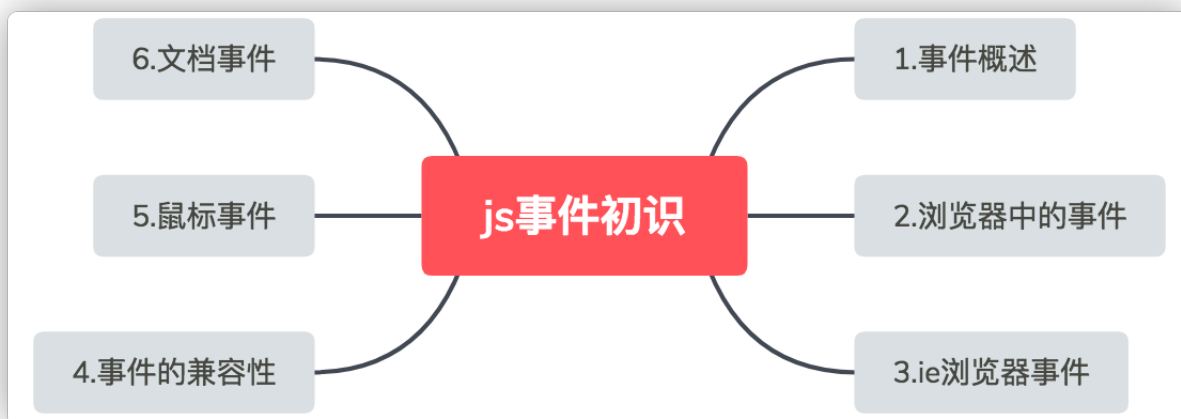


JS事件初识

0.1. 主要内容：



0.1. 学习目标：

节数	知识点	要求
第一节 事件概述	什么是事件	了解
	事件分类了解	了解
第二节 浏览器中的事件	HTML事件	了解
	DOM0级事件	了解
	DOM2级事件	了解
	事件流	了解
第三节 ie浏览器事件	ie浏览器事件	掌握
第四节 事件的兼容性	事件的兼容性	掌握
第五节 鼠标事件	事件分类	掌握

1. 关于事件

1.1. 什么是事件

事件就是在文档中或者在浏览器窗口中通过某些动作触发。比如，单击，鼠标经过，键盘按下等。事件通常和函数结合使用。

事件的作用：

(1)各个元素之间可以借助事件来进行交互

(2)用户和页面之间也可以通过事件来交互

(3)后端和页面之间也可以通过事件来交互（减缓服务器的压力）

onsubmit(提交事件和表单一起使用)

1.2. 事件分类

1、鼠标事件 onclick、ondblclick、onmouseover、onmouseout、onmousedown、onmouseup、onmousemove

2、HTML事件 onload、onscroll、onsubmit、onchange、oninput、onfocus(获取焦点)、onblur(失去焦点)

3、键盘事件 onkeydown: 键盘按下时触发 onkeypress: 键盘按下并松开的瞬间触发 onkeyup: 键盘抬起时触发

2. 事件的使用（重点）

2.1. HTML事件

绑定操作发生在HTML代码中的事件，称为HTML事件。

语法：on+事件='函数();函数();函数();.....'

HTML事件的移除方式：

元素.setAttribute('on+事件名', null);

HTML事件缺陷：耦合性太强了，修改一处另一处也要修改。

当函数没有加载成功时，用户去触发事件，则会报错。

```
6-事件流的介绍.html x 7-事件清除.html x 04:15
html body script
10 <body>
11 <button onclick=f1();f2()>单击</button>
12 <script>
13 var obt = document.querySelector("button");
14 //obt.onclick = null;事件清除的方法1
15 // obt.setAttribute("onclick", null); //事件清除的方法2
16 obt.removeAttribute("onclick"); //事件清除的方法3
17 function f1() {
18 console.log("f1");
19 }
20 function f2() {
21 console.log("f2");
22 }
23 </script>
24
25 </body>
26 </html>
```

