第七章-jQuery框架DOM操作

jQuery框架继承和优化了JavaScript访问DOM对象的特性，我们使用jQuery框架提供的api可以更加方便的操作DOM对象。

7.1 创建DOM节点

使用JavaScript原生方式创建DOM节点并添加到页面中的代码示例：

```
//01 创建DOM节点
 var oDiv = document.createElement("div");
 //02 设置DOM节点的内容
 oDiv.innerText = "测试的DIV标签";
 //03 把节点添加到页面中
 document.body.appendChild(oDiv);
```

使用jQuery框架创建DOM节点并添加的代码示例：

```
//01 创建DOM节点
 var oDom = $("<div></div>");
 //02 设置DOM节点的内容
 oDom.text("测试的div标签");
 //02 把DOM节点添加到页面中
 $("body").append(oDom);
```

更简单的创建及添加方式：

```
$("body").append($("<div>我是测试的div标签</div>"));
```

说明

【1】jQuery框架简化了DOM操作，直接使用jQuery构造函数$()来创建标签，在创建标签的时候只需要把HTML字符串传递给函数，jQuery框架会根据参数的内容来创建标签并包装成一个jQ实例对象返回。

【2】要明白jQuery框架的DOM操作本身是对JavaScript原生方式进行的封装，所以相对原生的DOM操作而言 效率更低 。

7.2 插入DOM节点

jQuery框架中提供了多个插入DOM节点的方法，我们可以通过调用这些方法方便的实现节点的插入操作。

在JavaScript原生的DOM操作中，我们通常使用 appendChild和insertBefore方法 来实现插入操作，下面的具体的代码示例。

```
<body>
<div>我是div标签1</div>
<div>我是div标签2</div>
<script>

    //appendChild方法使用
    //01 创建p标签
    var oP = document.createElement("p");
    oP.innerHTML = "我是p标签";

    //02 获取页面中第一个div标签
    var oDiv1 = document.getElementsByTagName("div")[0];

    //03 使用appendChild方法添加
    //把p标签插入到oDiv1标签内容的后面
    oDiv1.appendChild(oP);

    //insertBefore方法使用
    var oDiv2 = document.getElementsByTagName("div")[1];
    //把p标签插入到oDiv2标签内容的前面
    oDiv2.insertBefore(oP,oDiv2.firstChild);

</script>
</body>
```

jQuery框架中为我们提供了四个方法来提供对应的功能，它们分别是：

[1]append方法 :向每个匹配的元素内部追加内容。

[2]appendTo方法 ： 把所有匹配的元素追加到另一个指定的元素集合中，和append方法相反。

[3]prepend方法 :向每个匹配的元素内部前置内容。

[4]prependTo方法 ： 把所有匹配的元素前置到另一个指定的元素集合中，和append方法相反。

插入方法的代码示例

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script src="jQuery框架/jquery-2.0.0.js"></script>
</head>
<body>

<div class="cur">我是div1</div>
<div>我是div2</div>
<div>我是div3</div>

<ul>
    <li>我是第1个li</li>
    <li>我是第2个li</li>
    <li>我是第3个li</li>
    <li>我是第4个li</li>
    <li>我是第5个li</li>
</ul>

<button>按钮</button>
```

```
<script>
    $(function () {
        $("button").click(function () {

            //appendTo:把页面中所有的li标签都插入到所有的div标签内容的后面
            //$("li").appendTo($("div"));
            //append:把页面中所有的div标签都插入到所有的li标签内容的后面
            //$("li").append($("div"));

            //prependTo:把页面中所有的li标签都插入到所有的div标签内容的前面
            //$("li").prependTo($("div"));
            //prepend:把页面中所有的div标签都插入到所有的li标签内容的前面
            //$("li").prepend($("div"));
        })
    })

</script>
</body>
</html>
```

jQuery框架中还提供了多个外部插入内容的方法，它们分别是：

[1]after方法 ： 在每个匹配的元素之后插入内容。

[2]before方法 ： 在每个匹配的元素之前插入内容。

[3]insertAfter方法 ： 把所有匹配的元素插入到另一个指定的元素集合的后面。

[4]insertBefore方法 ： 把所有匹配的元素插入到另一个指定的元素集合的前面。

7.3 删除DOM节点

JavaScript原生的DOM操作中可以使用removeChild方法来删除指定的节点以及其包含的所有子节点，并返回这些删除的内容。

jQuery框架中定义了3个删除内容的方法：它们分别是remove（）、empty（）和detach（）。

remove方法 能够将匹配的元素从DOM中删除。

empty方法 用来清空元素包含的内容，该方法没有参数。

detach方法 能够将匹配的元素从DOM中分离出来。

注意

[1] 删除和清空是两个概念，清空操作执行后该标签还存在。

[2] detach方法和remove方法差不多，但detach方法能够保存所有jQuery数据与被移走的元素相关联，所有绑定在元素上的事件、附加的数据等都会保存下来。如果您在移走一个元素不久后，又需要将该元素重新插入DOM，那么推荐使用detach方法。

7.4 复制和替换DOM节点

① 节点的复制

在JavaScript原生方式操作DOM节点时，可以使用cloneNode方法来复制节点，具体的语法如下：

语法： `nodeObject.cloneNode(include_all)`
参数：

include_all参数本身为布尔类型的值。

- 如果为true，那么将会复制原有的节点以及所有的子节点。
- 如果为false，那么紧紧复制节点本身。

jQuery框架中，使用clones方法来复制节点，具体的语法如下：

语法： `jQ.clone ([widthDataAndEvents], [deepWithDataAndEvents])`
参数：
clone方法的两个参数都是可选的布尔值，如果不传递则默认全部为false。

- widthDataAndEvents参数表示是否复制该节点的事件处理数据。
- deepWithDataAndEvents参数表示是否复制子元素的事件处理数据。

② 节点的替换

在原生的DOM操作中，可以使用replaceChild方法来替换节点。
语法： `nodeObject.replaceChild(new_node,old_node)`
参数说明：new_node为指定的新节点，old_node为被替换的节点。如果替换成功，那么会返回被替换的节点，如果替换失败，那么会返回null。

jQuery框架中定义了replaceWith和replaceAll方法来替换节点。

[1] replaceWith方法

语法： `jQ.replaceWith (newContent)`
说明：replaceWith方法能够将所有匹配的元素都替换成指定的HTML或者是DOM元素。
示例： `$("p").replaceWith("<div>我是DIV标签<div>")`

[2] replaceAll方法

语法： `jQ.replaceAll(selector)`
说明：replaceAll方法和replaceWith是一对相反的操作。