

## 第5章 基础 DOM 和 CSS 操作

学习要点：

- 1.DOM 简介
- 2.设置元素及内容
- 3.元素属性操作
- 4.元素样式操作
- 5.CSS 方法

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

DOM 在 JavaScript 课程中我们详细的探讨过，它是一种文档对象模型。方便开发者对 HTML 结构元素内容进行展示和修改。在 JavaScript 中，DOM 不但内容庞大繁杂，而且我们开发的过程中需要考虑更多的兼容性、扩展性。在 jQuery 中，已经将最常用的 DOM 操作方法进行了有效封装，并且不需要考虑浏览器的兼容性。

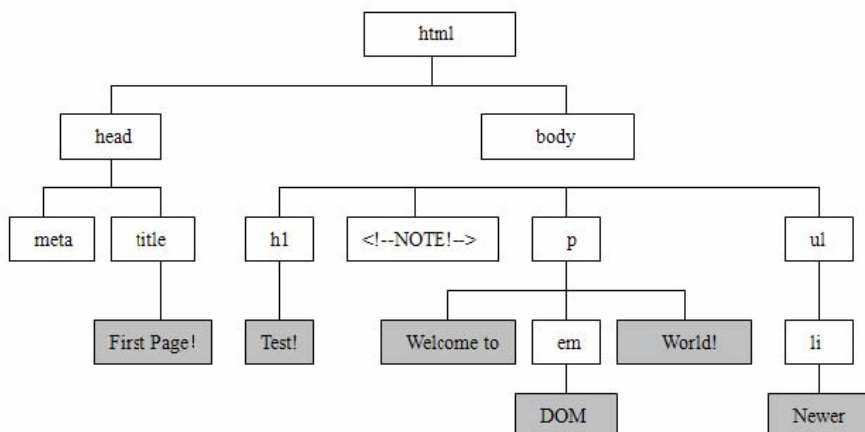
### 一. DOM 简介

由于课程是基于 JavaScript 基础上完成的，这里我们不去详细的了解 DOM 到底是什么。只需要知道几个基本概念：

1.D 表示的是页面文档 Document、O 表示对象，即一组含有独立特性的数据集合、M 表示模型，即页面上的元素节点和文本节点。

2.DOM 有三种形式，标准 DOM、HTML DOM、CSS DOM，大部分都进行了一系列的封装，在 jQuery 中并不需要深刻理解它。

3.树形结构用来表示 DOM，就非常的贴切，大部分操作都是元素节点操作，还有少部分是文本节点操作。



### 二. 设置元素及内容

我们通过前面所学习的各种选择器、过滤器来得到我们想要操作的元素。这个时候，我们就可以对这些元素进行 DOM 的操作。那么，最常用的操作就是对元素内容的获取和修改。

html()和 text()方法

方法名	描述
html()	获取元素中 HTML 内容
html(value)	设置元素中 HTML 内容
text()	获取元素中文本内容
text(value)	设置原生中文本内容
val()	获取表单中的文本内容
val(value)	设置表单中的文本内容

在常规的 DOM 元素中，我们可以使用 html()和 text()方法获取内部的数据。html()方法可以获取或设置 html 内容，text()可以获取或设置文本内容。

```

$('#box').html();           //获取 html 内容
$('#box').text();          //获取文本内容，会自动清理 html 标签
$('#box').html('<em>www.li.cc</em>'); //设置 html 内容
$('#box').text('<em>www.li.cc</em>'); //设置文本内容，会自动转义 html 标签

```

注意：当我们使用 html()或 text()设置元素里的内容时，会清空原来的数据。而我们期望能够追加数据的话，需要先获取原本的数据。

```

$('#box').html($('#box').html() + '<em>www.li.cc</em>'); //追加数据

```

如果元素是表单的话，jQuery 提供了 val()方法进行获取或设置内部的文本数据。

```

$('input').val();           //获取表单内容
$('input').val('www.li.cc'); //设置表单内容

```

如果想设置多个选项的选定状态，比如下拉列表、单选复选框等等，可以通过数组传递操作。

```

$("input").val(["check1","check2","radio1"]); //value 值是这些的将被选定

```

### 三. 元素属性操作

除了对元素内容进行设置和获取，通过 jQuery 也可以对元素本身的属性进行操作，包括获取属性的属性值、设置属性的属性值，并且可以删除掉属性。

attr()和 removeAttr()

方法名	描述
attr(key)	获取某个元素 key 属性的属性值
attr(key, value)	设置某个元素 key 属性的属性值
attr({key1:value2, key2:value2...})	设置某个元素多个 key 属性的属性值
attr(key, function (index, value) {})	设置某个元素 key 通过 fn 来设置

```
$('div').attr('title'); //获取属性的属性值
$('div').attr('title', '我是域名'); //设置属性及属性值
$('div').attr('title', function () { //通过匿名函数返回属性值
    return '我是域名';
});
$('div').attr('title', function (index, value) { //可以接受两个参数
    return value + (index+1) + ', 我是域名';
});
```

