



做真实的自己，用良心做教育

H5教学部

事件下

目录

做真实的自己，用良心做教育

1

阻止事件冒泡和默认行为

2

拖拽

3

事件监听器

4

练习

阻止事件冒泡和默认行为

事件流

事件流是描述的从页面接受事件的顺序，当几个都具有事件的元素层叠在一起的时候，那么点击其中一个元素，并不是只有当前被点击的元素会触发事件，而层叠在你点击范围的所有元素都会触发事件。事件流包括两种模式：冒泡和捕获。

事件冒泡

事件冒泡是从里往外逐个触发。事件捕获，是从外往里逐个触发。现代的浏览器默认情况下都是事件冒泡的模式。

阻止事件冒泡和默认行为

冒泡传递事件

```
document.onclick=function(){
    console.log('我是 document');
};
document.documentElement.onclick=function() {
    console.log('我是 html');
};
document.body.onclick= function(){
    console.log('我是 body');
};
document.getElementById('box').onclick=function() {
    console.log('我是 div');
};
document.getElementsByTagName('input')[0].onclick= function(){
    console.log('我是 input');
};
```

阻止事件冒泡和默认行为

但是一般我们只在指定的节点上添加事件, 而不想让其传递到下层节点触发事件, 这样我们就需要阻止事件冒泡;

阻止事件冒泡有两个方法:

(在指定不想再继续传递事件的节点的事件执行函数中使用)

1, 取消冒泡, IE

`e.cancelBubble = true;`

2, 停止传播, 非IE

`e.stopPropagation();`

例如:

```
document.getElementsByTagName('input')[0].onclick= function(evt){  
    var e = evt || window.event;  
    //可以通过下述两个方法取消事件冒泡  
    if(e.stopPropagation){  
        e.stopPropagation();  
    }else{  
        e.cancelBubble = true  
    }  
};
```

阻止事件冒泡和默认行为

阻止浏览器默认行为

```
if (e.preventDefault) {  
    e.preventDefault(); //非IE  
}  
else {  
    e.returnValue = false; // IE  
}
```

阻止右键菜单

在之前使用event对象的button属性时, 点击鼠标右键会弹出系统菜单, 如果我们想要创建自己的右键菜单, 则需要先阻止默认的右键菜单

```
document.oncontextmenu = function(){  
    console.log("右键被按下");  
    return false;  
}
```

阻止事件冒泡和默认行为

<a>标签有属性href, 在用户点击超链接标签<a>时, 实际上内部会调用onclick事件, 那么如果我们需要在超链接跳转前做一些判断处理, 则可以将onclick指向我们自己的函数, 并返回true或者false来决定是否允许超链接跳转;

例如:

```
var oA = document.getElementsByTagName("a")[0];
oA.onclick = function() {
    if(confirm("你确定要跳转吗?")) {
        return true;
    }
    else {
        return false;
    }
}
<a href= "http://www.baidu.com" >百度一下</a>
```

拖拽

所谓拖拽: 就是按住元素后移动位置, 最后松开的过程

1, 实现拖拽相关的三大事件:

onmousedown : 鼠标按下

onmousemove : 鼠标移动

onmouseup : 鼠标松开

拖拽

2, 实现拖拽思路：

1, 给目标元素添加onmousedown事件(鼠标按下事件)

在鼠标按下的瞬间, 记录鼠标所在位置与目标元素左边界的距离disX, 以及与上边界的距离disY

2, 当onmousedown事件发生以后(鼠标按下后), 就给document添加onmousemove事件(鼠标移动事件)

在onmousemove(鼠标移动事件)中, 根据以下公式不断刷新目标元素所在的位置:

公式: 目标元素的left = oEvent.clientX - disX + "px" ;

目标元素的top = oEvent.clientY - disY + "px" ;

3, 在onmousedown事件(鼠标按下事件)发生以后, 给document再添加onmouseup事件(鼠标松开事件), 且在onmouseup事件中, 取消document的onmousemove事件;

