[jQuery.AutoComplete (1.1.2) 简介](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html) [样式](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#style)

* - [width](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#width)
* - [maxHeight](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#maxHeight)
* - [itemHeight](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#itemHeight)
* - [listStyle](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#listStyle)
* - [listDirection](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#listDirection)

[数据](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#dataSource)

* - [data](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#data)
* - [ajaxDataType](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#ajaxDataType)
* - [ajaxParams](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#ajaxParams)
* - [ajaxTimeout](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#ajaxTimeout)
* - [ajaxType](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#ajaxType)
* - [maxItems](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#maxItems)

[事件](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#event)

* - [matchHandler](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#matchHandler)
* - [emphasisHandler](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#emphasisHandler)
* - [createItemHandler](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#createItemHandler)
* - [beforeLoadDataHandler](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#beforeLoadDataHandler)
* - [afterSelectedHandler](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#afterSelectedHandler)

[行为](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#event)

* - [async](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#async)
* - [emphasis](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#emphasis)

[调试](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#debug)

* - [onerror](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#onerror)

[ChangeLog](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#changelog)

* [v1.1.2](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.2)
* [v1.1.1](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.1)
* [v1.1.0](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.0)

jQuery.AutoComplete是一个基于jQuery的自动补全插件。借助于jQuery优秀的跨浏览器特性，可以兼容Chrome/IE/Firefox/Opera/Safari等多种浏览器。

特性一览：

* 支持补全列表的宽度设定。
* 支持补全列表的最大高度设定。
* 支持补全列表的行数限制。
* 支持补全列表的显示位置及方向的设定。
* 支持自定义匹配规则。
* 支持匹配文本的渲染。
* 支持自定义匹配文本的渲染样式。
* 支持补全列表的样式设定。
* 支持自定义补全列表项的创建。
* 支持多种数据源。
* 支持'json'和'xml'两种数据格式。
* 支持异步处理。
* 支持错误调试。
* 支持jQuery1.7.1+。

在Html文件的</head>标签之前，引入插件:

<script type="text/javascript" src="/path/to/jquery-1.7.1.min.js"></script>

<link rel="stylesheet" type="text/css" href="/path/to/jquery.autocomplete.css"></link>

<script type="text/javascript" src="/path/to/jquery.autocomplete.js"></script>

一个简单的例子: 

$('#sample').AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve']

});

源代码已托管在[Github](http://github.com/nswish/jquery.autocomplete)上，您可以把发现的bugs和希望扩展的features提交到[issues页面](http://github.com/nswish/jquery.autocomplete/issues)。

**直接下载: *(点击右键, 然后另存为)***

|  |  |
| --- | --- |
| [jquery.autocomplete.js (1.1.2)](http://www.jq22.com/demo/jQueryAutoComplete201704252204/src/jquery.autocomplete.js) | *16kb, 未经压缩，带有注释* |
| [jquery.autocomplete.min.js (1.1.2)](http://www.jq22.com/demo/jQueryAutoComplete201704252204/src/jquery.autocomplete.min.js) | *8kb, 使用*[*开源中国在线工具*](http://www.ostools.net/jscompress)*压缩* |
| [jquery.autocomplete.css (1.1.2)](http://www.jq22.com/demo/jQueryAutoComplete201704252204/src/jquery.autocomplete.css) | *2kb* |

**样式**

**width**列表的宽度。  
类型：数字，字符串默认值: 320单位: px   
当列表宽度不足以容纳下列表项的内容时，列表项的文本以折行的方式显示。可以通过设定足够大的列表宽度，避免折行的发生。  
[[since v1.1.0]](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.0)如果设定 width 为 'auto'，那么补全列表的宽度与输入框的宽度一致。

示例: 宽度设为数字 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'width': 300

});

}

示例: 宽度设为'auto' 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'width': 'auto'

});

}

**maxHeight**列表的最大高度。  
类型：数字默认值: null单位: px   
限制列表的高度不能大于某个值。当列表实际长度大于限制值时，显示纵向滚动轴。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'maxHeight': 100

});

}

**itemHeight**列表项的高度。  
类型：数字默认值: null单位: px

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'itemHeight': 50

});

}

**listStyle**列表的样式。  
类型：字符串默认值: 'normal'范围: ['normal', 'iconList', 'custom']   
'normal': 普通文本样式。   
'iconList': 左边是图标右边是文本的样式。   
'custom': 自定义样式。须配合[createItemHandler参数](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#createItemHandler)一起使用。

示例: 'normal': 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'listStyle': 'normal'

});

