

## jQuery UI 实例 - 自动完成 ( Autocomplete )

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。

如需了解更多有关 autocomplete 部件的细节，请查看 API 文档 [自动完成部件 \( Autocomplete Widget \)](#)。

本章节使用到 [search.php 下载](#)。

### 默认功能

当您在输入域中输入时，自动完成 ( Autocomplete ) 部件提供相应的建议。在本实例中，提供了编程语言的建议选项，您可以输入 "ja" 尝试一下，可以得到 Java 或 JavaScript。

数据源是一个简单的 JavaScript 数组，使用 source 选项提供给部件。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 默认功能</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <script>
$(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
    });
});
</script>
</head>
<body>

<div class="ui-widget">
    <label for="tags">标签: </label>
    <input id="tags">
</div>

</body>
</html>
```

[查看演示](#)

## 包含重音

autocomplete 域使用自定义的 source 选项来匹配带有重音字符的结果项，即使文本域不包含重音字符也会匹配。但是如果您在文本域中键入了重音字符，则不会显示非重音的结果项。

尝试键入 "Jo"，会看到 "John" 和 "Jörn"，然后 键入 "Jö"，只会看到 "Jörn"。

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI 自动完成 (Autocomplete) - 包含重音</title>
    <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
    <script src="//code.jquery.com/jquery-1.9.1.js"></script>
    <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
    <script>
    $(function() {
        var names = [ "Jörn Zaefferer", "Scott González", "John Resig" ];

        var accentMap = {
            "á": "a",
            "ö": "o"
        };

        var normalize = function( term ) {
            var ret = "";
```

```
for ( var i = 0; i < term.length; i++ ) {
    ret += accentMap[ term.charAt(i) ] || term.charAt(i);
}
return ret;
};

$( "#developer" ).autocomplete({
    source: function( request, response ) {
        var matcher = new RegExp( $.ui.autocomplete.escapeRegex( request.term ), "i" );
        response( $.grep( names, function( value ) {
            value = value.label || value.value || value;
            return matcher.test( value ) || matcher.test( normalize( value ) );
        }) );
    }
});
});
</script>
</head>
<body>

<div class="ui-widget">
    <form>
        <label for="developer">开发人员: </label>
        <input id="developer">
    </form>
</div>


</body>
</html>
```

[查看演示](#)

## 分类

分类的搜索结果。尝试键入 "a" 或 "n"。

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI 自动完成 (Autocomplete) - 分类</title>
    <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
    <script src="//code.jquery.com/jquery-1.9.1.js"></script>
    <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
    <style>
        .ui-autocomplete-category {
```

```
font-weight: bold;
padding: .2em .4em;
margin: .8em 0 .2em;
line-height: 1.5;
}
</style>
<script>
$.widget( "custom.catcomplete", $.ui.autocomplete, {
  _renderMenu: function( ul, items ) {
    var that = this,
        currentCategory = "";
    $.each( items, function( index, item ) {
      if ( item.category != currentCategory ) {
        ul.append( "<li class='ui-autocomplete-category'>" + item.category + "</li>" );
        currentCategory = item.category;
      }
      that._renderItemData( ul, item );
    });
  }
});
</script>
<script>
$(function() {
  var data = [
    { label: "anders", category: "" },
    { label: "andreas", category: "" },
    { label: "antal", category: "" },
    { label: "annhhx10", category: "Products" },
    { label: "annk K12", category: "Products" },
    { label: "annttop C13", category: "Products" },
    { label: "anders andersson", category: "People" },
    { label: "andreas andersson", category: "People" },
    { label: "andreas johnson", category: "People" }
  ];

  $( "#search" ).catcomplete({
    delay: 0,
    source: data
  });
});
</script>
</head>
<body>

