



jQuery教程

极客学院出版

前言

JQuery 是继 prototype 之后又一个优秀的 Javascript 库。

>

前端公开课基础群

Web 前端导学课



群名称：极客学院-前端公开课
群 号：185186017

Web前端介绍—课程概要

- 什么是前端
- Web前端是做什么的？
- Web前端开发需要哪些技能？
- Web前端有哪些职位及其职责
- Web前端开发的一般项目流程
- Web前端开发在企业中的发展
- 前端的未来前景

欢迎加 QQ 群：185186017

说明：专为 **0** 基础用户提供全技术服务交流，定期提供一线大牛**免费**公开课福利。

>

条件：首批开放给：零基础\转行从业者

jQuery 是一套跨浏览器的 [JavaScript 函式库](#)，简化 [HTML](#) 与JavaScript 之间的操作，下面为摘自 Wikipedia 上的 jQuery 说明：

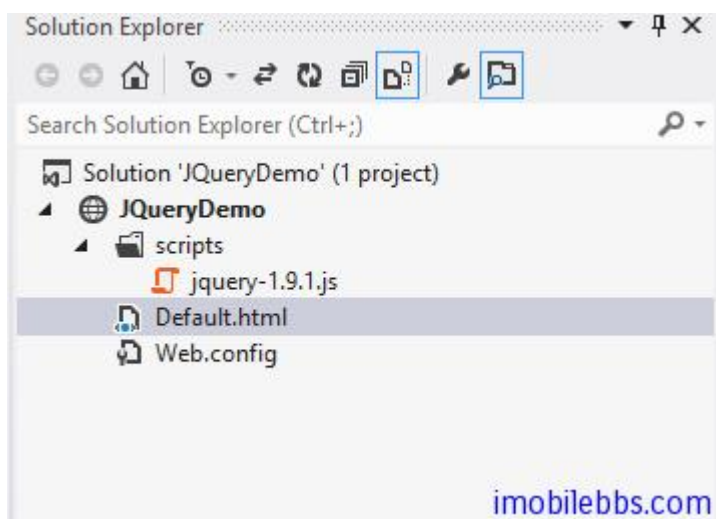
jQuery 是 [开源软件](#)，使用[MIT 许可证](#)授权。[\[4\]](#) jQuery 的语法设计使得许多操作变得容易，如操作文档对象（document）、选择 [DOM](#) 元素、创建[动画](#)效果、处理[事件](#)、以及开发 [Ajax](#) 程序。jQuery 也提供了给开发人员在其上创建插件的能力。这使开发人员可以对底层交互与动画、高级效果和高级主题化的组件进行抽象化。模块化的方式使 jQuery 函数库能够创建功能强大的[动态网页](#)以及[网络应用程序](#)。

- jQuery 有下列特色：
- 跨浏览器的 [DOM](#) 元素选择
- [DOM](#) 巡访与更改：支援 [CSS](#) 1-3与 基本的 [XPath](#)，jQuery 1.2版以后默认取消 [XPath](#) 支持，改为插件支持
- 事件（Events）
- [CSS](#) 操纵
- 特效和动画（移动显示位置、淡入、淡出）
- [Ajax](#)
- 延伸性（Extensibility）
- 工具：例如浏览器版本和each函数。
- [JavaScript](#) 插件
- 轻量级

有很多应用程序开发框架中已经集成了 jQuery，如前面介绍的 [PHP Yii Framework](#) 和 [Java EE Vaadin Web 应用开发框架](#)，这些框架可以不使用 JavaScript 而是使用 PHP 或是 Java 来使用 jQuery 的 UI 功能。但了解 jQuery JavaScripts 编程对于更好编写 Web 页面还是有很大的帮助的，[微软](#)和[诺基亚](#)已宣布在他们的平台上绑定 jQuery。[\[5\]](#) 微软最初在 [Visual Studio](#) 中集成了 jQuery[\[6\]](#) 以便在微软自己的 [ASP.NET AJAX](#) 框架和 [ASP.NET MVC Framework](#) 中使用。因此如果你从事 ASP.Net 或是 Sharepoint Web Part 或是 Application Page 开发了解 jQuery 尤为重要，本教程帮助初学者了解 jQuery 的基本开发，从而为进一步学习其它 Web 编程打好基础。

本教程使用 Visual Studio 为开发工具，你也可以使用你自己喜欢的开发 IDE，Visual Studio 支持 Javascript 的调试，jQuery 使用目前的最新版本 jquery-1.9.1.js. 下载到本地。

使用 Visual Studio 创建一个 Website，将下载下来的jquery-1.9.1.js 放在 scripts



添加的 Default.html 具有如下代码框架作为后续示例的基础：

```
<!doctype html>

<meta charset="utf-8" />
jQuery Demo

<script type="text/javascript" src="scripts/jquery-1.9.1.js"></script><script type="text/javascript">// <![CDATA[
    // Your code goes here

// ]]></script>
```

版本信息

书中演示代码基于以下版本：

| 语言/框架 | 版本信息 |
|--------|--------------|
| jQuery | jQuery 1.9.1 |

目录

| | |
|----------------------------|----|
| 前言. | 1 |
| 第 1 章 基本语法 | 9 |
| jQuery 的基本语法. | 11 |
| 文档准备好事件 | 12 |
| 第 2 章 Selectors | 13 |
| 选择 HTML 标记 | 15 |
| id 选择. | 16 |
| .class 选择器. | 17 |
| 第 3 章 Events | 18 |
| jQuery事件处理的语法 | 20 |
| 常见的jQuery事件 | 21 |
| 第 4 章 显示/隐藏内容. | 24 |
| 第 5 章 淡入淡出效果 | 27 |
| 第 6 章 滑动效果 | 31 |
| 第 7 章 动画效果 | 34 |
| 基本语法如下: | 36 |
| 使用 animate 修改多个属性. | 37 |
| 使用属性相对值 | 38 |
| 使用预定义的值 | 39 |
| 使用队列功能 | 40 |
| 第 8 章 终止动画 | 41 |
| 第 9 章 回调函数 | 43 |
| 第 10 章 方法链 | 45 |

| | | |
|--------|----------------------------------|-----|
| 第 11 章 | HTML Get | 47 |
| 第 12 章 | HTML Set | 50 |
| 第 13 章 | 添加HTML元素 | 53 |
| 第 14 章 | 删除HTML元素 | 58 |
| 第 15 章 | 设置或取得元素的 CSS class | 62 |
| 第 16 章 | 读写 HTML 元素的 css 属性 | 66 |
| 第 17 章 | 操作 HTML 元素的大小 | 69 |
| 第 18 章 | jQuery UI 概述 | 72 |
| | 互動 | 74 |
| | 控制項 | 75 |
| | 效果 | 76 |
| | 工具 | 77 |
| 第 19 章 | jQuery UI 基本工作过程 | 80 |
| | 初始化 | 82 |
| | 方法 | 83 |
| | 通用方法 | 84 |
| | 事件 | 86 |
| | 共有事件 | 87 |
| 第 20 章 | jQuery UI 示例 | 88 |
| 第 21 章 | jQuery UI Accordion 示例 | 92 |
| | 基本用法 | 94 |
| | 摺疊內容 | 97 |
| | 自定义图标 | 98 |
| | 设置 HeightStyle | 99 |
| | 滑鼠移动式自动打开内容 | 103 |
| | 支持调整顺序 | 104 |

| | | |
|--------|--|-----|
| 第 22 章 | jQuery UI Autocomplete 示例（一） | 107 |
| | 基本用法 | 94 |
| | 语调支持 | 111 |
| | 分类支持 | 113 |
| 第 23 章 | jQuery UI Autocomplete 示例（二） | 116 |
| 第 24 章 | jQuery UI Autocomplete 示例（三） | 120 |
| 第 25 章 | jQuery UI Button 示例（一） | 123 |
| | 基本用法 | 94 |
| | checkboxes | 126 |
| | 显示图标 | 127 |
| | Radio 单选钮 | 129 |
| 第 26 章 | jQuery UI Button 示例（二） | 130 |
| 第 27 章 | jQuery UI Datepicker 示例（一） | 134 |
| | 基本用法如下: | 136 |
| | 使用动画效果 | 137 |
| 第 28 章 | jQuery UI Datepicker 示例（二） | 139 |
| | 显示月份和年份选择下拉框 | 140 |
| | 显示多个月份 | 142 |
| 第 29 章 | jQuery UI Datepicker 示例（三） | 143 |
| 第 29 章 | 格式化日期 | 144 |
| | 本地化支持 | 146 |
| 第 30 章 | jQuery UI Datepicker 示例（四） | 148 |
| 第 31 章 | jQuery UI Datepicker 示例（五） | 151 |
| | 设置可以选择的日期范围 | 152 |
| | 设置日期范围 | 154 |
| 第 32 章 | jQuery UI Dialog 示例（一） | 156 |

