## DOM（三）

### 10.1.3 Element类型

要访问元素的标签名，可以用nodeValue，也可以用tagName。这两个属性都返回同样的值，都是大写的。只不过后者是为了清晰起见。

在HTML中两个值都是大写的，但是在XML或者XHTML中就不一定了，XML和XHYML会严格和标签的大小写情况保持一致，因此养成正确的好习惯，应该这样验证一个元素：

if(element.tagName.toLowerCase() == "div"){

// do something ...

}

什么是特性什么是属性？

在我看来，特性和属性都支持自定义，特性是特性，属性是属性。

凡是写在HTML标签里的都是特性，凡是在Js中可以通过DOM属性定义和访问的都是属性，正常来讲，属性和特性互不干预，你不可以直接访问DOM属性的方式来访问特性。但是却存在着一些特殊的特性可以通过这种方式访问，这些特殊的属性也就是下面所说的“脚踏两只船”的属性。

1. HTML元素

HTML元素普遍有以下几个脚踏两只船（即是特性也是属性）的特性或者说说是属性。这些属性都是经过公认的属性。

他们分别是：

Id,title,lang（语言，不常用）,dir（语言的方向，不常用，‘ltr’从左到右，‘rtl’从右到左）,className（class是保留字，用来访问class特性）。

以上这些DOM属性都可以通过访问特性的方式来访问，也就是说他们“脚踏两只船”。

那些一般的特性就是指除了以上介绍的特性以及自定义的特性，比如（name,data-xxx）。

1. 取得特性

操作DOM的特性的方法主要有三个。

查 ： getAttribute()

增，改： setAttribute()

删：removeAttribute()

(Tip：在我看来，类似的东西都有增删查改替换的功能)

具体用法不在赘述。这些方法都可以访问任一特性。无论是自定义的还是不是自定义的。但是不能访问DOM属性。

对于以下例子：

<div *id* = "fxm" *class* = "f" *title* = "这是一个隐形的div" *data-my\_attr* = "我的自定义新属性"></div>

上面的特性有id class title和my\_attr，前三个是“脚踏两只船”的公认特性，最后一个是自定属性（根据H5规范，应该在自定义特性前加上data-前缀）。

下面来访问这些特性。

var f = document.getElementById("fxm");

*console*.log(f.getAttribute("data-my\_attr"));

//可以通过访问DOM属性的方式访问这些公认的特性

*console*.log(f.id + "||" + f.title + "||" + f.className);

结果如下：

我的自定义新属性

fxm||这是一个隐形的div||f

下面为div添加DOM属性；

f.as = "asd"

*console*.log(f.as); // "asd"

这个属性并不是特性，如果想要让他成为特性，只需要setAttribute(“as”,”asd”) 即可。

值得一提的是，有两个特殊的特性，他们虽然也有对应的DOM属性，但是属性的值与通过getAttribute返回的值并不一样。

这两个就是style和onclick这样的事件处理函数。

直接打印style属性会返回style对象：

*console*.log(document.getElementById("app").style);

CSSStyleDeclaration {alignContent: "", alignItems: "", alignSelf: "", alignmentBaseline: "", all: "", …}

具体的用法和效果我也不再多赘述。

再来看看通过getAttribute访问；

<div *id* = "fxm" *style* = "height:30px; border:1px red solid;" *class* = "f" *title* = "这是一个隐形的div" *data-my\_attr* = "我的自定义新属性"></div>

返回结果为：

height:30px; border:1px red solid;

当然，这是嵌入式的css,如果是内部或者外部那肯定是获取不了，毕竟这已经不是DOM特性的范畴了。

事件处理程序也是如此，通过属性访问的结果人尽皆知。getAttribute则会返回的其中的代码。

*DOM中的 ：onclick* = "alert('你好')"

*console*.log(document.getElementById("fxm").getAttribute("onclick"));

返回结果

alert('你好')

1. attribute属性节点

属性节点是一个NamedNodeMap，与NodeList类似。也是一个动态的集合。有以下操作属性节点的方法。

增,：setNamedItem(node); 向列表中添加索引，以节点的nodeValue为索引。

查：getNamedItem(name);

删：removeNamedItem(name);

并没有改和替换。

查还有个经典的方法 item(pos)；也可以通过中括号语法来操作:[pos],[‘key’]。

因为是属性节点也是节点，nodeType为2,可以使用nodeValue，nodeName等查询节点信息的公用方法。

尽管属性节点也是节点，但是他们却不属于DOM文档树上的任何一个部分。

可以通过document.createAtrribute(“name”);来创建属性节点。（注意与之前讨论的区分，之前的相当于操作NamedNodeMap）

可以通过nodeValue或者value来这是属性节点的值。

刚刚创建出来的属性节点和新的DOM一样是一个“孤儿”。

通过setAttributeNode(node)来将属性节点附加给某个DOM。

当然还可以通过访问DOM的getAttributeNode()方法来获取属性节点。

Tip： 通常我们都是用getAttribute和setAttribute以及removeAttribute三个方法设置特性，一般不访问属性节点。但是要知道NamedNodeMap的方法和操作属性节点的方法都是获取和访问的对应的节点，而前者则获取和访问的是值。