<label for="search">搜索: </label>
<input id="search">
```

```
</body>
</html>
```

[查看演示](#)

## 组合框 ( Combobox )

一个由 Autocomplete 和 Button 创建的自定义部件。您可以键入一些字符，来获得基于您的输入过滤的结果，或者使用按钮从完整列表中选择。

该输入是从一个已有的 select 元素中读取，传递给带有自定义的 source 选项的 Autocomplete。

这是一个不被支持的不完美的部件。这里纯粹是为了演示 autocomplete 定制功能。[如需了解更多有关该部件工作原理的细节，请点击这里查看相关的 jQuery 文章。](#)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 组合框 (Combobox) </title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .custom-combobox {
      position: relative;
      display: inline-block;
    }
    .custom-combobox-toggle {
      position: absolute;
      top: 0;
      bottom: 0;
      margin-left: -1px;
      padding: 0;
      /* 支持: IE7 */
      *height: 1.7em;
      *top: 0.1em;
    }
    .custom-combobox-input {
      margin: 0;
      padding: 0.3em;
    }
  </style>
  <script>
  (function( $ ) {
    $.widget( "custom.combobox", {
      _create: function() {
        this.wrapper = $( "<span>" )
```

```
.addClass( "custom-combobox" )
.insertAfter( this.element );

this.element.hide();
this._createAutocomplete();
this._createShowAllButton();
},

_createAutocomplete: function() {
    var selected = this.element.children( ":selected" ),
        value = selected.val() ? selected.text() : "";

    this.input = $( "<input>" )
        .appendTo( this.wrapper )
        .val( value )
        .attr( "title", "" )
        .addClass( "custom-combobox-input ui-widget ui-widget-content ui-state-default ui-corner-left"
    )

    .autocomplete({
        delay: 0,
        minLength: 0,
        source: $.proxy( this, "_source" )
    })
    .tooltip({
        tooltipClass: "ui-state-highlight"
    });

    this._on( this.input, {
        autocompleteselect: function( event, ui ) {
            ui.item.option.selected = true;
            this._trigger( "select", event, {
                item: ui.item.option
            });
        },

        autocompletechange: "_removeIfInvalid"
    });
},

_createShowAllButton: function() {
    var input = this.input,
        wasOpen = false;

    $( "<a>" )
        .attr( "tabIndex", -1 )
        .attr( "title", "Show All Items" )
        .tooltip()
        .appendTo( this.wrapper )
        .button({
```

```
        icons: {
            primary: "ui-icon-triangle-1-s"
        },
        text: false
    })
    .removeClass( "ui-corner-all" )
    .addClass( "custom-combobox-toggle ui-corner-right" )
    .mousedown(function() {
        wasOpen = input.autocomplete( "widget" ).is( ":visible" );
    })
    .click(function() {
        input.focus();

        // 如果已经可见则关闭
        if ( wasOpen ) {
            return;
        }

        // 传递空字符串作为搜索的值, 显示所有的结果
        input.autocomplete( "search", "" );
    });
},

_source: function( request, response ) {
    var matcher = new RegExp( $.ui.autocomplete.escapeRegex(request.term), "i" );
    response( this.element.children( "option" ).map(function() {
        var text = $( this ).text();
        if ( this.value && ( !request.term || matcher.test(text) ) )
            return {
                label: text,
                value: text,
                option: this
            };
    }) );
},

_removeIfInvalid: function( event, ui ) {

    // 选择一项, 不执行其他动作
    if ( ui.item ) {
        return;
    }

    // 搜索一个匹配 (不区分大小写)
    var value = this.input.val(),
        valueLowerCase = value.toLowerCase(),
        valid = false;
    this.element.children( "option" ).each(function() {
        if ( $( this ).text().toLowerCase() === valueLowerCase ) {

```

```
        this.selected = valid = true;
        return false;
    }
});

// 找到一个匹配, 不执行其他动作
if ( valid ) {
    return;
}

// 移除无效的值
this.input
    .val( "" )
    .attr( "title", value + " didn't match any item" )
    .tooltip( "open" );
this.element.val( "" );
this._delay(function() {
    this.input.tooltip( "close" ).attr( "title", "" );
}, 2500 );
this.input.data( "ui-autocomplete" ).term = "";
},

_destroy: function() {
    this.wrapper.remove();
    this.element.show();
}
});
})( jQuery );

$(function() {
    $( "#combobox" ).combobox();
    $( "#toggle" ).click(function() {
        $( "#combobox" ).toggle();
    });
});
</script>
</head>
<body>

<div class="ui-widget">
    <label>您喜欢的编程语言: </label>
    <select id="combobox">
        <option value="">请选择...</option>
        <option value="ActionScript">ActionScript</option>
        <option value="AppleScript">AppleScript</option>
        <option value="Asp">Asp</option>
        <option value="BASIC">BASIC</option>
        <option value="C">C</option>
        <option value="C++">C++</option>
```



```
<option value="Clojure">Clojure</option>
<option value="COBOL">COBOL</option>
<option value="ColdFusion">ColdFusion</option>
<option value="Erlang">Erlang</option>
<option value="Fortran">Fortran</option>
<option value="Groovy">Groovy</option>
<option value="Haskell">Haskell</option>
<option value="Java">Java</option>
<option value="JavaScript">JavaScript</option>
<option value="Lisp">Lisp</option>
<option value="Perl">Perl</option>
<option value="PHP">PHP</option>
<option value="Python">Python</option>
<option value="Ruby">Ruby</option>
<option value="Scala">Scala</option>
<option value="Scheme">Scheme</option>
</select>
</div>
<button id="toggle">显示基础的选择框</button>

</body>
</html>
```