拖拽

注意:

onmousedown触发事件的对象: 目标元素(即要拖拽的元素);

onmousemove触发事件的对象: document

onmouseup触发事件的对象: document

onmousemove和onmouseup的触发事件对象都是document, 意味着鼠标在网页的任意位置移动鼠标或松开鼠标, 都会触发对应的事件;

onmousemove和onmouseup 都要写在onmousedown事件中, 这样就可以保证鼠标按下后才可以移动目标元素(onmousemove)或停止移动(onmouseup)

拖拽

```
onload = function() {  
    var box = document.getElementById("box");  
    //鼠标按下  
    box.onmousedown = function(evt) {  
        var e = evt || event;  
        //计算鼠标位置与div左边和上边的距离  
        var disX = e.clientX - box.offsetLeft;  
        var disY = e.clientY - box.offsetTop;  
        //鼠标移动  
        document.onmousemove = function(evt) {  
            var e = evt || event;  
            box.style.left = e.clientX - disX + "px";  
            box.style.top = e.clientY - disY + "px";  
        }  
        //鼠标松开  
        document.onmouseup = function() {  
            document.onmousemove = null;  
            document.onmouseup = null;  
        }  
    }  
}
```

拖拽

1, 窗口拖动和关闭, 只可以点击紫色区域才可以拖拽窗口, 点击关闭按钮就关闭窗口



获取style样式

获取style样式

```
if (window.getComputedStyle) {  
    style = window.getComputedStyle(box, null); //支持IE9+ 及非IE浏览器  
} else {  
    style = box.currentStyle; // IE8及以前  
}
```

事件监听器

事件监听器：

事件监听器, 是JS事件中的一种事件触发机制, 我们可以通过添加事件监听器的方式给元素添加事件及执行函数

1, 添加事件监听器

`box.addEventListener("click" , func, false)` : 给box元素添加点击事件(click), 以及事件执行函数func. 针对该函数的三个参数作说明:

第一个参数("click") : 事件名称(前面没有on)

第二个参数(func): 事件执行函数名称(没有双引号, 也没有())

第三个参数(false/true) : 是否使用捕捉(反向冒泡), 默认false , 为冒泡

2, 移除事件监听器

`box.removeEventListener("click" , func)` : 将box元素的点击事件(click), 以及对应的事件执行函数func移除
注: 这里只会删除使用box.addEventListener()方法添加的事件监听器

注: IE8及其以下不支持, 火狐和谷歌支持。

事件监听器

事件监听器：

在IE8及以下的浏览器中，我们可以使用以下方式：

```
box.attachEvent("onclick", fn); //添加事件
```

```
box.detachEvent("onclick", fn); //移除事件
```

封装事件监听器，兼容IE浏览器，谷歌，火狐

//添加事件

```
function addEvent(obj, type, fn, useCapture){  
    if (obj.addEventListener) {  
        obj.addEventListener(type, fn, useCapture);  
    }  
    else {  
        obj.attachEvent("on" + type, fn);  
    }  
}
```

事件监听器

//移除事件

```
function removeEvent(obj, type, fn, useCapture){  
    if (obj.removeEventListener) {  
        obj.removeEventListener(type, fn, useCapture);  
    }  
    else {  
        obj.detachEvent("on"+type, fn);  
    }  
}
```

//调用

```
addEvent(aBtn[0], "click", fn, false);  
function fn(){  
    alert("click 按钮1");  
}  
aBtn[1].onclick = function(){  
    removeEvent(aBtn[0], "click", fn, false);  
}
```


练习

1, iPhone解锁, 拖动滑块解锁, 让滑块在固定区域内左右拖动



练习

2, 360度全景展示, 在空白页面上左右拖动,让图片呈360旋转

(提示：在空白处实现拖拽功能，拖拽过程中 计算onmousemove的实时x坐标位置和mousedown的x坐标位置的差；根据差值的大小来改变图片的src路径，图片名称是由0.jpg-76.jpg)



THANK YOU



做真实的自己，用良心做教育