## js笔记

1. HTML：负责提供内容；
2. CSS：负责修饰页面；
3. JS：搞事情 --》事件

Js: es,dom,bom

事件的三要素：进到教室要开灯

1. 事件源：开关，界面元素-eg：按钮
2. 事件：摁一下，点击—滚轮—键盘敲入
3. 事件处理函数：接通火线，点亮灯泡，定义具体的操作

基本语法结构：

事件源**.**事件 = function(){

// 事件处理，可用this；

}

Eg: div.onclick=function(){ //……定义具体的操作}

小练习：标签页实现

**（※）DOM的基本概念**

DOM：Document Object Model 文档对象模型

DOM定义了表示和修改文档所需的方法。

DOM对象即为宿主对象document（区别于本地对象new String(),Object()），由浏览器厂商定义，用来操作Html和xml功能的一类对象的集合。

也有人称DOM是对HTML以及XML的标准编程接口。

//小练习：方块点击变色

为什么js可以获取到浏览器里面的元素？

因为浏览器支持，且提供了对应的API；

What:干什么和是什么，研究一下怎么去干？

1. **DOM操作初探**

package conctrol、install package、emmet、Jsprettify

//siblime需要安装的插件

**1.1 节点：**

document根节点对象，是表示和修改html的一系列方法的集合

我们称文档里的内容进行分块，称作节点(Node)，找节点querySelector(‘...’)等;

**1.2 节点类型：**

1. document节点：根节点； --类型值为9
2. 元素节点：div，p，b，span； --类型值为1
3. 属性节点：元素节点的属性getAttributeNode；--类型值为2
4. 文本节点：文字信息； --类型值为3
5. 注释节点； --类型值为8

类型节点三要素（属性）：nodeType、nodeName、

nodeValue(属性节点/文本节点/注释节点具有该属性，该属性可读可写，而nodeType/ nodeName属性只读)

**（二）节点的相关操作（查找、选择）：**

增删**查**改

**2.1 查找节点：**

id（元素id在IE8以下的浏览器不区分大小写，且返回可匹配name 属性的元素）

classname（类名，IE8 和IE8 以下的版本中没有，可以多个class）

tagname（标签名，兼容性好-最常用，可用通配符\*）

name（需注意，只有部分标签的name生效，例img/iframe/表单/表单元素）

queryselector()（css 选择器，在ie8 以下的版本中没有）

querySelectorAll()（css 选择器，在ie8 以下的版本中没有）

a1) dom方法中除了getElementById和querySelector方法，其他方法获取的都是类数组；

a2) 通过id进行选择的方法不常用；

a3) querySelector与querySelectorAll 方法慎重选择，它们是非实时的，选取的是静态副本。

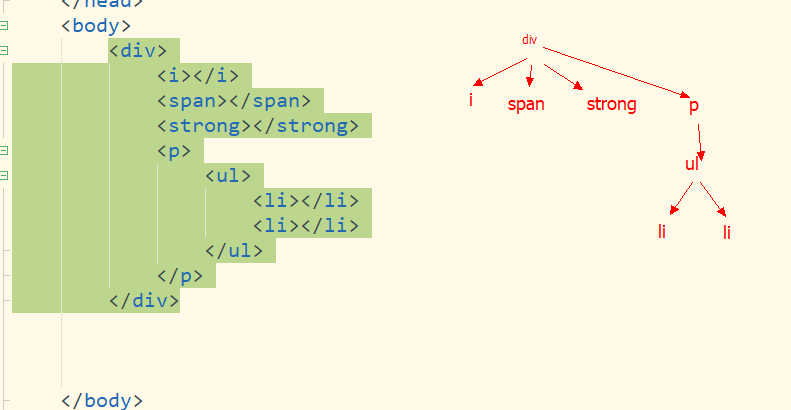
【IE 版本测试：

<meta http-equiv="X-UA-Compatible" content="IE=10" />】

**2.2 选择元素的非方法类操作：**

结合节点关系，可以找到文档里面所有的节点，节点关系??

