

1. 创建节点

- 创建节点: 使用 [jQuery](#) 的工厂函数 `$()`: `$(html)`; 会根据传入的 html 标记字符串创建一个 DOM 对象, 并把这个 DOM 对象包装成一个 jQuery 对象返回.
- 注意:
 - a. [动态创建的新元素节点不会被自动添加到文档](#)中, 而是需要使用其他方法将其插入到文档中;
 - b. 当创建单个元素时, 需[注意闭合标签和使用标准的 XHTML 格式](#). 例如创建一个 `<p>` 元素, 可以使用 `$("<p/>")` 或 `$("<p></p>")`, 但不能使用 `$("<p>")` 或 `$("<P>")`
- 创建文本节点就是在创建元素节点时直接把文本内容写出来; 创建属性节点也是在创建元素节点时一起创建

2. 查找节点

- 查找属性节点: 通过 jQuery 选择器完成.
- 操作属性节点: 查找到所需要的元素之后, 可以调用 [jQuery 对象的 attr\(\) 方法](#)来获取它的各种属性值
- 操作文本节点: 通过 [text\(\) 方法](#)

3. 插入节点

- 动态创建 HTML 元素并没有实际用处, 还需要将新创建的节点插入到文档中, 即成为文档中某个节点的子节点

方法	描述
<code>append()</code>	向每个匹配的元素的内部的结尾处追加内容
<code>appendTo()</code>	将每个匹配的元素追加到指定的元素中的内部的结尾处
<code>prepend()</code>	向每个匹配元素的内部的开始处插入内容
<code>prependTo()</code>	将每个匹配的元素插入到指定的元素内部的开始处

- 示例代码

```
1 http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
```

网打尽

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Untitled Document</title>

<script type="text/javascript" src="jquery-1.8.3.min.js"></script>
<script type="text/javascript">

    //测试使用 jQuery 创建节点并插入节点到指定的节点中
    /*
    1. 创建节点：使用 $(html) 方式即可，元素节点，文本节点，属性节点可以一
    返回对应节点的 jQuery 对象：
    $("<li id='atguigu'>尚硅谷</li>")

    2. 添加节点：
    1). appendTo 和 append：主语和宾语的位置不同：
    $("<li id='atguigu'>尚硅谷</li>").appendTo($("#city"));
    ($("#city").append("<li id='atguigu'>[尚硅谷]</li>"));

    2). prependTo 和 prepend：主语和宾语的位置不同：
    $("<li id='atguigu'>尚硅谷</li>").prependTo($("#city"));
    ($("#city").prepend("<li id='atguigu'>[尚硅谷]</li>"));
    */
    $(function(){
        $("#p1").click(function() {
            // $("<li id='gqc'>共青城</li>").appendTo($("#city"));
            $("#city").append("<li id='gqc'>共青城</li>");
        });

        $("#p2").click(function() {
            // $("#game").prepend("<li id='ds'>饥荒</li>");
            $("<li id='ds'>饥荒</li>").prependTo("#game");
        });
    });

</script>

</head>
<body>
    <p id="p1">你喜欢哪个城市?</p>
    <ul id="city"><li id="bj" name="BeiJing">北京</li>
        <li>上海</li>
        <li id="dj">东京</li>
        <li id="se">首尔</li>
    </ul>
```

```

49     <br><br>
50     <p id = "p2">你喜欢哪款单机游戏?</p>
51     <ul id="game">
52         <li id="r1">红警</li>
53         <li id="sk">实况</li>
54         <li>极品飞车</li>
55         <li>魔兽</li>
56     </ul>
57
58     <br><br>
59     gender:
60         <input type="radio" name="gender" value="male"/>Male
61         <input type="radio" name="gender" value="female"/>Female
62
63     <br><br>
64     name: <input type="text" name="username" value="atguigu"/>
65
66 </body>
67 </html>

```

