### 获取节点

在进行增、删、改的操作时，都需要指定到一个位置，或者找到一个目标，此时我们就可以通过Document对象提供的方法，查找、定位某个对象（也就是我们说的节点）。

注意：操作 dom 必须等节点初始化完毕后，才能执行。处理方式两种:

a. 把 script 调用标签移到html末尾即可；

b. 使用onload事件来处理JS，等待html 加载完毕再加载 onload 事件里的 JS。

window.onload = function() {

// 预加载html后执行

};

获取节点的四种方式

|  |  |
| --- | --- |
| **方法** | **描述** |
| getElementById() | 根据id获取dom对象，如果id重复，那么以第一个为准 |
| getElementsByTagName() | 根据标签名获取dom对象数组 |
| getElementsByClassName() | 根据样式名获取dom对象数组 |
| getElementsByName() | 根据name属性值获取dom对象数组，常用于多选获取值 |

### 创建节点和插入节点

很多时候我们想要在某个位置插入一个新的节点，此时我们首先需要有一个节点存在，可以通过以下几种方式创建新节点。

**创建节点**

|  |  |
| --- | --- |
| **方法** | **描述** |
| createElement() | 创建一个新的节点，需要传入节点的标签名称，返回创建的元素对象 |
| createTextNode() | 创建一个文本节点，可以传入文本内容 |
| innerHTML | 也能达到创建节点的效果，直接添加到指定位置了 |

**插入节点**

|  |  |
| --- | --- |
| **方法** | **描述** |
| write() | 将任意的字符串插入到文档中 |
| appendChild() | 向元素中添加新的子节点，作为最后一个子节点 |
| insertBefore() | 向指定的已有的节点之前插入新的节点  newItem：要插入的节点  exsitingItem：参考节点  需要参考父节点 |

### 间接查找节点

|  |  |
| --- | --- |
| **方法|属性** | **描述** |
| childNodes | 返回元素的一个子节点的数组 |
| firstChild | 返回元素的第一个子节点 |
| lastChild | 返回元素的最后一个子节点 |
| nextSibling | 返回元素的下一个兄弟节点 |
| parentNode | 返回元素的父节点 |
| previousSibling | 返回元素的上一个兄弟节点 |

### 替换克隆和删除节点

|  |  |
| --- | --- |
| **方法|属性** | **描述** |
| replaceChild(newNode, oldNode) | 用新的节点替换旧的节点 |
| cloneNode() | 复制节点 |
| removeChild() | 从元素中移除子节点 |

### 属性操作

在操作DOM对象时，除了可以操作对象整体之外，还可以更加灵活地操作对象中的各个属性。

|  |  |
| --- | --- |
| **方法|属性** | **描述** |
| getAttribute() | 返回指定元素的属性值 |
| getAttributeNode() | 返回指定属性节点 |
| element.id | 设置或者返回元素的 id |
| setAttribute() | 设置或者改变指定属性并指定值 |
| setAttributeNode() | 设置或者改变指定属性节点 |
| element.style | 设置或返回元素的样式属性 |
| element.className | 设置或返回元素的class属性 |
| element.classList | 返回元素的类名 |

案例:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style>

ul{

width:500px;

height:500px;

border:1px dashed deeppink;

}

input{

border:1px solid deeppink;

height:20px;

同行的行内元素的上下的对其方式

vertical-align:top;

}

input:nth-child(2){

background: deeppink;

color: #fff;

height:24px;

vertical-align:top;

}

</style>

</head>

<body>

<input type="text" name="text" id="text">

<input type="button" id="b" value="提交"><span id="span"></span>

<ul id="ul"></ul>

<script>

1.根据关键字取值

var b=document.getElementById("b")

var text=document.getElementById("text")

var span=document.getElementById("span")

var ul=document.getElementById("ul")

2.定义text框内的判断条件

function fn(){

3.框内值赋给str

var str=text.value;

4.判断值是否为空,如果为空提示不允许为空,如果不为空,把input置为空

if(str==""){

5.在span标签内用innerHTML赋值语句

span.innerHTML="不能为空";

6.return作用:条件成立则跳出判断

return;

}

7.条件不成立则框内值赋为空即清空(用于控制框内输入内容提交后自动清空框内内容)

else{

text.value="";

}

8.创建li标签节点

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

9.把str的值赋给li节点

li.innerHTML=str;

10.把li节点插入到ul中并首行置顶

ul.insertBefore(li,ul.firstElementChild);

