官方对于Events事件模块的手册我觉得写的不够详细和透彻，我们结合代码来解读一下node.js的Events模块到底如何运用的。

我们先看如下代码：

var events = require("events");

var x =new events.EventEmitter();

x.on('newListener',function(event, listener){

console.log(event + ' : '+listener);

});

x.on('y', function(a,b,c){

console.log('it\'s work1!'+a+b+c);

});

x.on('y', function(a,b,c){

console.log('it\'s work2!'+a+b+c);

});

x.on('y', function(a,b,c){

console.log('it\'s work3!'+a+b+c);

});

x.emit('y','111','222', '3333');

var z = x.listeners('y');

console.log(z+'');

代码很简单实例化events.EventEmitter这个类，然后就可以使用手册上的一些方法了。例如：

var x =new events.EventEmitter();

我们主要用到监听器和触发器

line1:x.on(事件名称a, 监听函数); //建立监听事件，当这个事件触发则执行监听函数

line2:x.emit(事件名称a, 参数1-n...); //用来触发某一个事件，然后将参数按顺序传给监听函数

所以当line2执行后，就相当于执行了 监听函数(参数1-n...) ，当然，你写了多个line1就执行多遍监听函数。

手册中的其他方法：

emitter.addListener(event, listener) ：建立事件监听俗称监听器，感觉还是监听器易懂一点

emitter.on(event, listener) ：同上

emitter.once(event, listener) ：建立一次性事件监听（监听器），执行listener后删除本监听器

emitter.removeListener(event, listener) ：从指定事件中除去某一个监听函数，说白了就是再触发这个事件，指定的监听函数不执行了。

emitter.removeAllListeners(event) ：删除所有对这个事件的监听函数，说白了就是再触发这个事件，没函数执行了

emitter.setMaxListeners(n) ： 设置这个emitter实例的最大事件监听数，默认是10个，设置0为不限制

emitter.listeners(event) ： 返回这个事件的监听函数的数组，例如返回：[function1, function2, function3]，修改会改变此事件的监听函数，例如emitter.listeners(event).length = 2;就相当于执行了：emitter.removeListener(event, function3) ;

emitter.emit(event, [arg1], [arg2], [...]) ：触发这个事件，让监听器工作的方法

emitter.on('newListener', function(event, listener){//dosomething}) ：当emitter实例有新的监听器建立，就返回事件名和监听函数

具体这个事件监听用在哪些场景中呢?

比如有一个秒杀活动，只有第一个点击按钮的人才有机会获取奖品，网站会记录本次活动有多少人点击这个按钮，同时又要根据用户ID记录此用户点过多少次这个按钮。

不使用事件监听代码可能这么写：

var prize=false; //false表示奖品未被领取

var isactive = true; //活动是否开启

if(有人点击抽奖按钮){

if(!isactive ) return false; //先判断活动是否已经关掉了

if(!prize){

getprize(此人id, 按钮id); //getprize是领奖的方法

prize = true;

}

addbutton(此人id, 按钮id); //此按钮点击统计+1

addclick(此人id, 按钮id); //此人点击此按钮统计+1

}

setTimeout(function(){

isactive = false;

}, 1000\*60\*60\*24); //活动时间1天，1天以后活动结束

如果用事件监听改写上面代码，看上去更清晰，而且可能效率更高一些。

var events = require("events");

var btclick =new events.EventEmitter();

btclick.once('bt0click',getprize) //getprize是领奖的方法，触发一次后，自动删除本触发器

btclick.on('bt0click', addbutton);//此按钮点击统计+1

btclick.on('bt0click', addclick);//此人点击此按钮统计+1

if(有人点击抽奖按钮){

btclick.emit('bt0click', 此人ID, 按钮ID);//等于触发了bt0click事件的所有监听器

}

setTimeout(function(){

btclick.removeAllListeners('bt0click'); //移除所有监听器

}, 1000\*60\*60\*24); //活动时间1天，1天以后活动结束

对于添加和删除监听器都非常方便，重要的是它不需要去加一些if判断，从而提高代码效率，而且貌似代码看上去更专业和装B一点，哈哈！