

千锋HTML5学院

第二阶段javascript课程课件



1

属性节点

2

DOM的节点的其他属性

3

忽略空白文本节点

4

节点操作

5

offsetParent参照物offsetTop、offsetLeft

属性节点

attributes属性: 返回该元素节点的属性节点集合。

`box.attributes` //[object NamedNodeMap]

`box.attributes.length`; //返回属性节点个数

`box.attributes[0]`; //[object Attr], 返回第一个属性节点

`box.attributes[0].nodeType`; //2, 节点类型

`box.attributes[0].nodeName`; //属性名称

`box.attributes[0].nodeValue`; //属性值

`box.attributes['id']`; //返回属性名称为id的节点

`box.attributes.getNamedItem('id')`; //返回属性名称为id的节点

firstChild 和 lastChild 属性

childNodes: 子节点集合

firstChild 用于获取当前元素节点的第一个子节点，相当于 `childNodes[0]`；

lastChild 用于获取当前元素节点的最后一个子节点，相当于 `childNodes[box.childNodes.length- 1]`。

```
//获取第一个子节点的文本内容  
console.log(box.firstChild.nodeValue);
```

```
//获取最后一个子节点的文本内容  
console.log(box.lastChild.nodeValue);
```

parentNode/ previousSibling/ nextSibling

parentNode、previousSibling、nextSibling 属性

parentNode 属性返回该节点的父节点，
previousSibling 属性返回该节点的前一个同级节点，
nextSibling 属性返回该节点的后一个同级节点。

```
alert(box.parentNode.nodeName); //获取父节点的标签名  
alert(box.lastChild.previousSibling); //获取前一个同级节点  
alert(box.firstChild.nextSibling); //获取后一个同级节点
```

ownerDocument 属性

ownerDocument 属性返回该节点的文档对象根节点，返回的对象相当于 document。

```
alert(box.ownerDocument === document); //true , 根节点
```

忽略空白文本节点

```
//获取 body 元素节点  
var body= document.getElementsByTagName('body')[0];  
  
//得到子节点个数  
alert(body.childNodes.length);
```

注: 在非 IE 中, 标准的DOM具有识别空白文本节点的功能, 而IE自动忽略了, 如果要保持一致的子元素节点, 需要手工忽略掉它。

忽略nodes数组中的空白节点

```
function filterSpaceNode(nodes) {  
    for(var i=0;i<nodes.length;i++) {  
        if(nodes[i].nodeType== 3 && /\s+$/.test(nodes[i].nodeValue)) {  
            //得到空白节点之后，移到父节点上，删除子节点  
            nodes[i].parentNode.removeChild(nodes[i]);  
        }  
    }  
    return nodes;  
}
```


节点操作

使用DOM创建节点、复制节点、插入节点、删除节点和替换节点。

节点操作方法

方法	说明
write()	这个方法可以把任意字符串插入到文档中
createElement()	创建一个元素节点
appendChild()	将新节点追加到子节点列表的末尾
createTextNode()	创建一个文件节点
insertBefore()	将新节点插入在前面
replaceChild()	将新节点替换旧节点
cloneNode()	复制节点
removeChild()	移除节点

write()方法

write()方法可以把任意字符串插入到文档中去。

```
document.write('<p>这是一个段落！ </p>'); //输出任意字符串
```

createElement()创建元素节点

createElement()方法可以创建一个元素节点。

```
//创建一个元素节点  
document.createElement('p');
```

appendChild()方法

appendChild()方法将一个新节点添加到某个节点的子节点列表的末尾上。

//获取某一个元素节点

```
var box=document.getElementById('box');
```

//创建一个新元素节点<p>

```
var p=document.createElement('p');
```

//把新元素节点<p>添加box节点的子节点末尾

```
box.appendChild(p);
```

createTextNode()方法

createTextNode() 方法创建一个文本节点。

//创建一个文本节点

```
var text = document.createTextNode('段落');
```

//将文本节点添加到子节点末尾

```
p.appendChild(text);
```

- 封装一个函数, 带两个参数(元素节点的名称, 该节点的文本内容)
返回一个带文本内容的元素节点

insertBefore()方法

insertBefore()方法可以把节点插入到指定节点的前面。

//通过父节点调用, 在box之前插入一个节点p;
`box.parentNode.insertBefore(p, box);`

insertBefore()方法

创建一个可以在节点target后插入newElement的函数
insertAfter(newElement, target)

```
function insertAfter(newElement, targetElement) {  
    //得到父节点  
    var parent = targetElement.parentNode;  
    //如果最后一个子节点是当前元素，那么直接添加即可  
    if (parent.lastChild === targetElement) {  
        parent.appendChild(newElement);  
    } else {  
        //否则，在当前节点的下一个节点之前添加  
        parent.insertBefore(newElement, targetElement.nextSibling);  
    }  
}
```