2.2. DOM0级事件

DOM0事件调用函数的几种方式：

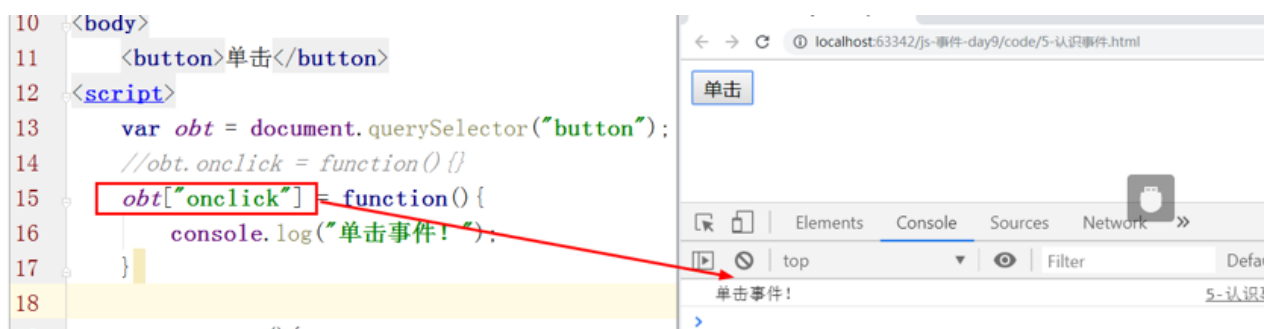
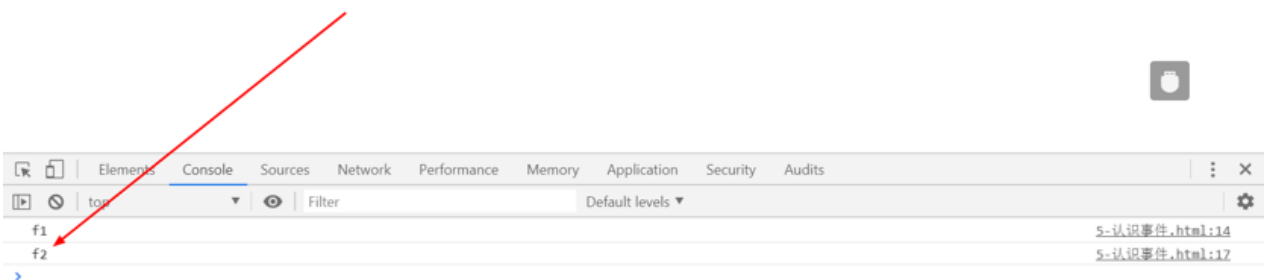
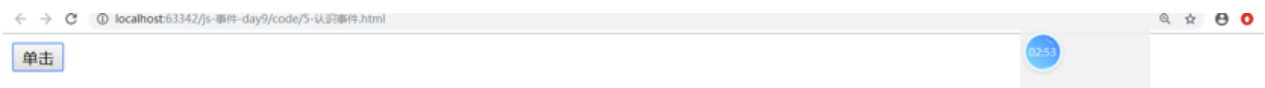
(1) 节点.事件=function(){}

(2) 节点.事件=函数名

```
11 <button onclick=f1();f2()>单击</button>
12 <script>
13 var obt = document.querySelector("button");
14 //方法2-调用函数
15 obt.onclick = function(){
16 f1();
17 }
18
19 // obt.onclick = f1;方法1: 调用函数, 直接写函数名称
20 function f1() {
21 console.log("f1");
22 }
23 function f2() {
24 console.log("f2");
25 }
26 </script>
27
```

(3)

```
5-认识事件.html x
html body script
7
8 </style>
9 </head>
10 <body>
11 <button onclick="f1();f2()">单击</button>
12 <script>
13 function f1(){
14     console.log("f1");
15 }
16 function f2(){
17     console.log("f2");
18 }
19 </script>
20
21 </body>
22 </html>
```



在js脚本中，直接通过【on+事件名】方式绑定的事件称为是DOM0级事件。

语法：

元素.on+事件名 = function(){需要执行的语句;}

元素[on+事件名] = function(){需要执行的语句;}

DOM0级事件的移除方式:

元素. on+事件名=null;

2.3. DOM2级事件

2.3.1. 冒泡

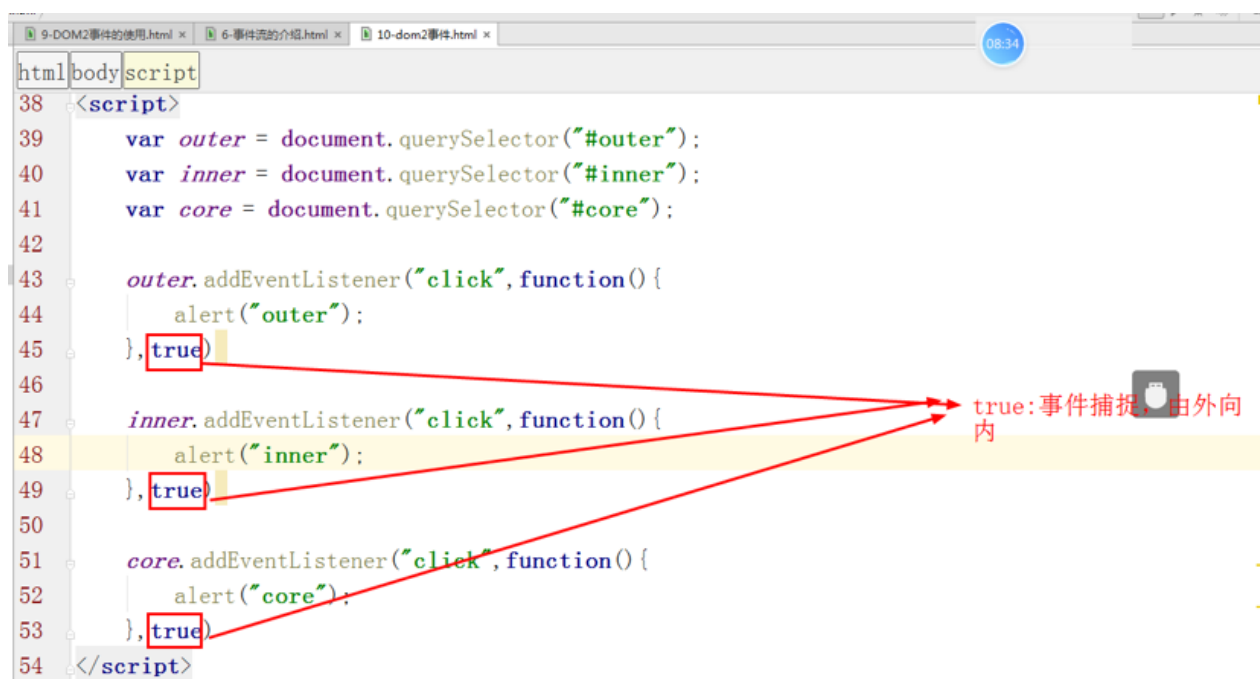
什么是事件冒泡?

