# 组件设计原理说明（jquery）

## 1、调用的方法

$.fn.jiraBox = function(options, param){...}

在js中通过$("#a").jiraBox(options[,param])就是先执行上面这个function。

首先判断options参数是不是字符串：

（1）如果是字符串就判断该key是否存在于$.fn.jiraBox.methods中，存在就调用相应的函数。不存在就弹窗说明错误。

（2）如果参数是json对象，就遍历选择器，为每一个dom节点进行数据绑定。$(dom).data()还没有jiraBox的属性就为它增加jiraBox的属性，并调用init函数。如果已经带有jiraBox属性就证明已经初始化，只需要把传进来的options继承到jiraBox.options属性上。

## 2、供外部使用的方法

$.fn.jiraBox.methods = {

options: function(){...},

getValue: function(){...},

...

}

在methods中定义一系列函数供用户使用，用户可以通过$("#a").jiraBox("方法名"[,参数])调用。

### 2.1、options

参数（无）。返回值（options数组）。

是在用户调用$().jiraBox("options");时调用。获取组件的options，包含用户提供的参数和函数、组件默认的参数和函数。

### 2.2、getValue

参数（无）。返回值（data数组）。

是在用户调用$().jiraBox("getValue");时调用。返回组件的已选值，组件采用options.data数组作为已选的值，如果输入框有内容，也会把输入框中当前的值追加在结果中。

### 2.3、setValue

参数（data数组）。返回值（jquery对象）。

是在用户调用$().jiraBox("setValue", data);时调用。用于修改组件的已选值，并重新渲染组件把已选值显示出来。如果有的值不在候选项中，同样会作为已选值。

### 2.4、loadData

参数（allData数组或者url字符串）。返回值（jquery对象）。

是在用户调用$().jiraBox("loadData", param);时调用。重新加载组件的所有候选项。如果用户提供的是数值，就直接把值赋给options.allData然后重新渲染组件。如果用户提交的是字符串，就执行异步请求，再把值赋给options.allData再重新渲染组件。

### 2.5、edit

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("edit");时调用。执行$.fn.jiraBox.defaults中的edit($elem)函数。只在普通状态下有效。

### 2.6、submit

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("submit");时调用。根据当前的options.data来还原为普通状态。只在编辑状态下有效。

### 2.7、cancel

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("cancel");时调用。根据原始的data来还原为普通状态。原始的data即originalData，会在组件进入编辑状态的时候备份一份data数据，然后之后的操作都是对data进行，如果这个时候取消，会根据originalData来还原，并把data赋值为originalData。只在编辑状态下有效。

### 2.8、enable

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("enable");时调用。让组件处于可使用状态。就是一些样式的改变。

### 2.9、disable

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("disable");时调用。让组件处于禁止使用状态。就是一些样式的改变。

### 2.10、showError

参数（msg字符串）。返回值（jquery对象）。

是在用户调用$().jiraBox("showError", msg);时调用。参数msg是要显示的信息字符串。只在编辑状态下有效。

### 2.11、hideError

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("hideError");时调用。把错误提示信息隐藏。只在编辑状态下有效。

### 2.12、showLoading

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("showLoading");时调用。显示加载中的等待图标。只在编辑状态下有效。

### 2.13、hideLoading

参数（无）。返回值（jquery对象）。

是在用户调用$().jiraBox("hideLoading");时调用。隐藏加载中的等待图标。只在编辑状态下有效。

## 3、组件默认的参数和方法

$.fn.jiraBox.defaults = {

originalData: [],

data: [],

allData: [],

url: "",

init: function(){...},

...

}

这部分的函数都是供组件内部使用，参数都是jquery对象，没有返回值

### 3.1、originalData

参数originalData代表原始数据，在组件进入编辑模式的时候对data的备份，如果组件执行取消操作，就会按originalData来还原组件的已选值。

### 3.2、data

参数data用来记录已选值，输入框的值不会保存进data，所以每次getValue都是获取data值再追加输入框的值。

### 3.3、allData

参数allData用来记录所有可选值，主要用来渲染下拉列表。

### 3.4、url

参数url用来获取远程数据。使用url的时候把allData参数置为空，因为需要allData长度来判断是否需要执行请求。

### 3.5、init

在初始化时执行的一些操作，主要是用来渲染和绑定事件。

### 3.6、autoSubmit

用于window点击的时候，执行的代码，也可以在options参数中提供自定义的autoSubmit函数来代替它。所以，如果不希望执行组件默认的autoSubmit，请在参数options中提供自定义的autoSubmit。默认的autoSubmit是使用originalData来还原组件为普通状态。

### 3.7、submit

默认的submit函数，可通过在options中编写自定义的submit函数来替换它。默认的submit函数把输入框的值追加到data参数中，然后根据data来还原组件为普通状态。

### 3.8、cancel

与submit说明相同，忽略输入框的值，采用originalData来还原组件为普通状态。

### 3.9、edit

这个函数名不是很贴切，其实主要是组件进入编辑状态立马做的一系列事情和绑定一系列的事件监听。有些需要注意不要重复绑定监听事件，否则会执行多次。最后一行需要取消组件的click事件，这样点击时就不会回到组件绑定的bindEditEvent事件。

