1. 鼠标事件
2. .hover();是mouseover、mouseout、mouseenter、mouseleave的集合

注意:在元素上移进移出切换其换色，一般通过2个事件配合就可以达到，这里用mouseenter与mouseleave，这样可以避免冒泡问题

二．表单事件

1. focusin(); 鼠标聚焦触发--->一般用于表单
2. focusout(); 鼠标失去焦点触发--->一般用于表单
3. focus和blur方法和focusin，focusout差不多区别是focus和blur不支持冒泡
4. change方法单表单value值发生改变时候或者下拉选项时触发
5. select方法 当 textarea 或文本类型的 input 元素中的文本被选择时，会发生 select 事件。
6. submit方法 提交表单是一个最常见的业务需求，比如用户注册，一些信息的输入都是需要表单的提交。同样的有时候开发者需要在表单提交的时候过滤一些的数据、做一些必要的操作（例如：验证表单输入的正确性，如果错误就阻止提交，从新输入）此时可以通过submit事件，监听下提交表单的这个动作

三．键盘事件

1.KeyPress主要用来接收字母、数字等ANSI字符。

On方法绑定事件

通过空格分离，传递不同的事件名，可以同时绑定多个事件，每一个事件执行自己的回调方法。 多个方法键值对的方式。

Off方法卸载事件

1. 根据on绑定事件的一些特性，off方法也可以通过相应的传递组合的事件名，名字空间，选择器或处理函数来移除绑定在元素上指定的事件处理函数。当有多个过滤参数时，只有与这些参数完全匹配的事件处理函数才会被移除
2. 快捷方式删除所有事件，这里不需要传递事件名了，节点上绑定的所有事件讲全部销毁--->不传参数
3. 事件委托和冒泡的利用。

设置其父元素事件，在设置e.target.执行的东西

1. trigger方法事件

众所周知类似于mousedown、click、keydown等等这类型的事件都是浏览器提供的，通俗叫原生事件，这类型的事件是需要有交互行为才能被触发。

在jQuery通过on方法绑定一个原生事件

$('#elem').on('click', function() {

alert("触发系统事件")

});

alert需要执行的条件：必须有用户点击才可以。如果不同用户交互是否能在某一时刻自动触发该事件呢？ 正常来说是不可以的，但是jQuery解决了这个问题，提供了一个trigger方法来触发浏览器事件

所以我们可以这样：

$('#elem').trigger('click');

在绑定on的事件元素上，通过trigger方法就可以调用到alert了，挺简单！

trigger事件还有一个特性：会在DOM树上冒泡，所以如果要阻止冒泡就需要在事件处理程序中返回false或调用事件对象中的.stopPropagation() 方法可以使事件停止冒泡

trigger事件是具有触发原生与自定义能力的，但是存在一个不可避免的问题： 事件对象event无法完美的实现，毕竟一个是浏览器给的，一个是自己模拟的。尽管 .trigger() 模拟事件对象，但是它并没有完美的复制自然发生的事件，若要触发通过 jQuery 绑定的事件处理函数，而不触发原生的事件，使用.triggerHandler() 来代替

triggerHandler不会触发浏览器的默认行为，.triggerHandler( "submit" )将不会调用表单上的.submit()

.trigger() 会影响所有与 jQuery 对象相匹配的元素，而 .triggerHandler() 仅影响第一个匹配到的元素

使用 .triggerHandler() 触发的事件，并不会在 DOM 树中向上冒泡。 如果它们不是由目标元素直接触发的，那么它就不会进行任何处理

与普通的方法返回 jQuery 对象(这样就能够使用链式用法)相反，.triggerHandler() 返回最后一个处理的事件的返回值。如果没有触发任何事件，会返回 undefined