

## day10-javascript BOM&DOM 下

笔记本： 凡尘：二阶段

创建时间： 2017/8/20 3:12

更新时间： 2017/9/27 17:47

作者： 437389128@qq.com

---

### 课程介绍

- 自定义属性及`getAttribute`方法
- `outerHTML/innerText` (非W3C)
- `childNodes`/过滤空白节点
- 高级选取`firstChild/lastChild/parentNode/previousSibling/nextSibling`
- 获取非行内样式 (兼容问题)
- `createTextNode`文本节点
- `box.insertBefore(newNode, existNode)`
- `offsetWidth/offsetHeight`
- `offsetLeft/offsetTop`

### 获取元素的样式 (重)

怎么获取元素的样式值? 比如获取元素的 宽度 高度 透明度等

元素.`style`.样式名 只能获取内嵌样式, 不能获取文件中或者`head`声明的样式

元素.`currentStyle`[样式名] (IE)  
`window.getComputedStyle(元素)[样式名]`

```
var div = document.getElementById();//获取元素
div.currentStyle.width
window.getComputedStyle(div).width
```

### 封装css、show、hide方法(重)

- 1、封装一个方法`css`, 可以获取元素的样式值  
`css(元素, 样式名)` 代表获取元素的样式值

`css(btn, 'width')` 获取`btn`的宽度

- 2、封装方法`show`, 用于显示某个元素  
`show(元素)` 让元素显示

```
function show(elem){
    //somecode...
}
```

- 3、封装方法hide，用于隐藏某个元素  
hide(元素) 让元素隐藏

## 自定义属性

### 1、html自定义属性(重)

- 1、什么是自定义属性？  
用户自己定义的属性（元素本身没有的）  
常用于保留数据
- 2、标签中的自定义属性  
一般自定义属性使用data-开头  
`<div data-url="imgs/large.jpg"></div>`  
  
自定义属性不能通过元素直接访问 元素.属性名  
  
扩展：所谓data-\*实际上就是data-前缀加上自定义的属性名，使用这样的结构可以进行数据存放。使用data-\*可以解决自定义属性混乱无管理的现状

### 2、js自定义属性操作(重)

- 0、为什么要进行属性操作？  
比如页面中往往需要添加很多自定义属性，那么怎么获取他们呢？
- 1、获取元素的属性值  
  
如果是元素自带的属性，我们可以通过 元素.属性名(div.className)获取，如果想获取自定义属性呢？  
元素.getAttribute(属性名);
- 2、设置元素的属性值  
元素.setAttribute(属性名,属性值); (IE7+)
- 3、删除元素的某个属性  
元素.removeAttribute(属性名);

### 3、关于IE的calss与className

class属性在W3C DOM中扮演着很重要的角色，但由于浏览器差异性仍然存在。

FF:使用setAttribute("class", vName)语句动态设置Element的class属性在firefox中是行的通的

IE:在IE中却不行。因为使用IE内核的浏览器不认识"class"，要改用"className"

### 4、封装一个attr方法（根据自己能力选择）

1、封装attr方法，用于获取和设置元素的属性值

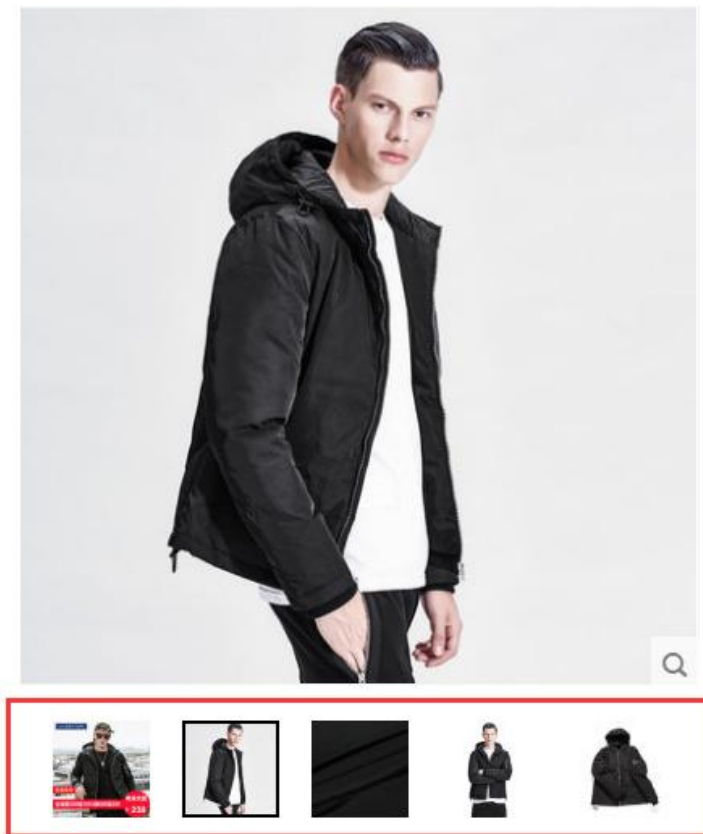
1、获取元素属性（当只传入元素和属性名时，返回元素的属性值）  
元素 `attr(div, 'data-url');` // 返回div的data-url属性 `div`是获取的页面上的元素

