JavaScript进阶

---JS事件及事件流

河北师范大学软件学院
Software College of Hebei Normal University

内容提纲

- **JS事件及事件对象**
- **JS事件响应**
- **JS事件流（冒泡、捕获）**

JS事件及事件对象

- 什么是JS事件
  - JS事件是浏览器或用户自身执行的某种动作（包括前端中的事件、Node中的事件等）
  - 前端事件主要包括BOM或DOM中发生的特定的交互
  - 常见事件（load、click、mouseover、keydown、keyup等）

```
window.onload = function () {
    console.log("window onload");
    var div2 = document.getElementById("div2");
    div2.onclick = function () {
        console.log("div2 click");
    }
}
```


参见实例index01.html 和 demo01.js

JS事件及事件对象

- 事件对象（包含事件中相应的信息）
  - 当事件发生时会产生对应的事件对象（如：鼠标事件对象、键盘事件对象等）
  - 事件对象（Event）包含对应事件的相关信息（如触发的元素、坐标信息、键值信息等）
  - 理解事件对象的继承关系（例如：Event--UIEvent--MouseEvent）

```
window.onload = function (e) {
    console.log("e:", e);
    var div1 = document.getElementById("div1");
    var eventHandler = function (e) {
        console.log(e);
        console.log(e.clientX,e.clientY);//坐标信息
    }
    div1.onclick = eventHandler;
}
```

除了默认的事件外，用户也可自定义事件对象

本节介绍基本的事件对象属性，与事件流相关的属性、方法参见事件流章节

参见实例index02.html 和 demo02.js

内容提纲

- JS事件及事件对象
- JS事件响应处理
- JS事件流（冒泡、捕获）

JS事件响应处理

- 事件响应处理方式
  - HTML事件响应处理
  - DOM0级事件响应处理
  - DOM2级事件响应处理

```
<div id="div1" onclick="div1click()">

</div>
<div id="div2" ondrag="console.log('drag')">

</div>
```


参见实例index03.html

JS事件响应

- 事件响应方式
  - HTML事件响应处理
  - DOM0级事件响应处理（比HTML事件响应处理的去耦合性要好很多）
  - DOM2级事件响应处理

```
window.onload = function (e) {
    var div1 = document.getElementById("div1");
    var eventHandler = function (e) {
        console.log(e.clientX,e.clientY);
    }
    div1.onclick = eventHandler;
    //div1.onclick = null;//取消事件响应
}
```


参见实例index03.html 和 demo03.js中的测试2

JS事件响应

- 事件响应方式
  - HTML事件响应处理
  - DOM0级事响应件处理
  - DOM2级事件响应处理（比DOM0级事响应件处理更强，可以重复，支持自定义事件）

```
window.onload = function (e) {
    var div1 = document.getElementById("div1");
    var eventHandler = function (e) {
        console.log(e.clientX,e.clientY);
    }
    div1.addEventListener("click",eventHandler);
    //div1.removeEventListener("click",eventHandler);
}
```


参见实例index03.html 和 demo03.js中的测试3

JS事件响应

- 事件响应的兼容性问题
  - 老版本IE不支持addEventListener、removeEventListener
  - 老版本IE支持attachEvent、detachEvent
  - 一些更特殊的浏览器可能两者都不支持

```
function addEvent(ele,type,hander){
    if(ele.addEventListener){
        ele.addEventListener(type,hander,false);
    }else if(ele.attachEvent){
        ele.attachEvent('on'+type,hander);
    }else{
        ele['on'+type]=hander;
    }
}
```