事件由具体某个元素（子节点）逐级向上传播（父节点，比如html）

2.3.2. 捕捉

什么是事件捕捉?

事件由某个具体的元素（父节点）逐级向下传播（子节点）。



2.3.3. 事件流

事件流：多个嵌套的标记或者包含的标记拥有相同的事件，其中一个元素的事件触发，同时影响其他元素同类型的事件的触发，我们称之为“事件流”。

2.3.4. 事件流语法

方式1：对象.addEventListener("事件类型","函数","事件流");**针对非ie浏览器的，事件类型：不加on**

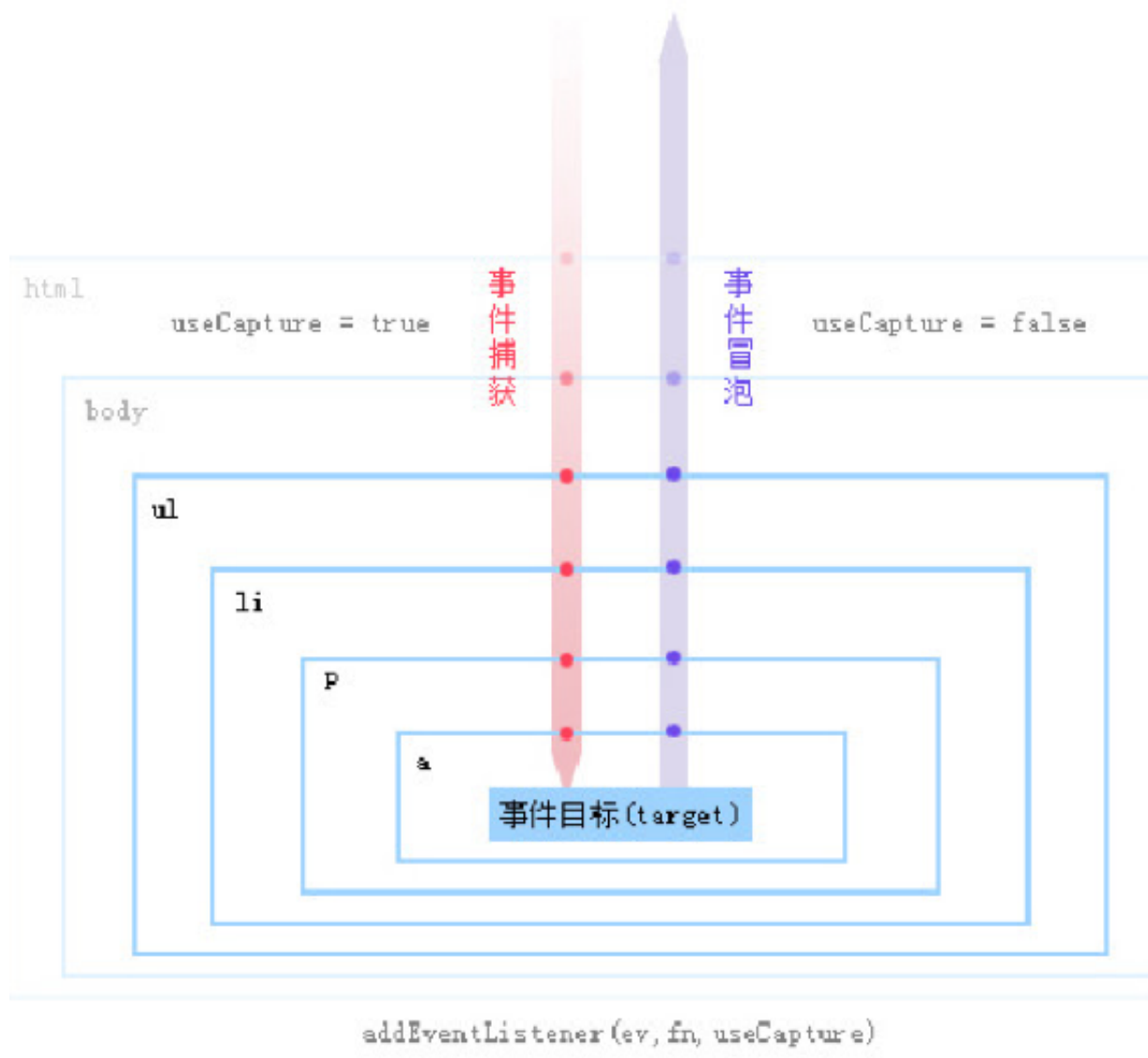
方式2：对象.attachEvent("事件类型","函数","事件流");**针对ie浏览器的,事件类型：加on**

```
38 <script>
39     var outer = document.querySelector("#outer");
40     var inner = document.querySelector("#inner");
41     var core = document.querySelector("#core");
42     outer.attachEvent("onclick",function(){
43         alert("ie事件!");
44     })
```

