

02jq的DOM

今日内容：

- 复习常用的选择器
- 学习DOM操作
 - 页面元素操作
 - 创建节点
 - 插入节点
 - 复制节点
 - 替换节点
 - 包裹节点
 - 遍历节点
 - 元素属性操作
 - 获取元素属性
 - 设置元素属性
 - 删除元素属性
 - 获取和设置元素
 - 获取和设置元素内容
 - 获取和设置元素值
 - 元素样式操作
 - 设置样式
 - 增加样式
 - 切换样式

- 删除样式

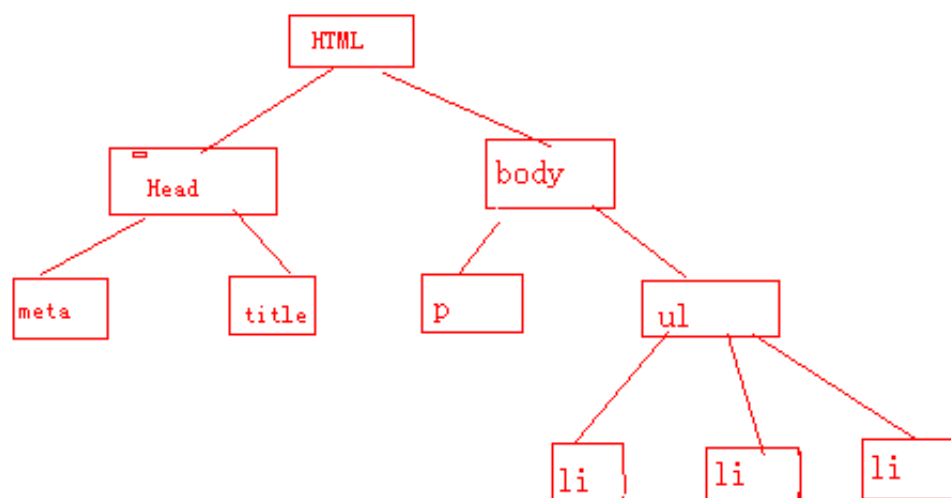
一、页面元素操作

DOM是Document Object Model的缩写，意思是文档对象模型。DOM是一种与浏览器、平台、语言无关的接口，使用该接口可以轻松访问页面中所有的标准组件。DOM操作可以分为三个方面即DOM Core(核心)、HTML-DOM和CSS-DOM。

每一个网页都可以用DOM表示出来，每个DOM都可以看作是一棵DOM树。下面的html页面结构可以构建出一棵DOM树，代码：

```
1 <!DOCTYPE html>
2 <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
4     <title>DOM Demo</title>
5 </head>
6 <body>
7 <p title="选择你最喜欢的水果.">你最喜欢的水果是？</p>
8 <ul>
9 <li title="苹果">苹果</li>
10 <li title="橘子">橘子</li>
11 <li title="菠萝">菠萝</li>
12 </ul>
13 </body>
14 </html>
```

构建出的DOM树如下：



JQuery中的DOM操作主要对包括：建【新建】、增【添加】、删【删除】、改【修改】、查【查找】【像数据库操作】。下面的DOM操作将围绕上面的DOM树进行学习JQueryDOM操作。

一、查--查找DOM节点

查找节点非常容易，使用选择器就能轻松完成各种查找工作。例：查找元素节点p返回p内的文本内容`$("p").text()`;例：查找元素节点p的属性返回属性名称对应的属性值`$("p").attr("title")`,返回p的属性title的值。

二、建--新建DOM节点

1、创建元素节点

创建元素节点并且把节点作为元素的子节点添加到DOM节点树上。先创建元素点，创建元素节点使用Jquery的工厂函数`$()`来完成，格式如下：`$(html)`,该方法会根据传入的html字符串返回一个DOM对象，并将DOM对象包装成一个JQuery对象后返回。创建一个元素节点JQuery代码如下：

```
$li1=$("<li></li>")
```

