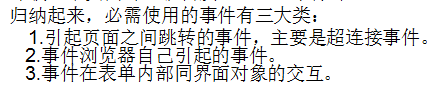
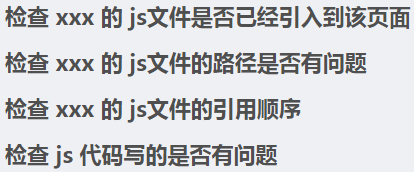
JS交互

事件是浏览器响应用户交互操作的一种机制，浏览器为了响应某个事件而进行的处理过程，叫做事件处理。



### Javascript 中Uncaught ReferenceError: xxx is not defined解决办法：



**在html中改变字体背景颜色：background-color**。

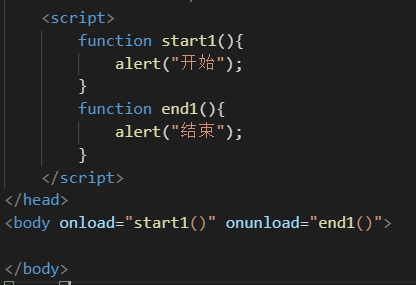
**在javascript中改变字体背景颜色：element.style.backgroundColor**。

**在html中表格的改变背景颜色可以用：bgcolor=” red”**,居中可用align=”center”。在html中改变表格的背景颜色也可用background-color，不过要用style=“background-color：red”,也可在css中的style样式中修改。直接用bgcolor不能改变文本如段落的背景颜色。

一、窗口事件：Window Events

**onload属性**：文档被载入时执行脚本(与某某取得联系)，其属性值为脚本。支持该事件的HTML标签：<body>、<frameset>、<frame>、<iframe>、<img>、<link>、<script>。属性值一般用函数调用。

**onunload属性**：文档被卸载时执行脚本(关闭资源)，其属性值为脚本。用户退出界面时发生。支持该事件的HTML标签：<body>、<frameset>。属性值一般用函数调用。



二、表单元素事件

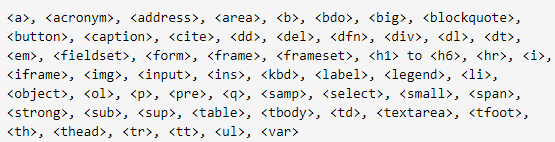
onchange：表元素修改时执行脚本。支持该事件的HTML标签：<input type=”text”>、<select>、<textarea>

**onsubmit**：当提交按钮被点击时，事件发生。支持该事件的HTML标签：<form>。onsubmit只用在表单中。

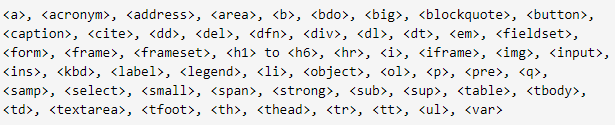
**onreset**：当重置按钮被点击时，事件发生。支持该事件的HTML标签：<form>

**onselect**：事件在文本框中的文本被选中时发生。支持该事件的HTML标签：<input type=”text”>、<textarea>。

**onblur**：事件在对象失去焦点时发生。支持的HTML标签如下：



**onfocus**：事件在对象获得焦点时发生。支持的HTML标签如下：





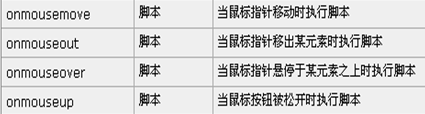


三、鼠标元素事件



**onclick**：支持该事件的HTML标签主要：<a>、<b>、<body>、<button>、<div>、<form>、<h1>to<h6>、<hr>、<i>、<img>、<input>、<li><、object>、<ol>、<p>、<pre>、<select>、<span>、<strong>、<small>、<table>、<td>、<textarea>、<th>、<tr>、<ul>、<var>。