| | | |
|--------|------------------------------------|-----|
| | 基本用法 | 94 |
| | 动画显示效果 | 159 |
| 第 33 章 | jQuery UI Dialog 示例 (二) | 161 |
| | 模式对话框 | 162 |
| | 添加确认和取消按钮 | 164 |
| 第 34 章 | jQuery UI Dialog 示例 (三) | 166 |
| 第 35 章 | jQuery UI Menu 示例 | 172 |
| | 基本用法 | 94 |
| | 添加图标 | 176 |
| 第 36 章 | jQuery UI Progressbar 示例 | 179 |
| | 基本用法 | 94 |
| | 显示进度 | 182 |
| | “中间过渡”状态条 | 184 |
| 第 37 章 | jQuery UI Slider 示例 (一) | 186 |
| | 基本用法 | 94 |
| | 多个滑块选择值域 | 189 |
| 第 38 章 | jQuery UI Slider 示例 (二) | 191 |
| | 基本用法 | 94 |
| | 垂直选择值域 | 195 |
| 第 39 章 | jQuery UI Spinner 示例 | 197 |
| | 基本用法 | 94 |
| | 显示地图 | 201 |
| 第 40 章 | jQuery UI Tab 示例 (一) | 203 |
| | 基本用法 | 94 |
| 第 41 章 | jQuery UI Tab 示例 (二) | 206 |
| | 支持收缩和展开 | 207 |

| | |
|---------------------------------------|-----|
| Ajax 支持. | 208 |
| 鼠标移动自动激活页面 | 209 |
| 在底部和左边显示标签头 | 210 |
| 第 42 章 jQuery UI Tooltip 示例 | 214 |



基本语法



学习 jQuery 之前需要你下面几个方面的基本知识

- HTML
- CSS
- JavaScript

jQuery 的基本语法

`$(selector).action()`

- `$` 符合定义这是一个 jQuery 语句
- `(selector)` 用来选择某个 HTML 元素，其语法和 CSS 的 selector 语法一样。
- `action()` 定义操作该 HTML 元素的方法。

比如：

`$(this).hide()` - 隐藏当前元素.

`$("p").hide()` - 隐藏所有元素.

`$(".test").hide()` - 隐藏所有类名为 test 的元素.

`$("#test").hide()` - 隐藏 ID 为 test 的元素。

文档准备好事件

几乎所有的 jQuery 都有如下的代码：

```
$(document).ready(function() {  
  
    // jQuery methods go here...  
  
});
```

这为 Document Ready 事件处理器以防止 jQuery 在页面没有完成载入前就执行。从而可以避免下面类似情况发生：

试图隐藏尚未创建好的元素

试图获取尚未载入的图片的大小

这个方法也可以使用下面简化的方法：

```
$(function() {  
  
    // jQuery methods go here...  
  
});
```

你可以选择你喜欢的方式，但通常还是采用 `$(document).ready(function() {` 的方式以便于代码的阅读。



2

Selectors



jQuery Selector 是 jQuery 库中非常重要的一个组成部分。

jQuery Selector 用来选择某个 HTML 元素，其基本语句和 CSS 的选择器 (Selector) 是一样的，所有 jQuery selector 都是以 `$()` 开始。

选择 HTML 标记

选择某个HTML元素的方法是直接使用该元素的标记名称，比如选择所有

<

p> 元素

```
$("#p")
```

下面的例子当用户点击一个按钮时，隐藏所有的

元素

```
$(document).ready(function() {  
    $("#button").click(function() {  
        $("#p").hide();  
    });  
});
```


id 选择

jQuery `#id` 选择器用来选择定义了 `id` 属性的元素，网页上元素的 `id` 应保证是唯一的，你可以使用 `id` 来选择单个唯一的元素。

比如下面的例子，当点击按钮时，只会隐藏 `id` 为 `test` 的元素。

```
<!DOCTYPE html>
html>
head>
script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js">
/script>
script>
$(document).ready(function() {
    $("button").click(function() {
        $("#test").hide();
    });
});
/script>/head>

body>
h2>This is a heading</h2>
p>This is a paragraph.</p>
p id="test">This is another paragraph.</p>
button>Click me</button>
/body>

/html>
```

.class 选择器

jQuery .class 选择器选择所有定义了 class 属性为制定值的所有元素，比如下面的例子 隐藏所有类名称为 test 的元素：

```
<!DOCTYPE html>
html>
head>
script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js">
/script>
script>
$(document).ready(function() {
    $("button").click(function() {
        $(".test").hide();
    });
});
/script>
/head>
body>

h2 class="test">This is a heading</h2>
p class="test">This is a paragraph.</p>
p>This is another paragraph.</p>
button>Click me</button>
/body>
/html>
```



Events



学习了 jQuery 的选择器，就可以针对选择的 HTML 标记或是元素添加事件处理，事件包括按键，鼠标，单击按钮等。

下面为常见的 DOM 事件：

jQuery事件处理的语法

在 jQuery 中，大部分的 DOM 事件都有对应的 jQuery 事件，比如为所有

<

p> 标记添加单击事件处理：

```
$("#p").click();
```

下一步添加单击事件的处理函数，你必须传一个回调函数给单击事件：

```
$("#p").click(function() {  
    // action goes here!!  
});
```

常见的jQuery事件

`$(document).ready()`

文档准备好事件，在文档完全载入后执行。

`click()`

用户单击某个 HTML 元素后调用，比如下面的例子中用户点击

<

p> 标记部分后隐藏该标记的内容：

```
$("#p").click(function() {  
    $(this).hide();  
});
```

`dblclick()`

双击某个 HTML 标记时调用，比如下面代码在双击

<

p> 元素时隐藏其内容：

```
$("#p").dblclick(function() {  
    $(this).hide();  
});
```

`mouseenter()`

在鼠标进入某个 HTML 元素时触发，例如：

```
$("#p1").mouseenter(function() {  
    alert("You entered p1!");  
});
```

mouseleave()

在鼠标进入离开某个 HTML 元素时触发，例如：

```
$("#p1").mouseleave(function() {  
    alert("Bye! You now leave p1!");  
});
```

mousedown()

在按下鼠标左键时触发，例如：

```
$("#p1").mousedown(function() {  
    alert("Mouse down over p1!");  
});
```

mouseup()

在某个元素上释放鼠标左键时触发，例如：

```
$("#p1").mouseup(function() {  
    alert("Mouse up over p1!");  
});
```

hover()

在鼠标指针放在某个元素之上时触发，为 `mouseenter` 和 `mouseleave` 的组合，它的第一个回调函数中鼠标进入时调用，第二个回调函数在鼠标离开时执行，例如：

```
$("#p1").hover(function() {  
    alert("You entered p1!");  
},  
function() {  
    alert("Bye! You now leave p1!");  
});
```

focus()

当某个表单输入域获得焦点时调用, 例如:

```
$("#input").focus(function() {  
    $(this).css("background-color", "#cccccc");  
});
```

blur()

当某个表单输入域失去焦点时调用, 例如:

```
$("#input").blur(function() {  
    $(this).css("background-color", "#ffffff");  
});
```




4

显示/隐藏内容



jQuery 的 `hide()` 和 `show()` 可以用来显示和隐藏内容。比如下面的例子：

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("#hide").click(function () {
        $("p").hide();
      });
      $("#show").click(function () {
        $("p").show();
      });
    });
  </script>
</head>
<body>
  <p>If you click on the "Hide" button, I will disappear.</p>
  <button id="hide">Hide</button>
  <button id="show">Show</button>
</body>
</html>
```

If you click on the "Hide" button, I will disappear.

Hide

Show

imobilebbs.com

基本语法