[查看演示](#)

## 自定义数据并显示

您可以使用自定义数据格式，并通过简单地重载默认的聚焦和选择行为来显示数据。

尝试键入 "j"，或者按向下箭头按键，即可得到一个项目列表。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 自定义数据并显示</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #project-label {
      display: block;
      font-weight: bold;
      margin-bottom: 1em;
    }
    #project-icon {
      float: left;
```

```
height: 32px;
width: 32px;
}
#project-description {
margin: 0;
padding: 0;
}
</style>
<script>
$(function() {
var projects = [
{
value: "jquery",
label: "jQuery",
desc: "the write less, do more, JavaScript library",
icon: "jquery_32x32.png"
},
{
value: "jquery-ui",
label: "jQuery UI",
desc: "the official user interface library for jQuery",
icon: "jqueryui_32x32.png"
},
{
value: "sizzlejs",
label: "Sizzle JS",
desc: "a pure-JavaScript CSS selector engine",
icon: "sizzlejs_32x32.png"
}
];

$( "#project" ).autocomplete({
minLength: 0,
source: projects,
focus: function( event, ui ) {
$( "#project" ).val( ui.item.label );
return false;
},
select: function( event, ui ) {
$( "#project" ).val( ui.item.label );
$( "#project-id" ).val( ui.item.value );
$( "#project-description" ).html( ui.item.desc );
$( "#project-icon" ).attr( "src", "images/" + ui.item.icon );

return false;
}
})

.data( "ui-autocomplete" )._renderItem = function( ul, item ) {
return $( "<li>" )
```

```
.append( "<a>" + item.label + "<br>" + item.desc + "</a>" )
.appendTo( ul );

});

});
</script>
</head>
<body>

<div id="project-label">选择一个项目（请键入 "j"）：</div>

<input id="project">
<input type="hidden" id="project-id">
<p id="project-description"></p>

</body>
</html>
```

[查看演示](#)

## 多个值

用法：键入一些字符，比如 "j"，可以看到相关的编程语言结果。选择一个值，然后继续键入字符来添加其他的值。

本实例演示如何使用 source 选项和一些事件来实现在一个单一的文本域输入多个自动完成的值。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 多个值</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <script>
$(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"
];
function split( val ) {
    return val.split( /\s*/ );
}
function extractLast( term ) {
    return split( term ).pop();
}

$( "#tags" )
    // 当选择一个条目时不离开文本域
    .bind( "keydown", function( event ) {
        if ( event.keyCode === $.ui.keyCode.TAB &&
            $( this ).data( "ui-autocomplete" ).menu.active ) {
            event.preventDefault();
        }
    })
    .autocomplete({
        minLength: 0,
        source: function( request, response ) {
            // 回到 autocomplete, 但是提取最后的条目
            response( $.ui.autocomplete.filter(
                availableTags, extractLast( request.term ) ) );
        },
        focus: function() {
            // 防止在获得焦点时插入值
            return false;
        },
        select: function( event, ui ) {
            var terms = split( this.value );
            // 移除当前输入
            terms.pop();
            // 添加被选项
            terms.push( ui.item.value );
            // 添加占位符, 在结尾添加逗号+空格
            terms.push( "" );
            this.value = terms.join( ", " );
            return false;
        }
    })
```

```
    });  
  });  
</script>  
</head>  
<body>  
  
<div class="ui-widget">  
  <label for="tags">编程语言: </label>  
  <input id="tags" size="50">  
</div>  
  
  
  
  
</body>  
</html>
```

[查看演示](#)