replaceChild()方法

replaceChild()方法可以把节点替换成指定的节点。

//通过父节点调用, p替换了div
`box.parentNode.replaceChild(p, div);`

cloneNode()方法

cloneNode()方法可以把子节点复制出来。

```
var box=document.getElementById('box');
```

```
//获取第一个子节点， true表示复制标签和内容，  
                        false表示只赋值标签
```

```
var clone=box.firstChild.cloneNode(true);
```

```
//添加到子节点列表末尾
```

```
box.appendChild(clone);
```

removeChild()方法

removeChild()方法可以删除指定子节点

//通过父节点调用, 来删除指定子节点
`box.parentNode.removeChild(box);`

练习

1, 动态表格的创建, 在输入框中输入内容后, 点击保存则在表格末尾添加一条记录, 点击删除则会删除对应行的记录

ID: 姓名: 电话:

Id	Name	Tel	操作
30	张三	135234234	删除
40	李四	14543423423	删除
222	王五	11112123	删除

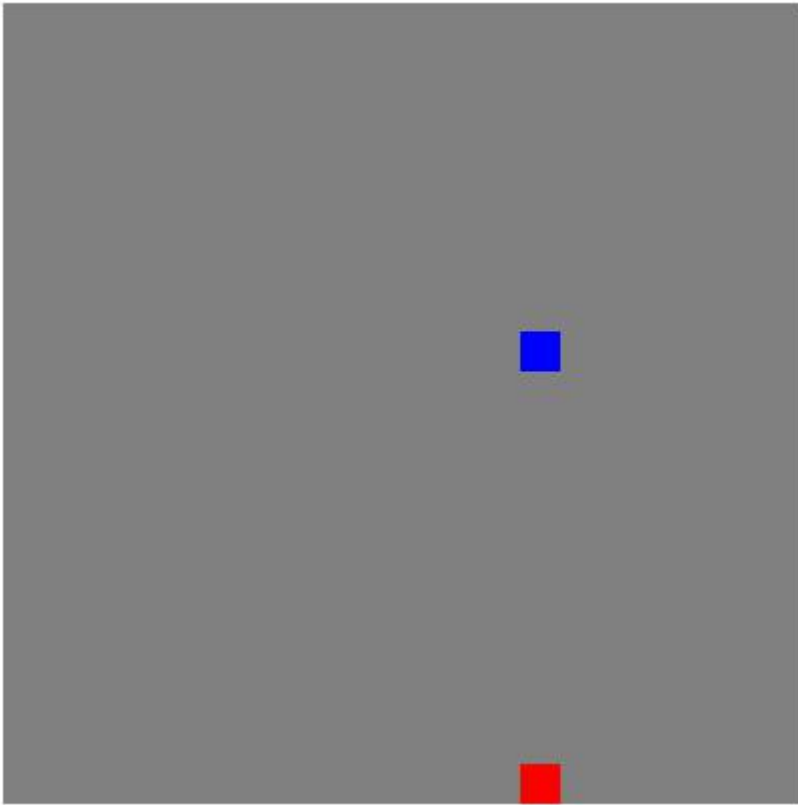
提示: `deleteRow(rowIndex): table`删除对应的行
`//rowIndex: 表中对应的行下标`
`oTable.deleteRow(tr标签.rowIndex);`

offsetParent参照物父元素

- 当某个元素的父元素或以上元素都未进行CSS定位时，则返回body元素，也就是说元素的偏移量（offsetTop、offsetLeft）等属性是以body为参照物的
- 当某个元素的父元素进行了CSS定位时（absolute或者relative），则返回父元素，也就是说元素的偏移量是以父元素为参照物的
- 当某个元素及其父元素都未进行CSS定位时，则返回距离最近的使用了CSS定位的元素

练习

1, 碰撞, 连个块在指定范围内运动, 碰壁后会反弹



作业

1, tab切换, 点击科技显示左图, 点击探索显示右图



