logo.png

DOM操作一

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# 一、事件

## 概述

JS是以事件驱动为核心的一门语言。

js是采用事件驱动(event-driven)响应用户操作的。比如通过鼠标或者按键在浏览器窗口或者网页元素(按钮，文本框...)上执行的操作，我们称之为事件(Event)。由鼠标或热键引发的一连串程序的动作，称之为事件驱动(Event-Driver)。对事件进行处理程序或函数，我们称之为事件处理程序(Event Handler)。

## 事件三要素

事件源、事件、事件驱动程序。

事件源 ：要触发的对象 （一般是名词，事件发起者，比如开关按钮）

事件：怎么触发这个事情 （一般是动词，触发事件，比如点击开关）

事件处理程序:发生了什么事情 （处理结果，比如灯亮了）

js做什么：获取事件源、绑定事件、书写事件驱动程序。

### 事件有哪些

参考附件。

# 二、DOM概述

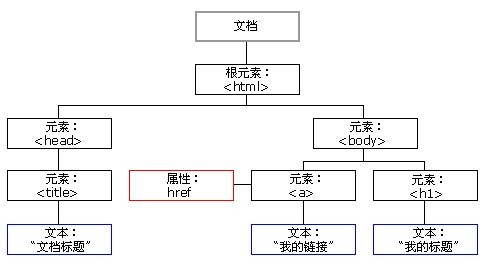
## 什么是DOM

HTML DOM是HTML Document Object Model(文档对象模型)的缩写，HTML DOM则是专门适用于HTML/XHTML的文档对象模型。熟悉软件开发的人员可以将HTML DOM理解为网页的API。它将网页中的各个元素都看作一个个对象，从而使网页中的元素也可以被计算机语言(js)获取或者编辑。 例如Javascript就可以利用HTML DOM动态地修改网页。

DOM 将文档解析为一个由节点和对象（包含属性和方法的对象）组成的结构集合, 是web页面的完全的面向对象表述，简言之，它会将web页面和脚本或程序语言连接起来。 它能够使用如 JavaScript等脚本语言进行修改。

## DOM的数据结构（树状）

文档结构图：



## HTML的组成部分为节点（ Node ）

在HTML当中一切都是节点……

由结构图中我们可以看到，整个文档就是一个文档节点。

每一个HMTL标签都是一个元素节点（标签）。

标签中的文字则是文字节点。（文本）

标签的属性是属性节点。（属性）

## DOM节点的获得

操作节点，必须首先找到该元素。有三种方法来做这件事：

* document.getElementById("demo");

最准确的 ， 需要获取的dom元素，必须有id，而且一个页面不能有重复的id

* document.getElementsByTagName("div");

效率低

没办法定位元素

* document.getElementsByClassName("a");

如果在移动使用，移动端浏览器完全支持。

目前先不考虑<ie8情况，之后学了jQuery或者html5（新的选择器）;

通过类名查找 HTML 元素在 IE 5,6,7,8 中无效

# 三、获取节点

DOM的节点并不是孤立的，因此可以通过DOM节点之间的相对关系对它们进行访问。

全世界你可以通过6个人联系到任意一个人。

节点的访问关系，是以属性的方式存在的。 A.parent a.parent();

## 3.1 父节点 （ parentNode ）

调用者就是节点。一个节点只有一个父节点。调用方式就是节点.parentNode.

案例：

通过子盒子设置父盒子的背景色。

关闭二维码

## 3.2 兄弟节点

* nextSibling：调用者是节点。（存在浏览器兼容问题）

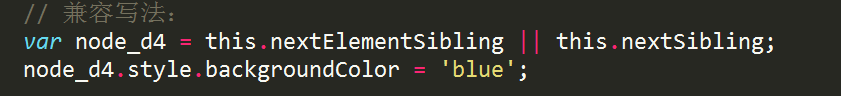
IE678中指下一个元素节点（标签、注释）。在火狐谷歌IE9+(标准)以后都指的是下一个节点（包括空文档和换行节点）。

* nextElementSibling：在火狐谷歌IE9都指的是下一个元素节点。

总结：在IE678中用nextSibling，在火狐谷歌IE9+ ie10以后用nextElementSibling

兼容写法：nextEle =节点.nextElementSibling || 节点.nextSibling

**注意：兼容写法位置不能交换**



* previousSibling：调用者是节点。IE678中指前一个元素节点（标签）。在火狐谷歌IE9+以后都指的是前一个节点（包括空文档和换行节点）。
* previousElementSibling：在火狐谷歌IE9都指的是前一个元素节点。