注意：attr()方法里的 function() {}，可以不传参数。可以只传一个参数 index，表示当前元素的索引(从 0 开始)。也可以传递两个参数 index、value，第二个参数表示属性原本的值。

注意：jQuery 中很多方法都可以使用 function() {} 来返回出字符串，比如 html()、text()、val()和上一章刚学过的 is()、filter()方法。而如果又涉及到多个元素集合的话，还可以传递 index 参数来获取索引值，并且可以使用第二个参数 value(并不是所有方法都适合，有兴趣可以自己逐个尝试)。

```
$('div').html(function (index) { //通过匿名函数赋值，并传递 index
    return '我是' + (index+1) + '号 div';
});

$('div').html(function (index, value) { //还可以实现追加内容
    return '我是' + (index+1) + '号 div: '+value ;
});
```

注意：我们也可以使用 attr()来创建 id 属性，但我们强烈不建议这么做。这样会导致整个页面结构的混乱。当然也可以创建 class 属性，但后面会有一个语义更好的方法来代替 attr()方法，所以也不建议使用。

删除指定的属性，这个方法就不可以使用匿名函数，传递 index 和 value 均无效。

```
$('div').removeAttr('title'); //删除指定的属性
```

#### 四. 元素样式操作

元素样式操作包括了直接设置 CSS 样式、增加 CSS 类别、类别切换、删除类别这几种操作方法。而在整个 jQuery 使用频率上来看，CSS 样式的操作也是极高的，所以需要重点掌握。

CSS 操作方法

方法名	描述
css(name)	获取某个元素行内的 CSS 样式
css([name1, name2, name3])	获取某个元素行内多个 CSS 样式
css(name, value)	设置某个元素行内的 CSS 样式
css(name, function (index, value) )	设置某个元素行内的 CSS 样式

<code>css({name1 : value1, name2 : value2})</code>	设置某个元素行内多个 CSS 样式
<code>addClass(class)</code>	给某个元素添加一个 CSS 类
<code>addClass(class1 class2 class3...)</code>	给某个元素添加多个 CSS 类
<code>removeClass(class)</code>	删除某个元素的一个 CSS 类
<code>removeClass(class1 class2 class3...)</code>	删除某个元素的多个 CSS 类
<code>toggleClass(class)</code>	来回切换默认样式和指定样式
<code>toggleClass(class1 class2 class3...)</code>	同上
<code>toggleClass(class, switch)</code>	来回切换样式的时候设置切换频率
<code>toggleClass(function () {})</code>	通过匿名函数设置切换的规则
<code>toggleClass(function () {}, switch)</code>	在匿名函数设置时也可以设置频率
<code>toggleClass(function (i, c, s) {}, switch)</code>	在匿名函数设置时传递三个参数

```

$('div').css('color');           //获取元素行内 CSS 样式的颜色
$('div').css('color', 'red');    //设置元素行内 CSS 样式颜色为红色

```

在 CSS 获取上，我们也可以获取多个 CSS 样式，而获取到的是一个对象数组，如果用传统方式进行解析需要使用 for in 遍历。

```

var box = $('div').css(['color', 'height', 'width']); //得到多个 CSS 样式的数组对象
for (var i in box) {                                //逐个遍历出来
    alert(i + ':' + box[i]);
}

```

jQuery 提供了一个遍历工具专门来处理这种对象数组，\$.each()方法，这个方法可以轻松遍历对象数组。

```

$.each(box, function (attr, value) {                //遍历 JavaScript 原生态的对象数组
    alert(attr + ':' + value);
});

```

使用\$.each()可以遍历原生的 JavaScript 对象数组，如果是 jQuery 对象的数组怎么使用\$.each()方法呢？

```

$('div').each(function (index, element) {           //index 为索引，element 为元素 DOM
    alert(index + ':' + element);
});

```

在需要设置多个样式的时候，我们可以传递多个 CSS 样式的键值对即可。

```

$('div').css({
    'background-color' : '#ccc',
    'color' : 'red',

```

```
});
```

如果想设置某个元素的 CSS 样式的值, 但这个值需要计算我们可以传递一个匿名函数。

```
$( 'div' ).css( 'width', function ( index, value ) {  
    return ( parseInt( value ) - 500 ) + 'px';  
});
```

除了行内 CSS 设置, 我们也可以直接给元素添加 CSS 类, 可以添加单个或多个, 并且也可以删除它。

```
$( 'div' ).addClass( 'red' );           //添加一个 CSS 类  
$( 'div' ).addClass( 'red bg' );       //添加多个 CSS 类  
$( 'div' ).removeClass( 'bg' );         //删除一个 CSS 类  
$( 'div' ).removeClass( 'red bg' );     //删除多个 CSS 类
```

我们还可以结合事件来实现 CSS 类的样式切换功能。

```
$( 'div' ).click( function () {        //当点击后触发  
    $( this ).toggleClass( 'red size' ); //单个样式多个样式均可  
});
```

