**Dom树**

Dom是根据html文档生成

Dom树是一个对象，其对象名为 document，是由element组成的树，



我们通过操作dom来操作html文档

**查找元素element**

想要操作某个元素（html标签），就必须要先找到

// 根据 id 查找元素

var x=document.getElementById("intro");

// 根据标签名查找元素

var x=document.getElementById("main");

var y=x.getElementsByTagName("p");

// 根据类名查找元素

var x=document.getElementsByClassName("intro");

**改变元素样式**

我们通过修改element的style对象来修改元素的样式

document.getElementById(id).style.property = newproperty

**Dom事件**

## 设置元素事件处理函数

我们可以通过如下2种方式设置元素的处理函数

在html标签种设置：

<button onclick="displayDate()">Try it</button>

通过dom设置

<script>

document.getElementById("myBtn").onclick=function(){displayDate()};

</script>

## onload 和 onunload 事件

onload 和 onunload 事件会在用户进入或离开页面时被触发。

<body onload="checkCookies()">

## onchange 事件

onchange在输入框的值发生改变时触发

<input type="text" id="fname" onchange="upperCase()">

## onmouseover 和 onmouseout 事件

onmouseover 和 onmouseout 事件可用于在用户的鼠标移至 HTML 元素上方或移出元素时触发函数。

## onmousedown、onmouseup 以及 onclick 事件

首先当点击鼠标按钮时，会触发 onmousedown 事件，当释放鼠标按钮时，会触发 onmouseup 事件，最后，当完成鼠标点击时，会触发 onclick 事件。

**Js生成事件和分发**

Js事件是由页面触发的，但js也可以触发事件

<script>

let div = document.createElement("div");

document.body.appendChild(div);

let event = document.createEvent("MouseEvent"); // 事件类型

event.initEvent(

"mouseover", // 鼠标事件类型

true // 是否冒泡

)

div.dispatchEvent(event)

</script>

**事件监听器EventListener**

我们可以通过设置元素的属性来设置事件处理函数，我们也可以通过设置元素的EventListener来设置事件处理函数

// 向元素 myBtn 添加 click 事件的监听器，处理函数为 displayDate

document.getElementById("myBtn").addEventListener("click", displayDate);

// 我们可以添加多个相同类型的监听器，不会覆盖

// 第 3 个参数为 true 时，表示捕获传递，即父元素的处理函数先运行，否则为冒泡传递，即子元素的处理函数先运行

document.getElementById("myBtn").addEventListener("click", displayNum, true);

// 移除监听器

document.getElementById("myBtn").removeEventListener("click", displayDate);

**创建和删除元素**

**创建元素**

// 创建元素 p

var para=document.createElement("p");

// 创建文本

var node=document.createTextNode("This is new.");

// 将文本追加到 p 元素

para.appendChild(node);

var element=document.getElementById("div1");

// 将 p 元素追加到 div1

element.appendChild(para);

**删除元素**

var parent=document.getElementById("div1");

var child=document.getElementById("p1");

// 删除 div1 下的 p1

parent.removeChild(child);

**改变 HTML 内容**

我们可以修改某个标签的html文本

document.getElementById(id).innerHTML="new HTML"

**改变 HTML 属性**

如需改变 HTML 元素的属性，请使用这个语法：

document.getElementById(id).attribute=new value