代码返回`$li1`就是一个由DOM对象包装成的JQuery对象。把新建节点添加到DOM树中JQuery代码如下：

```
$("ul").append($li1);
```

添加后页面中只能看到元素默认的"·",由于没有为节点添加文本所以只显示默认符号,下面创建文本节点。

PS:append()方法是添加DOM节点方法详见增--添加DOM节点。

2、创建文本节点

使用JQuery的工厂函数\$()同样能够创建文本节点,创建文本节点的JQuery代码如下:

```
$li2=$("<li>苹果</li>");
```

代码返回\$li2就是一个由DOM对象包装成JQuery对象,把新建的文本节点添加到DOM树中JQuery代码如下:

```
$("ul").append($li2);
```

添加后页面中能看到"·苹果",右键查看页面源码发现新加的文本节点没有title属性。下面方法创建带属性的节点。

3、创建属性节点

创建属性节点同元素节点、文本节点一样使用JQuery的工厂函数完成。创建属性节点的JQuery代码如下:

```
$li3=$("<li title='榴莲'>榴莲</li>");
```

代码返回\$li3也是一个由DOM对象包装成JQuery对象,把新建的属性节点添加到DOM树中JQuery代码如下:

```
$("ul").append($li3);
```

添加后页面中能看到"·榴莲",右键查看页面源码发现新加的属性节点有title='榴莲'属性。

三、增--添加DOM节点

动态新建元素不添加到文档中没有实际意义,将新建的节点插入到文档中有多个方法,如下:append()、appendTo()、prepend()、prependTo()、after()、insertAfter()、before()、insertBefore()。

1、append()方法

append()方法向匹配的元素内部追加内容，方法如下：
\$("target").append(element);例：

```
$("#ul").append("<li title='香蕉'>香蕉</li>");
```

该方法查找ul元素，然后向ul中添加新建的li元素。

2、appendTo()方法

appendTo()方法将所有匹配的元素追加到指定的元素中，该方法是append()方法的颠倒[操作主题的颠倒并非操作结果]操作。方法如下：
\$(element).appendTo(target);例：

```
$("#<li title='荔枝'>荔枝</li>").appendTo("ul");
```

该方法新建元素li，然后把li添加到查找到的ul元素中。

3、prepend()方法

prepend()方法将每匹配的元素内部前置要添加的元素，方法如下：
\$(target).prepend(element);例：

```
$("#ul").prepend("<li title='芒果'>芒果</li>")
```

该方法将查找元素ul然后将新建的li元素作为ul子节点，且作为ul的第一个子节点插入到ul中。

4、prependTo()方法

prependTo()方法将元素添加到每一个匹配的元素内部前置，方法如下：
\$(element).prependTo();例：

```
$("#<li title='西瓜'>西瓜</li>").prependTo("ul");
```

该方法将新建的元素li插入到查找到的ul元素中作为ul的第一个子节元素。

5、after()方法

after()方法向匹配的元素后面添加元素，新添加的元素做为目标元素后的紧邻的兄弟元素。方法如下：
\$(target).after(element);例：

```
$("p").after("<span>新加段新加段新加段新加段新加段</span>");
```

方法将查找节点p，然后把新建的元素添加到span节点后面做为p的兄弟节点。

6、insertAfter()方法

insertAfter()方法将新建的元素插入到查找到的目标元素后，做为目标元素的兄弟节点。方法如下：\$(element).insertAfter(target);例:

```
$("#<p>insertAfter操作</p>").insertAfter("span");
```