### 3.10、showList

展开下拉列表，在监听键盘操作的时候执行。如果组件处于disable状态就不执行。

实现改变下拉列表的位置，显示和隐藏的切换，然后根据输入框有没有内容，没有内容就显示所有未选的项，是根据data来决定是否未选。输入框有内容就根据内容，遍历所有下拉列表项，text包含输入内容串并且该项是没选的才让它显示，否则保持隐藏。最后判断如果全都隐藏了就把i标签“没有可选项”显示出来。

### 3.11、restore

编辑状态还原为普通状态。这里先判断是否有错误显示，有错误就不还原，避免用户不知道错误的存在。因为之前click事件都取消监听了，所以重新绑定点击编辑事件。

### 3.12、scrollDownHandler

用于在按下键移动下拉列表的高亮项的时候，滚动下拉列表，保证高亮项出现在可看见区域。

### 3.13、scrollUpHandler

用于在按上键移动下拉列表的高亮项的时候，滚动下拉列表，保证高亮项出现在可看见区域。

以下函数在render对象下。

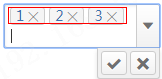
### 3.14、render.renderJiraBox

渲染普通状态下的组件，就是初始化之后就能看到的。

### 3.15、render.renderEdit

就是把组件由变为。然后还在最后根据是否需要进行获取远程数据来进行一些模拟操作（也可以把这个操作放到展开下拉列表才进行）。

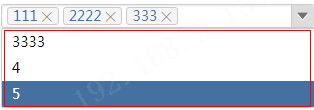
### 3.16、render.reRenderCursor

主要是设置textarea的padding，达到光标位置改变的效果。主要判断了4种情况：已选列表是否发生了换行，再判断光标是否需要换行。最后顺便更改一下下拉列表的位置。

### 3.17、render.reRenderBoxUl

根据data来渲染已选部分的节点。

### 3.18、render.reRenderSelectUl

渲染下拉列表。先把整个ul清空，然后根据allData，重新拼接li字符串，如果allData中的某项是已选的，增加hidden类让它隐藏，最后重新为li绑定mousemove事件。

以下函数在eventListener对象下

### 3.19、eventListener.aClick

绑定a标签的点击事件，主要是阻止冒泡。这样在点击a标签的时候就不会响应组件绑定的点击事件，是指中的三个超链接点击的时候不响应编辑事件。所有事件函数最后都要阻止冒泡，主要是防止window的click执行，下面就不再重复说明。

### 3.20、eventListener.bindEditEvent

的点击事件，方法中先判断一下组件是否有disabled类，有就不会继续执行。在css中对该类进行了此图标的隐藏，虽然隐藏了，但click还在（是对整个节点绑定的），所以需要判断是否有disabled类。里面还做了一件事：（1）每次进入编辑模式，都先对body中存在的组件，看他们是否处于编辑模式，（2）如果是再看看是否enable，（3）并且没有错误信息显示（4）并且不是自己，就触发submit函数，然后自己执行edit函数。

### 3.21、eventListener.submitByEvent

点击事件，如果用户在options中提供了submit函数，就执行自定义的，否则就执行组件默认的行为。

### 3.22、eventListener.cancelByEvent

同上，取消的事件

### 3.23、eventListener.showListByEvent

点击事件，如果组件处于disable状态就不执行。实现改变下拉列表的位置，显示和隐藏的切换，然后根据输入框有没有内容，没有内容就显示所有未选的项，是根据data来决定是否未选。输入框有内容就根据内容，遍历所有下拉列表项，text包含输入内容串并且该项是没选的才让它显示，否则保持隐藏。最后判断如果全都隐藏了就把i标签“没有可选项”显示出来。

### 3.24、eventListener.selectByEvent

选择，下拉列表的项点击，就会选择，主要是修改data的值，因为目前就是以data来作为已选值，已选的多了一项，也需要重新渲染光标的位置，最后把该项从下拉列表中隐藏。

### 3.25、eventListener.deleteSelectedByEvent

删除的点击事件，同理是修改data的值，然后遍历下拉列表中已隐藏的每一项，text相同的就让它取消隐藏，最后重新渲染光标的位置。

### 3.26、eventListener.inputPropertyChangeListener

textarea的内容改变监听函数，如果监听到内容为空，就收起下拉列表。如果监听到有内容，就根据内容显示包含该内容的候选项。

### 3.27、eventListener.keydownListener

键盘的监听事件。如果按回车，会把当前候选项中带有hover类的选中。如果按上键，会把hover类向上移动。如果按下键，就会先展开下拉列表并让第一个加上hover类，如果下拉列表已经展开就让hover向下移动。alt+s组合键就执行submit函数。alt+`组合键就执行cancel函数。

### 3.28、eventListener.windowClickHandle

全局window的点击事件，当点击的时候先判断点击的地方如果不是组件，就会遍历当前所有组件，如果组件处于编辑状态，并且没有错误提示，并且处于enable状态，就执行autoSubmit函数，用户有提供autoSubmit函数就执行用户的，没有提供就执行组件默认的。

## 4、全局函数

### 4.1、arrayIndexOf

参数（arr, obj）。

自定义的查找对象在数组中的位置，使用的是Underscore.js中的isEqual方法来判断对象是否相等。遍历数组arr，如果arr[i]与obj相等，就返回i，否则到最后没找到相等的就返回-1。