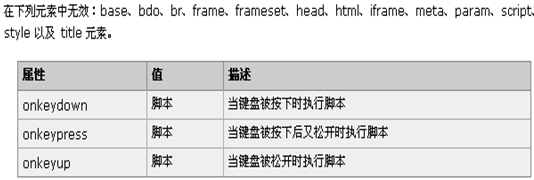


onmousedown：当鼠标按下时事件发生。

**onmouseover和onmouseout一起用的作用相当于:hover**的用法。



四、键盘元素事件

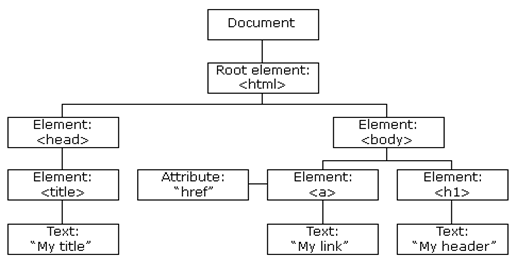




五、DOM：document object Model

DOM将整个HTML文档展现为内存中的一棵树状结构（浏览器内存中仅仅只有一棵这样的HTML DOM树，其**根节点为document**对象），每个元素、属性都是树上的一个节点。可以通过JavaScript来访问这棵DOM树，遍历树上的节点、动态添加、删除树上的节点、设置或修改某个节点的样式、设置或修改某个节点中保存的数值等等。通过JavaScript对于这棵DOM树所做的任何修改都会立即生效.

DOM节点图：



1.获得节点的方法：

①先通过Id值获得对象，再通过children获得子节点、通过parentNode获得父节点。注意每一个对象其子节点可以有多个，是数组类型。父节点有且仅有一个。

②直接通过class属性获得对象，获得的结果为数组类型，用数组元素表示就是所获得的节点。看所获得的对象是数组类型还是其他，可通过Elements看，Element有s表示多个，即为数组类型。

③直接通过标签名来获得对象，获得的结果为数组类型，用数组元素表示就是所获得的节点。



2.增加节点：

获得某一个标签内的文本：用**.innnerHTML**，获得的是字符串。xxx.innerHTML:表示xxx标签内的文本内容为.......。

document.creatElement()：创建节点对象

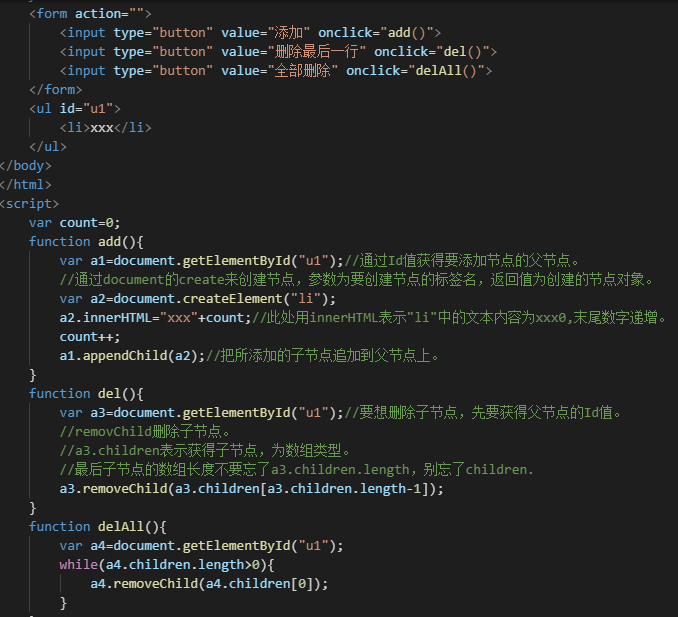
appendChild()：将节点对象添加到某个节点中

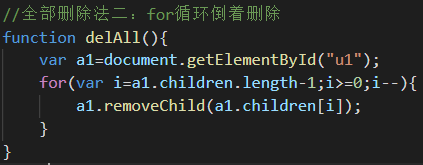


注意上面代码中获得添加结点的父结点不能用document.getElements.....类的，因为这得到的是一个集合类，不能直接用appendChild()方法。

3.删除节点：

removeChild()：删除子节点。注意这个Child的首字母是大写。子节点往往有多个，一般是数组类型，因此removeChild()括号内填数组类型的数据，如a3.children[a3.children.length-1]。



