千锋教育
www.mobiletrain.org

千锋HTML5学院

第二阶段javascript课程课件

1. 事件的冒泡
2. 阻止右键菜单
3. 超链接的拦截
4. 拖拽的原理
5. 事件监听器

事件的目标(target属性)

target: 目标对象,存放绑定事件的元素节点对象

```
document.onclick = function(evt) {
    var oEvent = evt || event;
    console.log("document: " + oEvent.target); //HTMLHtmlElement
}

box.onclick = function(evt) {
    var oEvent = evt || event;
    console.log("box: " + oEvent.target); //HTMLDivElement
}

oInput.onclick = function(evt) {
    var oEvent = evt || event;
    console.log("input: " + oEvent.target); //HTMLInputElement
}
```

事件的冒泡

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， 那么你点击其中一个元素，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

是从里往外逐个触发。事件捕获，是从外往里逐个触发。那么现代的浏览器 默认情况下都是冒泡模型

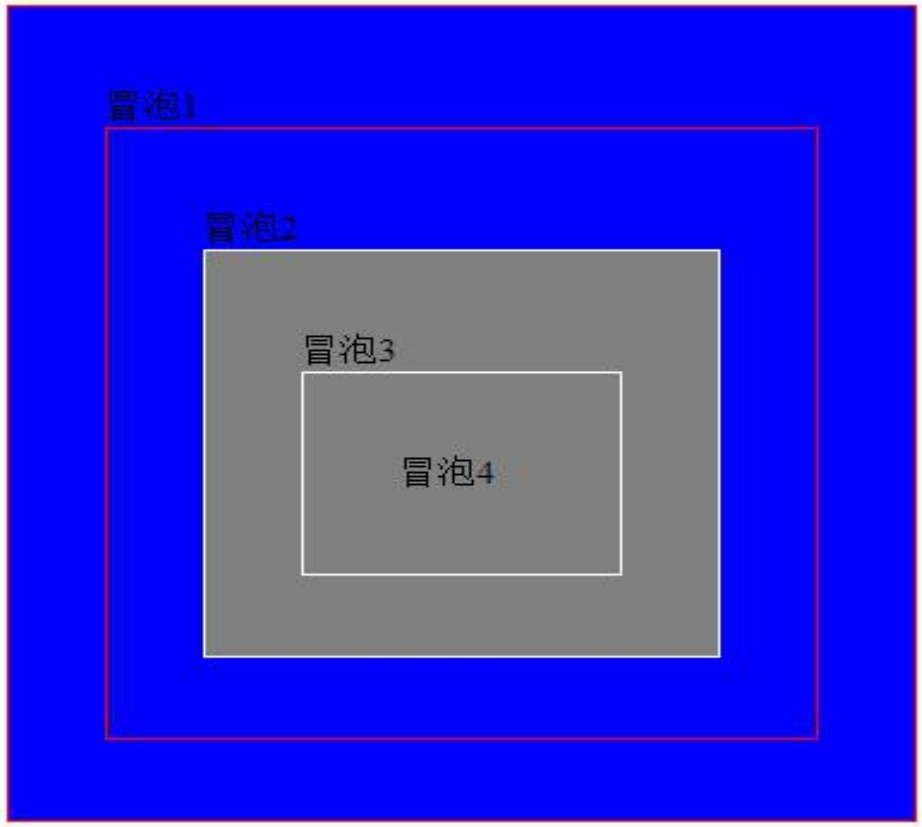
事件的冒泡

事件的冒泡: 指的是在页面上层节点触发的事件会继续传递给下层节点, 这种传递方式,我们称之为事件的冒泡传递;

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

事件的冒泡

点击冒泡2, 会触发到下层的, 如下图



事件的冒泡

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冒泡;

阻止冒泡的方式有两种:
(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
    oEvent.cancelBubble = true;
//2, 停止传播
    oEvent.stopPropagation();
```

