

20-第二十章 事件一

一、事件event对象

Event对象

用来获取事件的详细信息：鼠标位置、键盘按键

兼容

- **标准下**：事件对象是通过事件函数的第一个参数

传入 如果一个函数是被事件调用的那么，这个函数定义的第一个参数就是事件对象

- **ie**: event是一个内置全局对象

```
1. var obj.onclick = function(ev){
2.     var ev = ev || window.event;
3. }
4. function abc(ev){
5.     var ev = ev || window.event;
6. }
```

Event对象的兼容：ev = ev || window.event;

Event对象下的获取鼠标位置：e.clientX e.clientY

二事件冒泡

事件冒泡指子元素触发事件的时候，会 **冒泡**（触发）父级的相同的事件

非标准：`ev.stopPropagation()`；

非标准：`ev.cancelBubble=true`；

三 注册处理事件

1、标准：`obj.addEventListener(事件名称, 事件函数, 是否捕获)`；

是否捕获

`false` 冒泡

`true` 捕获

先捕获后冒泡

1.有捕获

2.事件名称 没有on

3.事件执行的顺序是 正序

4. `this` 触发该 事件的对象

2、低版ie：`obj.attachEvent(事件名称, 事件函数)`；

1.没有捕获

2.事件名称 有on

3.事件函数执行的顺序：标准ie-》 正序 非标准ie-》 倒序

4. `this` 指向 `window`

3、移除 `obj.removeEventListener(事件名称, 事件函数)`

4、移除 `object.detachEvent(事件名称,function)`；

四、拖拽

- `onmousedown`
- `onmousemove`
- `onmouseup`

解决：文字选中再的问题

标准：解决办法 `return false` (阻止默认事件)

非标准ie：全局捕获

1. `onselectstart="return false"` 处理文字选中
2. 或者添加`ondragstart="return false"` 处理图片拖动

标准下阻止默认事件，可以拖拽图片啦

五、鼠标滚轮事件

le/chrome：`onmousewheel`

event.`wheelDelta`

上：120

下：-120

firefox：`DOMMouseScroll` 必须用 `addEventListener()` 添加

fire:event.`detail`

上：-3

下：3

滚轮属性：`event.wheelDelta` \ `event.detail`

兼容：

```
1.         obj.onmousewheel = handle;//兼容ie chrome
2.         obj.addEventListener?obj.addEventListener('DOMMouseScroll',handle,boolean):false; //兼容firefox
3.
4.
5.         //滚动函数event对象处理
6.         function handle(e){
7.             e = e || window.event;
8.             e.wheel = e.wheelDelta?e.wheelDelta:-e.detail*40;
9.
10.        }
```

阻止默认事件：

标准：`event.preventDefault()`

非标准：`event.returnValue = false;`

return false阻止的是 obj.on事件名称=fn 所触发的默认行为
addEventListener绑定的事件需要通过event下面的event. preventDefault();
detachEvent绑定的事件需要通过event下面的event. returnValue=false;

鼠标滚轮与自定义滚动条结合

六、全兼容添加/删除事件 封装

```
1. //注册监听事件
2. function addEvent(obj,eventType,fn,boolean){
3.     //事件兼容处理
4.     boolean=boolean?boolean:false;
5.     obj[eventType] = handle;//把处理函数赋值给window，便于后面的清除
6.     obj.addEventListener?obj.addEventListener(eventType,obj[eventType],boolean):obj.attachEvent('on'+eventType,obj[eventType]);
7.
8.     //兼容:鼠标滚动
9.     if(eventType=='mousewheel'){
10.         obj.onmousewheel = obj[eventType];//兼容ie chrome
11.         obj.addEventListener?obj.addEventListener('DOMMouseScroll',obj[eventType],boolean):false; //兼容firefox
12.
13.     }
14.     //滚动函数event对象处理
15.
16.     function handle(e){
17.         e = e || window.event;
18.         console.log( 'ok')
19.         e.wheel = e.wheelDelta?e.wheelDelta:-e.detail*40;
20.         fn.call(obj,e);
21.     }
22.
23. }
24.
25.
26. //注册监听事件
27. function removeEvent(obj,eventType,fn,boolean){
28.     boolean=boolean?boolean:false;
29.     obj.removeEventListener?obj.removeEventListener(eventType,obj[eventType],boolean):obj.detachEvent('on'+eventType,obj[eventType]);
30.
31.     //兼容:鼠标滚动
32.     if(eventType=='mousewheel'){
33.         obj.onmousewheel = null;//兼容ie chrome
34.         eventType = 'DOMMouseScroll';
35.         obj.removeEventListener?obj.removeEventListener(eventType,obj[eventType],boolean):false; //兼容firefox
36.
37.     }
38.
39. }
```