事件流有两种：

- 1、冒泡型（事件由内到外），默认值：false
- 2、捕捉型（事件由外到内）值：true

JS事件捕获与事件冒泡原型图



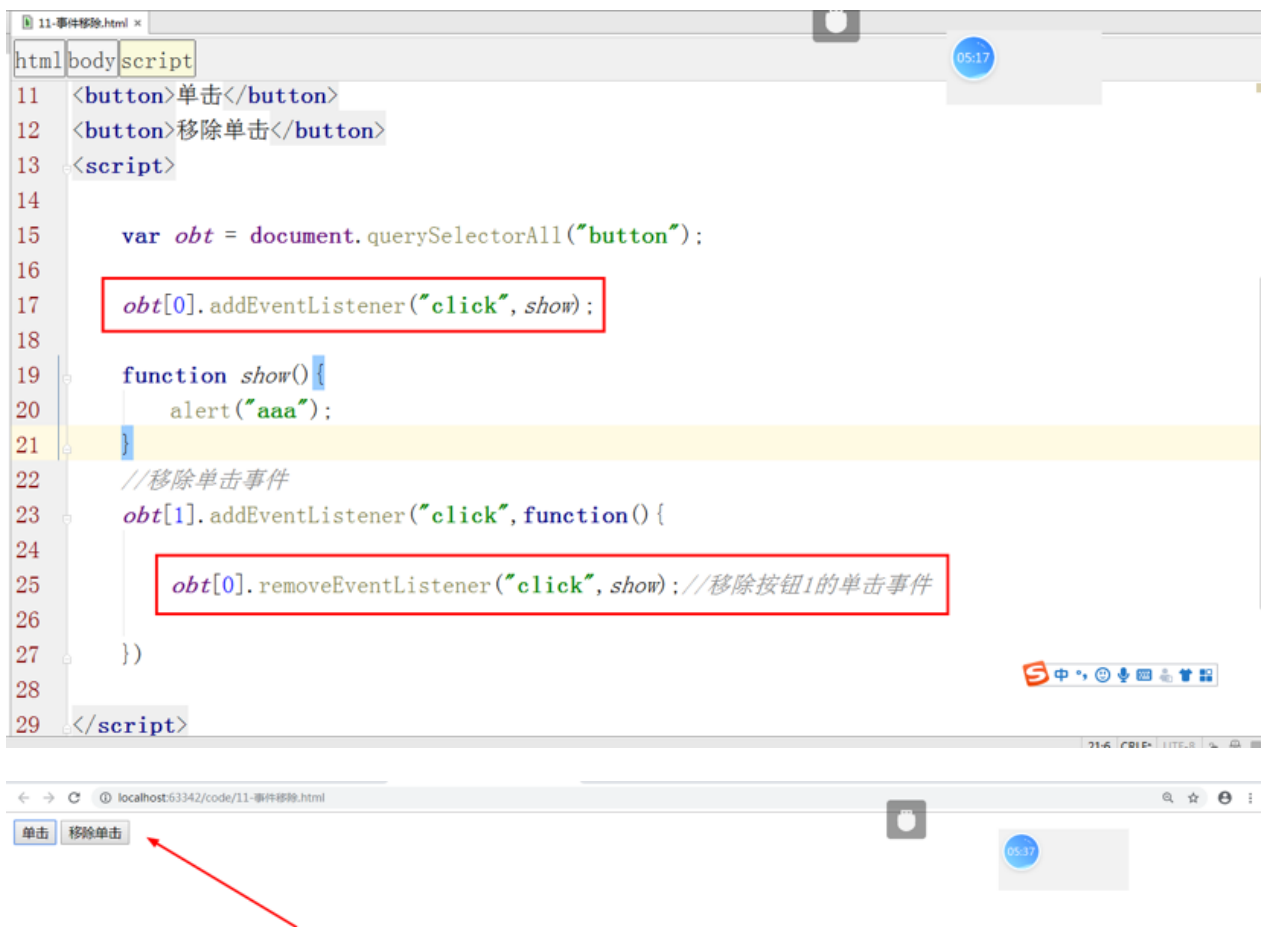
在js脚本中，通过`addEventListener`函数绑定的事件称为是DOM2级事件。

