

## 第 6 章 DOM 节点操作

学习要点：

1. 创建节点
2. 插入节点
3. 包裹节点
4. 节点操作

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

DOM 中有一个非常重要的功能，就是节点模型，也就是 DOM 中的“M”。页面中的元素结构就是通过这种节点模型来互相对应着的，我们只需要通过这些节点关系，可以创建、插入、替换、克隆、删除等等一些列的元素操作。

### 一. 创建节点

为了使页面更加智能化，有时我们想动态的在 html 结构页面添加一个元素标签，那么在插入之前首先要做的动作就是：创建节点。

```
var box = $('<div id="box">节点</div>'); //创建一个节点
$('body').append(box); //将节点插入到<body>元素内部
```

### 二. 插入节点

在创建节点的过程中，其实我们已经演示怎么通过.append()方法来插入一个节点。但除了这个方法之余呢，jQuery 提供了其他几个方法来插入节点。

内部插入节点方法

方法名	描述
append(content)	向指定元素内部后面插入节点 content
append(function (index, html) {})	使用匿名函数向指定元素内部后面插入节点
appendTo(content)	将指定元素移入到指定元素 content 内部后面
prepend(content)	向指定元素 content 内部的前面插入节点
prepend(function (index, html) {})	使用匿名函数向指定元素内部的前面插入节点
prependTo(content)	将指定元素移入到指定元素 content 内部前面

```
$('#div').append('<strong>节点</strong>'); //向 div 内部插入 strong 节点
$('#div').append(function (index, html) { //使用匿名函数插入节点，html 是原节点
    return '<strong>节点</strong>';
});
```

```
});
$('span').appendTo('div');           //讲 span 节点移入 div 节点内
$('span').appendTo($('div'));        //同上
$('div').prepend('<span>节点</span>'); //将 span 插入到 div 内部的前面
$('div').append(function (index, html) { //使用匿名函数，同上
    return '<span>节点</span>';
});
$('span').prependTo('div');          //将 span 移入 div 内部的前面
$('span').prependTo($('div'));        //同上
```

外部插入节点方法

方法名	描述
after(content)	向指定元素的外部后面插入节点 content
after(function (index, html) {})	使用匿名函数向指定元素的外部后面插入节点
before(content)	向指定元素的外部前面插入节点 content
before(function (index, html) {})	使用匿名函数向指定元素的外部前面插入节点
insertAfter(content)	将指定节点移到指定元素 content 外部的后面
insertBefore(content)	将指定节点移到指定元素 content 外部的前面

```

$('div').after('<span>节点</span>'); //向 div 的同级节点后面插入 span
$('div').after(function (index, html) { //使用匿名函数，同上
    return '<span>节点</span>';
});
$('div').before('<span>节点</span>'); //向 div 的同级节点前面插入 span
$('div').before(function (index, html) { //使用匿名函数，同上
    return '<span>节点</span>';
});
$('span').insertAfter('div'); //将 span 元素移到 div 元素外部的后面
$('span').insertBefore('div'); //将 span 元素移到 div 元素外部的前面

```

### 三. 包裹节点

jQuery 提供了一系列方法用于包裹节点，那包裹节点是什么意思呢？其实就是使用字符串代码将指定元素的代码包含着的意思。

包裹节点

方法名	描述
wrap(html)	向指定元素包裹一层 html 代码
wrap(element)	向指定元素包裹一层 DOM 对象节点

wrap(function (index) {})	使用匿名函数向指定元素包裹一层自定义内容
unwrap()	移除一层指定元素包裹的内容
wrapAll(html)	用 html 将所有元素包裹到一起
wrapAll(element)	用 DOM 对象将所有元素包裹在一起
wrapInner(html)	向指定元素的子内容包裹一层 html
wrapInner(element)	向指定元素的子内容包裹一层 DOM 对象节点
wrapInner(function (index) {})	用匿名函数向指定元素的子内容包裹一层

```

$('div').wrap('<strong></strong>');           //在 div 外层包裹一层 strong
$('div').wrap('<strong>123</strong>');         //包裹的元素可以带内容
$('div').wrap('<strong><em></em></strong>');     //包裹多个元素
$('div').wrap($('strong').get(0));           //也可以包裹一个原生 DOM
$('div').wrap(document.createElement('strong')); //临时的原生 DOM
$('div').wrap(function (index) {           //匿名函数
    return '<strong></strong>';
});
$('div').unwrap();                         //移除一层包裹内容，多个需移除多次
$('div').wrapAll('<strong></strong>');         //所有 div 外面只包一层 strong
$('div').wrapAll($('strong').get(0));       //同上
$('div').wrapInner('<strong></strong>');       //包裹子元素内容
$('div').wrapInner($('strong').get(0));     //DOM 节点
$('div').wrapInner(function () {           //匿名函数
    return '<strong></strong>';
});

```

注意：.wrap()和.wrapAll()的区别在前者把每个元素当成一个独立体，分别包含一层外层；后者将所有元素作为一个整体作为一个独立体，只包含一层外层。这两种都是在外层包含，而.wrapInner()在内层包含。

#### 四. 节点操作

除了创建、插入和包裹节点，jQuery 还提供了一些常规的节点操作方法：复制、替换和删除节点。

//复制节点

```

$('body').append($('div').clone(true));      //复制一个节点添加到 HTML 中

```

注意：clone(true)参数可以为空，表示只复制元素和内容，不复制事件行为。而加上 true 参数的话，这个元素附带的事件处理行为也复制出来。

//删除节点

```

$('div').remove();                          //直接删除 div 元素

```

注意：`.remove()`不带参数时，删除前面对象选择器指定的元素。而`.remove()`本身也可以带选择符参数的，比如：`$('#div').remove('#box');`只删除 `id=box` 的 `div`。

//保留事件的删除节点

`$('#div').detach();`

//保留事件行为的删除

注意：`.remove()`和`.detach()`都是删除节点，而删除后本身方法可以返回当前被删除的节点对象，但区别在于前者在恢复时不保留事件行为，后者则保留。

//清空节点

`$('#div').empty();`

//删除掉节点里的内容

//替换节点

`$('#div').replaceWith('<span>节点</span>');`

//将 `div` 替换成 `span` 元素

`$('#<span>节点</span>').replaceAll('div');`

//同上

注意：节点被替换后，所包含的事件行为就全部消失了。

# 感谢收看本次教程！

本课程是由北风网([ibeifeng.com](http://ibeifeng.com))

瓢城 Web 俱乐部([ycku.com](http://ycku.com))联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！