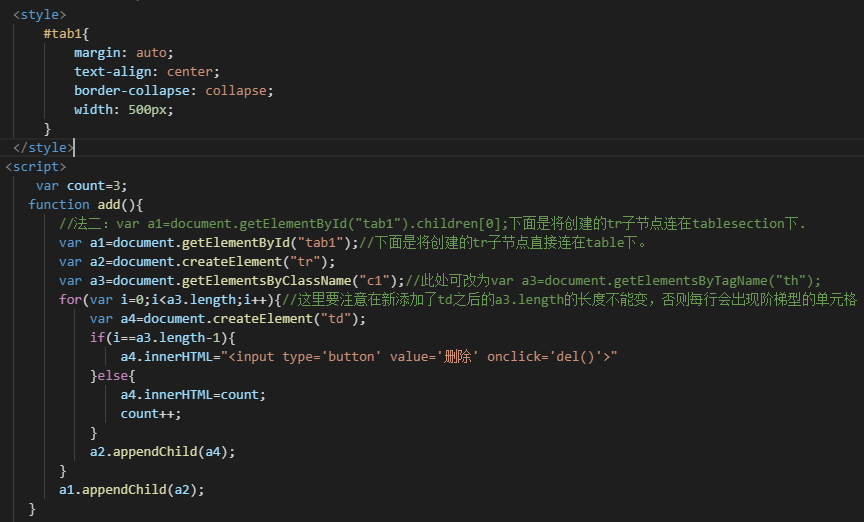


**特别注意自己在写children时老是写成chidren**，

注意table中的父节点与子节点之间的顺序：**Document-->HTML-->Body-->Table-->Table section-->Tabel row**

**-->Table cell(单元格)(td和th都是属于Table cell)**。

增删表格的行

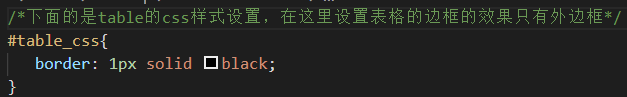






上面parentNode还没有弄明白：即input->form->td->tr之间的关系。



六、时间事件：

setTimeout(function,milliseconds)：在等待了指定的毫秒后开始执行函数。第一个参数为函数，第二个参数为毫秒。



clearTimeout()：停止执行setTimeout()中执行的函数。其括号里的参数为从setTimeout中返回的变量。如：



setInterval(function,milliseconds)：在等待了指定的毫秒后(间隔了指定的时间后)，开始重复执行函数。即按照指定的周期（以毫秒计）来调用函数或计算表达式

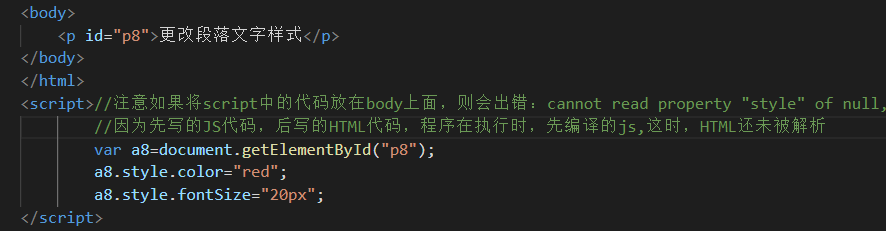
clearInterval()：停止执行setInterval()中执行的函数，括号内的参数为setInterval()的返回值。



七、JavaScript中修改HTML中的样式：

语法为：document.getElementById(id).style.property=new style。

更改图片的src：document.getElementById(id).src



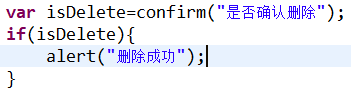


八、Window对象

widow对象在<body>或<framset>出现时自动创建，代表一个窗口或一个框架。

**alert()**：弹窗，window.alert()中的window可以省略。

window**.confirm()**：确认框，结果为布尔类型。如果点击确认，则为true，点击取消则为false。可加一个alert弹窗查看布尔值。



window**.prompt()**：提示用户输入的对话框，点确定，返回值为输入的内容，点取消，返回值为null。

window**.open()**：在新的标签页打开页面。

window.open()：在新标签打开页面

window**.location.href=“”**：在当前页面跳转页面

document**.getElementsByName()**、document**.getElementById()**、document**.getElementByTagName()**、document**.getElementsByClass()**在使用时要考虑用哪个更方便一点，灵活使用这几个方法。