```
$(selector).hide(speed, callback);
```

```
$(selector).show(speed, callback);
```

可选的参数 `speed` 给出显示或隐藏内容的速度，可以使用 “slow”，“fast” 或者数字代表微秒。第二个可选参数为回调函数，在显示或隐藏结束时调用。下面代码在1秒钟内隐藏内容：

```
$("#button").click(function() {
  $("p").hide(1000);
});
```

jQuery `toggle()` 方法

使用 `toggle()` 方法，可以实现交替显示和隐藏内容，如：

```
$("button").click(function() {  
    $("p").toggle();  
});
```

其基本语法如下：

```
$(selector).toggle(speed, callback);
```

可选的参数 `speed` 给出显示或隐藏内容的速度，可以使用 “slow”，“fast” 或者数字代表微秒。第二个可选参数为回调函数，在显示或隐藏结束时调用



5

淡入淡出效果



jQuery 提供了下面几种方法可以实现显示的淡入淡出效果：

- `fadeIn()`
- `fadeOut()`
- `fadeToggle()`
- `fadeTo()`

`fadeIn()` 方法

`fadeIn()` 实现淡入效果，其基本语法如下：

```
$(selector).fadeIn(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”，”fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `fadeIn()` 方法结束后调用。

```
$("#button").click(function() {  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```

`fadeOut()` 方法

`fadeOut()` 实现淡出效果，其基本语法如下：

```
$(selector).fadeOut(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”，”fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `fadeIn()` 方法结束后调用。

```
$("#button").click(function() {  
    $("#div1").fadeOut();  
    $("#div2").fadeOut("slow");  
    $("#div3").fadeOut(3000);  
});
```

`fadeToggle()` 方法

`fadeToggle()` 交替进行 `fadeIn()` 和 `fadeOut()`，如果元素是淡出的，那么 `fadeToggle()` 将淡入该元素，如果之前是淡入的，`fadeToggle` 将淡出该元素。

其基本语法如下：

```
$(selector).fadeToggle(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”, “fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `fadeIn()` 方法结束后调用。

下面例子，点击按钮之后，可以交替淡出淡入三个矩形。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("#div1").fadeToggle();
        $("#div2").fadeToggle("slow");
        $("#div3").fadeToggle(3000);
      });
    });
  </script>
</head>
<body>

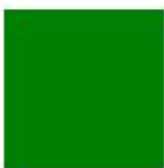
  <p>Demonstrate fadeToggle() with different speed parameters.</p>
  <button>Click to fade in/out boxes</button>
  <br>
  <br>

  <div id="div1" style="width: 80px; height: 80px;
    background-color: red;"></div>
  <br>
  <div id="div2" style="width: 80px; height: 80px;
    background-color: green;"></div>
  <br>
  <div id="div3" style="width: 80px; height: 80px;
    background-color: blue;"></div>

</body>
</html>
```

Demonstrate fadeToggle() with different speed parameters.

Click to fade in/out boxes



imobilebbs.com

fadeTo() 方法

fadeTo() 实现淡化到指定的透明度，其基本语法如下

```
$(selector).fadeTo(speed, opacity, callback);
```

必需参数 speed 给出了淡入效果的时间，可以使用 “slow”, “fast” 或是数值（微秒） 第二个必须参数为透明度，值域为0到1之间。

可选参数为回调函数，在 fadeIn() 方法结束后调用。

```
$("#button").click(function() {  
    $("#div1").fadeTo("slow", 0.15);  
    $("#div2").fadeTo("slow", 0.4);  
    $("#div3").fadeTo("slow", 0.7);  
});
```



滑动效果



jQuery 支持使用下面方法来实现 HTML 元素的滑动效果：

- `slideDown()`
- `slideUp()`
- `slideToggle()`

`slideDown` 方法

用来实现向下滑动动画效果，其基本语法为：

```
$(selector).slideDown(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”, “fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `slideDown()` 方法结束后调用。

下面例子点击按钮 `#panel` 将采用下滑方式显示出来：

```
<!DOCTYPE html>
<html>
<meta charset="utf-8">
<title>jQuery Demo</title>
<script src="scripts/jquery-1.9.1.js"></script>
<script>
    $(document).ready(function () {
        $("#flip").click(function () {
            $("#panel").slideDown("slow");
        });
    });
</script>

<style type="text/css">
    #panel, #flip
    {
        padding: 5px;
        text-align: center;
        background-color: #e5eccc;
        border: solid 1px #c3c3c3;
    }

    #panel
    {
        padding: 50px;
        display: none;
    }
</style>
```

```

</style>
</head>
<body>

    <div id="flip">Click to slide down panel</div>
    <div id="panel">Hello world!</div>

</body>
</html>

```

slideUp 方法

用来实现向上滑动动画效果，其基本语法为：

```
$(selector).slideUp(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”，“fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `slideUp()` 方法结束后调用。

```

$("#flip").click(function() {
    $("#panel").slideUp();
});

```

slideToggle 方法

用来实现交替显示向上向下滑动动画效果，如果之前是下滑显示，`slideToggle` 则显示 `slideUp()`，反之显示 `slideDown()`

其基本语法为：

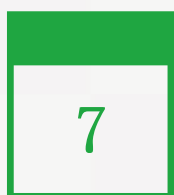
```
$(selector).slideToggle(speed, callback);
```

可选参数 `speed` 给出了淡入效果的时间，可以使用 “slow”，“fast” 或是数值（微秒） 第二个可选参数为回调函数，在 `slideUp()` 方法结束后调用。

```

$("#flip").click(function() {
    $("#panel").slideToggle();
});

```



动画效果



前面的 `hide/show, slide in/out` 其实也具有动画效果，本篇介绍使用 `animate()` 实现自定义动画效果。

基本语法如下：

```
$(selector).animate({params}, speed, callback);
```

必选的参数为 `params`, 定义 CSS 用于动画效果的属性。

可选参数 `speed` 给出了淡入效果的时间, 可以使用 “slow”, “fast” 或是数值 (微秒) 第二个可选参数为回调函数, 在 `animate()` 方法结束后调用。

比如下面的例子, 将一个

<

`div`> 标记移动到其 `Left` 属性等于250px.

```
$("#button").click(function() {  
    $("#div").animate({left:'250px'});  
});
```

要注意的是, 缺省情况下, 所有 HTML 元素的位置是固定的, 不能移动, 因此如果需要修改 HTML 元素的位置, 需要事先将它们的 CSS 属性的位置设置为 `relative`, `fixed`, 或 `absolute`。

使用 animate 修改多个属性

下面的例子，可以使用 animate 同时修改多个属性：

```
$("#button").click(function() {  
    $("#div").animate({  
        left:'250px',  
        opacity:'0.5',  
        height:'150px',  
        width:'150px'  
    });  
});
```

注意：基本所有的 CSS 属性都可以在 animation 中使用，颜色修改不在 jQuery 核心库中，需要Color Animation 插件来完成。

使用属性相对值

对于 CSS 属性，除了上面使用的绝对大小，也可以使用相对值来定义，使用 “+” “-” 。

```
$("#button").click(function() {  
    $("#div").animate({  
        left:'250px',  
        height:'+=150px',  
        width:'+=150px'  
    });  
});
```

使用预定义的值

也可以使用预定义的值属性赋值。比如 “show”，“hide”，或 “toggle”：

```
$("#button").click(function() {  
    $("#div").animate({  
        height: 'toggle'  
    });  
});
```


使用队列功能

缺省 jQuery 支持将多个 animation 功能串起来构成一个队列，然后一个一个的执行队列中的指令。

比如：

```
$("#button").click(function() {  
    var div=$("#div");  
    div.animate({height:'300px',opacity:'0.4'},"slow");  
    div.animate({width:'300px',opacity:'0.8'},"slow");  
    div.animate({height:'100px',opacity:'0.4'},"slow");  
    div.animate({width:'100px',opacity:'0.8'},"slow");  
});
```

或

```
$("#button").click(function() {  
    var div=$("#div");  
    div.animate({left:'100px'},"slow");  
    div.animate({fontSize:'3em'},"slow");  
});
```



T



终止动画



jQuery 的使用 `stop()` 方法在动画结束之前停止动画。

基本语法如下：

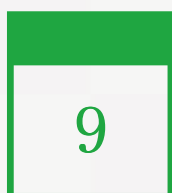
```
$(selector).stop(stopAll, goToEnd);
```

可选参数 `stopAll` 指明是否同时清除“动画队列”，缺省为 `false`. 意味着只停止当前活动的动画，之后的动画则继续运行。

可选参数 `goToEnd` 指明是否立即结束当前动画，缺省为 `false`.

