**一、功能**

根据用户输入值进行搜索和过滤，让用户快速找到并从预设值列表中选择。

官网地址：<http://jqueryui.com/autocomplete/>

**二、使用**

**1、引入文件**

<**link** rel=**"stylesheet"** href=**"css/jquery-ui.css"**>

<**script** src=**"js/jquery-1.11.3.min.js"**></**script**>

<**script** src=**"js/jquery-ui.js"**></**script**>

**2、html代码：**

<**div** class=**"ui-widget"**>

<**label** for=**"tags"**>Tags: </**label**>

<**input** id=**"tags"**>

</**div**>

**3、js代码：**

*$*( **function**() {

**var** availableTags = [

**"ActionScript"**,

**"AppleScript"**,

**"Asp"**,

**"BASIC"**,

**"C"**,

**"C++"**,

**"Clojure"**,

**"COBOL"**,

**"ColdFusion"**,

**"Erlang"**,

**"Fortran"**,

**"Groovy"**,

**"Haskell"**,

**"Java"**,

**"JavaScript"**,

**"Lisp"**,

**"Perl"**,

**"PHP"**,

**"Python"**,

**"Ruby"**,

**"Scala"**,

**"Scheme"**

];

*$*( **"#tags"** ).**autocomplete**({

**source**: availableTags

});

} );

**4、自定义下拉菜单高度：修改jquery.ui.css，添加红色标注部分代码**

.**ui-autocomplete** {

position: **absolute**;

top: 0;

left: 0;

cursor: **default**;

overflow-y: **auto**;

overflow-x: **hidden**;

max-height: 230**px**;

}

**5、点击下拉按钮，展示所有选项**

*$*( **".auto\_input"** ).**autocomplete**({

**minLength**:0

}

*$*(**".dropdown\_menu\_div"**).click(**function**(){

*$*( **".auto\_input"** ).**autocomplete**(**"search"**,**""**);

});

**三、API**

**1、键盘输入**

当下拉菜单展开时，可以使用下列键盘命令：

UP：将焦点移至上一个选项

DOWN：将焦点移至下一个选项

ESCAPE：关闭下拉菜单

ENTER：选择下拉菜单中当前处于焦点的选项并关闭下拉菜单

TAB：选择下拉菜单中当前处于焦点的选项，关闭下拉菜单并将焦点移至下一个可聚焦的元素

PAGE UP/PAGE DOWN：将焦点置顶/置底

当下拉菜单关闭时，可以使用下列键盘命令：

UP/DOWN：展开下拉菜单，如果minLength属性已定义

**2、属性**

2.1 appendTo：下拉菜单被添加至哪个元素，默认值为input标签的class为"ui-front"祖先元素

初始化代码：

$( ".selector" ).autocomplete({

appendTo: "#someElem"

});

2.2 autoFocus：设置为true值时，当下拉菜单展开时第一个选项将自动获取焦点

$( ".selector" ).autocomplete({

autoFocus: true

});

2.3 delay：延迟时间，单位为毫秒

$( ".selector" ).autocomplete({

delay: 500

});

2.4 disabled：禁止插件，默认值为false

$( ".selector" ).autocomplete({

disabled: true

});

2.5 minLength：展示下拉菜单需要的最小输入字符数

$( ".selector" ).autocomplete({

minLength: 0

});

2.6 source：

类型：Array，String，Function

默认值：无，必须指定

Array：数组可以被使用为本地数据

字符串数组：["Choice1","Choice2"]，

有label和value属性的对象数组：[{label:"Choice1",value:"value1"},...]

String：指向一个可以转换为JSON数据的URL

$( ".selector" ).autocomplete({

source: [ "c++", "java", "php", "coldfusion", "javascript", "asp", "ruby" ]

});

**3、方法Methods**

3.1 close()：关闭下拉菜单

$( ".selector" ).autocomplete( "close" );

3.2 destroy()：彻底移除自动完成插件功能

$( ".selector" ).autocomplete( "destroy" );

3.3 disable()：禁用插件

$( ".selector" ).autocomplete( "disable" );

3.4 enable()：启用插件

$( ".selector" ).autocomplete( "enable" );

3.5 option(optionName)：获取特定属性值

var isDisabled = $( ".selector" ).autocomplete( "option", "disabled" );

3.6 option()：获取所有属性值，返回一个包含属性键值对的对象

var options = $( ".selector" ).autocomplete( "option" );

3.7 option(optionName,value)：给特定属性赋值

$( ".selector" ).autocomplete( "option", "disabled", true );

3.8 option( options )：给一个或多个属性赋值

$( ".selector" ).autocomplete( "option", { disabled: true } );

**4、扩展点Extension Points**

4.1 \_renderItem(ul,item)：控制下拉菜单每个选项的生成

\_renderItem: function( ul, item ) {

return $( "<li>" )

.attr( "data-value", item.value )

.append( item.label )

.appendTo( ul );

}

4.2 \_renderMenu(ul,items)：控制生成下拉菜单的方法

\_renderMenu: function( ul, items ) {

var that = this;

$.each( items, function( index, item ) {

that.\_renderItemData( ul, item );

});

$( ul ).find( "li:odd" ).addClass( "odd" );

}

4.3 \_resizeMenu()：下拉菜单展开之前控制其大小

\_resizeMenu: function() {

this.menu.element.outerWidth( 500 );

}

**5、事件Events**

5.1 change(event,ui)

当区域失去焦点，且输入值发生改变时触发事件

初始化：

$( ".selector" ).autocomplete({

change: **function**( event, ui ) {}

});

绑定事件监听：

$( ".selector" ).on( "autocompletechange", **function**( event, ui ) {} );

5.2 close(event,ui)

当下拉菜单隐藏时触发

5.3 create(event,ui)

创建插件时触发

5.4 focus(event,ui)

当焦点移动至选项时触发。默认操作是将处于焦点选项的值代替输入区域的值

5.5 open(event,ui)

当下拉菜单展开时触发

5.6 select(event,ui)

当选中下拉菜单的某个选项时触发