方法将新建的p元素添加到查找到目标元素span后面，做为目标元素后面的第一个兄弟节点。

7、before()方法

before()方法在每一个匹配的元素之前插入，做为匹配元素的前一个兄弟节点。方法如下:\$(target).before(element);例：

```
$("#p").before("<span>下面是个段落</span>");
```

before方法查找每个元素p，将新建的span元素插入到元素p之前做为p的前一个兄弟节点。

8、insertBefore()方法

insertBefore()方法将新建元素添加到目标元素前，做为目标元素的前一个兄弟节点，方法如下:\$(element).insertBefore(target);例:

```
$("#<a href='#>锚</a>").insertBefore("ul");
```

insertBefore()新建a元素，将新建的a元素添加到元素ul前，做为ul的前一个兄弟节点。

增加元素的方法前四个是添加到元素内部，后四个是添加到元素外部的操作，有这些方法可以完成任何形式的元素添加。

四、删--删除DOM节点操作

如果想要删除文档中的某个元素jQuery提供了两种删除节点的

方法：remove()和empty();

1、remove()方法

remove()方法删除所有匹配的元素，传入的参数用于筛选元素，该方法能删除元素中的所有子节点，当匹配的节点及后代被删除后，该方法返回值是指向被删除节点的引用，因此可以使用该引用，再使用这些被删除的元素。方法如下：\$(element).remove();例:

```
$span=$("span").remove();
```

```
$span.insertAfter("ul");
```

该示例中先删除所有的span元素，把删除后的元素使用\$span接收，把删除后的元素添加到ul后面做为ul的兄弟节点。该操作相当于将所有的span元素以及后代元素移到ul后面。

2、empty()方法。

empty()方法严格来讲并不是删除元素，该方法只是清空节点，它能清空元素中的所有子节点。方法如下:\$(element).empty();例：

```
$("ul li:eq(0)").empty();
```

该示例使用empty方法清空ul中第一个li的文本值。只能下li标签默认符号”。“。

五、改--修改DOM节点操作

修改文档中的元素节点可以使用多种方法:复制节点、替换节点、包裹节点。

1、复制节点\$(element).clone()

复制节点方法能够复制节点元素，并且能够根据参数决定是否复制节点元素的行为。方法如下:\$(element).clone(true);例：

```
$("ul li:eq(0)").clone(true);
```

该方法复制ul的第一个li元素，true参数决定复制元素时也复制元素行为，当不复制行为时没有参数。

2、替换节点\$(element).replaceWith()、\$(element).replaceAll()

替换节点方法能够替换某个节点，有两种形式实现：replaceWith()和replaceAll().使用replaceWith方法使用后面的元素替换前面的元素，replaceAll方法使用前面的元素替换后面的元素，方法如下：

\$(oldelement).replaceWith(newelement);\$(newelement).replaceAll(oldelement);例：

\$("#p").replaceWith("我要留下");该方法使用strong元素替换p元素。

\$("#<h3>替换strong</h3>").replaceAll("strong");该例使用h3元素替换所有的strong元素。

3、包裹节点\$(element).wrap()、\$(element).wrapAll()、\$(element).wrapInner()

包裹节点方法使用其他标记包裹目标元素从而改变元素的显示形式等，并且该操作不会破坏原始文档的词义。包裹节点有三种实现形式：wrap();wrapAll();wrapInner();

