# webApi第五天--事件详解

（事件这部分的主要内容 事件的绑定形式 事件对象 事件流的三个阶段（事件冒泡 事件目标 事件捕获）键盘事件 事件的默认行为 事件拖拽）

1. 事件冒泡：
   1. **事件冒泡概念**：（和html嵌套结构有关系）
      1. 当一个对象身上发生一个事件（单击事件 鼠标移上事件）的时候，它本身执行完这个事件以后，会继续向上（向父级们）传递该事件，不断向上传递，直到window为止；在传递过程中 不管父级有没有做事情，它都会收到下面传上来的事件；这样一个事件传递的机制：冒泡机制（w3c定义的一种理论规范）
      2. 代码演示
   2. 事件冒泡到底好还是不好？有好也有坏处
      1. 案例1：分享到（利用事件冒泡 将子元素的事件交给父元素去做）
      2. 案例2：下拉菜单展开收缩 （我们需要阻止事件冒泡）

如果阻止：在你不希望它冒泡的那个事件触发函数里面 使用

(这里的ev必须是兼容后的事件对象)**ev.cancelBubble = true;** 阻止事件冒泡

* 1. 事件委托：
     1. 原理：利用事件冒泡机制，将子元素们的事件委托给父级来代理执行，通过判断事件源，从而做事情，这样做的好处，不需要关系父级内部元素如果变化，

对于内部子元素动态变化尤其适用

* + 1. 案例：留言框删除

1. 事件流--事件捕获
   1. 事件捕获概念：事件传递的过程是从外往里 （IE浏览器不支持事件捕获） 标准浏览器支持事件捕获
      1. 演示画图
      2. addEventListener 和 attachEvent 对于捕获的支持
      3. 代码执行顺序和书写顺序无关和事件流的顺序有关 （代码演示）
   2. 事件流（三个阶段）
      1. 事件捕获---->事件目标---->事件冒泡
2. 总结一下我们目前常用的事件
3. 关于js中一些尺寸问题：
   1. 页面可视区域的宽高
      1. document.documentElement.clientWidth
      2. document.documentElement.clientHeight
   2. A.offsetParent 找到距离A最近的具有定位属性的父级，如果没有直接指向body 以及 offsetLeft offsetTop 这两个属性 要去参考它的offsetParent ，相对于offsetParent的左侧偏移量和顶部偏移量
   3. offsetWidth offsetHeight 这两个属性 直接可以获取元素的实际的宽/高 + 左右/上下 padding + 左右/上下 border
   4. clientLeft clientTop 盒子内部距离外部的偏移量（边框厚度） 实际效果就是一个边框距离
   5. clientWidth clientHeight这两个属性 直接可以获取元素的实际的宽/高 + 左右/上下 padding
   6. scrollLeft scrollTop
      1. 如果获取的是页面的滚动条垂直滚动距离需要兼容一下

var scrollT = document.documentElement.scrollTop || document.body.scrollTop

Ii. 普通元素.scrollTop 可以设置也可以获取

* 1. scrollWidth scrollHeight

1. 键盘事件：
   1. keydown keyup
   2. keyCode
   3. 辅助键 ctrlKey altKey shiftKey
   4. 案例：
      1. qq发送信息案例
      2. 键盘控制小飞机走
      3. 进阶版 小飞机可以斜着走 以及限制范围
2. 事件的默认行为
   1. 概念：
   2. 浏览器的默认行为：空格 图片的拖拽 右键菜单
   3. return false 以及其他的方法
   4. 案例： 自定义右键菜单
3. 事件拖拽：
   1. Div拖拽
   2. 图片拖拽
4. 作业：放大镜案例