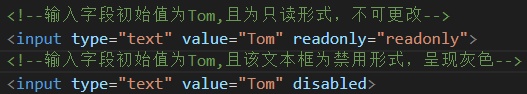
九、form表单

form表单的reset()、submit()函数。注意这两个函数都是form表单的，要获取id值都是form的id值。

input的属性：value属性、readonly属性、disabled属性。其中：value属性规定输入字段的初始值，readonly规定输入字段为只读，不能修改，disabled属性规定输入字段是禁用的。readonly=“readonly”，disabled=“disabled”。

注意：设置readonly属性时，在javascript中O要大写，而在html标签中o不用大写。如：

(在script中设置readonly属性)





setAttribute()：添加指定的属性，并为其赋指定的值，其语法为：，注意括号内参数属性名和属性值都是string形式，即字符串形式，即都用引号括起。

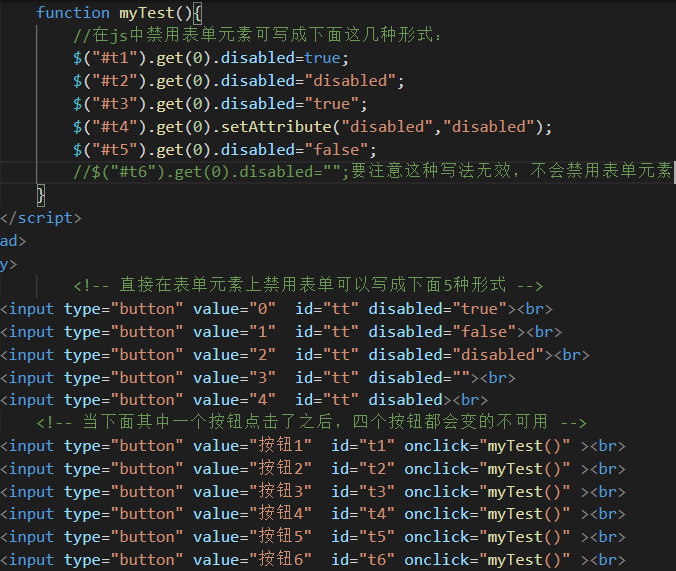
removeAttribute()：删除指定的属性，其语法为：

，参数为string形式。

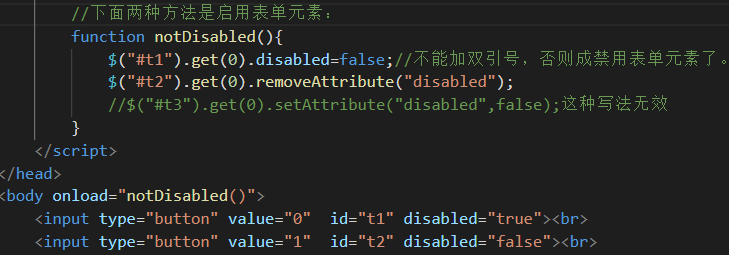


注意disabled没有true或者false属性。以下是将表单元素禁用、启用的几种方式：

**禁用表单元素**：



**启用表单元素**：



十、补充内容

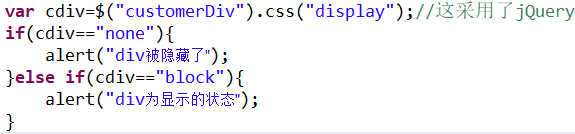
在javascipt中显示或隐藏div用display。

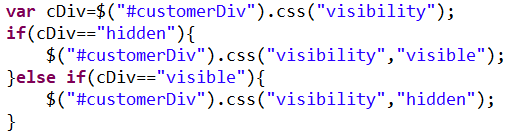
element.style.display=“block”：显示div

element.style.display=“none”：隐藏div

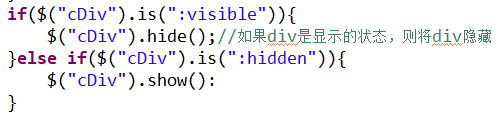
**判断div是否是隐藏或者显示的状态**：

**法一**：使用css属性和jquery结合





法二：使用jQuery的方法



JavaScript中：.value用在form表单的属性中，.innerHTML用在其它地方要在文本中添加内容中。如：要更改form表单中input输入中value的内容时如value=“按行输入”，改为 “完成 “时，用Object.value=“完成”。form表单中只要是<input>中文本框中的内容都用value表示。如下图所示：只要是要获得或者改变框内的“文本框中的内容”和按钮中的“删除”文字都要用value。