```

1  $("<tr></tr>").append($("<td>"+name+"</td>"))
2      .append($("<td>"+email+"</td>"))
3      .append($("<td>"+salary+"</td>"))
4      .append($("<td><a href='deleteEmp?id=XXX'>Delete</a>
</td>"))
5
6      .appendTo($("#employeetable tbody"))
7      .find("a").click(function() {
8          removeTr(this);
9          return false;
        });

```

方法	描述
after()	向每个匹配的元素 的 之后 插入内容
insertAfter()	向每个匹配的元素插入到指定的元素 之后
before()	向每个匹配的元素 的 之前 插入内容
insertBefore()	向每个匹配的元素插入到指定的元素 之前

• 示例代码

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title>Untitled Document</title>
6
7          <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
8          <script type="text/javascript">
9
10             //测试使用 jQuery 插入节点
11             $(function(){
12                 //1. 创建一个 <li id="atguigu">尚硅谷</li>
13                 //2. 并把其加入到 #bj 的后面
14                 $("#p1").click(function() {
15                     $("<li id='gqc'>共青城</li>").insertAfter($("#dj"));
16                     // $("#dj").before("<li id='gqc'>共青城</li>");
17                 });
18
19                 $("#p2").click(function() {
20                     $("<li id='ds'>饥荒</li>").insertBefore($("#sk"));
21                     // $("#sk").after("<li id='ds'>饥荒</li>");
22                 });
23             });
24
25
26         </script>
27
28     </head>
29     <body>
30         <p id="p1">你喜欢哪个城市?</p>
31         <ul id="city"><li id="bj" name="BeiJing">北京</li>
32             <li>上海</li>

```

```

33         <li id="dj">东京</li>
34         <li id="se">首尔</li>
35     </ul>
36
37     <br><br>
38     <p id="p2">你喜欢哪款单机游戏?</p>
39     <ul id="game">
40         <li id="rl">红警</li>
41         <li id="sk">实况</li>
42         <li>极品飞车</li>
43         <li>魔兽</li>
44     </ul>
45
46     <br><br>
47     gender:
48         <input type="radio" name="gender" value="male"/>Male
49         <input type="radio" name="gender" value="female"/>Female
50
51     <br><br>
52     name: <input type="text" name="username" value="atguigu"/>
53
54 </body>
55 </html>

```

4. 删除节点

- `remove()`: 从 DOM 中删除所有匹配的元素, 传入的参数用于根据 jQuery 表达式来筛选元素. 当某个节点用 `remove()` 方法删除后, 该节点所包含的所有后代节点将被同时删除. 这个方法的返回值是一个指向已被删除的节点的引用。
-
- `empty()`: 清空节点 – 清空元素中的所有后代节点(不包含属性节点)。

• 示例代码

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title>Untitled Document</title>

```

```

6
7     <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
8     <script type="text/javascript">
9
10        //测试使用 jQuery 删除节点
11        $(function(){
12            //1. 为 #city 的每一个 li 添加 click 响应函数：点击时，li 被删除
13            $("#city li").click(function() {
14                $(this).remove();
15            });
16
17            //jQuery 对象的 remove() 方法：将把 jQuery 对象对应的
18            //DOM 节点直接删除。
19            // $("#bj").remove();
20
21            //2. 清空 #game 节点
22            //jQuery 对象的 empty() 方法：清空 jQuery 对象对应的
23            //DOM 对象的所有的子节点。
24            $("#p2").click(function() {
25                alert("要清空了!");
26                $("#game").empty();
27            })
28
29        })
30
31
32    </script>
33
34    </head>
35    <body>
36        <p>你喜欢哪个城市?</p>
37        <ul id="city"><li id="bj" name="BeiJing">北京</li>
38            <li>上海</li>
39            <li id="dj">东京</li>
40            <li id="se">首尔</li>
41        </ul>
42
43        <br><br>
44        <p id="p2">你喜欢哪款单机游戏?</p>
45        <ul id="game">
46            <li id="r1">红警</li>
47            <li>实况</li>
48            <li>极品飞车</li>
49            <li>魔兽</li>
50        </ul>
51

```