2、设置元素属性  
`attr(div, 'data-id', 10)` // 将元素的data-id属性设置为10

提示：根据传入参数的个数进行不同的操作

## 实战

实现图片的切换



## innerHTML、innerText和outerHTML、outerText的区别

- `innerHTML` 设置或获取位于对象起始和结束标签内的 HTML
- `outerHTML` 设置或获取对象及其内容的 HTML 形式
- `innerText` 设置或获取位于对象起始和结束标签内的文本
- `outerText` 设置(包括标签)或获取(不包括标签)对象的文本

`innerText`和`outerText`在获取时是相同效果，但在设置时，`innerText`仅设置标签内的文本，而`outerText`设置包括标签在内的文本。

不同之处：

简单的说`innerHTML`和`outerHTML`、`innerText`与`outerText`的不同之处在于：

1)、`innerHTML`与`outerHTML`在设置对象的内容时包含的HTML会被解析，而`innerText`与`outerText`则不会。

2)、在设置时，`innerHTML`与`innerText`仅设置标签内的文本，而`outerHTML`与`outerText`设置包括标签在内的文本。

3)、`outerHTML` `innerText`(非W3C)

## DOM节点

节点的关系从这里真正的体现出来



## 节点属性

### 文本节点

`childNodes`: 获取所有的子节点(包括文本节点)

## 节点属性

1、节点有哪一种？

节点可以分为元素节点、属性节点和文本节点

2、`node`三剑客

3个非常有用的节点属性（`nodeName`、`nodeType`、`nodeValue`）

**nodeName**：节点名称（）

**nodeType**：节点类型（判断节点是元素节点还是文本节点或者属性节点）  
1 代表元素节点 2 代表属性节点 3 代表文本节点

用途：通常用于判断一个节点是不是元素节点

**nodeValue**：获取节点的文本

如果您希望返回元素的文本，请记住文本始终位于文本节点中，并且您必须返回文本节点的值

和**innerHTML**的区别：**nodeValue**只能获取和设置文本，**innerHTML**可以获取和设置元素的内容

`<div><span>这是一段文本</span></div>`

## 创建文本节点

```
var txt = document.createTextNode('');
```

## 过滤文本节点

## 案例

隔行变色

# 高级节点属性

父子节点：

兄弟节

点：



父（**parent**）、子（**child**）和同胞（**sibling**）等术语用于描述这些关系。父节点拥有子节点。  
同级的子节点被称 为同胞（兄弟或姐妹）

- 1、`children` 获取元素的所有子元素节点
- 2、`firstChild` 获取当前元素节点的第一个子节点  
`firstElementChild` 第一个子元素节点
- 3、`lastChild` 获取当前元素节点的最后一个子节点  
`lastElementChild` 最后一个子元素节点
- 4、`parentNode` 获取当前节点的父节点
- 5、`previousSibling` 获取当前节点的前一个同级节点  
`previousElementSibling` 前一个兄弟元素 （如果没有就是`null`）
- 6、`nextSibling` 获取当前节点的后一个同级节点  
`nextElementSibling` 后一个兄弟元素 （如果没有是`null`）
- 7、`ownerDocument` 获取该节点的文档根节点，相当与 `document`
- 8、`attributes` 获取当前元素节点的所有属性节点集合