wrap()方法如下：\$(dstelement).wrap(tag);例：

\$("#p").wrap("");该示例方法使用b标签包裹所有的p元素每个元素都使用b标签包裹。

wrapAll()方法如下：\$(dstelement).wrapAll(tag);例：

\$("#p").wrapAll("");该示例方法使用b标签包裹所有的p元素，所有的p元素标签用一个b标签包裹。

wrapInner()方法如下：\$(dstelement).wrapInner(tag);例：

\$("#strong").wrapInner("");该示例使用b标签包裹每一个strong元素的子元素。

Dom元素的其他操作:属性操作、样式操作、设置和获取HTML,文本和值、遍历节点操作、Css-Dom操作。


```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <meta charset="UTF-8">
6         <title></title>
7         <script src="jquery-1.10.1.min.js" type="text/javascript"
charset="utf-8"></script>
8         <script type="text/javascript">
9             $(function(){
10
11                 $('li:last').html('香蕉');//节点查找
12
13                 $li=$('<li title="liulian">榴莲</li>');//节点的创建
14
15                 $('ul').append($li);//节点的添加1: 添加到后面
16
17                 $li2=$('<li>葡萄</li>');
18
19                 $li2.appendTo('ul');//节点的添加2: 添加到后面第二种写法
20
21                 $li3=$('<li>殷桃</li>');
22
23                 $('ul').prepend($li3);//节点的添加3: 添加到前面
24
25                 $span=$('<span>fds</span>');
26
27                 $('p').after($span);//节点的添加4: 加到后面当兄弟节点
28
29                 $('p').before('<span>fddfdfs</span>');//节点的添加5: 加
到前面当兄弟节点
30
31                 $('li:eq(2)').remove(); //删除节点
32
33                 $('li:eq(2)').empty();//删除内容
34
35                 $('li:eq(0)').clone(true).appendTo('ul');//复制节点
36
37                 $('p').replaceWith('<span>5566</span>');//替换节点1
38
39                 $('li').wrap('<b></b>');//用b标记包含每一个li
```

```

40
41         $('ul').wrapAll('<div></div>');//用div标记包含所有的ul
42
43         $('li').wrapInner('<a href=""></a>');//用a标记包含li里
面的子元素
44     });
45     </script>
46 </head>
47
48 <body>
49     <p title="选择你最喜欢的水果.">你最喜欢的水果是? </p>
50
51     <ul>
52         <li title="苹果">苹果</li>
53         <li title="橘子">橘子</li>
54         <li title="菠萝">菠萝</li>
55     </ul>
56     <ul>
57         <li title="橘子">橘子</li>
58     </ul>
59     <p title="选择你最喜欢的水果.">水果? </p>
60 </body>
61
62 </html>

```

二、元素属性操作、获取和设置元素、元素样式操作

1、属性操作attr()和removeAttr()

attr()方法能够获取元素属性，也能够设置元素属性。方法如下，当attr(para1)方法有个参数时候用于获得当前元素的para1的属性值，当attr(para1,attrValue)有两个参数时候设置当前元素的属性名为para1的属性值为attrValue;例：

`$("p").attr("title");`该示例用于获得p元素的title属性值。

`$("p").attr("title","你最喜欢的水果");`该示例设置p元素的title属性值为"你最喜欢的水果";

如果一次设置多个属性值可以使用“名/值”对形式，例：

`$("#p").attr({"title":"你最喜欢的水果","name":"水果"})`。该示例一次设置两个属性值。

`removeAttr()`方法用于删除特定的属性，方法是在参数中指定属性名。例：

`$("#p").removeAttr("name");`该方法就是移除p元素的name属性。

2、样式操作addClass()、removeClass()、toggleClass()和hasClass()

添加样式addClass()方法，使用该方法对目标元素添加相应的样式，方法如下：`$(element).addClass();`例：

`$("#p").addClass("ul");`该示例设置元素p的样式为ul。

移除样式removeClass()方法，使用该方法移除目标元素的指定样式，方法如下：`$(element).removeClass();`例：

`$("#p").removeClass("ul");`该救命去除掉p元素的ul类样式。

切换样式toggleClass()方法，使用该方法切换目标元素的样式，方法如下：`$(element).toggleClass();`例：

`$("#p").toggleClass("ul");`该方法来回切换【添加删除实现切换】元素p的样式ul。

判断元素是否使用了样式`$(element).hasClass()`,方法如下：`$(element).hasClass(class);`例：

`alert($("#p").hasClass("ul"));`打印出p元素是否有ul样式。

PS:addClass()和attr()方法设置样式的不同，attr方法把元素的属性名对应的属性值设为方法中的参数值，addClass()则把属性值添加到属性名对应的属性值中。例：已有元素<p class='class1'>元素样式</p>,使用attr()和addClass()分别添加新样式。

