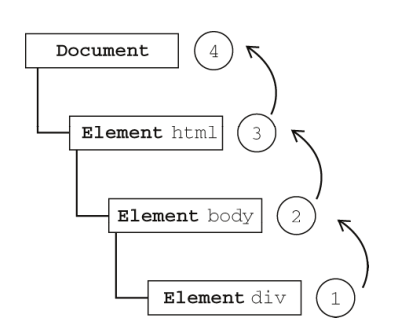
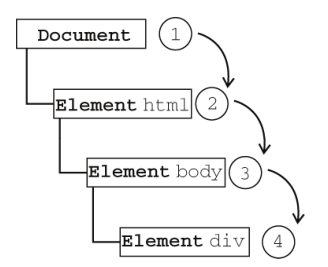
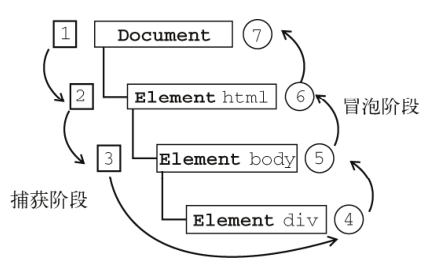
**学习目标：**

1、事件冒泡

**事件冒泡** :







演示文档：

div {padding: 40px;}

#div1 {background:red}

#div2 {background:green}

#div3 {background:blue; /\* position: absolute; top: 300px;\*/}

<div id="div1">

<div id="div2">

<div id="div3"></div>

</div>

</div>

var oDiv1 = document.getElementById('div1');

var oDiv2 = document.getElementById('div2');

var oDiv3 = document.getElementById('div3');

function fn1() {

alert( this.id );

}

oDiv1.onclick = fn1;

oDiv2.onclick = fn1;

oDiv3.onclick = fn1;

说明：当点击div3时，同时触发了div1和div2上绑定的事件。同时如果把div3定位出来，还是会出发。所以说明：跟它的样式表现没有关系，跟页面html结构有关系。

**事件冒泡的解释** : 当一个元素接收到事件的时候，会把他接收到的所有传播给他的父级，一直到顶层window。这种机制我们称之为”事件冒泡机制”

如果这个元素它有事件函数绑定，哪么在冒泡时，就会执行它绑定的函数。如果它没有事件函数绑定，也不妨碍它接收它的子级传过来的事件，以及它向它的父级传播事件。

oDiv1.onclick = fn1;

//oDiv2.onclick = fn1;

oDiv3.onclick = fn1;

//事件函数绑定

oDiv1.onclick = fn1;//告诉div1，如果他接收到了一个点击事件，那么他就去执行fn1

事件冒泡的机制默认会存在。

例子：下拉菜单：

参考：下拉菜单.html

讲明白点不出菜单的问题，（因为点击oBtn，同时会冒泡到document上面，也会触发它的点击，其实菜单会出来，接着就隐藏了，因为太快，所以看不到。）

可以引入一个延时定时器，展示清楚一点。

讲明白阻止冒泡的问题。

案例：分享到：

参考：分享到.html

说明：我们可以把滑上滑离分别给每个元素添加，但是有冒泡机制了，就不用给每个元素都加了，只需要给父级添加即可。这时，我们只需要获取div1即可。这样就可以少写很多代码。