Bind()方法：

给元素绑定事件

形式：

.bind(eventType[,eventData],handler(eventObject))

eventType,string类型，一个或多个DOM事件的名称，比如：click、submit

eventData，Object，将会传递给handler函数的数据，也就是eventObject的data属性中，如果要调用那么使用eventObject.data

Handler，function，当事件被触发时执行的函数

.bind(eventType[,eventData][,preventBubble])

eventType,string,一个或多个DOM事件的名称，比如：click、submit

eventData，Object，将会传递给handler函数的数据，也就是eventObject的data属性中，如果要调用那么使用eventObject.data

preventBubble,boolean,默认为true，当设置为false的时候将阻止浏览器默认事件发生，并且阻止事件冒泡。.

.bind(events)

Events,object,使用json的形式，为当前元素定义多个事件，例如：

{

click: **function**() {

$( **this** ).addClass( "active" );

},

mouseenter: **function**() {

$( **this** ).addClass( "inside" );

},

mouseleave: **function**() {

$( **this** ).removeClass( "inside" );

}

}

说明：

bind可以给当前存在的节点添加事件，即便使用标签选择器选择一类标签，但是也不能给新加的标签绑定目标时间。当只是简单的给固定的元素绑定事件时，可以使用该方法。但是如果是较为复杂的会动态添加执行事件元素的情况下，bind方法不适用。

bind中定义的evnetType，可以是任意的字符串，各个字符串用空格隔开，当该字符串是默认是DOM事件时，调用DOM事件，如果是不默认的DOM事件，那么就调用自定义事件，这些自定义事件不会被浏览器触发。但是可以通过trigger()方法手动触发。

当evnetType中包含"."的时候，.后边的字符串表示命名空间，也就是当前事件绑定在某命名空间中，那么命名空间中的事件不会影响其他事件，对于命名了解不深所以对于这块有更深的了解后再说明。

对于一个元素可以bind多个同一事件，事件触发按照绑定的顺序触发。

在handler事件中可以使用this关键词来表示触发当前事件的当前DOM对象，注意是DOM对象不是JQuery对象，要转化为JQuery对象使用$(this)。

Live()方法：

该方法已经在Jquery 1.7不推荐使用，1.9中删除。可以使用on()、delegate()方法替代。

不使用live()方法的原因：

1，live方法执行需要先通过选择器选择元素，而在大文档中，会耗费大量时间在选择特定元素的操作上。

2，live()方法不支持链接使用,例如 $( "a" ).find( ".offsite, .external" ).live(... );不能很好的工作。

3，由于所有的live()事件都附着到document对象上，事件会在执行前通过最远、最慢的传播路径。

4，移动设备上click时间不会产生冒泡，所以live()方法不会起作用。

还有其他原因。所以live事件已经被删除了。

Delegate()方法：

该方法可以基于某些特定元素(通过选择器得到)，附加一个或多个事件到特定元素内部选择器指定的元素上，无论选择出的目标元素是当前存在还是将来创建。

形式：

.deletgate(selector,evnetType,handler(evnetObject))

selector，string，用来过滤触发事件的元素，即目标元素。

eventType，string，与bind中介绍的eventType一样，多个事件只用空格隔开。

Handler(eventObject)，被触发的事件。

.delegate(selector,eventType,eventData,handler(eventObject))

selector，string，用来过滤触发事件的元素，即目标元素。

eventType，string，与bind中介绍的eventType一样，多个事件只用空格隔开。

eventData，Object，将会传递给handler函数的数据，也就是eventObject的data属性中，如果要调用那么使用eventObject.data

Handler(eventObject)，被触发的事件。

.delegate(selector,events):

selector，string，用来过滤触发事件的元素，即目标元素。

Events,object,

使用json的形式，为当前元素定义多个事件，例如：

{

click: **function**() {

$( **this** ).addClass( "active" );

},

mouseenter: **function**() {

$( **this** ).addClass( "inside" );

},

mouseleave: **function**() {

$( **this** ).removeClass( "inside" );

}

}

说明：

在JQuery1.7之后delegate已经被on取代，在1.4-1.7之间，delegate任然是最有效的给元素绑定事件的方式。

on方法(JQuery推荐使用)：

附加一个事件处理函数的一个或多个事件到选定的元素，无论选择出的目标元素是当前存在还是将来创建。

形式：

.on(events[,selector][,data],handler(evnetObject))

Events,string,事件名，可以是空格隔开的多个事件也可是带有命名空间的事件。Click submit 或 click.myPlugin

Selector,string,用来指明调用on方法的JQuery对象的特殊子元素，如果没有指定那么就是绑定到调用on方法的JQuery对象。

data，object，通过eventObject.data传递到触发事件中的数据

Handler(eventObject),被触发的函数，这里也可以使用false，false==returnfalse。

.on(events[,selector][,data])

Events,object,

使用json的形式，为当前元素定义多个事件，例如：

{

click: **function**() {

$( **this** ).addClass( "active" );

},

mouseenter: **function**() {

$( **this** ).addClass( "inside" );

},

mouseleave: **function**() {

$( **this** ).removeClass( "inside" );

}

}

Selector,string,用来指明调用on方法的JQuery对象的特殊子元素，如果没有指定那么就是绑定到调用on方法的JQuery对象

Data,object,通过eventObject.data传递到触发事件中的数据

说明：

on方法是当前JQuery推荐使用的事件绑定方法，相反方法是off()，附加只运行一次就删除函数的方法是one()。