

# 一、事件

**事件对象：**事件被触发时产生的数据集 {key:value} 。

**事件监听：** 监听事件状态的方法 `addEventListener`

## 主流浏览器

```
1 dom.addEventListener("click", function (event) {
2     // 使用事件对象，记得填写
3     var event = event || window.event; // 兼容写法
4     event.stopPropagation();
5 }, true);
```

## 低版本IE (5678)

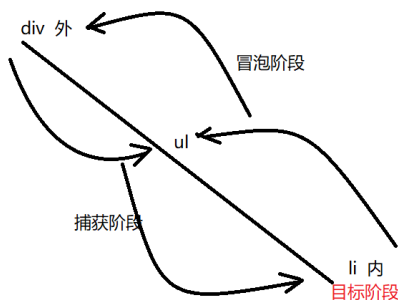
```
1 // 兼容的写法
2 // if(li.addEventListener){
3 //     li.addEventListener()
4 // }else {
5 //     li.attachEvent();
6 // }
7 dom.attachEvent("onclick", function(event){
8     var event = event || window.event;
9     event.cancelBubble = true;
10 });
```

**冒泡阶段：** 事件从内层标签往外层标签传递，  
事件是从子元素往父级元素递进，一直到window。  
(这个现象不明显，但是真实存在的)

**捕获阶段**

**目标阶段**

**目标阶段**



事件由内往外顺序执行 这个是事件冒泡现象 冒泡阶段

事件由外往内顺序执行 这个是事件捕获现象 捕获阶段

在最内部的元素（目标元素） 目标阶段

## 阻止事件冒泡

```
1 dom.onclick = function(event){
2   var event = event || window.event;
3   if(event.stopPropagation){
4     event.stopPropagation();
5     console.log("主流浏览器");
6   }else {
7     // 低版本IE 5678
8     event.cancelBubble = true;
9     console.log("低版本IE 5678");
10  }
11 }
```

## 阻止浏览器的默认行为

```
1 例如：页面跳转、页面刷新、选中图片文字
2 dom.onclick = function(event){
3   // event
4   // 阻止默认行为发生
5   // event.preventDefault();
6   return false // 阻止默认行为发生else;
7   //return 有终止代码，后续的代码就不执行了
8 }
```

## 事件委托

**事件委托：是通过事件冒泡原理，给父级元素（祖先元素）绑定事件，其后代元素同时也注册了事件。**

```
1 <!-- div > ul > li（目标阶段） -->
2 <!-- 外 > 中 > 内 -->
3 <div class="div">
```

```
4     <ul class="ul">
5         <li class="li"></li>
6     </ul>
7 </div>
```

```
1 var div = document.getElementsByTagName("div")[0];
2 // console.log(div)
3 // 2.0 事件给div标签绑定
4 // 可以给标签设置不同属性是
5 // class = "li"
6 // class = "ul"
7 // class = "div"
8 // 区别点击的元素
9 div.onclick = function(event){
10     // console.log("click")
11     // 通过事件对象 可以找到事件源（目标元素）
12     // 当前点击的标签（目标元素）事件原
13     var ele = event.target;
14     // console.log(ele.className);
15     if(ele.className == "li"){
16         document.body.style.backgroundColor = "yellow";
17         console.log(ele);
18     }
19
20     }else if(ele.className == "ul"){
21         document.body.style.backgroundColor = "skyblue";
22         console.log(ele);
23     }
24
25     }else if(ele.className == "div"){
26         document.body.style.backgroundColor = "lightgreen";
27         console.log(ele);
28     }
29 }
30
31
32 // 业务：
33 // 点击 li 标签设置 页面yellow背景色
34 // 点击 ul 标签设置 页面skyblue背景色
35 // 点击 div 标签设置 页面lightgreen背景色
```

```
36
37
38
39 // 优点:
40     // 减少使用for循环语句
41     // 减少web中DOM操作
42     // 还可以动态的给予元素（后代元素）注册事件
```

## 遇到的问题:

1、属性绑定不知道怎么用

记录数据

2、事件委托应用哪些地方

不使用循环语句

减少事件的注册（给父级元素、祖先元素绑定事件即可）

动态的方式给后代元素注册事件