## 多个值，远程

用法：键入至少两个字符来获取鸟的名称。选择一个值，然后继续键入字符来添加其他的值。

本实例演示如何使用 source 选项和一些事件来实现在一个单一的文本域输入多个自动完成的值。

```
<!doctype html>  
<html lang="en">  
<head>  
  <meta charset="utf-8">  
  <title>jQuery UI 自动完成 (Autocomplete) - 多个值，远程</title>  
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">  
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>  
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>  
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">  
  <style>  
    .ui-autocomplete-loading {  
      background: white url('images/ui-anim_basic_16x16.gif') right center no-repeat;  
    }  
  </style>  
<script>  
$(function() {  
  function split( val ) {  
    return val.split( /,\s*/ );  
  }  
  function extractLast( term ) {  
    return split( term ).pop();  
  }  
  
  $( "#birds" )  
    // 当选择一个条目时不离开文本域  
    .bind( "keydown", function( event ) {
```

```
if ( event.keyCode === $.ui.keyCode.TAB &&
    $( this ).data( "ui-autocomplete" ).menu.active ) {
    event.preventDefault();
}
})
.autocomplete({
    source: function( request, response ) {
        $.getJSON( "search.php", {
            term: extractLast( request.term )
        }, response );
    },
    search: function() {
        // 自定义最小长度
        var term = extractLast( this.value );
        if ( term.length < 2 ) {
            return false;
        }
    },
    focus: function() {
        // 防止在获得焦点时插入值
        return false;
    },
    select: function( event, ui ) {
        var terms = split( this.value );
        // 移除当前输入
        terms.pop();
        // 添加被选项
        terms.push( ui.item.value );
        // 添加占位符，在结尾添加逗号+空格
        terms.push( "" );
        this.value = terms.join( ", " );
        return false;
    }
});
});
</script>
</head>
<body>

<div class="ui-widget">
    <label for="birds">鸟: </label>
    <input id="birds" size="50">
</div>

</body>
</html>
```

[查看演示](#)

## 远程 JSONP 数据源

当您在文本域中键入字符时，Autocomplete 部件给出建议结果。在本实例中，当您在文本域中至少键入两个字符时，将显示相关城市的名称。

在本实例中，数据源是 [geonames.org webservice](http://www.geonames.org/web/service)。虽然选择一个元素后文本域中是该城市名称，但是会显示更多的信息以便找到正确的条目。数据也可以回调，显示在下面的结果框中。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 远程 JSONP 数据源</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .ui-autocomplete-loading {
      background: white url('images/ui-anim_basic_16x16.gif') right center no-repeat;
    }
    #city { width: 25em; }
  </style>
  <script>
    $(function() {
      function log( message ) {
        $( "<div>" ).text( message ).prependTo( "#log" );
        $( "#log" ).scrollTop( 0 );
      }

      $( "#city" ).autocomplete({
        source: function( request, response ) {
          $.ajax({
            url: "http://ws.geonames.org/searchJSON",
            dataType: "jsonp",
            data: {
              featureClass: "P",
              style: "full",
              maxRows: 12,
              name_startsWith: request.term
            },
            success: function( data ) {
              response( $.map( data.geonames, function( item ) {
                return {
                  label: item.name + (item.adminName1 ? ", " + item.adminName1 : "") + ", " + item.country
                  Name,
```

```
        value: item.name
    }
    }));
    }
    });
    },
    minLength: 2,
    select: function( event, ui ) {
        log( ui.item ?
            "Selected: " + ui.item.label :
            "Nothing selected, input was " + this.value);
    },
    open: function() {
        $( this ).removeClass( "ui-corner-all" ).addClass( "ui-corner-top" );
    },
    close: function() {
        $( this ).removeClass( "ui-corner-top" ).addClass( "ui-corner-all" );
    }
    });
});
</script>
</head>
<body>

<div class="ui-widget">
    <label for="city">您的城市: </label>
    <input id="city">
    Powered by <a href="http://geonames.org" target="_blank">geonames.org</a>
</div>

<div class="ui-widget" style="margin-top:2em; font-family:Arial">
    结果:
    <div id="log" style="height: 200px; width: 300px; overflow: auto;" class="ui-widget-content"></div>
</div>

</body>
</html>
```

[查看演示](#)

## 远程数据源

当您在文本域中键入字符时，Autocomplete 部件给出建议结果。在本实例中，当您在文本域中至少键入两个字符时，将显示相关鸟的名称。

在本实例中，数据源是可返回 JSON 数据的服务器端脚本，通过一个简单的 source 选项来指定。另外，minLength 选项设置为 2，避免查询返回太多的结果，select 事件用于显示一些反馈。



```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 远程数据源</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .ui-autocomplete-loading {
      background: white url('images/ui-anim_basic_16x16.gif') right center no-repeat;
    }
  </style>
  <script>
    $(function() {
      function log( message ) {
        $( "<div>" ).text( message ).prependTo( "#log" );
        $( "#log" ).scrollTop( 0 );
      }