语法：元素.`addEventListener`(type,listener,useCapture)

type:事件类型。【没有on! 没有on! 没有on! 】 listener:监听函数，绑定的函数 useCapture:是否使用捕获机制。如果不写，默认值为false false:冒泡机制 true: 捕获机制 注意:DOM2级事件可以绑定多个函数，执行顺序按照函数书写的顺序。

DOM2级事件的移除方式：

node.removeEventListener(type,外部函数名,useCapture)



2.3.5. Dom2事件的传参数



2.4. DOM0和DOM2的区别

2.4.1. DOM0如果你写了多个事件，只应用最后一个！

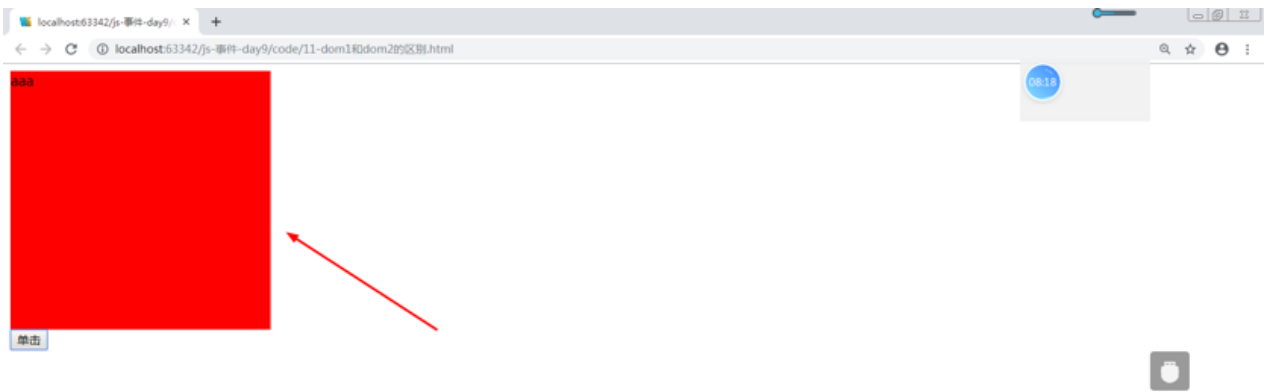
2.4.2. Dom2如果您写了多个事件，它会都应用

实例：单击按钮，给div标签添加样式，宽，高，背影颜色分别定义三个函数，分别使用DOM0和dom2的方式来做。

```
10 <body>
11 <div>aaa</div>
12 <button onclick="f1();f2();f3()">单击</button>
```

```
11-dom1和dom2的区别.html x 07:52
html body script
13 <script>
14   var obt = document.querySelector("button");
15   var odiv = document.querySelector("div");
16   //dom1事件的使用
17   /*
18      obt.onclick = f1;
19      obt.onclick = f2;
20      obt.onclick = f3;
21   */
22
23   function f1() {
24       odiv.style.width = "300px";
25   }
26   function f2() {
27       odiv.style.height = "300px";
28   }
29   function f3() {
30       odiv.style.background= "red";
31   }
```

```
22
23   function f1() {
24       odiv.style.width = "300px";
25   }
26   function f2() {
27       odiv.style.height = "300px";
28   }
29   function f3() {
30       odiv.style.background= "red";
31   }
32   //dom2事件的用法
33   obt.addEventListener("click", f1);
34   obt.addEventListener("click", f2);
35   obt.addEventListener("click", f3);
36 </script>
```



2.5. 怎样阻止冒泡

stopPropagation(): 针对非ie浏览器阻止冒泡,阻止事件的派发

cancelBubble=true: 针对ie浏览器

```
html body script
38 <script>
39     var outer = document.querySelector("#outer");
40     var inner = document.querySelector("#inner");
41     var core = document.querySelector("#core");
42
43     core.onclick = function() {
44         alert("core!");
45         // event.stopPropagation();//方法1-阻止冒泡
46         //event: 是js的内置事件对象
47         event.cancelBubble = true;//方法2-阻止冒泡
48     }
49     inner.onclick = function() {
50         alert("inner!");
51         // event.stopPropagation();//方法1-阻止冒泡
52         event.cancelBubble = true;//方法2-阻止冒泡
53     }
54     outer.onclick = function() {
55         alert("outer!");
56     }
57 }
```

```
html body script
38 <script>
39     var outer = document.querySelector("#outer");
40     var inner = document.querySelector("#inner");
41     var core = document.querySelector("#core");
42
43     core.onclick = function ea {
44         alert("core!");
45         //ea:代表event事件的一个简写
46         ea.stopPropagation();
47         //event.stopPropagation();//方法1-阻止冒泡
48         //event: 是js的内置事件对象
49         //event.cancelBubble = true;//方法2-阻止冒泡
50     }
```