父子、兄弟、后代



所有dom对象都具有以下两大类属性，允许链式操作。

遍历节点树：

parentNode-->父节点（最顶端的parentNode为#document）

childNodes-->子节点们（判断有无方法：div.hasChildNodes()）

firstChild-->第一个子节点

lastChild-->最后一个子节点

nextSibling-->后一个兄弟节点

previousSibling-->前一个兄弟节点

基于元素节点树的遍历：

parentElement-->返回当前元素的父元素节点（最顶端为html，ie9不兼容）

children-->只返回当前元素的元素子节点

// node.childElementCount === node.children.length当前元素节点的子元素节点个数（ie9不兼容）

firstElementChild-->返回的是第一个元素节点（ie9不兼容）

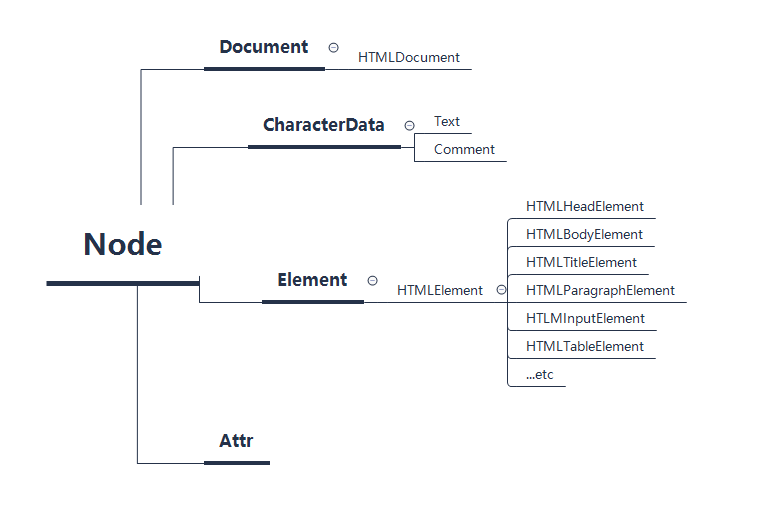
lastElementChild-->返回的是最后一个元素节点（ie9不兼容）

nextElementSibling/previousElementSibling-->返回后一个/前一个兄弟元素节点（ie9不兼容）

（三） **DOM结构树**

1. 要学习DOM结构树，首先复习下原型链知识（构造函数原型，对象的原型，构造函数原型的原型）
2. 了解document与Document、HTMLDocument（浏览器的构造函数，不能手动去new）三者之间的关系。
3. 浏览器构造函数不能手动new实例对象
4. document-HTMLDocument.prototype-Document.prototype

**3.1 DOM结构树，如下图所示：**



以上树节点均为浏览器构造函数，它们之间的枝干展示了具体的继承关系。

HTMLElement的子节点们均为文档中的元素(标签)节点，对应着每一种元素都有自己的特定义构造方法。

可自定义（abc/bcd）属性在构造函数的原型上，验证继承关系。

可通过document的\_\_proto\_\_往上追溯验证顶端构造函数。

**3.2 DOM结构树上定义的属性和方法：**

1. getElementById方法定义在Document.prototype上，即Element节点不能使用，不可链式调用。

b) getElementByName方法定义在HTMLDocument.prototype上，即非html中的document不能使用（eg：xml document,Element）。c) getElementsByTagName方法定义在Document.prototype和Element.prototype上，可链式调用且XML也可用。

d) HTMLDocument.prototype定义了一些常用的属性，例body、head，分别指代HTML文档中的<body><head>标签。document

e) Document.prototype上定义了documentElement属性，指代文档的根元素；在HTML文档中，它总是指代<html>元素。

f) getElementsByClassName、querySelectorAll、querySelector定义在Document.prototype和Element.prototype上。

**（四） 练习习题**

1、 遍历元素节点树（在原型链上编程Element.prototype.traverseNode）

2、封装函数，返回元素e的第n层祖先元素节点

3、封装函数，返回元素e的第n个兄弟元素节点，n为正，返回后面的兄弟元素节点；n为负，返回前面的；n为0，返回自己。

4、编辑函数，在原型链Element.prototype上封装myChildren功能，解决浏览器的兼容性问题, 不可用children属性

5、自己封装hasChildren()方法，不可用children属性，childNodes

**（五） 节点的相关操作（增删改）：**

**增删**查**改**

**5.1 增加节点：**

创建新节点：

document.createElement("p")

document.createTextNode();

document.createComment();

新增节点：

PARENTNODE.appendChild(); // 类似于push,相对于已有节点来说是一种“剪切-粘贴”操作

PARENTNODE.insertBefore(a,b); // 向PARENTNODE中插入节点a,插入后a在b的前边

**5.2 删除和复制节点：**

在指定的父节点里面删除子节点：

parent.removeChild(‘删除的节点’);//相当于他杀，被“剪切”出来

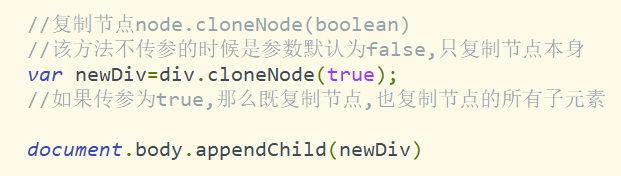
自己删除自己：

node.remove()**;**//相当于自杀，彻底销毁

替换节点：

parent.replaceChild(new, ); //上镜率不高，origin被“剪切”出来

复制节点：cloneNode(Boolean)



**5.3 修改节点：**

获取或修改内容：

1. innerHTML：包括内部html标签，修改符：= 和 += ；
2. outerHTML: 包括内部html标签，还包括自己
3. innerText：返回文本，没有html标签，赋值要小心

修改类名：

className：修改元素的classList；

classList：节点的classList提供了操作class的更专业的方法：

A： classList.add()：在不影响原始class的情况下追加新的class；

B：classList.remove()：删除指定的class，没指定的不会删除；

C：classList.contains()：检查是否存在指定的class；

修改各种属性：

A，标准属性：一个盒子的 id，input的name，img的src，a的href，checkbox的checked；

标准属性可以直接操作：node.attr = ‘新值’;( class属性不能直接操作。)

如： Input.value,img.src,a.href

B，自定义属性：自己定义的属性;

标准的w3c推荐的自定义属性的定义方式：

data-myattr = ‘value’

获取自定义属性的值：

dataset.myattr;

C，任意~~的自定义~~属性：

getAttribute()：获取指定的属性名的值;

setAttribute(‘属性名’，‘属性值’)：设置指定的属性名的值;

hasAttribute()：检查一个属性是否存在;

removeAttribute()：删除指定的属性;

小练习：给body的子标签元素统一设置一个”this-name”的属性，属性值为这些元素自己的标签名nodeName

**（六）练习习题**

1、请编写一段JavaScript脚本生成下面这段DOM结构。要求--使用标准的DOM方法或属性。

<div class="example">

<p class="slogan">XXX，你最帅</p>

</div>

Document.body.innerHTML=”<div class=‘example’> <p class=’slogan’>贺子航，最好看！</p> </div>”

2、封装函数insertAfter，功能类似insertBefore()，直接在Element. prototype上编程;

3、将目标节点内部的节点顺序逆序reverse()。

eg：<div>

<a></a>

<em></em>

</div>

<div> <em></em> <a><a> </div>