### 任务2：制作图片提示效果

#### 2.1 事件对象

JavaScript 在事件处理函数中默认传递了event 对象，也就是事件对象。但由于浏览器的兼容性，开发者总是会做兼容方面的处理。jQuery 在封装的时候，

解决了这些问题，并且还创建了一些非常好用的属性和方法。JQuery中的事件对象就是event 对象，通过处理函数默认传递接受，如：

//通过处理函数传递事件对象

$(“#btn”).bind(“click”, function (e) { //接受事件对象参数

alert(e);

});

event对象常用的属性如表所示。

表 event对象常用属性

|  |  |
| --- | --- |
| **属性名称** | **描述** |
| type | 获取这个事件的事件类型，例如：click |
| target | 获取绑定事件的DOM 元素 |
| data | 获取事件调用时传入的额外参数 |
| relatedTarget | 对于鼠标事件, 标示触发事件时离开或者进入的DOM元素 |
| currentTarget | 冒泡前的当前触发事件的DOM对象, 等同于this |
| pageX/pageY | 鼠标事件中, 事件相对于页面原点的水平/垂直坐标 |
| result | 上一个事件处理函数返回的值 |
| timeStamp | 事件发生时的时间戳 |
| which | 获取鼠标的左中右键(1,2,3)，或获取键盘按键 |

#### 2.2 事件冒泡

如果在页面中重叠了多个元素，并且重叠的这些元素都绑定了同一个事件，那么就会出现冒泡问题。

事件冒泡可能会引起预料之外的效果，为了解决这些问题，jQuery 提供了一个事件对象的方法：event.stopPropagation()；

这个方法设置到需要触发的事件上时，所有上层的冒泡行为都将被取消。

$('input').click(function () {

$("#msg").html($("#msg").html()+"按钮被触发了！<br/>");

e.stopPropagation();

});

当然还有一种简写方案代替，就是直接用return false代替e.stopPropagation()。