**移动端触屏事件**

**写法：要用Dom2的形式绑**

var T = document.querySelector('.touchA')

//手指按下去触发的事件

T.addEventListener('touchstart',function(){

console.log('start')

})

//手指按下去触发的事件

T.addEventListener('touchmove',function(){

console.log('move')

})

//手指按下去触发的事件

T.addEventListener('touchend',function(){

console.log('end')

})

移动端的默认事件  
有字体会被选，右键菜单等等  
**用ev.peventDefault();**

**橡皮筋效果**

**//pc端的禁止行为**

**//比如右击的菜单 吧禁止行为绑定给document的时候，他的子元素的右击默认行为也会被禁止**

document.oncontextmenu = function(ev){

ev = ev||event;

ev.preventDefault();

}

**//如果希望不冒泡影响给字元素的话给子元素加一个ev.stopPropagation();这样这个子元素就不会禁止默认行为了**

wrap.oncontextmenu = function(ev){

ev = ev||event;

**ev.stopPropagation();**

}

**//如果希望孙子也能冒泡到 就给这个孙子加ev.preventDefault();**

inner.oncontextmenu = function(ev){

ev = ev||event;

ev.preventDefault();

}

**移动端全面禁止行为—隐患—移动端的滚动条全部都会失效**

//全面禁止移动端事件的默认行为

document.addEventListener("touchstart",function(ev){

ev=ev||event;

ev.cancelable=true;//静默失败

//ev.preventDefault在ev.cancelable为true的时候才有作用

ev.preventDefault();

})

testNode.addEventListener("touchstart",function(ev){

ev=ev||event;

ev.preventDefault();

})

innerNode.addEventListener("touchstart",function(ev){

ev=ev||event;

ev.stopPropagation();

})

**自定义一个右键菜单**

document.oncontextmenu=function(ev){

ev=ev||event;

//首先获得当前ev手指所在的地方，阻止默认菜单之后把当前偏移量给div

var x = ev.clientX;

var y = ev.clientY;

var menu = document.querySelector("#menu");

menu.style.display="block";/将菜单显示出来

menu.style.left = x+"px";//把这个div的偏移量赋给这个div

menu.style.top = y+"px";

ev.preventDefault();

}

document.onclick=function(){

var menu = document.querySelector("#menu");

menu.style.display="none";

}

**事件点透：**

**点透出现的场景**

刚才举例说明了什么是点透，其实点透的出现场景可以总结如下：

1.A/B两个层上下z轴重叠。

2.上层的A点击后消失或移开。（这一点很重要）

3.B元素本身有默认click事件（如a标签） 或 B绑定了click事件。

在以上情况下，点击A/B重叠的部分，就会出现点透的现象。

**为什么会出现点透**

 click延迟，延迟，还是延迟。

在**移动端**不使用click而用**touch事件**代替触摸是因为click事件有着明显的延迟300毫秒，具体touchstart与click的区别如下：

1.touchstart：在这个DOM（或冒泡到这个DOM）上手指触摸开始即能立即触发

2.click：在这个DOM（或冒泡到这个DOM）上手指触摸开始，且手指未曾在屏幕上移动（某些浏览器允许移动一个非常小的位移值），且在这个在这个dom上手指离开屏幕，且触摸和离开屏幕之间的间隔时间较短（某些浏览器不检测间隔时间，也会触发click）才能触发

也就是说，事件的触发时间按由早到晚排列为：touchstart 早于 touchend 早于 click。亦即click的触发是有延迟的，这个时间大概在300ms左右。

由于我们在touchstart阶段就已经隐藏了罩层A，当click被触发时候，能够被点击的元素则是其下的B元素，根据click事件的触发规则：

只有在被触发时，当前有click事件的元素显示，且在面朝用户的最前端时，才触发click事件。

由于B绑定了click事件（或者B本身默认存在click事件），所以B的click事件被触发，产生了点透的情况。

**三、解决方案**

1.对于B元素本身没有默认click事件的情况（无a标签等），应统一使用touch事件，统一代码风格，并且由于click事件在移动端的延迟要大很多，不利于用户体验，所以关于触摸事件应尽量使用touch相关事件。

2.对于B元素本身存在默认click事件的情况,应及时取消A元素的默认点击事件，从而阻止click事件的产生。即应在上例的handle函数中添加代码如下：

if(eve == "touchend") e.preventDefault();

 3.对于遮盖浮层，由于遮盖浮层的点击即使有小延迟也是没有关系的，反而会有疑似更好的用户体验，所以这种情况，可以针对遮盖浮层自己采用click事件，这样就不会出现点透问题。

**四、现有解决方案框架（库）**

1. 众所周知，zepto的tap事件是有点透问题的，但是最新版的zepto已经修复了这个问题。

2. 在zepto修复问题之前，有fastclick、hammer等通用库可以使用。

**页面跳转方案 //防止误触**

// **PC端事件在执行时会有300ms的延迟**

document.addEventListener("touchstart",function(ev){

ev=ev||event;

ev.preventDefault();

})

//移动端的跳转方案 解决误触

var aNodes = document.querySelectorAll("a");

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

aNodes[i].addEventListener("touchstart",function(){

this.ismoved=false;

})

aNodes[i].addEventListener("touchmove",function(){

if(!this.ismoved){

this.ismoved=true;

}

})

aNodes[i].addEventListener("touchend",function(){

if(this.ismoved){

return;

}

window.location.href = this.href;

})

}

**移动端手指事件的几个API**

**ev.如下**

changedTouches:触发当前事件的手指列表

targetTouches:触发当前事件时,元素身上的手指列表（）

touches:触发当前事件时,屏幕上的手指列表（屏幕有多少就多少）

var testNode = document.querySelector("#test");

testNode.addEventListener("touchend",function(ev){

ev=ev||event;

console.log(ev)

testNode.innerHTML="change:"+ev.changedTouches.length+"<br>"+"target:"+ev.targetTouches.length+"<br>"+"touches:"+ev.touches.length;

})

**禁止电话和邮箱**

<p>1383838438</p>

<p>snw@qq.com</p>

<a href="tel:110">1383838438</a>

<a href="mailTo:snw@qq.com">snw@qq.com</a>

**链接&按钮背景高亮问题.html**

/\*处理背景高亮\*/背景会有高亮要取消

-webkit-tap-highlight-color: rgba(0,0,0,0);

/\*处理圆角渲染太过分\*/只有有圆角，移动端都是圆的

-webkit-appearance: none;