http://stylechen.com/attribute-property.html

DOM元素的attribute和property很容易混倄在一起，分不清楚，两者是不同的东西，但是两者又联系紧密。很多新手朋友，也包括以前的我，经常会搞不清楚。

attribute翻译成中文术语为“特性”，property翻译成中文术语为“属性”，从中文的字面意思来看，确实是有点区别了，先来说说attribute。

attribute是一个特性节点，每个DOM元素都有一个对应的attributes属性来存放所有的attribute节点，attributes是一个类数组的容器，说得准确点就是NameNodeMap，总之就是一个类似数组但又和数组不太一样的容器。attributes的每个数字索引以名值对(name=”value”)的形式存放了一个attribute节点。

|  |  |
| --- | --- |
| 1 | <div **class**="box" id="box" gameid="880">hello</div> |

上面的div元素的HTML代码中有class、id还有自定义的gameid，这些特性都存放在attributes中，类似下面的形式：

|  |  |
| --- | --- |
| 1 | [ **class**="box", id="box", gameid="880" ] |

可以这样来访问attribute节点：

|  |  |
| --- | --- |
| 1 | var elem = document.getElementById( 'box' ); |

|  |  |
| --- | --- |
| 2 | console.log( elem.attributes[0].name ); // class |

|  |  |
| --- | --- |
| 3 | console.log( elem.attributes[0].value ); // box |

但是IE6-7将很多东西都存放在attributes中，上面的访问方法和标准浏览器的返回结果又不同。通常要获取一个attribute节点直接用getAttribute方法：

|  |  |
| --- | --- |
| 1 | console.log( elem.getAttribute('gameid') ); // 880 |

要设置一个attribute节点使用setAttribute方法，要删除就用removeAttribute：

|  |  |
| --- | --- |
| 1 | elem.setAttribute('testAttr', 'testVal'); |

|  |  |
| --- | --- |
| 2 | console.log( elem.removeAttribute('gameid') ); // undefined |

attributes是会随着添加或删除attribute节点动态更新的。

property就是一个属性，如果把DOM元素看成是一个普通的Object对象，那么property就是一个以名值对(name=”value”)的形式存放在Object中的属性。要添加和删除property也简单多了，和普通的对象没啥分别：

|  |  |
| --- | --- |
| 1 | elem.gameid = 880; // 添加 |

|  |  |
| --- | --- |
| 2 | console.log( elem.gameid ) // 获取 |

|  |  |
| --- | --- |
| 3 | delete elem.gameid // 删除 |

之所以attribute和property容易混倄在一起的原因是，很多attribute节点还有一个相对应的property属性，比如上面的div元素的id和class既是attribute，也有对应的property，不管使用哪种方法都可以访问和修改。

|  |  |
| --- | --- |
| 1 | console.log( elem.getAttribute('id') ); // box |

|  |  |
| --- | --- |
| 2 | console.log( elem.id ); // box |

|  |  |
| --- | --- |
| 3 | elem.id = 'hello'; |

|  |  |
| --- | --- |
| 4 | console.log( elem.getAttribute('id') ); // hello |

但是对于自定义的attribute节点，或者自定义property，两者就没有关系了。

|  |  |
| --- | --- |
| 1 | console.log( elem.getAttribute('gameid') ); // 880 |

|  |  |
| --- | --- |
| 2 | console.log( elem.gameid ); // undefined |

|  |  |
| --- | --- |
| 3 | elem.areaid = '900'; |

|  |  |
| --- | --- |
| 4 | console.log( elem.getAttribute('areaid') ) // null |

对于IE6-7来说，没有区分attribute和property：

|  |  |
| --- | --- |
| 1 | console.log( elem.getAttribute('gameid') ); // 880 |

|  |  |
| --- | --- |
| 2 | console.log( elem.gameid ); // 880 |

|  |  |
| --- | --- |
| 3 | elem.areaid = '900'; |

|  |  |
| --- | --- |
| 4 | console.log( elem.getAttribute('areaid') ) // 900 |

很多新手朋友估计都很容易掉进这个坑中。

DOM元素一些默认常见的attribute节点都有与之对应的property属性，比较特殊的是一些值为Boolean类型的property，如一些表单元素：

|  |  |
| --- | --- |
| 1 | <input type="radio" **checked**="checked" id="raido"> |

|  |  |
| --- | --- |
| 2 | var radio = document.getElementById( 'radio' ); |

|  |  |
| --- | --- |
| 3 | console.log( radio.getAttribute('checked') ); // checked |

|  |  |
| --- | --- |
| 4 | console.log( radio.**checked** ); // true |

对于这些特殊的attribute节点，只有存在该节点，对应的property的值就为true，如：

|  |  |
| --- | --- |
| 1 | <input type="radio" **checked**="anything" id="raido"> |

|  |  |
| --- | --- |
| 2 | var radio = document.getElementById( 'radio' ); |

|  |  |
| --- | --- |
| 3 | console.log( radio.getAttribute('checked') ); // anything |

|  |  |
| --- | --- |
| 4 | console.log( radio.**checked** ); // true |

最后为了更好的区分attribute和property，基本可以总结为attribute节点都是在HTML代码中可见的，而property只是一个普通的名值对属性。

|  |  |
| --- | --- |
| 1 | // gameid和id都是attribute节点 |

|  |  |
| --- | --- |
| 2 | // id同时又可以通过property来访问和修改 |

|  |  |
| --- | --- |
| 3 | <div gameid="880" id="box">hello</div> |

|  |  |
| --- | --- |
| 4 | // areaid仅仅是property |

|  |  |
| --- | --- |
| 5 | elem.areaid = 900; |

原载于：雨夜带刀's Blog