1. HTML DOM Node对象

HTML文档中的每一个成员都是一个节点对象，节点对象具有自己的属性和方法。

* 1. 找到HTML节点和Body节点

document.documentElement -> html标签

可返回存在于 XML 以及 HTML 文档中的文档根节点

document.body ->body标签

对 HTML 页面的特殊扩展，提供了对 <body> 标签的直接访问。

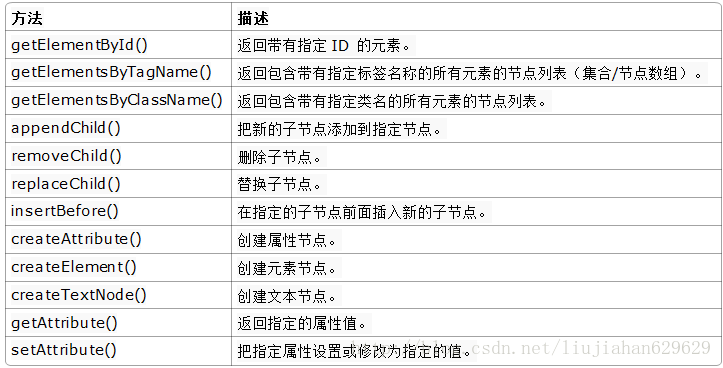
如果我们要对页面上添加显示元素,一般来说会使用这种方式

我们要注意的区别:

document.documentElement.clientHeight:它拿到的是整个窗口高度

document.body.clientHeight:拿到内容的高度

* 1. Node对象的属性与方法总表



|  |  |  |  |
| --- | --- | --- | --- |
|  | **属性名称** | **名称** | **返回类型** |
| **节点列表** | parentNode | 父节点 | Node |
| firstChild | 列表中的第一个节点 | Node |
| lastChild | 列表中的最后一个节点 | Node |
| childNodes | 所有子节点的列表 | NodeList |
| previousSibling | 上一个兄弟节点 | Node |
| nextSibling | 下一个兄弟节点 | Node |
| 节点信息 | nodeName | 节点名称 | String |
| nodeValue | 节点值 | String |
| nodeType | 节点类型 | Number |
| **~~元素属性~~** | ~~attributes~~ | ~~一个元素的属性对象~~ | ~~NamedNodeMap~~ |
| **方法** | hasChildNodes() | 当childNodes包含一个或多个节点时，返回真 | Boolean |
| appendChild(node) | 将node添加到childNodes的末尾 | Node |
| removeChild(node) | 从childNodes中删除node | Node |
| replaceChild(newnode, oldnode) | 将childNodes中的oldnode替换成newnode | Node |
| insertBefore(newnode, refnode) | 在childNodes中的refnode之前插入newnode | Node |
| **其他**  **方法** | reateElement(TagName) | 创建一个元素节点 |  |
| createTextNode(text) | 创建文本节点 |  |
| setAttribute(属性名称，属性的值) | 设置属性 |  |
| getAttribute(属性名称) | 得到属性 |  |

* 1. 节点列表

parentNode，firstChild，lastChild，childNodes，previousSibling，nextSibling

|  |
| --- |
| parentNode : 父节点 |
|  |

|  |
| --- |
| firstChild: 拿到第一个孩子 |
|  |
| 注意:空格换行也是一个元素 |

|  |
| --- |
| lastChild: 拿到第一个孩子 |
|  |

|  |
| --- |
| childNodes: 拿到孩子们 |
|  |

|  |
| --- |
| previousSibling: 上一个兄弟 |
|  |

|  |
| --- |
| nextSibling: 下一个兄弟 |
|  |

* 1. 不同类型的元素节点信息

nodeName，nodeValue，nodeType

|  |  |  |  |
| --- | --- | --- | --- |
|  | nodeName | nodeValue | nodeType |
| Element(元素) | 标签名称 | null | 1 |
| Attribute(属性) | 属性的名称 | 属性的值 | 2 |
| Text(文本) | #text | 文本的内容 | 3 |

