1. 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,他将文档的内容呈现在就是面前,并赋予了js操作文档的能力.

DOM与JavaScript的关系

一个web页面是一个文档,该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JavaScript脚本语言进行修改.

DOM节点

加载html页面时,web浏览器生成一个树形结构,用来表示页面的内部结构.DOM将这种结构理解为节点结构,根据w3c的html DOM标准,htnl找那个的所有内容都是节点.

> 整个文档是一个节点是一个文档节点

> 每个html元素是元素节点

> html元素内的文本是文本节点 空格也是一个文本节点

> 每个html属性都是属性节点

> 注释也是节点,叫注释节点

DOM树

DOM数体现着html页面的层级结构,而DOM树有DOM文档数和DOM元素树两种,DOM元素树包含的只有元素节点,而DOM文档数则包括DOM文档中的所有内容.

1. 获取元素

<1>用id获取元素名

getElementById()的方法,接受一个参数:获取元素的id.如果找到相应的元素,则返回该元素,否则反回null.

var d = document.getElementById('d')

<2>用标签名获取元素

getElementsByTagName();可以获取该元素名下所有的元素,返回一个伪数组,或者说是一个节点列表.

var lis = document.getElementsByTagName('li');

console.log(lis);

<3>用类名获取元素

var a = document.getElementsByClassName('a');

console.log(a);

<4>用name属性值获取元素

getElementsByName()方法可以获取相同名称的name元素,返回一个伪数组对象.

<input type="radio" name="sex" value="0">男

<input type="radio" name="sex" value="1">女

var sex = document.getElementsByName('sex');

console.log(sex);

<5>用选择器获取元素

querySelector()和querySelectorAll()可以依靠选择器找到元素,但是前者只能找到元素列表的第一个元素,而后者可以全部找到.注意,该方法性能没有直接利用标签寻找高.

var ul = document.getElementById('ul')

var x = document.querySelector('#div>div a');

var lis = ul.querySelectorAll('li');

console.log(lis);

1. 获取和设置元素中的其他信息

<1>获取元素名

当我们使用id获取元素名的时候,元素的名称会和整个元素一起显示出来,如果想要拿到该元素的元素名,也就是标签名则需要用tagName属性.该属性只能获取,不能设置.

var div = document.getElementById('mydiv');

console.log(div.tagName);

<2>获取元素的节点里的内容

当获取了元素之后,如果我们需要拿到元素中的内容(所有东西),则需要另一种方法得到,元素里的内容可能包括:文本或元素.需要用innerHTML属性.

var div = document.getElementById('mydiv');

console.log(div.innerHTML);

<3> 获取元素节点中的文本

利用inner Text属性获取的只能是文本节点,不能是其他,当然,innerText属性除了获取,也可以设置元素中的文本.

var div = document.getElementById('mydiv');

console.log(div.innerText);

div.innerText = 'hello';

<4>获取元素类名

利用className属性获取元素的类名,以字符串的形式返回.同时可以设置新的class类名,需要注意的是,返回值是一个字符串,如果更换其中一个类,则需要考虑该字符串中其他类是否应该一起设置.

var div = document.getElementsById('mydiv');

console.log (div.className);

<5> 获取元素样式

style属性可以获取元素内联样式的所有属性,当然如果在style中找属性名如:backgroundColor,就可以得到该属性的值,一字符串的形式返回,同时也可以设置该属性,从而达到增加或更改样式.需要注意的是,以js形式增加的样式的优先级大于css优先级.

<div style="background-color:blue;"></div>

<script>

var div = document.getElementsByTagName('div');

console.log(div[0].style.backgroundcolor);

</script>

var div =document.getElementsById('mydiv');

<6> 获取元素的属性

可以获取元素的某个属性,要将属性名称放在括号中,记得要用引号括起来,返回值就是字符串形式的属性值

var a =document.getElementsByTagName('a');

console.log(div.getAttribute('data-lj'))

console.log(a[0].getAttribute('herf'))

<7>设置元素属性

setAttribut() 可以设置元素的某种属性，第一个参数是属性名，第二个参数是属性值，都需要用引号括起来用逗号隔开

var div =document.getElementsByTagName('div');

var a =document.getElementsByTagName('a');

div[0].setAttribute('id','mydiv');

a[0].setAttribute('herf','http://www.baidu.com');

<8>删除元素属性

使用removeAttribute（），可以删除某个元素的某个属性，括号中放入需要删除的属性名，用引号包裹.

var div = document.getElementById('mydiv')

div.removeAttribute('id');