```
52     <br><br>
53     gender:
54         <input type="radio" name="gender" value="male"/>Male
55         <input type="radio" name="gender" value="female"/>Female
56
57     <br><br>
58     name: <input type="text" name="username" value="atguigu"/>
59
60 </body>
61 </html>
```

5. 克隆结点

- `clone()`: 克隆匹配的 DOM 元素, 返回值为克隆后的副本. 但此时复制的新节点不具有任何行为。
- `clone(true)`: 复制元素的同时也复制元素中的的事件。
- 示例代码:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Untitled Document</title>
6
7     <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
8     <script type="text/javascript">
9
10      //测试使用 jQuery clone 方法：复制节点
11      $(function(){
12
13          $("li").click(function(){
14              alert($(this).text());
15          });
16
17      //复制 #bj 节点，并添加到 #r1 节点的后面
18      /*
19      1. clone 节点时需要注意克隆后的节点的 id 属性。若原节点有 id
20      属性，则克隆后，会出现在一个文档中有两个 id 相同的节点的情况。
21      2. clone(true)：在克隆节点的同时，克隆节点包含的事件。
```

```

22         */
23         $("#dj").clone(true)
24             .attr("id", "xrb")
25             .insertAfter($("#se"));
26
27     });
28
29
30     </script>
31
32 </head>
33 <body>
34     <p>你喜欢哪个城市?</p>
35     <ul id="city">
36         <li id="bj" name="BeiJing">北京</li>
37         <li>上海</li>
38         <li id="dj">东京</li>
39         <li id="se">首尔</li>
40     </ul>
41
42     <br><br>
43     <p>你喜欢哪款单机游戏?</p>
44     <ul id="game">
45         <li id="r1">红警</li>
46         <li>实况</li>
47         <li>极品飞车</li>
48         <li>魔兽</li>
49     </ul>
50
51     <br><br>
52     gender:
53         <input type="radio" name="gender" value="male"/>Male
54         <input type="radio" name="gender" value="female"/>Female
55
56     <br><br>
57     name: <input type="text" name="username" value="atguigu"/>
58
59 </body>
60 </html>

```

6. 替换节点

- `replaceWith()`: 将所有匹配的元素都替换为指定的 HTML 或 DOM 元素。

-
- replaceAll(): 颠倒了的 replaceWith() 方法。
-
- 注意: 若在替换之前, 已经在元素上绑定了事件, 替换后原先绑定的事件会与原先的元素一起消失。
- 示例代码

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title>Untitled Document</title>
6
7          <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
8          <script type="text/javascript">
9
10             //测试使用 jQuery replaceWith (replaceAll) 方法: 替换节点
11             /*
12             1. replaceWith , replaceAll 一对方法, 可以完成一件事. 就是主语宾语
13             哪个在前面的问题.
14             2. 以上的两个方法还有移动节点的功能
15             3. 节点互换需要先克隆节点.
16             4. var $r1 = $("#r1").replaceWith($bj2);
17             */
18             $(function(){
19
20                 //1. 创建一个 <li>尚硅谷</li> 节点, 替换 #city 的最后一个 li 子节
21                 点
22                 $("#p1").click(function() {
23                     $("<li>南昌</li>").replaceAll($("#city :last"));
24                 });
25
26                 //2. 创建一个 <li>[尚硅谷]</li> 节点,
27                 //替换 #city 的第二个 li 子节点
28                 $("#p2").click(function() {
29                     $("#city :eq(1)").replaceWith("<li>南昌</li>");
30                 });
31
32                 //3. 互换以下两个节点: #r1 和 #bj. 还有移动的功能.
33                 //节点互换需要先克隆节点.

```