2.6. 怎样阻止默认行为

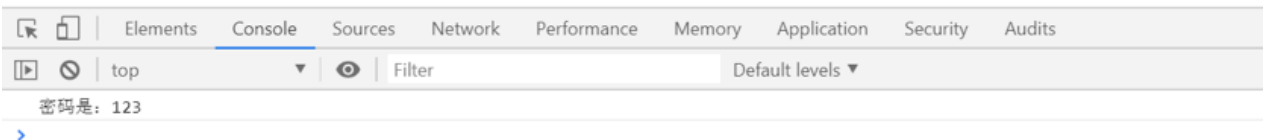
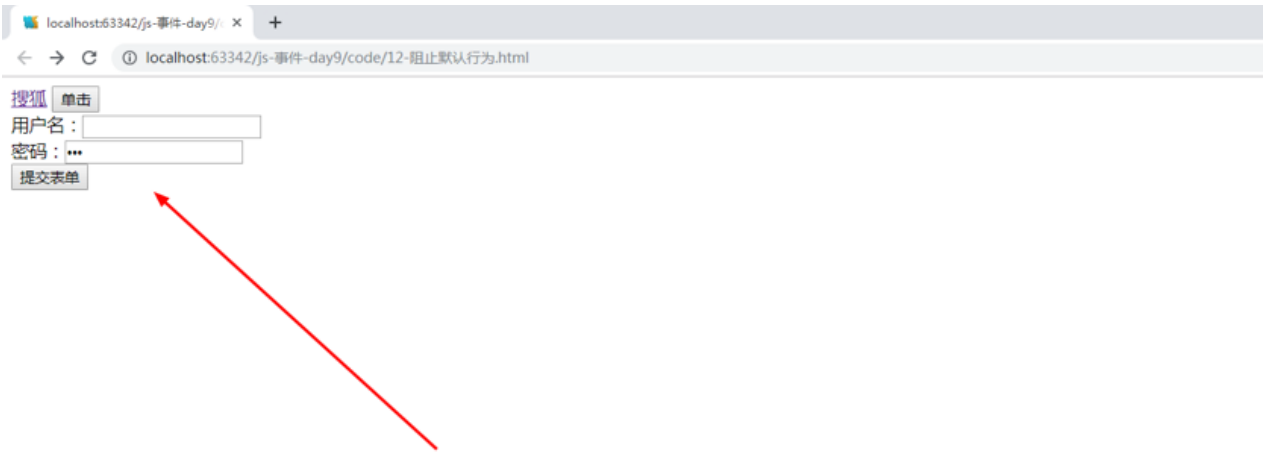
preventDefault()

returnValue = false

```
html body script
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4     <meta charset="UTF-8">
5     <title></title>
6     <style>
7
8     </style>
9 </head>
10 <body>
11 <a href="http://www.sohu.com">搜狐</a>
12 <button>单击</button>
13 <form action="reg.php" method="post" name="form1">
14     用户名: <input type="text" name="username"/><br/>
15     密码: <input type="password" name="pwd"/><br/>
16     <input type="submit" value="提交表单"/>
17 </form>
```

```
11-dom1和dom2的区别.html x 12-阻止默认行为.html x 31:46
html body script
18 <script>
19   var oa = document.querySelector("a");
20   var oform = document.querySelector("form");
21   var username = document.getElementsByName("username")[0];
22   oform.onsubmit = function() {
23     //if(document.form1.username.value == "")
24     //if(document.forms[0].username.value == "")
25     if(username.value == "") {
26       alert("哥们，用户名不能为空！");
27       document.form1.username.focus(); //获取焦点
28       event.preventDefault(); //阻止默认提交
29     }
30     console.log("密码是: " + document["form1"]["pwd"]["value"]);
31   }
32 }
```

```
// oa.onclick = function() {
//   return false;
// }
oa.addEventListener("click", function(e) {
  //event.preventDefault(); //阻止默认链接-方法1
  //e.preventDefault();
  e.returnValue = false; //阻止默认链接-方法2
});
```



2.7. 实例：单选按钮选颜色

```
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4   <meta charset="UTF-8">
5   <title></title>
6 </head>
7 <body>
8 <div>请选择您喜欢的颜色: </div>
9 <input type="radio" value="red" name="color"/>红色<br>
10 <input type="radio" value="green" name="color" checked="checked"/>绿色<br>
11 <input type="radio" value="blue" name="color"/>蓝色<br>
12 <button>提交颜色</button>

13 <script>
14   var ocolor = document.getElementsByName("color");//数组-获取单选按钮组
15   var obt = document.querySelector("button");
16   var odiv = document.querySelector("div");
17   obt.addEventListener("click", choiceColor);
18
19   function choiceColor() {
20     for(var i = 0; i < ocolor.length; i++) {
21       if(ocolor[i].checked) {
22         alert("你喜欢的颜色是: "+ocolor[i].value);
23       }
24     }
25   }
26 </script>
27 </body>
28 </html>
```