总结：在IE678中用previousSibling，在火狐谷歌IE9+以后用previousElementSibling。

## 3.3 单个子节点

* firstChild：调用者是父节点。IE678中指第一个子元素节点（标签）。在火狐谷歌IE9+以后都指的是第一个节点（包括空文档和换行节点）。
* firstElementChild:在火狐谷歌IE9都指的第一个元素节点。
* lastChild:调用者是父节点。IE678中指最后一个子元素节点（标签）。在火狐谷歌IE9+以后都指的是最后一个节点（包括空文档和换行节点）。
* lastElementChild：在火狐谷歌IE9都指的最后一个元素节点。

## 3.4 所有子节点

* **childNodes**：它是标准属性，嫡出，它返回指定元素的子元素集合，包括HTML节点，所有属性，文本节点 （他还是W3C的亲儿子 ）

火狐 谷歌等高本版会把换行也看做是子节点

nodeType==1时才是元素节点(标签)

nodeType == 1 表示的是元素节点 记住 元素就是标签

nodeType == 2 表示是属性节点 了解

nodeType == 3 是文本节点 了解

nodeType = 8 注释 了解

* **children**：非标准属性，庶出，它返回指定元素的子元素集合。

但它只返回HTML节点，甚至不返回文本节点，虽然不是标准的DOM属性，但它和innerHTML方法一样，得到了几乎所有浏览器的支持。

children在IE6/7/8中包含注释节点

在IE678中，注释节点不要写在里面。

思考 怎么处理兼容问题？

|  |
| --- |
| // 封装好的兼容ie8和标准浏览器的获取孩子节点的方法  function myChildren(node){  var myChildrenArr = [] ;  var childrenArr = node.children; //拿到的是node的所有的孩子节点，包括ie中的注释节点  for(var n in childrenArr){  if(childrenArr[n].nodeType == 1){ //选择出所有的标签节点  myChildrenArr.push(childrenArr[n]);  }  }  return myChildrenArr;  } |

### 思考：

求除了自己的所有兄弟节点。

# 四、DOM节点操作

**节点的访问关系都是属性。节点的操作都是函数或者方法。**

## 4.1 创建节点

使用方法是这样的document.createElement();

## 4.2 插入节点（使用节点）

使用方法： 父节点.appendChild();

使用方法：父节点.insertBefore(要插入的节点，参考节点)；

如果参考节点为null，那么他将在节点最后插入一个节点。

## 4.3 删除节点

用法：父节点.removeChild（子节点）。

节点自己删除自己：

不知道父级的情况下，可以这么写：node.parentNode.removeChild(node)

## 4.4 复制节点

用法：oldNode.cloneNode（true）

想要复制的节点调用这个函数cloneNode()，得到一个新节点。 方法内部可以传参，入股是true，深层复制，如果是false，只复制节点本身。

## 4.5 节点属性（节点.属性）

获取：getAttribute(名称)

设置：setAttribute(名称, 值)

删除：removeAttribute(名称)

# 五、案例

1.切换图片（a连接+图片）

2.显示和隐藏盒子

3.相册走廊（placehold.it）

## value和innerHTML和innerText和~~textContent~~

1.老版本的火狐不支持innerText，支持textContent

Value只对表单有效果，是属性

innerHTML插入可执行的标签，标签和样式会被解析，常用于动态生成页面元素

innerText 插入文本内容，标签和样式会被当做文本内容处理。

# 附件：

|  |  |
| --- | --- |
| **属性** | **当以下情况发生时，出现此事件** |
| onabort | 图像加载被中断 |
| onblur | 元素失去焦点 |
| onchange | 用户改变域的内容 |
| onclick | 鼠标点击某个对象 |
| ondblclick | 鼠标双击某个对象 |
| onerror | 当加载文档或图像时发生某个错误 |
| onfocus | 元素获得焦点 |
| onkeydown | 某个键盘的键被按下 |
| onkeypress | 某个键盘的键被按下或按住 |
| onkeyup | 某个键盘的键被松开 |
| onload | 某个页面或图像被完成加载 |
| onmousedown | 某个鼠标按键被按下 |
| onmousemove | 鼠标被移动 |
| onmouseout | 鼠标从某元素移开 |
| onmouseover | 鼠标被移到某元素之上 |
| onmouseup | 某个鼠标按键被松开 |
| onreset | 重置按钮被点击 |
| onresize | 窗口或框架被调整尺寸 |
| onselect | 文本被选定 |
| onsubmit | 提交按钮被点击 |
| onunload | 用户退出页面 |