```
36         var $bjTemp = $("#bj").clone(true);
37         //返回值是被替换掉的结点
38         var $r1 = $("#r1").replaceWith($bjTemp);
39         // $temp.prependTo("#city");
40         // alert("换");
41         $("#bj").replaceWith($r1);
42
43     })
44
45
46     </script>
47
48 </head>
49 <body>
50     <p id="p1">你喜欢哪个城市?</p>
51     <ul id="city">
52         <li id="bj" name="BeiJing">北京</li>
53         <li>上海</li>
54         <li id="dj">东京</li>
55         <li id="se">首尔</li>
56     </ul>
57
58     <br><br>
59     <p id="p2">你喜欢哪款单机游戏?</p>
60     <ul id="game">
61         <li id="r1">红警</li>
62         <li>实况</li>
63         <li>极品飞车</li>
64         <li>魔兽</li>
65     </ul>
66
67     <br><br>
68     gender:
69         <input type="radio" name="gender" value="male"/>Male
70         <input type="radio" name="gender" value="female"/>Female
71
72     <br><br>
73     name: <input type="text" name="username" value="atguigu"/>
74
75 </body>
76 </html>
```

7. 包裹结点

- wrap(): 将指定节点用其他标记包裹起来. 该方法对于需要在文档中插入额外的结构化标记非常有用, 而且不会破坏原始文档的语义。
-
- wrapAll(): 将所有匹配的元素用一个元素来包裹. 而 wrap() 方法是将所有的元素进行单独包裹。
-
- wrapInner(): 将每一个匹配的元素子内容(包括文本节点)用其他结构化标记包裹起来。
- 示例代码

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title>Untitled Document</title>
6
7          <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
8          <script type="text/javascript">
9
10             //测试使用 jQuery wrap, wrapAll, wrapInner
11             $(function(){
12
13                 //包装 li 本身
14                 $("p").wrap("<font color='darkblue'></font>");
15
16                 //包装所有的 li
17                 $("#city li").wrapAll("<font color='darkyellow'></font>");
18
19                 //包装 li 里边的文字.
20                 $("#language li").wrapInner("<font color='lightred'></font>");
21             })
22
23
24         </script>
25
26     </head>
27     <body>
28         <p>你喜欢哪个城市?</p>
29         <ul id="city">
30             <li id="bj" name="BeiJing">北京</li>
```

```

31         <li>上海</li>
32         <li id="dj">东京</li>
33         <li id="se">首尔</li>
34     </ul>
35
36     <br><br>
37     <p>你喜欢哪款单机游戏?</p>
38     <ul id="game">
39         <li id="r1">红警</li>
40         <li>实况</li>
41         <li>极品飞车</li>
42         <li>魔兽</li>
43     </ul>
44
45     <br><br>
46     <p>你喜欢哪种开发语言?</p>
47     <ul id="language">
48         <li>C</li>
49         <li>Java</li>
50         <li>.NET</li>
51         <li>PHP</li>
52     </ul>
53
54     <br><br>
55     gender:
56         <input type="radio" name="gender" value="male"/>Male
57         <input type="radio" name="gender" value="female"/>Female
58
59     <br><br>
60     name: <input type="text" name="username" value="atguigu"/>
61
62 </body>
63 </html>

```

8. 属性操作

- attr(): 获取属性和设置属性
 - a. 当为该方法传递一个参数时, 即为某元素的获取指定属性。
 - b. 当为该方法传递两个参数时, 即为某元素设置指定属性的值。
- jQuery 中有很多方法都是一个函数实现获取和设置. 如: attr(), html(), text(), val(), height(), width(), css() 等。
-

- removeAttr(): 删除指定元素的指定属性。
- 示例代码

```
1  <script type="text/javascript">
2
3      //测试使用 html() 方法.
4      $(function(){
5
6          alert($("#city").html());
7
8          $("#city").html("<li id='nc'>南昌</li>");
9
10     })
11
12
13     </script>
```

9. 设置和获取 HTML, 文本和值

-
- 读取和设置某个元素中的 **HTML 内容**: html(). 该方法可以用于 XHTML, 但不能用于 XML 文档。
-
- 读取和设置某个元素中的 **文本内容**: text(). 该方法既可以用于 XHTML 也可以用于 XML 文档。
-
- 读取和设置某个元素中的值: val() --- 该方法类似 JavaScript 中的 value 属性. 对于文本框, 下拉列表框, 单选框该方法可返回元素的值(**多选框只能返回第一个值**). 如果为多选下拉列表框, 则返回一个包含所有选择值的数组。