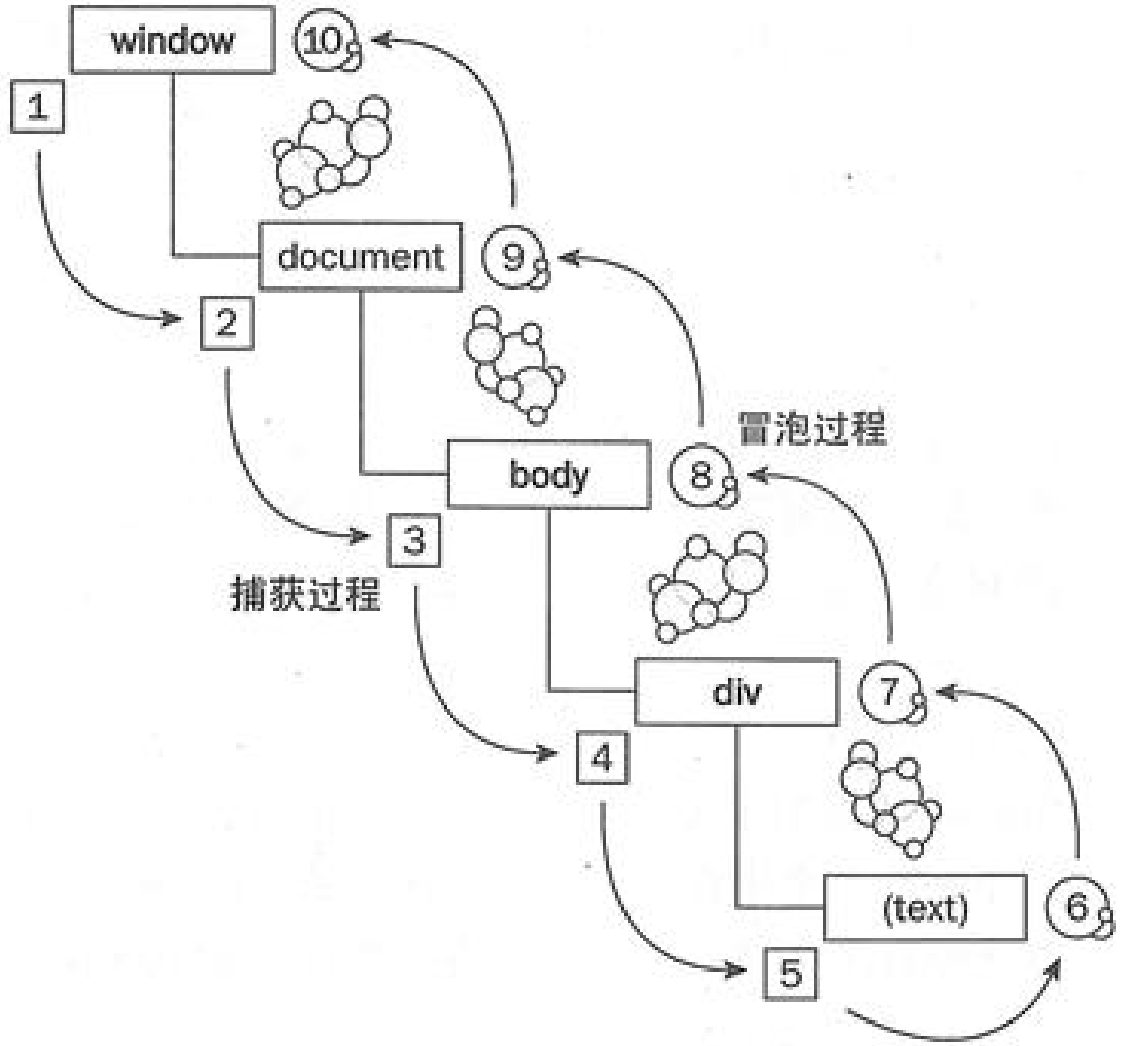
解决去除事件监听兼容性问题，与解决添加事件监听兼容性问题的方法类似：
写一个可写一个removeEvent函数，然后
将addEventListener改为removeEventListener
将attachEvent改为detachEvent
思考：最后else里怎么改

内容提纲

- JS事件及事件对象
- JS事件响应
- JS事件流（冒泡、捕获）

JS事件流

- 什么是事件流（冒泡方式，捕获方式）



事件流指从页面中接收事件的顺序包括（冒泡流和捕获流）

IE 提出的是冒泡流，而 Netscape 提出的是捕获流

当页面中发生某种事件（比如鼠标点击，鼠标滑过等）时，子元素和父元素都会接收到该事件，具体顺序是怎样的呢？冒泡和捕获则描述了两种不同的顺序

冒泡：从最具体的节点到最不具体节点
捕获：从最不具体的节点到最具体节点

参见实例index04.html 和 demo04.js

JS事件流

- 再谈DOM2级事件响应（在不同阶段，对事件的响应）

```
window.onload = function (e) {
    var div1 = document.getElementById("div1");
    var div2 = document.getElementById("div2");
    div1.addEventListener("click",function (e) {
        console.log("div1 click");
    },false);//改成true会怎样
    div2.addEventListener("click",function (e) {
        console.log("div2 click");
    },false);//改成true会怎样
}
```


参见实例index05.html 和 demo05.js

JS事件流

- 事件对象的属性及方法与事件流（查看Event.prototype）
  - 事件对象的target属性、bubbles（表示该事件是否冒泡）
  - 事件对象的stopPropagation( )方法（用于阻止事件冒泡）

```
div1.addEventListener("click",function (e) {
    console.log("div1 click--red");
    console.log("target:",e.target);
    console.log("this:",this);
    console.log(e.bubbles,e.cancelable,e.cancelBubble);
    e.stopPropagation();
    e.preventDefault();
},false);//改成true会怎样
```

stopPropagation 阻止事件冒泡

preventDefault 阻止默认响应

参见实例index06.html 和 demo06.js

Thank You !

河北師范大学软件学院
Software College of Hebei Normal University

补充：事件相关参考

- JS事件对象兼容性问题（对于老IE浏览器）
  - srcElement属性 用于获取事件的目标对象
  - cancelBubble属性 用于阻止事件冒泡
  - returnValue属性 用于阻止事件的默认行为
  - 解决事件对象兼容问题的方法（同解决事件响应的兼容问题）
- JS事件参考链接
  - https://developer.mozilla.org/zh-CN/docs/Web/API/Event/Event
  - https://developer.mozilla.org/zh-CN/docs/Web/API/EventTarget

作业

- 查看慕课网上的视频（1-3章必看，4、5章选看）

  https://www.imooc.com/learn/138

- 安装Node.js（安装成功后，查看Node和npm的版本）