localhost:63342 显示
你喜欢的颜色是: green

2.8. 实例：多选按钮选颜色

```
html head
3 <head lang= en >
4   <meta charset="UTF-8">
5   <title></title>
6 </head>
7 <body>
8 <div>请选择您喜欢的颜色(多选): </div>
9 <input type="checkbox" value="red" name="color"/>红色<br>
10 <input type="checkbox" value="green" name="color" checked="checked"/>绿色<br>
11 <input type="checkbox" value="blue" name="color"/>蓝色<br>
12 <input type="checkbox" value="yellow" name="color" checked="checked"/>黄色<br>
13 <input type="checkbox" value="orange" name="color"/>橙色<br>
14 您喜欢的颜色有: <input type="text" class="re"/>
15 <button>提交颜色</button>
```

```
html head
16 <script>
17   var ocolor = document.getElementsByName("color");//数组-获取多选按钮组
18   var obt = document.querySelector("button");
19   var odiv = document.querySelector("div");
20   var ore = document.querySelector(".re");
21   var str = "";
22   obt.addEventListener("click", choiceColor);
23
24   function choiceColor() {
25     for(var i = 0; i < ocolor.length; i++) {
26       if(ocolor[i].checked) {
27         str+=ocolor[i].value+","; //str = str+ocolor[i].value
28       }
29     }
30     ore.value = str; //将累计的结果赋值给文本框
31   }
32 </script>
33 </body>
34 </html>
```

请选择您喜欢的颜色(多选) :

☐ 红色

☒ 绿色

☒ 蓝色

☒ 黄色

☐ 橙色

您喜欢的颜色有:

2.9. 实例：树形菜单阻止冒泡

```
html body script
4 <meta charset= UTF-8 >
5 <title></title>
6 <style>
7     ul{
8         display: none;
9     }
10 </style>
11 </head>
12 <body>
13 <div>
14     <h2>商品展示</h2>
15     <ul>
16         <li>11</li>
17         <li>22</li>
18         <li>33</li>
19         <li>44</li>
20         <li>55</li>
21     </ul>
22 </div>
```

```
html body script
24 <script>
25     var oh2 = document.querySelector("h2");
26     var oul = document.querySelector("ul");
27     //单击空白处让子菜单隐藏
28     document.addEventListener("click",function(){
29         oul.style.display = "none";
30     });
31     oh2.addEventListener("click",shownav);
32
33     function shownav(){
34         if(oul.style.display == "none"){
35             oul.style.display = "block";
36             event.stopPropagation();//阻止冒泡
37         }
38         else{
39             oul.style.display = "none";
40         }
41     }
42 }
```

localhost:63342/code/6-下拉菜单.html

商品展示

- 11
- 22
- 33
- 44
- 55

3. IE浏览器中的事件

3.1. 事件绑定

在js脚本中，通过attachEvent函数绑定事件

语法：元素.attachEvent(type,listener)

type:事件类型。【有on! 有on! 有on! 】 listener:监听函数，绑定的函数 注意:如果绑定多个函数，按照函数书写的倒叙执行。

3.2. 事件解除

IE下DOM2级事件的移除方式：

元素.detachEvent(type,listener)

ps：匿名函数无法被移除。

4. 事件兼容性问题

4.1. 兼容性问题

由于【IE浏览器中的事件绑定】和【非IE浏览器中的事件绑定】方式方法都有所不同。所以单一的某种函数都不能完美解决不同浏览器下的方法绑定问题。

4.2. 兼容性封装

```
10 <body>
11 <button>单击</button>
```

```
13 //封装一个函数，让各个浏览器都能给元素添加事件
14 //element:元素, type:事件类型, method:函数方法
15 var obt = document.querySelector("button");
16 function addEvent(element, type, method){
17     if(element.addEventListener){
18         element.addEventListener(type, method);
19     }
20     else if(element.attachEvent){
21         element.attachEvent("on"+type, method);
22     }
23     else{
24         element["on"+type] = method;//obt["onclick"] = function() {}
25     }
26 }
27 function show() {
28     alert("我是show方法!");
29 }
30 addEvent(obt, "click", show);
31 </script>
```

5. 鼠标事件

5.1. 关于鼠标事件

鼠标单击是触发：click 鼠标双击是触发：dblclick

鼠标按下时触发：mousedown 鼠标抬起时触发：mouseup

鼠标移动时触发：mousemove 鼠标移入时触发(不冒泡)：

mouseenter

鼠标移出时触发(不冒泡)：mouseleave 鼠标移入时触发(冒泡)：

mouseover

鼠标移出时触发(冒泡)：mouseout

语法：元素.on+鼠标事件名称 = 调用函数

例如：d1.ondblclick = function () { console.log('这是d1');}

5.2. mouseover和mouseenter的区别

```

23
24 <div class="box1">
25   <p>onmouseenter: <br> <span id="demo1">鼠标移动到我这!</span></p>
26 </div>
27
28 <div class="box2">
29   <p>onmouseover: <br> <span id="demo2">鼠标移动到我这!</span></p>
30 </div>
31