因此缺省的 `stop()` 动作为终止指定 HTML 元素的当前动画效果。如：

```
$("#stop").click(function() {  
    $("#panel").stop();  
});
```



回调函数



JavaScript 语句是一行一行执行的，然而对于前面的动画效果语句，后面的语句可以在动画效果完成前就执行，因此可能会造成错误的结果。所以之前的 `hide`, `show`, `fadeIn`, `fadeOut`, `slideIn`, `slideOut`, `animation` 都支持一个 `callback` 可选参数，支持为这些方法添加一个回调函数，在动画完成之后调用。

典型的语法如下：

```
$(selector).hide(speed, callback);
```

比如，下面的例子中

<

p>隐藏之后执行：

```
$("#button").click(function() {  
    $("#p").hide("slow", function() {  
        alert("The paragraph is now hidden");  
    });  
});
```

而下面的例子的 `Alert` 在

隐藏完成之前就显示。

```
$("#button").click(function() {  
    $("#p").hide(1000);  
    alert("The paragraph is now hidden");  
});
```



10

方法链



jQuery 支持将多个方法链接在一起，比如介绍的多个 `animation()`，这个链接也可以应用到其它方法，构成一个方法链。

使用这个方法，对于同一个元素，链接多个方法，浏览器无需多次查找 HTML 元素。

下面的方法，可以把 `css`, `slideUp`, `slideDown` 串接在一起：

```
$("#p1")  
  .css("color", "red")  
  .slideUp(2000)  
  .slideDown(2000);
```



T



11

HTML Get



jQuery 库包含了很多用来改变和操作 HTML 元素及其属性的方法。

其中一个非常重要的部分就是 jQuery 可以用来操作 DOM。

本篇介绍使用 jQuery 来取得 DOM 节点元素的值或属性。

其中三个简单而有用的方法如下：

- `text()` - 设置或取得指定元素的文本内容。
- `html()` - 设置或取得指定元素的内容（包括 HTML 标记）
- `val()` - 设置或取得表单某个输入域的值。

例如，下面代码使用 `html()` 和 `text()` 方法取得 HTML 元素的内容：

```
$("#btn1").click(function() {
    alert("Text: " + $("#test").text());
});
$("#btn2").click(function() {
    alert("HTML: " + $("#test").html());
});
```

下面的代码取得 Form 中 Input 的内容：

```
$("#btn1").click(function() {
    alert("Value: " + $("#test").val());
});
```

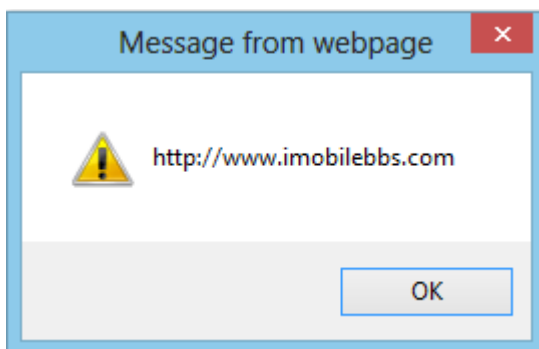
除了上面的方法外，`attr()` 方法用来取得某个元素的属性：

下面代码用来取得链接的 `href` 属性：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>jQuery Demo</title>
<script src="scripts/jquery-1.9.1.js"></script>
</script>
<script>
    $(document).ready(function () {
        $("#button").click(function () {
            alert($("#guidebee").attr("href"));
        });
    });
</script>
```

```
</head>

<body>
<p><a
  href="http://www.imobilebbs.com"
  id="guidebee">
  imobilebbs.com
</a></p>
<button>Show href Value</button>
</body>
</html>
```





12

HTML Set



上篇介绍 HTML Get, jQuery 其实使用上面同样的三个方法来赋值。

- `text()` - 设置或取得指定元素的文本内容。
- `html()` - 设置或取得指定元素的内容（包括 HTML 标记）
- `val()` - 设置或取得表单某个输入域的值。

比如下面代码就是使用上面三种方法给 HTML 元素或 Form 赋值

```
$("#btn1").click(function() {
    $("#test1").text("Hello world!");
});
$("#btn2").click(function() {
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function() {
    $("#test3").val("Dolly Duck");
});
```

用于 `text()`, `html()` 和 `val()` 的回调函数

`text()`, `html()` 和 `val()` 方法可以和一个回调函数配合使用，这个回调函数具有两个参数，一个是选择中元素的序号，第二个为当前值（旧值），然后你可以通过回调函数的返回值做为元素的新值。

例如：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery Demo</title>
    <script src="scripts/jquery-1.9.1.js"></script>
    <script>
        $(document).ready(function () {
            $("#btn1").click(function () {
                $("#test1").text(function (i, origText) {
                    return "Old text: " + origText
                        + " New text: Hello world! (index: " + i + ")";
                });
            });

            $("#btn2").click(function () {
                $("#test2").html(function (i, origText) {
                    return "Old html: " + origText
                        + " New html: Hello <b>world!</b> (index: " + i + ")";
```

```

        });
    });

    });
</script>
</head>

<body>
    <p id="test1">This is a <b>bold</b> paragraph.</p>
    <p id="test2">This is another <b>bold</b> paragraph.</p>
    <button id="btn1">Show Old/New Text</button>
    <button id="btn2">Show Old/New HTML</button>
</body>
</html>

```

Old text: This is a bold paragraph. New text: Hello world! (index: 0)

Old html: This is another bold paragraph. New html: Hello world! (index: 0)

Show Old/New Text

Show Old/New HTML

[imobilebbs.com](http://www.imobilebbs.com)

同样为元素的属性赋值也使用 `attr()` 方法，例如：

```

$("button").click(function() {
    $("#guidebee").attr({
        "href" : "http://www.imobilebbs.com",
        "title" : "imobilebbs jQuery Tutorial"
    });
});

```



13

添加HTML元素



使用 jQuery 可以方便的添加新的 HTML 元素。

下面的方法用于添加 HTML 元素：

- `append()` - 在指定的元素的尾部添加一个新内容。
- `prepend()` - 在指定的元素里前部添加新内容。
- `after()` - 在指定元素前添加新内容
- `before()` - 在指定元素的后面添加新内容。

乍一看 `append`, `prepend` 和 `after`, `before` 似乎功能一样，但 `append`, `prepend` 指在选中的元素本身（内部）的前面和后面，而 `after`, `before` 指在选择中的元素的前面和后面。

可以参考下面的 `append` 例子：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("#btn1").click(function () {
        $("p").append(" <b>Appended text</b>.");
      });

      $("#btn2").click(function () {
        $("ol").append("<li>Appended item</li>");
      });
    });
  </script>
</head>

<body>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
  <ol>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
  </ol>
  <button id="btn1">Append text</button>
  <button id="btn2">Append list items</button>
```

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

This is a paragraph. **Appended text.**

This is another paragraph. **Appended text.**

1. List item 1
2. List item 2
3. List item 3
4. Appended item
5. Appended item

Append text

Append list items

imobilebbs.com

prepend 示例:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("#btn1").click(function () {
        $("p").prepend("<b>Prepended text</b>. ");
      });
      $("#btn2").click(function () {
        $("ol").prepend("<li>Prepended item</li>");
      });
    });
  </script>
</head>
<body>

  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
  <ol>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
  </ol>

  <button id="btn1">Prepend text</button>
  <button id="btn2">Prepend list item</button>
```



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

Prepended text. This is a paragraph.

Prepended text. This is another paragraph.

