg0)）
57 // 第三个选择器：.box 类选择器（选择所有的class属性中包含该class的元
58 /* $(".box") <div></div> // 不被选择
59 <div class="box"></div> // 被选择
60 <div class="box xxx yyy"></div> // 被选择
61 <div class="inbox xxx yyy"></div> // 不被选择
62 <p class="box xxx yyy"></p> // 被选择 */
63
64 // 条件
65 // 所有的div元素 AND 不满足.box选择器（换言之，也就是说所有class属性不包含box的
66 //ZG --end
67
68 // 4. 选择第二个和第三个li元素 :gt > :lt <
69 // $('li:gt(0):lt(2)').css('background','pink') //多个选择器不是同时执行，是
70 $('li:lt(3):gt(0)').css('background','pink')
71 // 5. 选择内容为BBBBB的li
72 $('li:contains("BBBBB)').css('background','black')

```

```

73 // 6. 选择隐藏的li
74 console.log($('li:hidden').length,$('li:hidden')[0])
75 // 7. 选择有title属性的li元素
76 $('li[title]').css('background','black')
77 // 8. 选择所有属性title为hello的li元素
78 // $('li[title=hello]').css('background','green')
79 $('li[title="hello"]').css('background','green') //加不加双引号都行
80 </script>
81 </body>
82
83 </html>

```

2.4 表单选择器

2.4.1 表单

2.4.2 表单对象属性

JavaScript



D3 (material)



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>08_表单选择器</title>
6 </head>
7
8 <body>
9 <form>
10  用户名: <input type="text"/><br>
11  密 码: <input type="password"/><br>
12  爱 好:
13  <input type="checkbox" checked="checked"/>篮球
14  <input type="checkbox"/>足球
15  <input type="checkbox" checked="checked"/>羽毛球 <br>
16  性 别:
17  <input type="radio" name="sex" value='male'/>男
18  <input type="radio" name="sex" value='female'/>女<br>
19  邮 箱: <input type="text" name="email" disabled="disabled"/><br>
20  所在地:
21  <select>

```

```

22     <option value="1">北京</option>
23     <option value="2" selected="selected">天津</option>
24     <option value="3">河北</option>
25 </select><br>
26 <input type="submit" value="提交"/>
27 </form>
28 <!--
29 表单选择器
30 1). 表单
31 2). 表单对象属性
32 -->
33 <script src="../../../../课件/jQuery库/jquery-1.10.1.js" type="text/javascript">
34 <script type="text/javascript">
35     /*
36     需求:
37     1. 选择不可用的文本输入框
38     2. 显示选择爱好 的个数
39     3. 显示选择的城市名称
40     */
41
42     // 1. 选择不可用的文本输入框    邮箱    :text:disabled
43     $(':text:disabled').css('background','red')
44     // 2. 显示选择爱好 的个数    :checkbox:checked
45     console.log($('checkbox:checked').length)
46     // 3. 显示选择的城市名称
47     //问题1: selected为啥写成:selected?? 是哪个selected??    已解决
48     //:selected: 伪类选择器, 选择下拉列表中被选中的
49     //selected属性: 一般只有下拉列表被选中项的option元素才有!!!!
50     $(':submit').click(function(){
51         var city = $('select>option:selected').html()
52         alert(city)
53     })
54     //4. 显示选择的城市ID
55     //定位submit的四种写法
56     //$('#:submit'):伪类
57     //$('[type='submit']') 属性选择器
58     //$('input[type='submit']')
59     //$('input:submit')
60
61     //问题2: select为啥就能定位到是天津对应的??为啥不写 selected??    已解决
62     //$('select'):标签选择器
63     //val():对于select标签来说会返回选中项的option标签的value值
64     $(':submit').click(function(){

```



```
65 //select和option标签具有特殊性，被选中的option标签无需定位到它，
66 //可通过其父标签<select>直接取到它的value值！！！！
67 city=$('#select').val() //选择的option的value的属性值 是下面写法的省略，
68 //city=$('#select>option:selected').val() 等于上面的写法
69 alert(city)
70 })
71
72
73 </script>
74 </body>
75 </html>
```