.toggleClass() 方法的第二个参数可以传入一个布尔值, true 表示执行切换到 class 类, false 表示执行回默认 class 类(默认的是空 class), 运用这个特性, 我们可以设置切换的频率。

```
var count = 0;  
$( 'div' ).click( function () {        //每点击两次切换一次 red  
    $( this ).toggleClass( 'red', count++ % 3 == 0 );  
});
```

默认的 CSS 类切换只能是无样式和指定样式之间的切换, 如果想实现样式 1 和样式 2 之间的切换还必须自己写一些逻辑。

```
$( 'div' ).click( function () {  
    $( this ).toggleClass( 'red size' ); //一开始切换到样式 2  
    if ( $( this ).hasClass( 'red' ) ) { //判断样式 2 存在后  
        $( this ).removeClass( 'blue' ); //删除样式 1  
    } else {  
        $( this ).toggleClass( 'blue' ); //添加样式 1, 这里也可以 addClass  
    }  
});
```

上面的方法较为繁琐, .toggleClass() 方法提供了传递匿名函数的方式, 来设置你所需要切换的规则。

```
$( 'div' ).click( function () {  
    $( this ).toggleClass( function () { //传递匿名函数, 返回要切换的样式  
        return $( this ).hasClass( 'red' ) ? 'blue' : 'red size';  
    }  
});
```

```
});  
});
```

注意：上面虽然一句话实现了这个功能，但还是有一些小缺陷，因为原来的 class 类没有被删除，只不过被替代了而已。所以，需要改写一下。

```
$('#div').click(function () {  
    $(this).toggleClass(function () {  
        if ($(this).hasClass('red')) {  
            $(this).removeClass('red');  
            return 'green';  
        } else {  
            $(this).removeClass('green');  
            return 'red';  
        }  
    });  
});
```

我们也可以在传递匿名函数的模式下，增加第二个频率参数。

```
var count = 0;  
$('#div').click(function () {  
    $(this).toggleClass(function () {  
        return $(this).hasClass('red') ? 'blue' : 'red size';  
    }, count++ % 3 == 0); //增加了频率  
});
```

对于.toggleClass()传入匿名函数的方法，还可以传递 index 索引、class 类两个参数以及频率布尔值，可以得到当前的索引、class 类名和频率布尔值。

```
var count = 0;  
$('#div').click(function () {  
    $(this).toggleClass(function (index, className, switchBool) {  
        alert(index + '!' + className + '!' + switchBool); //得到当前值  
        return $(this).hasClass('red') ? 'blue' : 'red size';  
    }, count++ % 3 == 0);  
});
```

## 五. CSS 方法

jQuery 不但提供了 CSS 的核心操作方法，比如.css()、.addClass()等。还封装了一些特殊功能的 CSS 操作方法，我们分别来了解一下。

width()方法

方法名	描述
width()	获取某个元素的长度

width(value)	设置某个元素的长度
width(function (index, width) {})	通过匿名函数设置某个元素的长度

```

$('div').width();           //获取元素的长度，返回的类型为 number
$('div').width(500);        //设置元素长度，直接传数值，默认加 px
$('div').width('500pt');     //同上，设置了 pt 单位
$('div').width(function (index, value) {
    return value - 500;
});                          //index 是索引，value 是原本值
                              //无须调整类型，直接计算
    
```

#### height()方法

方法名	描述
height()	获取某个元素的长度
height(value)	设置某个元素的长度
height(function (index, width) {})	通过匿名函数设置某个元素的长度

```

$('div').height();          //获取元素的高度，返回的类型为 number
$('div').height(500);       //设置元素高度，直接传数值，默认加 px
$('div').height('500pt');   //同上，设置了 pt 单位
$('div').height(function (index, value) {
    return value - 1;
});                          //index 是索引，value 是原本值
                              //无须调整类型，直接计算
    
```

#### 内外边距和边框尺寸方法

方法名	描述
innerWidth()	获取元素宽度，包含内边距 padding
innerHeight()	获取元素高度，包含内边距 padding
outerWidth()	获取元素宽度，包含边框 border 和内边距 padding
outerHeight()	获取元素高度，包含边框 border 和内边距 padding
outerWidth(true)	同上，且包含外边距
outerHeight(true)	同上，且包含外边距

```

alert($('div').width());     //不包含
alert($('div').innerWidth()); //包含内边距 padding
alert($('div').outerWidth()); //包含内边距 padding+边框 border
alert($('div').outerWidth(true)); //包含内边距 padding+边框 border+外边距 margin
    
```

元素偏移方法

方法名	描述
offset()	获取某个元素相对于视口的偏移位置
position()	获取某个元素相对于父元素的偏移位置
scrollTop()	获取垂直滚动条的值
scrollTop(value)	设置垂直滚动条的值
scrollLeft()	获取水平滚动条的值
scrollLeft(value)	设置水平滚动条的值

`$('strong').offset().left;`

//相对于视口的偏移

`$('strong').position().left;`

//相对于父元素的偏移

`$(window).scrollTop();`

//获取当前滚动条的位置

`$(window).scrollTop(300);`

//设置当前滚动条的位置

# 感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！