* 1. 操作元素方法

hasChildNodes()，appendChild(node)，removeChild(node)

replaceChild(newnode, oldnode)，insertBefore(newnode, refnode)

createElement(“元素名称”)

|  |
| --- |
| **添加元素** appendChild(node) |
| 从内存中创建一个标签,追加到页面上 |
|  |
|  |
| 拿到页面上的某一个标签,放到另一个位置 |
| 注意:我们是把原来的标签拿过去(原来的标签就不存在了) |

|  |
| --- |
| 修改元素:replaceChild() |
|  |

|  |
| --- |
| 删除元素:removeChild() |
|  |

* 1. innerHTML与innerText

innerHTML,innerText: 修改与得到一段HTML标签中的内容

|  |
| --- |
|  |
|  |
| innerText:会把值原封不动显示(火狐中不支持)  innerHTML:会将值解析成HTML的格式显示 |

* 1. 设置和获取属性

如果我元素上有的属性,都可以通过

**元素对象.属性名** 或者 **元素对象["属性名"];**

|  |
| --- |
| **1.元素上的原始属性和自定义属性的获取值和设置值**  操作元素的原始属性:  获取属性值:  **元素对象.属性名;**  **元素对象["属性名"];**  设置属性值:  **元素对象.属性名=值;**  **元素对象["属性名"]=值;**    操作元素**自定义**属性:  获取属性值:  元素对象.getAttribute("属性名");  设置属性值:  元素对象.setAttribute("属性名",值); |

1. 事件与事件处理函数
   1. 什么是事件与事件处理函数

事件：是用户与计算机交互的方法。比如：单击、鼠标放上、双击等，当然有的事件是由对象发出,比如onload事件；

事件处理函数：就是JS中，当事件发生时，去调用函数。在事件前加“on”，比如：onclick、ondblclick(双击事件)、onchange(改变事件)

事件例子：

HTML 事件的例子：

当用户点击鼠标时

当网页已加载时 onload（文档加载完毕）

当网页被卸载时 onbeforeunload（文档即将从浏览器中卸载）

当图片已加载时

当鼠标移动到元素上时

当输入字段被改变时

当 HTML 表单被提交时

当用户触发按键时

* 1. 常用的事件

鼠标事件：

onclick: 点击事件

ondblclick: 双击事件

onmouseover: 鼠标进入“某对象区域”

onmouseout: 鼠标离开“某对象区域”

onmousedown: 鼠标按下

onmouseup: 鼠标抬起

onmousemove: 鼠标移动。

表单事件：

onsubmit： 表单提交事件

onfocus： 一个表单项获得焦点。

onblur: 一个表单项失去焦点

onchange: 一个表单项的内容的改变（通常用于select选项值的改变）

onreset: 表单重置事件

键盘事件：

onkeydown： 按键按下去（尚未抬起来）

onkeyup： 按钮抬起来。

onkeypress: 按键一次(不包含功能键，比如退格键)。

窗口事件：

onload: 网页一打开时发生——准确点说，是网页加载完毕时发生。

onload有两种用法：

第一种：

<body onload = “函数名()”>………

第二种：

<script>

window.onload = 函数名; //注意，这里没有引号和小括号

</script>

练习鼠标事件:



登录表单提示文字



* 1. 事件处理函数返回值（阻止浏览器的默认行为）

某些事件发生时，浏览器会自动执行默认的动作。而事件句柄的返回值会影响浏览器默认动作的执行。如果事件句柄不返回值，或者返回true，默认动作被执行；如果事件句柄返回false，则默认动作不发生。

**例如：当单击删除链接时**

<a href="http://www.sina.com.cn" onclick="return confirm('是否确认删除操作？')">删除商品</a>

**例如：提交表单时的验证**