}

示例: 'iconList': 

function(input) {

var world = ['Cambodia', 'Cameroon', 'Canada', 'Cape-Verde', 'Cayman-Islands', 'Central-African-Republic', 'Chad', 'Chile', 'China', 'Colombia', 'Commonwealth', 'Comoros', 'Costa-Rica', "Cote-d'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czech-Republic'],

result = [];

$.each(world, function(index, name) {

new Image()

.src = 'demo/image/worldFlag/' + name + '.png'; // 预加载图片

result.push({

'value': name,

'label': name,

'image': 'demo/image/worldFlag/' + name + '.png'

});

});

$(input)

.AutoComplete({

'data': result,

'listStyle': 'iconList',

'itemHeight': 24,

'width': 280,

'onerror': function(msg) {

alert(msg);

}

});

}

示例: 'custom': 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'listStyle': 'custom',

'emphasis': false,

'createItemHandler': function(index, data) {

return "<span style='color:green;'>--" + data.label + "--</span>"; // 文本显示为绿色，并在前后使用'--'包裹

},

'onerror': function(msg) {

alert(msg);

}

});

}

**listDirection**列表的位置和方向。  
类型：字符串默认值: 'down'范围: ['down', 'up']   
'down': 列表会显示在输入框的下方。   
'up': 列表会显示在输入框的上方。*(注:在某些情况下，通过设定列表项高度[itemHeight]，可帮助正确定位列表的显示位置)*

示例: 'down': 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'listDirection': 'down'

});

}

示例: 'up': 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'listDirection': 'up'

});

}

**数据**

**data**数据源。  
类型：数组，函数，Url字符串默认值: []   
输入框中每次发生按件抬起事件(除某些被忽略的按键，如上下左右键)时，AutoComplete都会从数据源中搜寻符合匹配规则的数据作为列表的项。

* 数组: 可以直接把数组作为数据源。数组中的元素可以是字符串，也可以是一个js对象(包含label和value属性。在列表样式是'iconList'时需要image属性)。
* 函数: 将数组作为返回值的函数。
* Url字符串: 远程或本地的Url字符串。AutoComplete以ajax方式访问Url指向的数据服务。

示例: 数组作为数据源 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve']

});

}

示例: 函数作为数据源 

function(input) {

$(input)

.AutoComplete({

'data': function() {

return ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'];

}

});

}

示例: Url字符串作为数据源 

function(input) {

$(input)

.AutoComplete({

'data': "../test/data.json",

'ajaxDataType': 'json',

'onerror': function(msg) {

alert(msg);

}

});

}

**ajaxDataType**Ajax返回数据的格式。  
类型：字符串默认值: 'json'范围: ['json', 'xml']   
指导AutoComplete如何对返回数据做解析，同时也作为jQuery.ajax函数的dataType参数,。

* 'json': 样例格式

["One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve"]

* 'xml': 样例格式
* <data>
* <item>One</item>
* <item>Two</item>
* <item>Three</item>
* <item>Four</item>
* <item>Five</item>
* <item>Six</item>
* <item>Seven</item>
* <item>Eight</item>
* <item>Nine</item>
* <item>Ten</item>
* <item>Eleven</item>
* <item value="Twelve" label="Twelve"></item>

</data>

示例: json 

function(input) {

$(input)

.AutoComplete({

'data': "../test/data.json",

'ajaxDataType': 'json',

'onerror': function(msg) {

alert(msg);

}

});

}

示例: xml 

function(input) {

$(input)

.AutoComplete({

'data': "../test/data.xml",

'ajaxDataType': 'xml',

'onerror': function(msg) {

alert(msg);

}

});

}

**ajaxParams**Ajax发送给服务器的数据。  
类型：对象，函数，字符串默认值: {'keyword': 输入框中的值}   
作为jQuery.ajax函数中的data参数。

* 对象: 请参考jQuery.ajax函数的data参数的描述。
* 函数: 把js对象作为返回值的函数。
* 字符串: 请参考jQuery.ajax函数的data参数的描述。

**ajaxTimeout**Ajax的超时。  
类型：数字默认值: 3000单位: 毫秒   
作为jQuery.ajax函数中的timeout参数。

**ajaxType**Ajax访问服务器的HTTP(s)方法(method)。  
类型：字符串默认值: 'GET'范围: ['GET', 'POST']   
作为jQuery.ajax函数中的type参数。

**maxItems**列表的最大项数。  
类型：数字默认值: 20   
作用类似[maxHeight参数](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#maxHeight)，用来限制列表的高度。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'maxItems': 3

});