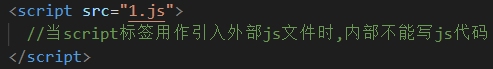
**获取元素的值或者内容：**

**innerHTML：起始标记和结束标记之间的内容**

**value ：表单元素的内容**

注意：当用Object.innerHTML或者Object.value给另一个赋值时，若出现undefined这种情况，可以考虑一下Object是不是数组，是不是要改成Object[i].value等情况。

JavaScript的引用：



CSS的样式：



十一、表格

tableObj**.insertRow(index)**：为表格添加一行，参数为添加位置的索引号，-1代表末尾添加，返回值为tr对象。语法：

tableObj**.insertCell(index)**：为表格添加空的单元格，在行里面添加单元格。注意只能添加<td>，若要添加<th>则要用document.createElement(“th”)。其语法为：

。

tableObj **.deleteRow(index)**：删除行，-1代表末尾删除。语法为：

。

tableObj**.deleteCell(index)**：删除表格中行的单元格。语法：。

**rowIndex**属性：返回某一行在表格行中的索引位置。语法为：



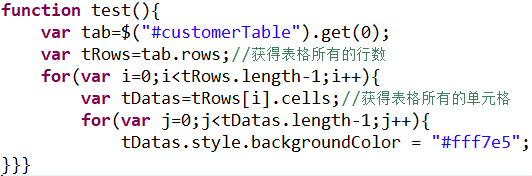
**rows**集合：返回表格中所有行的一个数组，语法为：

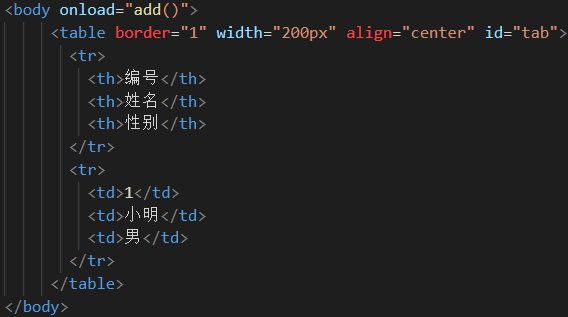
。**查看表格的某行：tableObject.rows[index]。**

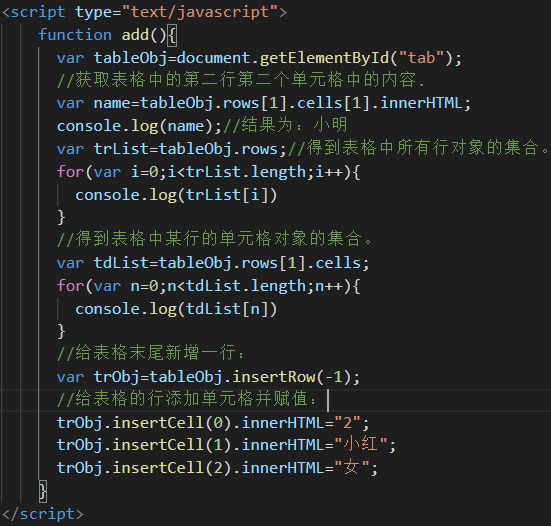
**查看所有行的长度：tableObject.rows.length。**

**cells**集合：返回表格中所有行的单元格的数组。查看表格中某行的内的某个单元格：tableObject.rows[index1].cells[index2]。查看某行的单元格长度：tableObject.rows[index1].cells.length。

