DOM高级编程

2019年8月21日

10:24

一、DOM(Document Object Model)文档对象模型

可以将HTML理解为一颗DOM树，文档里的所有标签及属性都是DOM树的节点。

DOM编程：利用JS改变树的结构，动态地为整个文档增加或删除节点，或修改某个节点。

<html></html>是根节点。

二、DOM编程

1、查看节点

访问指定节点的方法：

getElementById():返回一个节点对象

getElementByTagName():返回多个节点对象(节点数组)

getElementByName:返回多个节点对象(节点数组)

查看节点属性：

getAttribute("属性名");

img.getAttribute("src");相对路径

img.src;绝对路径

修改节点属性:

setAttribute("属性名","属性值");

img.setAttribute("src","img/img1");

或 img.src = "img/img1";

if(img1.style.display=="block"){

img2.style.display="block";

img1.style.display="none";

}

2、创建和增加节点的方法

createElement("标签")；创建节点

cloneNode()；克隆节点

appendChild(插入节点)；添加到子节点

insertBefore(插入节点1，节点2)；在指定节点前添加//末尾式插入

function addImg(){

//找到旧节点

var img1=document.getElementById("img1");

//创建图片节点

var img=document.createElement("img");

//设置属性

img.setAttribute("src","img/img2.png");

//插入节点

//document.body.appendChild(img);

document.body.insertBefore(img,img1);

}

function clone(){

//找到要克隆的节点

var oldNode=document.getElementById("img1");

var newNOde=oldNode.cloneNode();

document.body.appendChild(newNOde);

}

3、删除和替换节点

removeChild(old):删除节点

replaceChild(new, old):替换节点

function deleteImg(){

//找到图片

var img=document.getElementById("img1");

//删除图片

document.body.removeChild(img);

}

function replaceImg(){

//找到要替换的节点

var oldNode=document.getElementById("img1");

var newNode=document.createElement("img");//创建新节点

newNode.src="img/img2.png";

//替换节点

document.body.replaceChild(newNode,oldNode)

}

三、操作table对象

table对象：

代表一个HTML表格

属性：

rows(返回行数组） cells（返回单元格数组）

方法：

insertRow(index):在指定位置处插入行

deleteRow(index):删除指定位置处的行

tableRow对象：

代表一个表格行

属性：

cells：返回单元格数组

rowIndex

innerHTML

方法：

insertCell(index)

deleteCell(index)

* tableCell对象：

代表一个单元格

属性：

innerHTML、cellIndex、colspan、rowspan

function addRow(){

//找到table

var myTable=document.getElementById("myTable");

//插入行

var newRow=myTable.insertRow(2);

//插入单元格1

var cell1=newRow.insertCell(0);

//给单元格添加内容

cell1.innerHTML="幸福从天而降";

//插入单元格2

var cell2 = newRow.insertCell(1);

cell2.innerHTML="￥43.3";

}

//删除第二行

function delRow(){

var myTable=document.getElementById("myTable");

myTable.deleteRow(2);

}

//

function updateTitle(){

//找到table

var myTable=document.getElementById("myTable");

//找到第一行

var firstRow=myTable.rows[0];

//设置第一行的类样式为title

firstRow.className="title";

}

**获取节点的某个兄弟节点**

tr.forEach(node => {

//node为元节点之一

const nodes = node.parentNode.childNodes;

for(let i = 0; i < nodes.length; i++){

if(nodes[i] === node){

console.log(nodes[i+2]);//通过和元节点进行判断,获取元节点后的第二个节点

}

}

});

**HTML宽高获取**

**// 首先是取到元素**

**var main = document.getElementById('main');**

**// 第一种方式**

**var mainWidth = main.offsetWidth,**

**mainHeight = main.offsetHeight;**

**// 第二种方式**

**var mainWidth2 = main.style.width,**

**mainHeight2 = main.style.height;**

**元素尺寸和位置获取**

**获取元素的border + padding + 内容 宽高 :**

element.offsetHeight element.offsetWidth

**获取元素的滚动容器宽高(滚动条会占用padding和content宽高)**

element.scrollHeight element.scrollWidth

**获取元素的border和padding宽高**

// 考虑 IE 的兼容性

function getStyle(el) {

if(window.getComputedStyle) {

return window.getComputedStyle(el, null);

}else{

return el.currentStyle;

}

}

parseFloat(getStyle[element]['paddingLeft']) //获取左内边距

parseFloat(getStyle[element]['borderLeftWidth']) //获取左边框宽度

getStyle[element]['borderTopColor'] //获取上边框颜色

getStyle[element]['borderTopStyle'] //获取上边框样式

**获取元素的位置**

**element.getBoundingClientRect()** //返回一个对象,包括元素的位置的大小

{

x : 450 //距离左窗口位置

y: 150 //距离上窗口位置

bottom:108 //相对父元素底边框

height:100 //元素高,等同于element.offsetHeight

left:13 //相对父元素左边框

right:113 //相对父元素右边框

top:8 //相对父元素上边框

width:100 //元素宽,等同于element.offsetWidth

}

**注意:**

JS触发事件不能使select下拉菜单打开

监听DOM元素变化

