# DOM 和 BOM

## 事件绑定和解绑（要对应）

### 1.DOM 0 1 2 3

dom 0 和 dom2 有自己独立的事件绑定和解绑方式

dom0事件所有的浏览器都是通用的

dom2事件高级浏览器和低级浏览器用的方法不同

dom1和dom3 没有事件绑定方式;

### 2.Dom0事件绑定和解绑

不可以同时添加同一类事件多次，如果添加后面覆盖前面；

dom0事件解绑 本质上就是把事件回调函数和事件对象的事件属性断开指向；

box.onclick = null;

### 3.Dom2事件绑定和解绑

可以添加同一类事件多次

高级浏览器和ie绑定方式（兼容封装还有解绑）

//dom2事件解绑，函数必须放在外面去定义,如果不放在外面定义，直接在参数当中写匿名函数表达式，那么绑定和解绑

传的回调函数不是同一个，所以解绑不了；

## 事件流（事件传播）\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 1.事件流的发展：

1）捕获事件流（网景） 最终很少用几乎不用

2）冒泡事件流（ie） 最终我们所用的事件传播都是冒泡

3）标准DOM事件流 这个是我们现用的标准事件流，里面包含三个阶段： 有捕获 再去获取目标元素 最后冒泡，这个三个阶段当中的捕获和冒泡可以由程序员自己选择。但是通常情况我们都是使用默认 （冒泡）；

### 2.事件冒泡和事件捕获

案例演示冒泡和捕获

### 3.阻止事件冒泡

事件流（事件传播）每个事件都是必不可免的，也就是说每个事件都会进行冒泡；冒泡的情况下，如果是父子元素都添加了相同的事件监听，那么事件处理会从内到外依次执行；但是有些时候，我们确实是这样的结构，但是又不想让父元素事件进行处理；此时：我们就要用到阻止事件冒泡；

阻止冒泡的时候，想在哪个元素阻止，需要在哪个元素的事件回调处理函数当中加上这一行；

event.stopPropagation();//专门用来阻止事件冒泡用的；

## 事件委派\*\*\*\*\*\*\*\*\*\*\*\*

### 1.什么是事件委派

事件委派过程当中依赖了事件冒泡；

阻止事件冒泡//是为了解决冒泡给我们带来的困扰

事件冒泡的好处就是可以进行事件委派（事件委托，事件代理）；把子元素的事件添加给父（祖先）元素,把子元素发生的事件委托给父元素进行处理；

### 2.事件委派用法，

1）什么时候用：

1、当一个元素内部儿子很多，并且每个儿子都要添加相同的事件的时候，我们可以使用事件委派来提高效率；

2、出现新添加的东西，并且新添加的东西要和老的拥有同样的行为；此时我们就想事件委派；不用事件委派，老的身上会有想要的行为，而新添加的没有；

2）用法：

事件委派的做法： 给爹添加事件监听，不给元素本身添加，事件发生后通过event去找真正发生事件的目标元素进行处理；

3）好处

事件委派的好处：可以大大降低内存的占用，并且可以提高效率。

4）总结

事件委派其实是借用事件冒泡去做的，因为事件冒泡导致内部所有的元素发生事件都会冒泡到祖先身上，我们不在子元素身上去添加事件监听和处理，而是在共同的祖先身上去添加，让祖先去处理子元素发生的事件；祖先去处理其实就是通过事件对象当中的target 去获取到真正发生事件的子元素；对子元素进行处理；

### 3.两对移入移出事件的区别

onmouseenter onmouseleave

如果是一个父子元素模型，对父元素添加移入和移出，当鼠标移入父元素里面的子元素的时候， 事件并没有移出然后再移入。也就是说事件元素没有切换；

onmouseover onmouseout

如果是一个父子元素模型，对父元素添加移入和移出，当鼠标移入父元素里面的子元素的时候，事件会移出然后再移入。也就是说事件元素会有切换；事件委派的时候，必须使用这一对；

## bom

### 1.window对象（bom的顶级对象）

var a; window.a

function fn(){} window.fn()

document window.document

window.onload

window.onresize

window.onscroll

### 2.location

window.location可以让用户获取当前页面地址以及重定向到一个新的页面。window.location.href 可以读也可以写，写的时候相当于转向另外一个页面

### 3.history

window.history对象包含浏览器的历史记录，window可以省略。这些历史记录以栈的形式保存。页面前进则入栈，页面返回则出栈。

### 4.navigator

是一个只读对象，它用来描述浏览器本身的信息，包括浏览器的名称、版本、语言、系统平台、用户特性字符串等信息。

### 5.screen

提供了用户显示屏幕的相关属性，比如显示屏幕的宽度、高度，可用宽度、高度。

## event 对象

### 1.event概念，作用

系统给我们封装的，任何事件都会有这个event对象，这个对象当中封装了和这个事件相关的一切信息；

### 2.event兼容性处理

如果是高级浏览器去调用函数的回调函数，它会把事件对象封装好传给回调函数的第一个形参;

如果是低版本浏览器去调用，它会把事件对象封装好作为window的一个属性 window.event;

所以我们在去拿事件对象的时候，要兼容性去拿

event = event || window.event;

### 3.目标元素节点兼容处理

event.target || event.srcElement

### 4.三种鼠标位置

clientX & clientY

拿的是鼠标相对视口的 水平距离和垂直距离 相对的是视口的左上角（以视口左上角为原点）

pageX pageY

拿的是鼠标相对页面的 水平距离和垂直距离 相对的是页面的左上角（以页面左上角为原点）

offsetX offsetY

拿的是鼠标相对自身元素的 水平距离和垂直距离 相对的是自身元素左上角（以自身元素左上角为原点）

## 案例

鼠标跟随

学生管理系统

作业

1. BOM操作

2. 事件委托练习