}

11.设置提交按钮的点击功能

b.onclick=fn;

12.设置框内聚焦span设置为空(用于控制点击框内时原先不允许为空提示清空)

text.onfocus=function(){

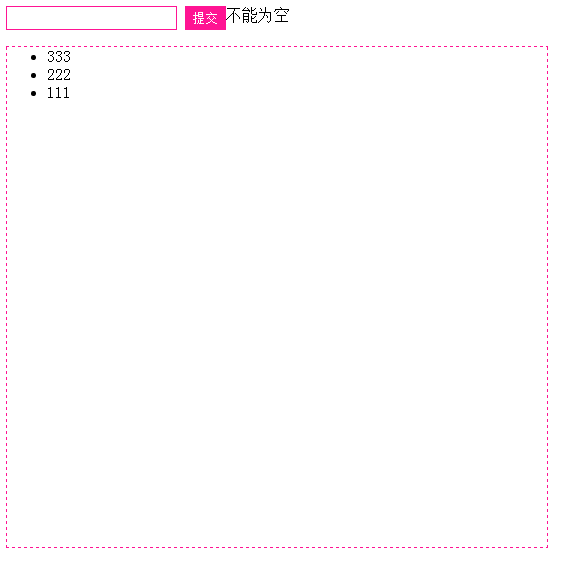
span.innerHTML="";

};

</script>

</body>

</html>



点击子节点隐藏父节点.

思路:

可以先从一个做起成功后再添加循环遍历

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<span id="span">span1</span>

<script>

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

span.onclick=function(){

用style.display="none"来隐藏span1

span.style.display="none";

}

</script>

</body>

</html>

接下来开始添加ui和li

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<ui>

<li><span>span1</span></li>

<li><span>span2</span></li>

<li><span>span3</span></li>

<li><span>span4</span></li>

<li><span>span5</span></li>

</ui>

<script>

1.获取到所有的span和li

此处用.getElementsByTagName获取数组(与上面测试区别开)

var lis = document.getElementsByTagName("li")

var spans = document.getElementsByTagName("span")

2.遍历数组

for(var i=0;i<spans.length;i++){

3. 为spans添加一个点击事件

spans[i].onclick=function(){

4.用this确定span位置然后找到当前被点击的span的父级li最后为li标签设置display:none

console.log(this); 此句用于测试找到 被点击span的对象

this.parentNode.style.display="none";

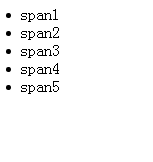
}

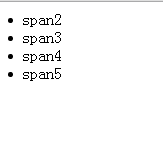
}

</script>

</body>

</html>





第二种方法

for(var i=0;i<spans.length;i++){

//给每一个span自定义一个属性 index,存储的是这个span的索引-->对应着这个span的父节点li在lis中的索引

spans[i].index=i;

console.log(spans[i].index);

spans[i].onclick=function(){

console.log(this.index);

//3.找到当前被点击的这个span

lis[this.index].style.display="none";

}

}

实现每隔一秒刷新时间

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style>

div{

width: 500px;

height: 30px;

text-align: center;

background-color: black;

color: white;

font-size: 30px;

}

</style>

</head>

<body >

<div id="time"><span></span></div>

<script>

1.设置重复执行定时器

setInterval(function() {

2.创建当前时间

var d = new Date();

3. 从 Date 对象返回年月周日等的值

var year = d.getFullYear();

4.因为月份是从0开始计算的所以需要加一

var month = d.getMonth() + 1;

var day = d.getDay();

var dd = d.getDate();

var h = d.getHours();

var m = d.getMinutes();

var s = d.getSeconds();

5.设定输出的内容

document.getElementById("time").innerHTML = year + "年" + month + "月" + dd + "日 " + "星期" + day + " " + h + "点" + m + "分" + s + "秒";

6.1000单位毫秒即一秒

},1000);

</script>

</body>

</html>





用js创建插入图片标签(两种方法)

第一种方法

1.创建图片

var img=new Image();

2.可以理解为src为img的内置属性系统中默认存在的所以可以用img.src来调用

img.src="img/2.jpg";

3.将图片插入到body中

document.body.appendChild(img);

第二种方法

1.创建图片标签

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

2.用setAttribute 方法添加Attribute属性

img.setAttribute("src","img/2.jpg");

3.将图片插入到body中

document.body.appendChild(img);