1. Prepended item
2. Prepended item
3. List item 1
4. List item 2
5. List item 3

Prepend text

Prepend list item

`append()`, `prepend()` 支持同时插入多个元素，下面的例子添加三个使用不同方法创建的新元素：

```
function appendText()
{
    // Create element with HTML
    var txt1="<p>Text.</p>";
    // Create with jQuery
    var txt2=$("<p></p>").text("Text.");
    // Create with DOM
    var txt3=document.createElement("p");
    txt3.innerHTML="Text.";
    // Append the new elements
    $("p").append(txt1,txt2,txt3);
}
```

下面的例子使用 `after`, `before` 在指定的元素前后面添加新内容：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery Demo</title>
    <script src="scripts/jquery-1.9.1.js"></script>
    <script>
        $(document).ready(function () {
            $("#btn1").click(function () {
                $("img").before("<b>Before</b>");
            });

            $("#btn2").click(function () {
```

```

        $("img").after("<i>After</i>");
    });
});
</script>
</head>

<body>
    
    <br>
    <br>
    <button id="btn1">Insert before</button>
    <button id="btn2">Insert after</button>
</body>
</html>

```

**Before***After*

Insert before

Insert after

同样 after, before 也支持同时插入多个元素:

```

function afterText()
{
    // Create element with HTML
    var txt1="<b>I </b>";
    // Create with jQuery
    var txt2=$("<i></i>").text("love ");
    // Create with DOM
    var txt3=document.createElement("big");
    txt3.innerHTML="jQuery!";
    // Insert new elements after img
    $("img").after(txt1,txt2,txt3);
}

```



14

删除HTML元素



jQuery 使用下面两个方法来删除或是清空某个 HTML 元素。

- `remove()` - 删除指定的元素（包括其子元素）
- `empty()` - 清空指定元素的子元素

例如：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("#div1").remove();
      });
    });
  </script>
</head>
<body>

  <div id="div1" style="height: 100px; width: 300px;
    border: 1px solid black; background-color: yellow;">
    This is some text in the div.
    <p>This is a paragraph in the div.</p>
    <p>This is another paragraph in the div.</p>

  </div>
  <br>
  <button>Remove div element</button>

</body>
</html>
```

This is some text in the div.

This is a paragraph in the div.

This is another paragraph in the div.

Remove div element



Remove div element

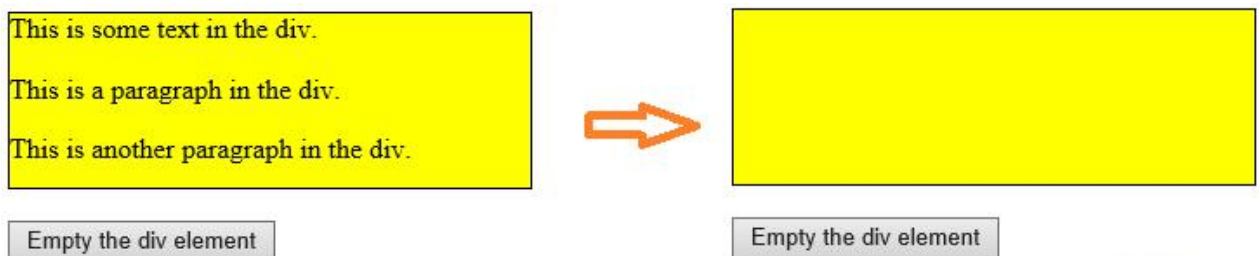
empty() 示例:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("#div1").empty();
      });
    });
  </script>
</head>
<body>

  <div id="div1" style="height: 100px; width: 300px;
    border: 1px solid black; background-color: yellow;">
    This is some text in the div.
    <p>This is a paragraph in the div.</p>
    <p>This is another paragraph in the div.</p>

  </div>
  <br>
  <button>Empty the div element</button>

</body>
</html>
```



imobilebbs.com

jQuery 的 remove() 方法也支持一个参数，可以用于过滤一些需要删除的 HTML 元素。这个参数可以为任何有效的 jQuery selector。比如下面代码只删除 class="italic" 的

<

p>元素:

```
$("#p").remove(".italic");
```



15



设置或取得元素的 CSS class



jQuery 支持方法用来操作 HTML 元素的 CSS 属性

下面的方法为 jQuery 提供的 CSS 操作方法：

- `addClass()` - 为指定的元素添加一个或多个 CSS 类。
- `removeClass()` - 为指定的元素删除一个或多个 CSS 类。
- `toggleClass()` - 为指定的元素在添加/删除 CSS 类之间切换。
- `css()` - 设置或是取得 CSS 类型属性。

下面的 StyleSheet 为后面例子使用的 CSS 风格：

```
.important
{
font-weight:bold;
font-size:xx-large;
}

.blue
{
color:blue;
}
```

jQuery addClass 示例

下面的例子为给定的元素添加 CSS 风格类

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery Demo</title>
  <script src="scripts/jquery-1.9.1.js"></script>

  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("h1,h2,p").addClass("blue");
        $("div").addClass("important");
      });
    });
  </script>
  <style type="text/css">
    .important
    {
```



```

        font-weight: bold;
        font-size: xx-large;
    }

    .blue
    {
        color: blue;
    }
</style>
</head>
<body>

    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <div>This is some important text!</div>
    <br>
    <button>Add classes to elements</button>

</body>
</html>

```

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements



Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements

imobilebbs.com

你也可以在 `addClass` 添加多个类的名称，如：

```

$("button").click(function() {
    $("#div1").addClass("important blue");
});

```

jQuery `removeClass` 示例

```
$("#button").click(function() {  
    $("h1,h2,p").removeClass("blue");  
});
```

jQuery toggle() 示例，下面的例子使用 toggle 为 HTML 元素在添加/删除 CSS 类 blue 之间切换

```
$("#button").click(function() {  
    $("h1,h2,p").toggleClass("blue");  
});
```

下一篇介绍 css() 的用法。



16

读写 HTML 元素的 css 属性



jQuery 的 `css()` 方法用来设置或读取 HTML 元素的 `css` 属性。

读取元素的 CSS 语法语法如下：

```
css(“propertyname “);
```

比如下面代码取得第一个

```
<
```

`p`>元素的背景颜色。

```
$(“p”).css(“background-color”);
```

使用下面的语法来设置 HTML 元素的 CSS 属性：

```
css(“propertyname “, “ value “);
```

例如，下面代码为所有

```
<
```

`p`>元素设置背景色为黄色。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=“utf-8”>
  <title>jQuery Demo</title>
  <script src=“scripts/jquery-1.9.1.js”></script>
  <script>
    $(document).ready(function () {
      $(“button”).click(function () {
        $(“p”).css(“background-color”, “yellow”);
      });
    });
  </script>
</head>

<body>
  <h2>This is a heading</h2>
  <p style=“background-color: #ff0000”>This is a paragraph.</p>
  <p style=“background-color: #00ff00”>This is a paragraph.</p>
  <p style=“background-color: #0000ff”>This is a paragraph.</p>
  <p>This is a paragraph.</p>
  <button>Set background-color of p</button>
```

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

This is a heading

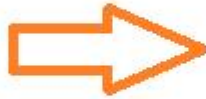
This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Set background-color of p



This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

Set background-color of p

imobilebbs.com

css() 也支持同时多个 CSS 属性：其语法如下：