}

**事件**

**matchHandler**发生匹配处理时的回调函数。  
类型：函数   
通过此回调函数，可以实现自定义的匹配规则。在匹配的过程中，数据源中的每一个数据项都会触发一次此回调函数。

function matchHandler(keyword, data)

* 参数keyword是输入框中的值。
* 参数data是数据源中的一个数据项。
* 返回true，表示当前数据项匹配。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'matchHandler': function(keyword, data) {

return data.value.indexOf(keyword) == 0; // 匹配规则: 以输入框中的值开头且大小写敏感

}

});

}

**emphasisHandler**在渲染匹配内容时的回调函数。  
类型：函数   
通过此函数，可以实现自定义的匹配内容渲染。每个匹配成功的数据项都会触发一次此回调函数。

function emphasisHandler(keyword, data)

* 参数keyword是输入框中的值。
* 参数data是数据源中的一个数据项。
* 修改data.label中的内容，即可改变列表项的内容显示。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'emphasisHandler': function(keyword, data) {

var regex = RegExp("(" + keyword.replace(/([.?\*+^$[\]\\(){}|-])/g, "\\$1") + ")", 'ig');

data.label = data.label.replace(regex, "<span style='font-weight:bold;color:blue;'>$1</span>");

}

});

}

**createItemHandler**在创建列表项时的回调函数。  
类型：函数   
当[listStyle参数](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#listStyle)设置为'custom'时，AutoComplete会调用此回调函数获取列表项的内容。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'listStyle': 'custom',

'emphasis': false,

'createItemHandler': function(index, data) {

return "<span style='color:green;'>--" + data.label + "--</span>"; // 文本显示为绿色，并在前后使用'--'包裹

},

'onerror': function(msg) {

alert(msg);

}

});

}

**beforeLoadDataHandler**在每次输入之后装载数据之前的回调函数。[[since v1.1.1]](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.1)  
类型：函数   
在输入框中每输入一个有效字符，控件都会发起一次数据请求并装载之。beforeLoadDataHandler在输入发生之后，数据请求和装载之前被调用，输入框中的值会作为参数传递给回调函数。如果回调函数的返回值为false，则控件取消随后的数据请求和装载动作。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': "../test/data.json",

'async': true,

'beforeLoadDataHandler': function(keyword) {

return keyword.length > 3;

},

'onerror': function(msg) {

alert(msg);

}

});

}

**afterSelectedHandler**在列表项被选择之后的回调函数。[[since v1.1.2]](http://www.jq22.com/demo/jQueryAutoComplete201704252204/doc/index.html#v1.1.2)  
类型：函数   
无论是使用键盘还是鼠标来选择列表项，都会触发这个回调函数。控件首先将用户选中列表项的值放置到输入框中，然后调起回调函数，将绑定在这个列表项上的数据传递给它。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': ['One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Eleven', 'Twelve'],

'afterSelectedHandler': function(data) {

alert(data.value);

},

'onerror': function(msg) {

alert(msg);

}

});

}

**行为**

**async**是否异步。  
类型：布尔值默认值: false   
AutoComplete将“输入框按键抬起 => 从数据源获取数据 => 匹配处理 => 匹配渲染处理 => 列表生成并显示”这5步作为一次处理过程。   
在同步的设置下，一个过程完成之后，才会开始第二个过程。如果数据源来自较慢的网络，亦或过程较为耗时，用户会感到一定的延时阻塞；   
在异步的设置下，不用等待第一个过程完成，第二个过程就可以立即开始，因此没有阻塞发生且有助于改善用户的体验。多个过程同时执行时，AutoComplete会自动判别哪个是有效过程，忽略其他无效过程。

示例: 

function(input) {

$(input)

.AutoComplete({

'data': "../test/data.json",

'async': true,

'onerror': function(msg) {

alert(msg);

}

});

}

**emphasis**是否渲染匹配文本。  
类型：布尔值默认值: true

**调试**

**onerror**遇到错误时的回调函数  
类型：函数   
遇到错误时，AutoComplete默认采取静默方式处理。通过设置此回调函数，可以获取到错误发生的信息。

function onerror(msg)

jQuery.AutoComplete使用[Sublime Text 2](http://www.sublimetext.com/)完成所有代码、样例及文档的编写。

**ChangeLog**

**v1.1.2**

* 增加了一个列表项被选择之后的事件

**v1.1.1**

* 增加了一个输入发生之后数据请求装载之前的事件

**v1.1.0**

* 当参数width设定为'auto'时，补全列表的宽度可自动适配输入框的宽度。