`$("#p").attr("class","another");`结果是<p class='another'>元素样式</>

`$("#p").addClass("class","another");`结果是<p class='class1 another'>元素样式</p>

3、设置和获取HTML【html()】 ,文本【text()】 和值【val()】

html()方法获得或设置某个元素的html元素。方法如下:\$(selector).html();例:

`$("#p").html();`该示例获得元素p的html内容。

`$("#p").html("添加html内容");`该示例设置p的html内容为"**添加html内容**";

PS : 该方法可以用于XHTML文档 , 不能用于XML文档。

text()方法获得或设置某个元素的文本值。方法如下:\$(selecotr).text();例 :

`$("#p").text();`该示例获得元素p的text文本内容。

`$("#p").text("重新设置的文本内容");`该示例设置元素p的text文本为"重新设置的文本内容";

PS:该方法对html和XML文档都适用。

val()方法获得或设置某个元素的值 , 如果元素值是单选则以数组形式返回 , 方法如下 : \$(selector).val();例:文本元素

`<input type="text" id="userName" value="请输入用户名" />`

`$("#userName").val();`获得input元素的值。

`$("#userName").val('响马');`设置input元素的值为'响马'。

val()方法的不仅能操作input , 最重要的一个用途用于select【下拉列表框】、checkbox【多选框】、radio【单选框】。例 : 在下拉框下的多选赋值应用`<select id="fruits" multiple="multiple"><option>苹果</option><option>香蕉</option><option>西瓜</option></select>`

`$("#fruits").val(['苹果','香蕉']);`该示例使select中苹果和香蕉两项被选中。

4、遍历节点操作children()、next()、prev()、siblings()和closest()

children()方法用于取得匹配元素的子元素集合 , **只匹配子元素不考虑任何后代元素**。方法如下 : \$(selector).children();例 :

`$("#body").children().length;`该示例获得body元素的子元素

个数；

next()方法用于匹配元素的下一个兄弟节点，方法如下：
\$(selector).next();例：

\$("p").next().html();该示例获得p元素的下一个兄弟节点的html内容。

prev()方法用于匹配元素的上一个兄弟节点，方法如下：
\$(selector).prev();例：

\$("ul").prev().text();该示例获得ul元素的上一个兄弟节点的文本内容。

siblings方法()用于匹配目标元素的所有兄弟元素，方法如下：
\$(selector).siblings();例：

\$("p").slibings();示例获得p元素的所有兄弟节点元素。

closest()方法()用来取得最近的匹配元素，首先检查当前元素是否匹配如果匹配则直接返回，否则继续向上查找父元素中符合条件的元素返回，如果没有匹配的元素则返回空jQuery对象。

5、CSS-Dom操作css()、offset()、position()、scrollTop()和scrollLeft()

css()方法用于获取、设置元素的一个或多个属性。方法如下：
\$(selector).css();例：

\$("p").css("color","red");该示例用于设置元素的颜色属性为红色；

\$("p").css("color")该示例用于获得元素的color样式值；

\$("p").css({"font-size":"30px","backgroundColor","#888888"});该示例用于设置元素的多个样式。

offset()方法用于获取元素相对当前窗体的偏移量，其返回对象包括两个属性：top和left。方法如下：\$(selector).offset()

var offset= \$("p").offset(); var left=offset.left;var top=offset.top;该示例用于获得元素p的偏移量。

PS:offset()只对可见元素有效。

position()方法用于获取元素于最近的个position样式属性设置为relative或者absolute的祖先节点的相对偏移量。方法如下：
\$(selector).position();例：

var postion = \$("p").positon();var left=positon.left;var top=positon.top;该示例用于获得元素p的位置。

scrollTop()和scrollLeft()方法用于获取元素的滚动条距顶端的距离和距左侧的距离。方法如下:\$(selector).scrollTop();\$(selector).scrollLeft();
例：

var scrollTop=\$("p").scrollTop();var scrollLeft=\$("p").scrollLeft();该示例用于获得元素的滚动条的位置。

