事件基础1

**目录：**

1. 什么是事件，事件的作用，事件分类

2. 事件函数、事件监听器、事件处理器

2.1 事件触发原理

2.2 事件对象

3. 触发事件的两种方法，和区别

3.1 事件绑定——on的事件名

3.2 事件监听——addEventListen

4. 第二种方法addEventListen事件监听，所需注意的事项，存在的坑

4.1 坑1——同一个函数绑定多次，只处理一次

4.2 坑2——绑定事件套绑定事件，如果不是传函数，内部绑定事件会触发多次

4.3 坑3——兼容性，ie8及以下不兼容，用attachEvent来处理，这个this指向window

5. 取消事件的方法——有名函数，匿名函数

5.1 on

5.2 addEventListen

6. addEventListen的实现原理

1. 什么是事件，事件的作用，事件分类

1.1 什么是事件

事件是是文档或浏览器窗口发生一些特定交互的交互瞬间；

事件的作用是产生交互，能够使页面（浏览器）与其他的（比如人）进行互动；

js可以侦听这些交互，在交互发生时执行相应的一些代码；

在js中可以对各种不同的事件进行监听（侦听、绑定），当该事件发生的时候可以通过js做某些不同的处理。

比如，当用户点击了页面中某个元素的时候，就会触发一个事件（点击事件，不同的交互可能产生不同的事件），js可以监听该元素的点击事件，当改元素的点击事件绑定了一个函数，绑定的函数就会被执行。

1.2 事件分类

参考文档：https://developer.mozilla.org/en-US/docs/Web/Events

资源事件

网络事件 online offline

焦点事件

CSS动画事件 animation

表单 reset submit

视图事件 resize scroll

拖拽事件 drag

值改变事件、处理进度 hashchang input CheckboxsStateChange)键盘和点击

自定义事件

MouseEvent 与**鼠标**相关的事件

click

dblclick

mouseover

mouseenter

mouseout

mouseleave

mousemove

mouseup

mousedown

mousewheel 鼠标滚轮

KeyboardEvent 与**键盘**相关的事件

ProcessEvent 与**处理进度**有关的事件

2. 事件函数、事件监听器、事件处理器——事件触发原理

2.1 事件发生有两个重要的因素

触发事件主体——什么元素身上

触发事件类型

浏览器内置了很多各种各样的不同事件，当某件事件发生的时候，浏览器会监听这些事件交互，并且触发一些定义好的事件。我们可以为这些事件绑定一些行为，我们可以在某个事件发生的时候做某些事情；

2.2 事件对象

如果一个函数是被**事件调用**的，那么这个函数的执行是由浏览器在触发该事件的时候执行，内部在调用该函数的时候，会自动传入一个参数，这个参数是一个对象，我们把这个参数称为：事件对象；

如果一个函数是事件调用的函数，那么这个函数的第一个参数就是事件对象，这个对象里面包含了与当前触发的事件有关的所有详细的细节数据；
事件对象的数据有很多
clientX,clientY——当前事件发生的时候，鼠标相对于屏幕可视区的距离

**事件对象的兼容性**

标准下：使用事件函数的第一个参数

ie：是一个全局的对象 -> event

具体：

低版本ie：全局event，没有参数event

高版本ie9+/chrome：全局event，同时参数event也有

firefox：只有参数event

3. 触发事件的两种方法，和区别

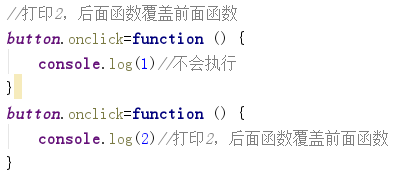
**3.1 事件绑定**

**DOM0**，这种方式是的DOM的第0版的时候定义的

**元素对象.on事件名称 = 函数**

与事件对应的属性，只能接收两种类型的值——**null**和**function**，如果复制的是其他值的话，会转出null。

**特征：**我们通过on属性的方式绑定的函数只能同时**绑定一个**，后面绑定的会覆盖前面绑定的。



解释：在对象上有一些以on开头的属性，这些属性有一些特殊的作用，默认情况下这些为null，我们给可以这个属性赋值函数，大概我们给某个对象的某个on属性赋值一个函数的时候，那么当这个对应的事件被触发的时候，浏览器会调用该对象的对应的on属性，如果该属性是函数，则执行。

注意：我们给某某添加一个事件 这说法不太准确；

因为事件能不能发生，不是div.onclick决定的，它能决定的只是事件发生之后会处理的什么事件。某个行为出现了，对应的事件是一定会触发的，我们只能给这个事件绑定对应的要处理函数 ，我们把这个过程称为：事件绑定、事件函数绑定、事件监听、事件侦听，同时我们把这个与事件有关的函数称为：事件函数、事件侦听器，事件处理器 event handler。

**3.2 事件监听**

**DOM2**，这种方式是的DOM的第2版的时候定义的

**对象.addEventListener('事件名称', 绑定的函数，捕获/冒泡)**

true的永远先执行，同一个阶段的，要看哪个先绑定，先绑定先执行

**特征：**给一个对象的同一个事件同时**绑定多个**函数；

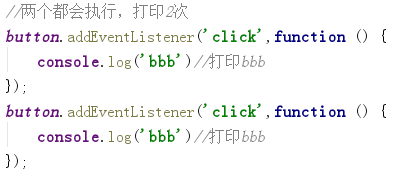
addEventListener绑定的函数与on方式绑定函数不冲突；

addEventListener绑定的函数必须是不同的（地址不同），对于相同的函数只能绑定一个。

4. 第二种方法addEventListen事件监听，所需注意的事项，存在的坑

**4.1 坑1**——同一个事件相同函数只能绑定一次，如果绑定多次，只处理一次

情况1：当传的是匿名函数内容相同的时候，两个都会执行，因为只是内容相同，实质上是两个不同的对象。



情况2：当传入的是申明函数，只会执行一次，因为对象不能复制，只能引用



**4.2 坑2**——绑定事件套绑定事件，如果不是传函数，内部绑定事件会触发多次





**解决方法：**不用匿名函数绑定，将绑定函数提取出来，封装成函数进行调用



**4.3 坑3——兼容性**，ie8及以下不兼容，用attachEvent来处理，这个this指向window

attachEvent和addEventListen

功能相似，但是也有一些不同的地方：

事件名称：

addEventListener 没有on

attachEvent 有on

执行顺序：

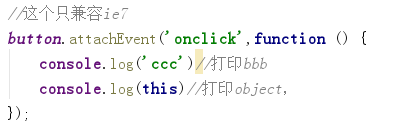
addEventListener 正序执行：队列

attachEvent 倒叙执行：栈

this的指向：

addEventListener 指向元素

attachEvent 指向window



5. 取消绑定事件的方法——有名函数，匿名函数

5.1 on

无论是有名函数还是匿名函数，都是用null。

***box1***.**onclick**=**null**;

5.2 addEventListen

5.2.1 有名函数

***box1***.removeEventListener(**'click'**,*fn2*)

5.2.2 匿名函数

匿名函数是无法直接取消绑定的，我们可以通过自定义函数，自定义一个和addEventListen一样的事件，以后再说

6. addEventListen的实现原理

大致过程：

js监听button按钮addEventListen方法的click事件上面绑定的事件，把不同的事件放到exe函数的数组中，组成fns数组。并且把不同函数中的事件对象存到e对象中，e的默认只有cancelBubble：false。

循环fns数组，执行数组里面的每个函数。

e中没有阻止冒泡事件，就往上冒泡。