扩展：

```
document.getElementById('box').attributes // 获取所有，该节点的属性信息；

document.getElementById('box').attributes.length; // 返回属性节点个数

document.getElementById('box').attributes[0]; // 返回第一个属性节点

document.getElementById('box').attributes[0].nodeType; // 2，属性

document.getElementById('box').attributes[0].nodeValue; // 属性值

document.getElementById('box').attributes['id']; // 返回属性为 id 的节点

document.getElementById('box').attributes.getNamedItem('id'); // 获取 id 的节点；

一般只用作获取；
```

## childNodes与children的区别

以后写js的时候，如果想获取到子元素的`element`节点，最好使用`children`方法，`childNodes`方法以及`firstChild`方法在现代浏览器中使用，都会把元素标签中的空白节点检测出来，一般我们使用这两个方法都是为了获取到元素的元素节点，空白节点会给我们造成很多不必要的bug，而`children`方法则是只检测`element`元素节点，防范于未然，所以推荐大家以后使用`children`方法来替代`childNodes`

## 实战

## 1、进度条

¥259.00	- 5 + 有货	¥1295.00 13kg	删除 移到我的关注
¥299.00	- 1 + 有货	¥299.00 4.92kg	删除 移到我的关注

2、

## 封装方法next/prev(常用方法)

`prev` 获取前一个元素节点

`next` 获取后一个元素节点

## DOM操作（非常重要）

前言：在前一天我们学习了关于dom的增删改查。除了这4种以外dom还可以创建节点 复制节点 插入节点 删除节点和替换节点

1、`document.createElement('标签名')`；//创建一个元素的节点 只有document才能创建

2、`document.createTextNode('文本')`；//创建一个文本节点

3、`box.appendChild(node)`；//把node节点插入到box的内部最后的位置

栗子：

```
<div class="container">
  <p></p>
```



```
</div>
var con = document.createElement('');
con.appendChild(div);
```

4、`box.insertBefore(node,existNode)`//把node节点插入到box中已经存在的节点前面

5、`box.removeChild(node)` 从元素中移除某个子元素

6、`cloneNode()`:复制空标签 `cloneNode(true)` 复制整个元素

7、`box.replaceChild(子元素,要被替换的元素)`

栗子:

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

oDiv.parentNode.replaceChild(p,oDiv);
```

7、`hasChildNodes()`;判断一个节点是否有子节点

## 练习

1、动态表格（学生管理系统-信息录入）

2、弹幕

## 文档碎片（扩展）

1、js操作dom时发生了什么？

每次对dom的操作都会触发"重排"，这严重影响到能耗，一般通常采取的做法是尽可能的减少dom操作来减少"重排"

2、什么是文档碎片？

```
document.createDocumentFragment()
一个容器，用于暂时存放创建的dom元素
```

3、文档碎片有什么用？

将需要添加的大量元素 先添加到文档碎片 中，再将文档碎片添加到需要插入的位置，大大减少dom操作，**提高性能（IE和火狐比较明显）**

**比如需要往页面上放100个元素：**

**普通方式：（操作了100次dom）**

```
for(var i=100; i>0; i--){
    var elem = document.createElement('div');
    document.body.appendChild(elem);//放到body中
}
```

**文档碎片：（操作1次dom）**

```
var df = document.createDocumentFragment();
for(var i=100; i>0; i--){
    var elem = document.createElement('div');
    df.appendChild(elem);
}
//最后放入到页面上
document.body.appendChild(df);
```

## offset

### offsetWidth和offsetHeight (重)

怎么获取元素占据的空间大小 `width + padding + border`  
`css(div, 'width');` 只能获取元素的宽度 (不包含padding、border)

元素.offsetWidth	元素占据的宽度	<code>width + padding + border</code>
元素.offsetHeight	元素占据的高度	<code>height + padding + border</code>

### offsetLeft/offsetTop/offsetParent(重)

## 作业

- 1、网页换肤
- 2、简易年历
- 3、点击按钮切换图片
- 4、tab切换
- 5、QQ延迟提示框
- 6、苹果菜单
- 7、密码强度

## 预习

- 0、常用的事件有哪些 (百度或者查看文档)
- 1、事件对象是什么? 都有哪些属性?
- 2、事件对象中常用属性

```
e.which e.keyCode
e.clientX e.clientY
e.pageX e.pageY
e.offsetX e.offsetY
```
- 3、键盘事件keydown keypress keyup