```
css({ "propertyname ":" value ", "propertyname ":" value ", ... });
```

比如：

```
$("p").css({"background-color":"yellow","font-size":"200%"})
```



17

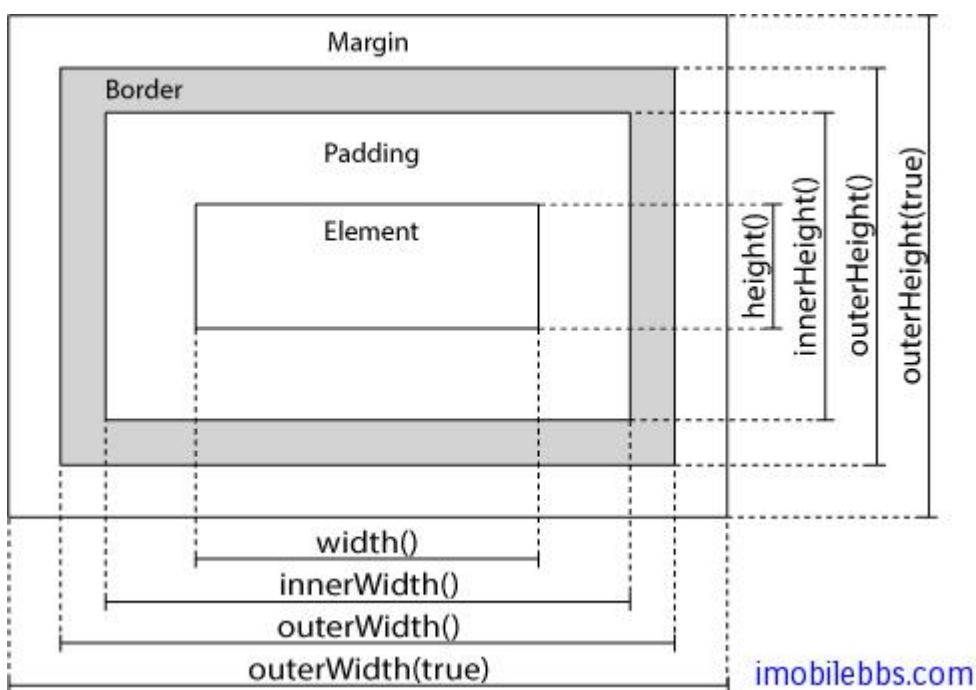
操作 HTML 元素的大小



jQuery 提供下面方法来控制 HTML 元素的大小：

- width()
- height()
- innerWidth()
- innerHeight()
- outerWidth()
- outerHeight()

一般影响 HTML 元素 大小各部分的示意图如下所示：



jQuery width() 和 height() 方法

width() 用来设置或取得元素的宽度，height() 设置和取得元素的高度。

下面代码取得

<

div>元素的高度和宽度。

```
$("#button").click(function() {
    var txt="";
    txt+="Width: " + $("#div1").width() + "<br>";
    txt+="Height: " + $("#div1").height();
});
```

```
$("#div1").html(txt);  
});
```

jQuery 的 `innerWidth()` 和 `innerHeight()` 方法

`innerWidth()` 返回元素包括 `Padding` 的宽度, `innerHeight()` 返回包括 `Padding` 的高度。

jQuery 的 `outerWidth()` 和 `outerHeight()` 方法

`outerWidth()` 返回包括 `padding` 和 `border` 的宽度, `outerHeight()` 返回包括 `padding` 和 `border` 的高度。

而 `outWidth(true)` 和 `outHeight(true)` 返回包括 `padding`, `border` 和 `margin` 的高度和宽度。

下面的例子设置指定

元素的宽度和高度:

```
$("button").click(function() {  
    $("#div1").width(500).height(500);  
});
```




18

jQuery UI 概述



前面介紹了 jQuery 的基本用法，jQuery UI 是一套 [JavaScript 函式庫](#)，提供抽象化、可自訂主題的 [GUI 控制項](#)與動畫效果。基於 [jQuery](#) JavaScript 函式庫，可用來建構互動式的 Web 應用。

它的基本功能有：

互動

- 拖曳 (Draggable) - 讓元素可以用滑鼠拖曳。
- 拖放 (Droppable) - 讓控制項可以接受其他拖曳進來的元素。
- 調整大小 (Resizable) - 讓元素可以調整大小。
- 選取 (Selectable) - 提供進階的大量元素選擇功能。
- 排序 (Sortable) - 讓列表可以更容易排序。

控制項

jQuery UI 的所有控制項都能自訂主題樣式。

- 手風琴式選單 (Accordion) - 如手風琴般可伸縮的控制項。
- 自動完成 (Autocomplete) - 根據使用者的輸入來自動完成文字欄。
- 按鈕 (Button) - 增強按鈕外觀，將單選與複選控制項轉變成按鈕型式。
- 日期選擇器 (Datepicker) - 進階的日期選擇工具。
- 對話框 (Dialog) - 在頁面最上層顯示對話框。
- 選單 (Menu) - 顯示多階層式的選單。
- 進度條 (Progressbar) - 動態與靜態的進度指示條。
- 滑動條 (Slider) - 完全可以自訂的滑動條與各種功能。
- 微調選擇器 (Spinner) - 上下控制的數字微調輸入欄。
- 頁簽 (Tabs) - 頁簽切換控制項，可以內嵌或動態載入內容。
- 工具提示 (Tooltip) - 彈出式提示框。

效果

- 顏色動畫 (Color Animation) - 產生顏色轉變的動畫效果。
- 切換 Class、新增 Class、移除 Class、開關 Class - 讓頁面上元素樣式轉變時有動畫效果。
- 效果 (Effects) - 各種效果 (顯示、下拉、爆炸、淡入等等)。
- 切換 (Toggle) - 切換效果開關。
- 隱藏、顯示 - 使用上述的效果。

工具

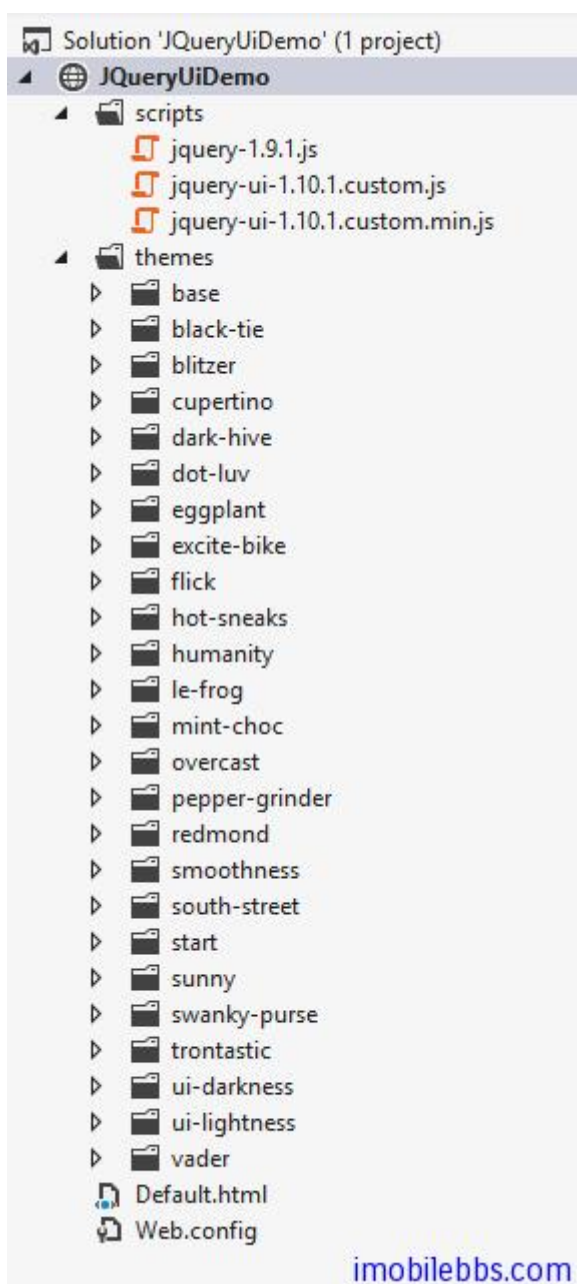
- 位置 (Position) - 根據另一個元素來設定目標元素的位置 (對齊)。

下圖為使用 jQuery UI 一些 UI 組件



imobilebbs.com

後面的例子，我們依然是 visual studio 作為開發 IDE，從 <http://jqueryui.com> 現在最新版的 jQuery 庫，並支持的主題，創建一個 jQueryUIDemo 項目：



scripts 目錄存放 jquery core, jquery ui 庫文件， themes 目錄存放jQuery 自帶的主題，你可以選擇你喜歡的主題，本教程使用其中的「trontastic」主題。創建一個預設的 Default.html 作為測試頁面，如下定義：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
```

```
<script>
    $(function () {

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

其中 指明使用的主题，後面引用 jQuery core, jQuery UI 庫。



19

jQuery UI 基本工作过程



本篇介绍 JQuery UI 组件的基本工作过程，以进程条 (Progressbar) 为例介绍 JQuery UI 组件工作的基本过程。

初始化

大部分 JQuery Ui 组件都可以保持其状态，因此为了能够跟踪 UI 组件的状态，jQuery UI 组件也有一个生命周期，这个生命周期从初始化开始，为了初始化一个 UI 组件，一般在某个 HTML 元素调用 UI 组件（插件）方法。，比如

