

## 18.5.3 设置属性

【示例 2】下面示例定义了一个文本节点和元素节点，并为一级标题元素设置 title 属性，最后把它们添加到文档结构中。

```
var hello = document.createTextNode("Hello World! "); // 创建一个文本节点
var h1 = document.createElement("h1");           // 创建一个一级标题
h1.setAttribute("title", "你好，欢迎光临！");    // 为一级标题定义 title 属性
h1.appendChild(hello);                           // 把文本节点增加到一级标题中
document.body.appendChild(h1);                   // 把一级标题增加到文档
```

【示例 3】也可以通过快捷方法设置 HTML DOM 文档中元素的属性值。

```
<label id="label1">文本框:
  <input type="text" name="textfield" id="textfield" />
</label>
<script>
var label = document.getElementById("label1");
label.className="class1";
label.htmlFor="textfield";
</script>
```

DOM 支持使用 `getAttribute()` 和 `setAttribute()` 方法读写自定义属性，不过 IE 6.0 及其以下版本浏览器对其支持不是很完善。

【示例 4】直接使用 `className` 添加类样式，会覆盖掉元素原来的类样式。这时可以采用叠加的方式添加类。

```
<div id="red">红盒子</div>
<script>
var red = document.getElementById("red");
red.className = "red";
red.className += " blue";
</script>
```

【示例 5】使用叠加的方式添加类也存在问题，这样容易添加大量重复的类。为此，定义一个检测函数，判断元素是否包含指定的类，然后再决定是否添加类。

```
<script>
function hasClass(element,className){           //类名检测函数
  var reg =new RegExp('(\\s|^)+' + className + '(\\s|$)');
  return reg.test(element.className);           //使用正则检测是否有相同的样式
}
function addClass(element,className){           //添加类名函数
  if(!hasClass(element, className))
    element.className += ' ' + className;
}
</script>
<div id="red">红盒子</div>
<script>
var red = document.getElementById("red");
addClass(red,'red');
addClass(red,'blue');
</script>
```