```

```

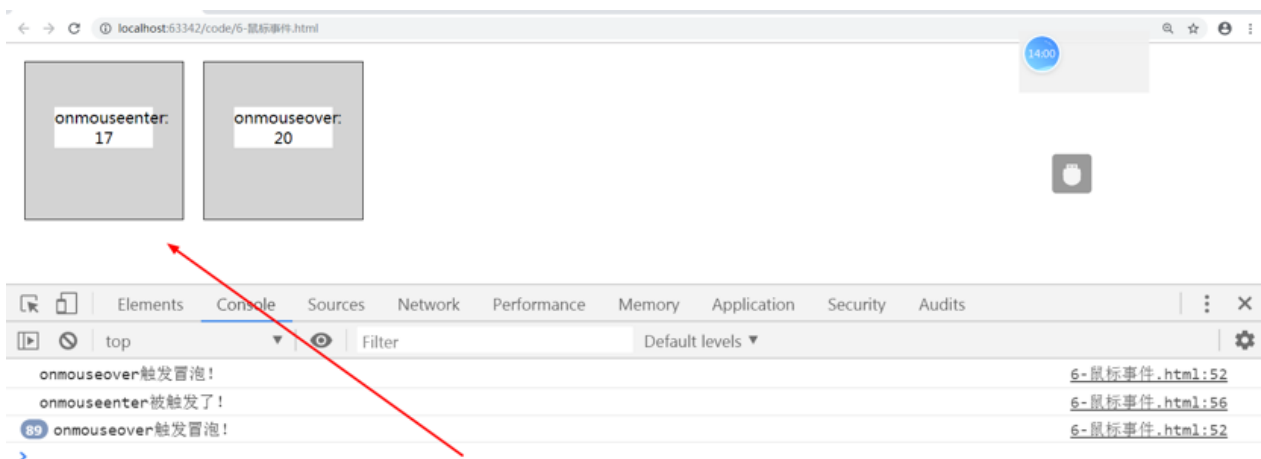
html body script
5   <title></title>
6   <style>
7     div {
8       width: 100px;
9       height: 100px;
10      border: 1px solid black;
11      margin: 10px;
12      float: left;
13      padding: 30px;
14      text-align: center;
15      background-color: lightgray;
16    }
17    p {
18      background-color: white;
19    }
20  </style>
21 </head>

```

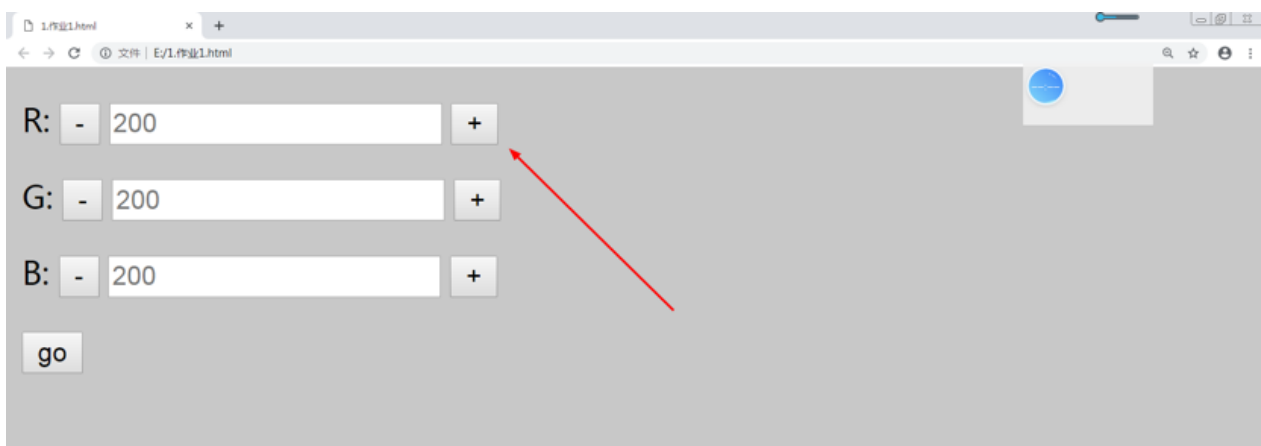
```

html body script
32 <script>
33   var box1 = document.querySelector(".box1");
34   var box2 = document.querySelector(".box2");
35   var span1 = document.querySelector("#demo1");
36   var span2 = document.querySelector("#demo2");
37   var x = 0, y = 0;
38   //mouseenter进入子元素的时候, 不实现累加
39   //mouseenter:没有冒泡
40   box1.onmouseenter = function () {
41     y++;
42     span1.innerHTML = y;
43   }
44   //mouseover进入子元素的时候, 会实现累加
45   //mouseover:有冒泡
46   box2.onmouseover = function () {
47     x++;
48     span2.innerHTML = x;
49   }
50

```



5.3. 实例：调色板



5.4. 实例：div跟随鼠标移动



```
7-事件的属性坐标.html x 8-跟随鼠标移动.html x
html body div. box
16 </head>
17 <body>
18 <div class="box">
19
20 </div>
21 <script>
22     var odiv = document.querySelector(".box");
23     document.onmousemove = function(e) {
24         odiv.style.left = e.clientX - odiv.offsetWidth/2 + "px"; //块距离左边的距离
25         odiv.style.top = e.clientY - odiv.offsetHeight/2 + "px"; //块距离顶部的距离
26     }
27
28 </script>
29 </body>
30 </html>
```

