



**一、事件冒泡：**

对document、body、container、item都绑定了事件监听后，在点击item的时候，事件会从item开始往上冒泡。依次触发item、container、body、document的监听函数。

**二、事件代理：**

container绑定事件监听，来处理item的点击事件。通过e.target来判断点击对象.

container.addEventListener(function(e){

if(e.target.nodeName == ‘A’ && e.target.classList.contains(‘item’)){ }

}, false)

**三、拖拽案例：**

item通过事件代理mousedown 记录初始位置数据——鼠标、item

document绑定mousemove 事件，获取最新的pos，修改item的left top

document绑定mouseup事件，取消mousemove mouseup事件监听

1. mouseover/mouseout 与 mouseenter/mouseleave 的区别：

<div class=”outter”>

<div class=”inner”></div>

</div>

1. 现象

当鼠标在outter和inner之间来回移动的时候，移入inner和移出到outter均会触发outter的mousout和mouseover事件，但不会触发mouseleave/mouseenter。只有在鼠标移到outter外面，或从外面进入outter的时候，才会触发mouseleave/mouseenter

outter与inner的层级关系只与dom结构有关，跟显示位置无关

1. 顺序

先执行over/out，再执行enter/leave

1. 原理

在鼠标从outter移入inner时，会触发outter的mouseouter事件和inner的mouseover事件。因为冒泡机制，inner的mouseover事件被outter的监听捕获，故会触发一次out和over。

1. 让mouseover不执行冒泡

判断outter是否包含移入的对象，从而执行对应的方法

outter.onmouseover = function(e){

if (e) {

if (outter.contains(e.relatedTarget)) return;

} else if (event) {

if (outter.contains(event.fromElement)) return;

}

console.log('over');

}

1. **keypress与keydown的区别**

keydown只要有按键按下就会触发，即使是window键

keypress只有能输出内容的按键按下才会触发，(Enter会输出\n，故也支持)。支持组合键（如shift+1输出!，在输出!后触发），但不支持中文输入法（包含中文标点，如中文的！不会触发）

键盘事件执行顺序：keydown、keypress、input、keyup