```
$( "#elem" ).progressbar();
```

这个方法初始化 id=elem 的元素，因为我们调用 progressbar 没有带参数，因此将使用缺省属性来初始化进程条。我们可以通过传入配置的方法为 jQuery UI 组件修改缺省值。比如：

```
$( "#elem" ).progressbar({ value: 20 });
```

也可以一次传入多个参数来初始化 jQuery UI 组件，其它没有配置的属性还是使用其缺省值。这些属性也属于 jQuery UI 组件状态的一部分。初始化之后，如果需要修改这些属性，可以通过 option 方法来改变。

方法

初始化 jQuery UI 组件之后，我们可以查询 UI 组件的状态属性，每个初始化过的 UI 组件都可以调用其方法，调用方法是通过传入方法名称，比如，调用进程条的 value 方法如下：

```
$( "#elem" ).progressbar( "value" );
```

如果这个方法可以有参数，直接在方法名之后传入参数，比如，设置 value 值为40

```
$( "#elem" ).progressbar( "value", 40 );
```

和前面介绍的 jQuery 方法链一样，jQuery UI 的方法也支持多个方法串接在一起，比如：

```
$( "#elem" )  
  .progressbar( "value", 90 )  
  .addClass( "almost-done" );
```

通用方法

对于大多数 jQuery UI 组件来说，有些方法是都支持的，比如：

option 方法

正如前面说的，在 UI 组件初始化之后，如果需要修改一些属性值，可以通过 option 方法，比如，修改 progressbar 的 value 值，

```
$( "#elem" ).progressbar( "option", "value", 30 );
```

注意这和前面调用 value 方法不同，这个例子是想修改配置项 value 为30。如果需要取得某个配置项的值，则使用不带参数的调用：如：

```
$( "#elem" ).progressbar( "option", "value" );
```

除此之外，如果需要一次修改多个配置项，可以通过传入对象的方法，如：

```
$( "#elem" ).progressbar( "option", {  
    value: 100,  
    disabled: true  
});
```

disable 方法

这个方法 disable 某个 UI 组件，比如：

```
$( "#elem" ).progressbar( "disable" );
```

这个方法 disable 进程条，并修改其 Style 使其看起来无效。这个方法等同于设置的 disable 属性为 true.

enable 方法

enable UI 组件，比如：

```
$( "#elem" ).progressbar( "enable" );
```

这个方法等同于设置 disable 属性为 false.

destroy 方法

如果你不再需要某个 UI 组件，可以调用 `destroy()` 方法，这将使的对应的 HTML 元素恢复原状（没有使用 jQuery 之前的标记显示）。这也终结 jQuery UI 的生命周期。一旦你终止 UI 组件，你就不能再调用 UI 组件的方法。如果你删除该 HTML 元素，jQuery 自动终止。

widget 方法

某些 UI 组件创建一个 wrapper 元素或者和原先元素无关联的新元素。在这种情况下，widget 方法返回生成的新元素。对于 Progressbar 来说，没有生成的 HTML 元素，这个方法返回原先的 HTML 元素。

```
$( "#elem" ).progressbar( "widget" );
```

事件

所有 UI 组件都具有事件来通知其状态发生变化。对于大部分 UI 组件来说，某个事件发生时，事件名以 UI 组件名为前缀。比如当 Progressbar 值发生变化时绑定一个事件处理方法：

```
$( "#elem" ).bind( "progressbarchange", function() {  
    alert( "The value has changed!" );  
});
```

每个事件都有对应的回调函数，是作为一个配置项来定义的，因此我们可以直接挂到 Progressbar 的 change 回调函数而不要绑定到 Progressbarchange 事件，比如：

```
$( "#elem" ).progressbar({  
    change: function() {  
        alert( "The value has changed!" );  
    }  
});
```

共有事件

虽然每个 UI 组件支持的事件不同，但 `create` 事件是每个 UI 组件都支持的，这个事件在 UI 组件创建好后立即触发。



20

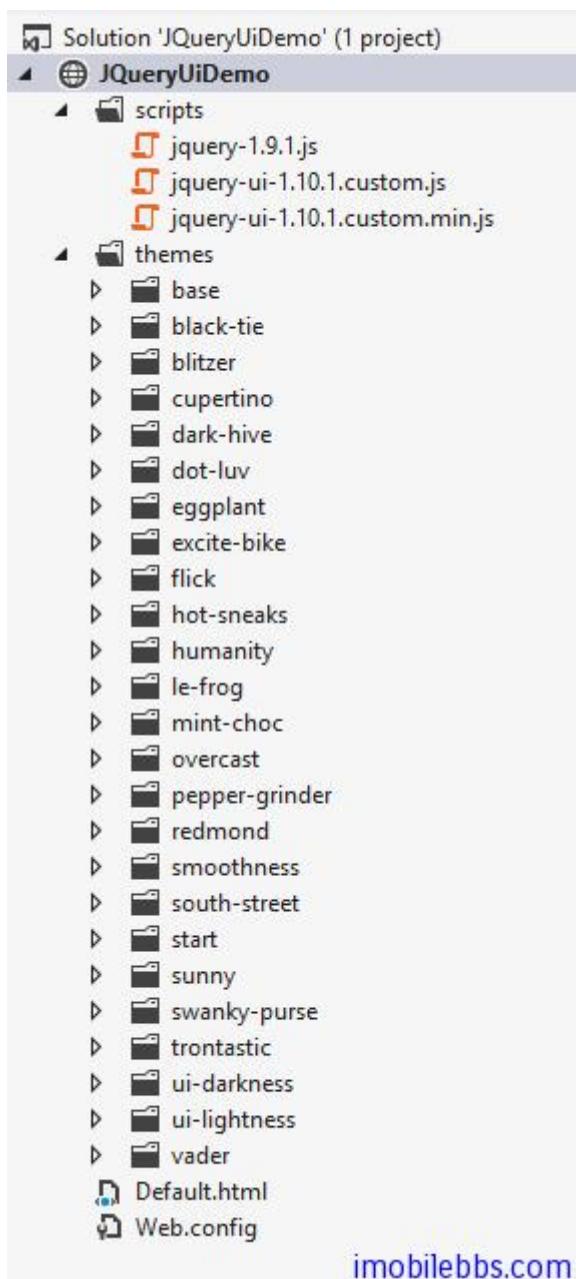
jQuery UI 示例



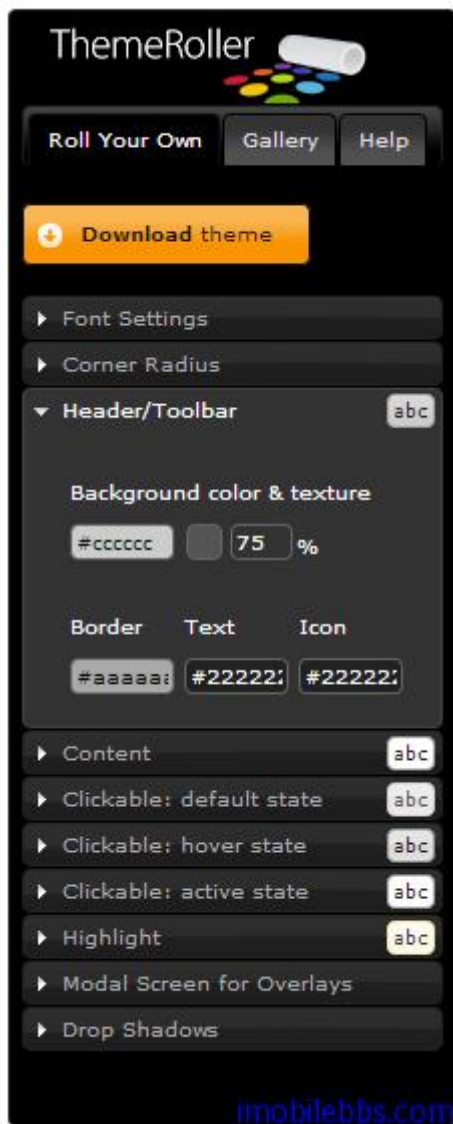
上篇介绍了使用 jQuery UI 基本工作过程，后面就逐个介绍 jQuery UI 库内置的 UI 组件，支持的拖放，动画效果等，如果你之前看过 [Yii Framework 教程](#)，PHP Yii Framework 封装了 jQuery UI 组件，有兴趣的可以看一看。

- [Yii Framework 开发教程\(29\) Zii 组件-Menu 示例](#)
- [Yii Framework 开发教程\(30\) Zii 组件-ListView 示例](#)
- [Yii Framework 开发教程\(31\) Zii 组件-DetailView 示例](#)
- [Yii Framework 开发教程\(32\) Zii 组件-GridView 示例](#)
- [Yii Framework 开发教程\(33\) Zii 组件-Accordion 示例](#)
- [Yii Framework 开发教程\(34\) Zii 组件-AutoComplete 示例](#)
- [Yii Framework 开发教程\(35\) Zii 组件-Button 示例](#)
- [Yii Framework 开发教程\(36\) Zii 组件-DatePicker 示例](#)
- [Yii Framework 开发教程\(37\) Zii 组件-Dialog 示例](#)
- [Yii Framework 开发教程\(38\) Zii 组件-ProgressBar 示例](#)
- [Yii Framework 开发教程\(39\) Zii 组件-Slider 示例](#)
- [Yii Framework 开发教程\(40\) Zii 组件-SliderInput 示例](#)
- [Yii Framework 开发教程\(41\) Zii 组件-Tabs 示例](#)
- [Yii Framework 开发教程\(42\) Zii 组件-Draggable 示例](#)
- [Yii Framework 开发教程\(43\) Zii 组件-Droppable 示例](#)
- [Yii Framework 开发教程\(44\) Zii 组件-Resizable 示例](#)
- [Yii Framework 开发教程\(45\) Zii 组件-Selectable 示例](#)
- [Yii Framework 开发教程\(46\) Zii 组件-Sortable 示例](#)

本教程面的示例基本翻译自 jQuery UI 英文网站，部分例子可能有些修改，测试环境依然是 Visual Studio IDE (2012)。



jQuery UI 也支持多种显示主题。





21

jQuery UI Accordion 示例



本篇介绍 Accordion 组件（类似手风琴可以摺叠的 UI 组件）。

基本用法

jQuery Accordion UI 组件可以把一个包含有具有 Header（标题头）和 Content（内容）对的容器转变成 Accordion 组件。

比如如下一个 Id="accordion" Div HTML 元素。