```
document.getElementsByTagName('input')[0].onclick= function(){
    var oEvent = evt || window.event;
    //可以通过下述两种方式取消事件冒泡
    oEvent.cancelBubble = true; //1, 取消冒泡
    oEvent.stopPropagation(); //2, 停止传播
};
```

阻止右键菜单事件

阻止右键菜单

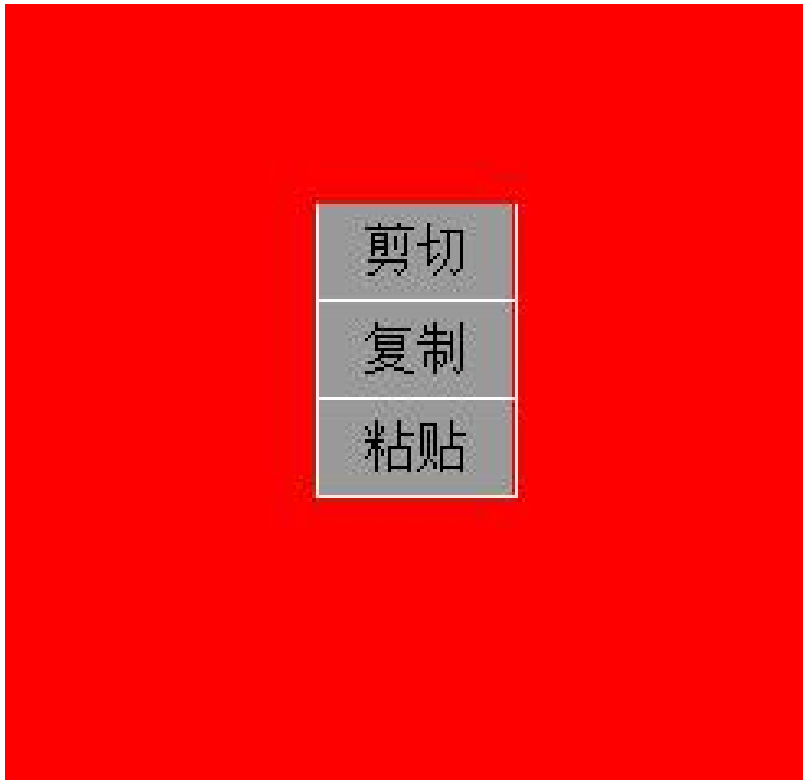
在之前使用event对象的button属性时，点击鼠标右键会弹出系统菜单，如果我们想要创建自己的右键菜单，则需要先阻止默认的右键菜单

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

自定义右键菜单

自定义右键菜单

右键弹出如下图的自定义菜单:



超链接<a>的拦截

<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;

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

练习

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

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

注: IE8及其以下不支持，火狐和谷歌支持。

事件监听器

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

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
}
```

练习

练习1

1, 鼠标跟随一串效果: 在鼠标移动过程中,在其后面跟随一串效果

练习

练习2

iPhone解锁，拖动滑块解锁

练习

练习3

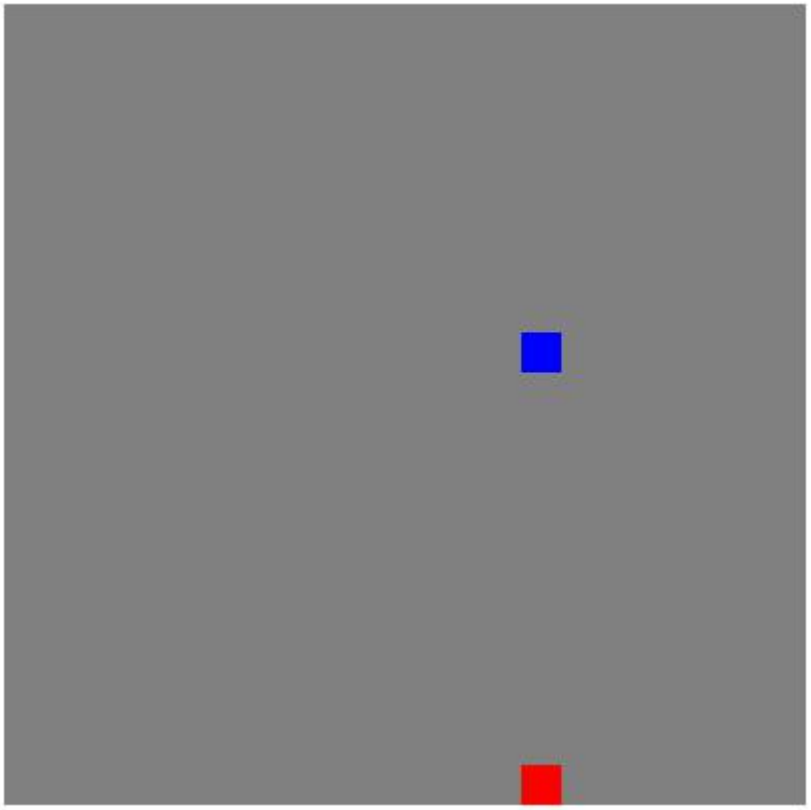
360度全景展示, 在空白页面上左右拖动,让图片呈360旋转

offsetParent参照物父元素

- 当某个元素的父元素或以上元素都未进行CSS定位时，则返回body元素，也就是说元素的偏移量（offsetTop、offsetLeft）等属性是以body为参照物的
- 当某个元素的父元素进行了CSS定位时（absolute或者relative），则返回父元素，也就是说元素的偏移量是以父元素为参照物的
- 当某个元素及其父元素都未进行CSS定位时，则返回距离最近的使用了CSS定位的元素

练习

1, 碰撞, 两个块在指定范围内运动, 碰壁后会反弹



作业

放大镜，移动鼠标，在小图上移动时显示大图,并跟随鼠标移动，当不在小图上时，隐藏大图，效果如下图