注意用insertRow(index)和insertCell(index)时，不用再写appendChild。







**下面这段代码好好学习：getElementsByTagName()、getElementsByName()的特殊使用。**



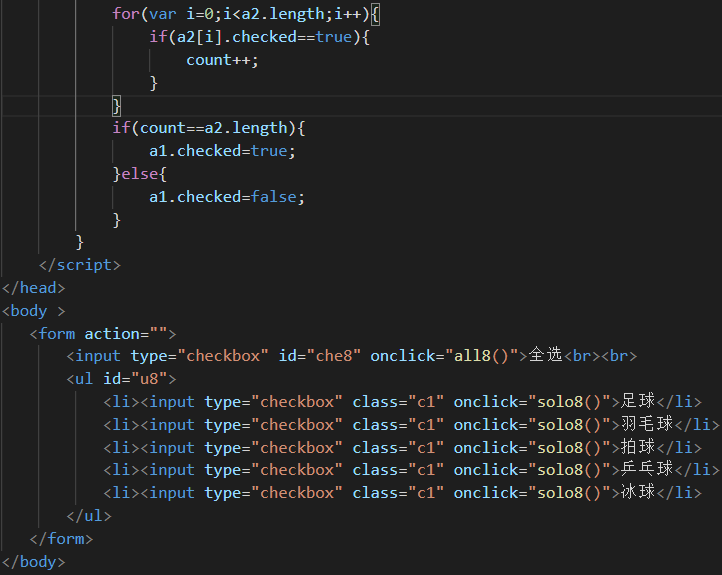


十二、复选框

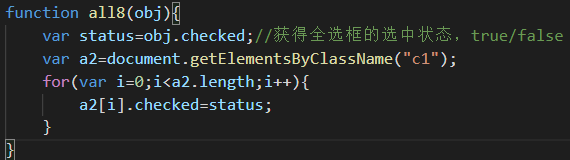
javascript中checkbox属性是否被选中，其语法为：checkboxObject**.checked**=true(或者false)。

无序列表去掉圆圈：list-style:none；





上面的function all8()函数可以改为下面这种更简洁：



Javascipt中好像不能用break语句跳出if语句。

十三、控制台

**console.log()**：在页面控制台上输出内容。

**console.error()**：在页面控制台上输出错误内容。

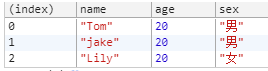
**console.warn()**：在页面控制台上输出警告内容。

**console.info(**)：输出提示信息。



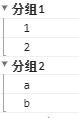
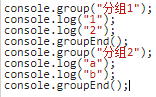
console.table()：将数据表格化。





**统计代码的执行时间**：在代码开头写console.time()，代码结尾写console.timeEnd()。结果为毫秒。括号中无参数。

**分组输出信息**：分组开始处console.group(String args)、分组结尾处console.groupEnd()。



**console.count()**：统计代码执行的次数。



**console支持的占位符**：**%s(字符)、%d或%i(整数)、%f(浮点数)、%o(对象**)。

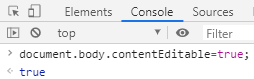




**清空控制台**：下面这个按钮，点击即可清空控制台上的内容。



**把页面变成可以直接编辑**：有时候我们需要临时改变页面上的文字、图案等信息，一种常见的方法是打开Elememts，然后在修改页面的源码，这种做法比较麻烦。可以通过Chrome的控制台Console，在控制台上输入**document.body.contentEditable=true**，回车之后即可直接修改页面上的文字等信息。

这是可直接修改页面上的文字等信息。

通过F12或右键点击检查进入页面的开发者模式，不能显示全部菜单的功能，在Mac下通过快捷键Cmd + Shift + P，在Windows,Linux下通过**Ctrl + Shift + P**，可以打开所有的菜单选项，然后点击选择快速进入到对应的菜单中。

十四、报错

当在执行js代码时，报错：Uncaught SyntaxError: Unexpected token xxx时，可能有以下原因：

①js的引入时，路径没写对。

②执行了空方法体，如定义了一个onclick=“test()”，这个test()方法是一个空方法体。

③<script></script>放置的位置不对，因为js放置在不同地方，它的执行顺序是不同的，也就是加载顺序不同。

④js中使用El表达式时，没写对。

如：一开始加载页面时，作用域中没有isDelete的数据，因此若将下面的el表达式去掉引号写成这样：，则页面一加载时，结果会变成这样：**var del=;**。等号后面直接跟着一个分号，这样就会报语法错误。因此El中获得的数据是字符串则一定要加引号。下面加了引号，即使isDelete中没有获得数据，结果也是**var del=””;**空字符串。

