DOM  
常用  
元素  
文本  
属性  
节点的属性  
- console.log(div.nodeName);  
- console.log(div.nodeValue;  
-  console.log(div.nodeType);  
层次节点  
父节点  
子节点、兄弟节点  
获取所有子节点  
childNodes  
获取第一个和最后一个子节点  
firstChild  
lastChild  
获取父节点  
parentNode  
获取兄弟节点  
previousSibling  
nextSiblin  
元素的宽高属性  
- offsetWidth  
  
- 它包含了内边距和边框  
  
- offseHeight  
  
- 它包含了内边距和边框  
  
- clientHeight  
  
- 不包含边框  
  
- clientWidth  
  
- 不包含边框  
子元素与父元素的距离  
offsetLefte  
  
body左边界和上边界的距离使用了定位属性,则offsetLefte和offsetTop参照的就不再是body而是距离它最近的父级元素  
offsetTop  
  
获取元素  
用id获取元素  
  
   getElementById()  
用标签名获取元素  
  
getElementsByTagName()  
用类名获取元素  
  
getElementsByClassName()  
用选择器获取元素  
  
querySelector()  
querySelectorAll()  
获取和设置元素其他信息  
获取元素名  
  
               tagName      
只能获取不能设置  
获取元素节点里的内容  
  
innderHTML  
  
可以获取标签内的属性，还可以设置标签内的内容  
获取元素节点中的文本  
  
innerText  
  
除了获取,也可以设置元素中的文本.  
获取元素的类名  
  
className  
  
返回值是一个字符串  
获取元素样式  
  
style  
  
background-color  
获取元素属性  
  
getAttribute（）  
设置元素属性  
  
setAttribute（）  
删除元素属性、  
  
  - removeAttribute   
查找元素  
创建元素节点  
  
document.createElement()  
创建文本节点  
  
document.createTextNode(&apos;一段文字&apos;  
css样式赋予  
  
style.cssText  
在某元素前,插入创建的新元素.  
  
insertBefore()  
删除和替换元素  
元素的替换  
  
replaceChild()  
删除节点  
  
removeChild()  
DOM的概述  
DOM的概念  
DOM文档对象模型，DOM对象不仅仅是一个普通的内置对象，它还是一个巨大的API的核心对象，它将文档呈现在js面前，并赋予了js操作文档能力，  
DOM和JavaScript的关系  
一个web页面是一个文档,这文档可以在浏览器窗口或者作为html源代码显示出来.但上述两种情况都是同一个文档,DOM提供了对同一份文档的另一种表现,存储和操作的方式.DOM能够使用JavaScript脚本语言进行修改.  
6.1.3DOM节点  
  
DOM节点  
整个文档是一个文档节点  
每个html元素是元素节点  
HTML元素内的文本是文本节点  
每个html属性都是属性节点  
注释也是节点，叫注释节点  
事件绑定  
鼠标和键盘 事件  
事件种类  
  
鼠标事件  
  
鼠标移入  
  
会触发冒泡或者说（从内逐级向外依次触发）会触发多次取决于嵌套和鼠标为准  
  
onmouseover  
  
onmouseenter  
  
不会触发多次  
鼠标移出  
  
会触发冒泡或者说（从内逐级向外依次触发） 会触发多次取决于嵌套和鼠标为准  
  
onmouseout  
  
onmouseleave  
  
不会触发多次  
键盘事件  
  
键盘的按钮按下  
  
- onkeydown  
  
-   
键盘的按钮抬起  
  
- onkeyup  
Event事件对象  
Event事件对象  
  
键盘案件的状态  
鼠标的位置  
鼠标按钮  
Event事件对象的方式  
  
通过事件触发式的函数以参数的形式传递该函数内  
event 对象常用属性  
  
鼠标的坐标值：  
  
e.clientX和e.clientY  
键盘的值  
  
e.keyCode  
对象事件  
页面加载完成后执行事件  
  
window.onload  
窗口变化大小时触发事件  
  
indow.onresize  
表单事件  
获取焦点  
  
onfocus  
失去焦点  
  
onblur  
用户输入触发  
  
oninput  
内容改变时触发  
  
onchange  
事件绑定  
事件绑定概念  
  
希望运用js让用户操作dom元素，先要让dom元素处理一个事件函数，将这个函数绑定在该dom元素上  
事件绑定的方法  
  
子主题 1  
  
<button id="btn2"><button>  
<script>  
     var btn1=document.getElementById(&apos;btn1&apos;);  
     var btn2=document.getElementById(&apos;btn2&apos;);  
    //1种  
       function a(){  
           alert(&apos;按钮1&apos;)  
       }  
       //第二种  
      btn2.onclick= function a(){  
           alert(&apos;按钮2&apos;)  
       }  
</script>  
解除绑定方法  
  
子主题 1  
  
 <button id="btn2"><button>  
     <script>  
     var btn2=document.getElementById(&apos;btn2&apos;);  
      btn2.onclick= function a(){  
           alert(&apos;按钮2&apos;)  
       }  
      //解除绑定方法  
      btn2.onclick=null;  
</script>  
事件监听  
事件监听的方法  
  
方法有三个参数(监听事件,方法,冒泡),冒泡false,监听事件不用加on  
  
监听有兼容性问题,IE的监听需要addEvent(&apos;onclick&apos;,c)第一个参数为监听的事件,需要加on,第二个参数是触发的事件函数.  
解绑和封装  
  
子主题 1  
  
<script>  
  var  mydiv=document.getElementById(&apos;mydiv&apos;)  
              //封装监听  
    function addE(target,type,fn){  
        if(target,addEventListener){  
            target.addEventListener(type,fn);  
        }else {  
            target.attachEvent(&apos;on&apos;+type,&apos;fn&apos;);  
        }  
    }  
    addE(mydiv,&apos;click&apos;,a);  
    //封装解绑  
    function removeE(target,type,fn){  
        if(target,removeEventListener){  
            target.removeEventListener(type,fn);  
        }else {  
            target.datachEvent(&apos;on&apos;+type,&apos;fn&apos;);  
        }  
    }  
</script>