也可以添加参数将元素滚动到指定的位置。例：

\$("#textarea").scrollTop(300);\$("#textarea").scrollLeft(300);

```
1  <script>
2
3  $(function(){
4
5      $(document).click(function(){
6
7          alert( $(window).scrollTop() ); //滚动距离
8
9      });
10
11 });
12
13
14
15 </script>
```

拓展：

•width() innerWidth() outerWidth()

```
1 <script>
2
3
4 $(function(){
5
6     alert( $('div').width() ); //width
7
8     alert( $('div').innerWidth() ); //width + padding
9
10    alert( $('div').outerWidth() ); //width + padding + border
11
12    alert( $('div').outerWidth(true) ); //width + padding + border +
margin
13
14 });
15
16
17
18 </script>
```

•on() off()

```
1 <!DOCTYPE html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>无标题文档</title>
5
6 <script type="text/javascript" src="jquery-1.10.1.min.js"></script>
7 <script>
8
9 $(function(){
10
11     /*$('div').click(function(){
12         alert(123);
13     });*/
```

```
14
15     /*$('div').on('click',function(){
16         alert(123);
17     });*/
18
19     /*$('div').on('click mouseover',function(){
20         alert(123);
21     });*/
22
23     /*$('div').on({
24         'click' : function(){
25             alert(123);
26         },
27         'mouseover' : function(){
28             alert(456);
29         }
30     });*/
31
32     $('div').on('click mouseover',function(){
33         alert(123);
34         $('div').off('mouseover');
35     });
36
37 });
38
39
40
41 </script>
42 </head>
43
44 <body>
45 <div>div</div>
46 <span>span</span>
47 </body>
48 </html>
```

弹窗：


```
1 <!DOCTYPE html >
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>无标题文档</title>
5 <style>
6 *{ margin:0; padding:0;}
7 #login{ width:300px; height:300px; border:1px #000 solid;
   position:absolute;}
8 #close{ position:absolute; right:0; top:0;}
9 </style>
10 <script type="text/javascript" src="jquery-1.10.1.min.js"></script>
11 <script>
12
13 $(function(){
14
15     //var oDiv = $('<div>div</div>');
16     //$('body').append( oDiv );
17
18     $('#input1').click(function(){
19
20         var oLogin = $('<div id="login"><p>用户名:<input type="text" />
   </p><p>密码:<input type="text" /></p><div id="close">X</div></div>');
21
22         $('body').append( oLogin );
23
24         oLogin.css('left' , ($(window).width() -
   oLogin.outerWidth())/2 );
25         oLogin.css('top' , ($(window).height() -
   oLogin.outerHeight())/2 );
26
27
28         $('#close').click(function(){
29
30             oLogin.remove();
31
32         });
33
34
35         $(window).on('resize scroll',function(){
36
37             oLogin.css('left' , ($(window).width() -
```

```

oLogin.outerWidth())/2 );
38     oLogin.css('top' , ($(window).height() -
oLogin.outerHeight())/2 + $(window).scrollTop() );
39
40     });
41
42     });
43
44 });
45
46 </script>
47 </head>
48
49 <body style="height:2000px;">
50 <input type="button" value="点击" id="input1" />
51 <!--<div id="login">
52     <p>用户名:<input type="text" /></p>
53     <p>密码:<input type="text" /></p>
54     <div id="close">X</div>
55 </div>-->
56 </body>
57 </html>

```

窗口发现改变就触发事件：

```

1 <html>
2 <head>
3 <script type="text/javascript" src="/jquery/jquery.js"></script>
4 <script type="text/javascript">
5 x=0;
6 $(document).ready(function(){
7 $(window).resize(function() {
8     $("span").text(x+=1);
9 });
10
11 });
12 </script>
13 </head>
14 <body>
15 <p>窗口大小被调整过 <span>0</span> 次。</p>

```

```
16 <p>请试着重新调整浏览器窗口的大小。</p>
17 </body>
18 </html>
```