      $( "#birds" ).autocomplete({
        source: "search.php",
        minLength: 2,
        select: function( event, ui ) {
          log( ui.item ?
            "Selected: " + ui.item.value + " aka " + ui.item.id :
            "Nothing selected, input was " + this.value );
        }
      });
    });
  </script>
</head>
<body>

  <div class="ui-widget">
    <label for="birds">鸟: </label>
    <input id="birds">
  </div>

  <div class="ui-widget" style="margin-top:2em; font-family:Arial">
    结果:
    <div id="log" style="height: 200px; width: 300px; overflow: auto;" class="ui-widget-content"></div>
  </div>

</body>
</html>
```

[查看演示](#)

## 远程缓存

当您在文本域中键入字符时，Autocomplete 部件给出建议结果。在本实例中，当您在文本域中至少键入两个字符时，将显示相关鸟的名称。

为了提高性能，这里添加了一些本地缓存，其他与远程数据源实例相似。在这里，缓存只保存了一个查询，并可以扩展到缓存多个值，每个条目一个值。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 远程缓存</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .ui-autocomplete-loading {
      background: white url('images/ui-anim_basic_16x16.gif') right center no-repeat;
    }
  </style>
  <script>
    $(function() {
      var cache = {};
      $( "#birds" ).autocomplete({
        minLength: 2,
        source: function( request, response ) {
          var term = request.term;
          if ( term in cache ) {
            response( cache[ term ] );
            return;
          }

          $.getJSON( "search.php", request, function( data, status, xhr ) {
            cache[ term ] = data;
            response( data );
          });
        }
      });
    });
  </script>
</head>
<body>
```

```
<div class="ui-widget">
  <label for="birds">鸟: </label>
  <input id="birds">
</div>

</body>
</html>
```

[查看演示](#)

## 可滚动的结果

当显示一个长列表的选项时，您可以简单地为 autocomplete 菜单设置 max-height 来防止菜单显示太长。尝试键入 "a" 或 "s" 来获得一个可滚动的长列表的结果。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - 可滚动的结果</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .ui-autocomplete {
      max-height: 100px;
      overflow-y: auto;
      /* 防止水平滚动条 */
      overflow-x: hidden;
    }
    /* IE 6 不支持 max-height
    * 我们使用 height 代替，但是这会强制菜单总是显示为那个高度
    */
    * html .ui-autocomplete {
      height: 100px;
    }
  </style>
  <script>
    $(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
});
});
</script>
</head>
<body>

<div class="ui-widget">
  <label for="tags">标签: </label>
  <input id="tags">
</div>


</body>
</html>
```

[查看演示](#)

## XML 数据

本实例演示如何获取一些 XML 数据，并使用 jQuery 的方法解析它，然后把它提供给 autocomplete 作为数据源。

本实例也可作为解析远程 XML 数据源的参考 - 解析在每次 source 回调请求时发生。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 自动完成 (Autocomplete) - XML 数据</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
```

```
<script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
<style>
.ui-autocomplete-loading { background: white url('images/ui-anim_basic_16x16.gif') right center no-repeat; }
</style>
<script>
$(function() {
    function log( message ) {
        $( "<div/>" ).text( message ).prependTo( "#log" );
        $( "#log" ).attr( "scrollTop", 0 );
    }

    $.ajax({
        url: "london.xml",
        dataType: "xml",
        success: function( xmlResponse ) {
            var data = $( "geoname", xmlResponse ).map(function() {
                return {
                    value: $( "name", this ).text() + ", " +
                        ( $.trim( $( "countryName", this ).text() ) || "(unknown country)" ),
                    id: $( "geonameId", this ).text()
                };
            }).get();
            $( "#birds" ).autocomplete({
                source: data,
                minLength: 0,
                select: function( event, ui ) {
                    log( ui.item ?
                        "Selected: " + ui.item.value + ", geonameId: " + ui.item.id :
                        "Nothing selected, input was " + this.value );
                }
            });
        }
    });
</script>
</head>
<body>

<div class="ui-widget">
    <label for="birds">London 匹配: </label>
    <input id="birds">
</div>

<div class="ui-widget" style="margin-top:2em; font-family:Arial">
    结果:
    <div id="log" style="height: 200px; width: 300px; overflow: auto;" class="ui-widget-content"></div>
</div>
```

```
</body>
</html>
```

查看演示

← jQuery UI 实例 – 折叠面板 ( Accordion )

jQuery UI 实例 – 按钮 ( Button ) →

 点我分享笔记