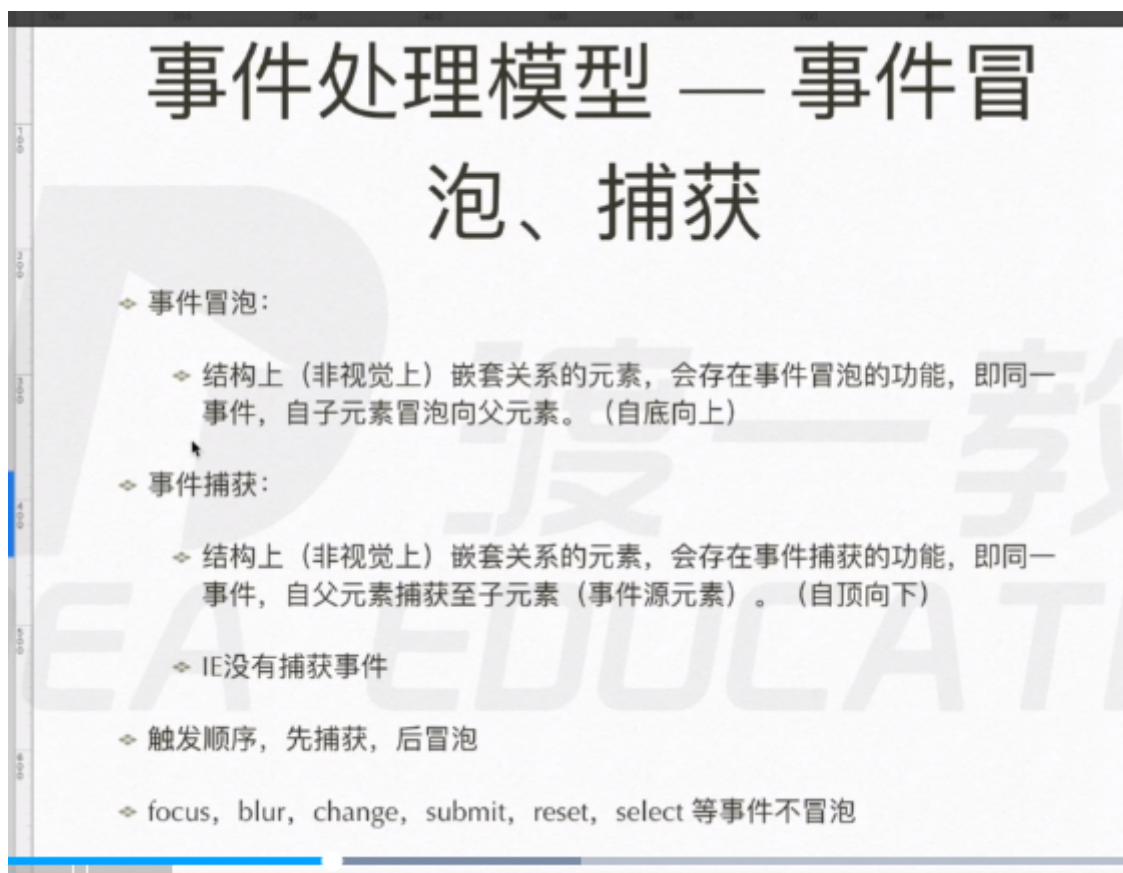


# 一：事件处理模型---事件冒泡 /捕获



1:冒泡，里层的元素触发的事件，（即点击了里层的元素，也相当于点击了外层的元素，会触发外层元素所绑定的函数） 常规模型

2:支持事件捕获的 只有谷歌/火狐/opera（新版本）， 自顶向下

事件捕获，即触发里层元素的时候，会先触发外层元素所绑定的相应的事件，再去执行最里面的事件（外层元素是事件捕获，最里面的事情是 事件执行）

3:ie没有事件捕获

4:addEventListenr的第三个参数为false的时候，是触发捕获事件，写成false是冒泡

5:如果给同一个元素的同一个事件同时绑定的冒泡事件和捕获事件，则会正常的事件是 先捕获，后冒泡的执行顺序

## 二：取消事件冒泡

事件对象：在每一个事件处理函数上我们都可以写一个形参，这个形参记载了事件发生时的一系列的信息，事件类型啊，时间啊，坐标点啊

```
domdiv.addEventListener("click", function(e) {  
    console.log(e);
```

阻止事件冒泡的方法1:

```
e.stopPropagation();
```

阻止事件冒泡的方法2:

```
e.cancelBubble = true;
```

```
});
```

### 封装的阻止冒泡的函数

```
function stopBubble(e) {  
    if (e.stopPropagation) {  
        e.stopPropagation();  
    } else {  
        e.cancelBubble = true;  
    }  
}
```

右键出菜单事件

onContextmenu

## 三：阻止默认事件

阻止默认事件:

- ❖ 默认事件 — 表单提交, a标签跳转, 右键菜单等
- ❖ 1.return false; 以对象属性的方式注册的事件才生效
- ❖ 2.event.preventDefault(); W3C标注, IE9以下不兼容
- ❖ 3.event.returnValue = false; 兼容IE
- ❖ 封装阻止默认事件的函数 cancelHandler(event);

**return false; 注意：以属性的方式注册使事件才生效**  
**event.preventDefault();**

```
event.returnValue = false;
```

```
<a href="javascript:void(false)" >demo</a>
```

## 四：事件对象

e / 谷歌

window.event    ie

### 事件源对象

，如果是冒泡或者捕获的话，则可以知道是谁触发的，就就叫”源”对象

事件源对象是 事件对象的一个属性

event.target    火狐

event.srcElement    ie

谷歌都有

事件委托

- ❖ 利用事件冒泡，和事件源对象进行处理
- ❖ 优点：
  - ❖ 1. 性能 不需要循环所有的元素一个个绑定事件
  - ❖ 2. 灵活 当有新的子元素时不需要重新绑定事件