```
<div id="accordion">
  <h1>Section 1</h1>
  <div>
    <p>Description 1</p>
  </div>
  <h2>Section 2</h2>
  <div>
    <p>Description 3</p>
  </div>
  <h3>Section 3</h3>
  <div>
    <p>Description 3</p>
    <ul>
      <li>List item one</li>
      <li>List item two</li>
      <li>List item three</li>
    </ul>
  </div>
</div>
```

对于 `<div id="accordion">` 裡面的元素，According 允许使用任意的标记，只要是一个 Header 紧接著一个 Content，比如上面的 H3，你可以换成 H1，H2等其它标记，或者通过 Header 选项来配置。有了这样的 HTML 元素，然后对其使用 Accordion 方法，就将该 HTML 元素变成 Accordion 样式了

```
<script>
  $("#accordion").accordion();
</script>
```

本例完整代码如下：

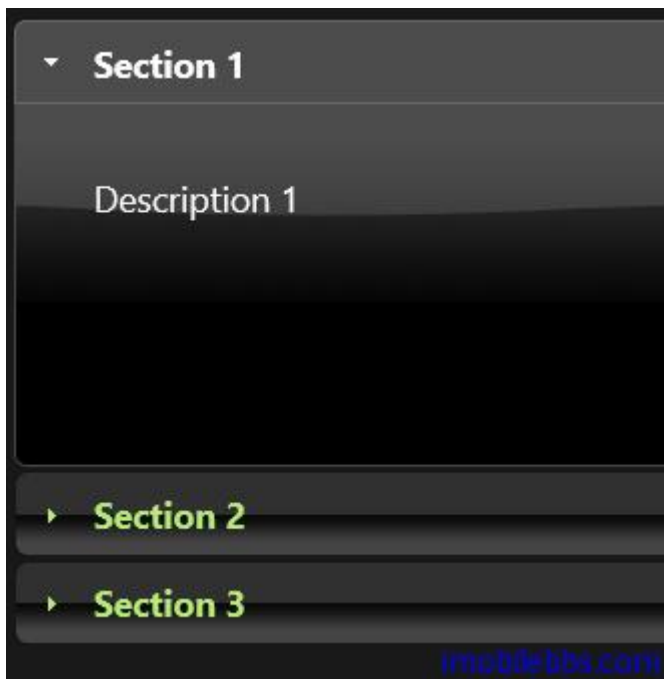
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
```

```
<script src="scripts/jquery-1.9.1.js"></script>
<script src="scripts/jquery-ui-1.10.1.custom.js"></script>

<script>
    $(function () {

    });
</script>
</head>
<body>
    <div id="accordion">
        <h1>Section 1</h1>
        <div>
            <p>Description 1</p>
        </div>
        <h2>Section 2</h2>
        <div>
            <p>Description 3</p>
        </div>
        <h3>Section 3</h3>
        <div>
            <p>Description 3</p>
            <ul>
                <li>List item one</li>
                <li>List item two</li>
                <li>List item three</li>
            </ul>
        </div>
    </div>

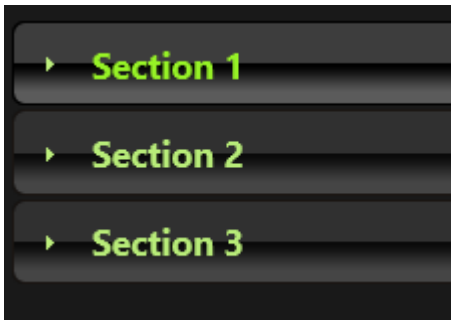
    <script>
        $("#accordion").accordion();
    </script>
</body>
</html>
```

摺叠内容

Accordion 组件预设情况下总有一项内容是展开的，如说图 Section1 的内容是可见的，点击其它部分的标题，该部分内容可见，但无法将所有内容都摺叠起来，如果要支持所有部分都可以摺叠，可以通过配置 collapsible 属性，如：

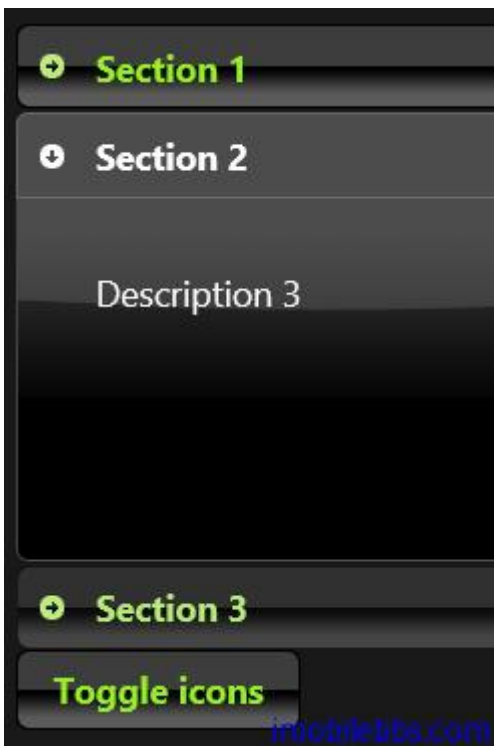
```
$(function() {  
    $( "#accordion" ).accordion({  
        collapsible: true  
    });  
});
```



自定义图标

可以自定义 Accordion 组件标题前的图标，预设的图标为箭头（未展开是箭头向右，展开时向下），使用 jQuery 和 CSS 框架中定义类或是通过自定义背景图像类，可以重新配置这两个图标：

```
$(function() {
    var icons = {
        header: "ui-icon-circle-arrow-e",
        activeHeader: "ui-icon-circle-arrow-s"
    };
    $( "#accordion" ).accordion({
        icons: icons
    });
    $( "#toggle" ).button().click(function() {
        if ( $( "#accordion" ).accordion( "option", "icons" ) ) {
            $( "#accordion" ).accordion( "option", "icons", null );
        } else {
            $( "#accordion" ).accordion( "option", "icons", icons );
        }
    });
});
```



设置 HeightStyle

因为 Accordion 由「Block 级」元素组成（可以参加 CSS 的 display），因此它预设在水方向占据父元素的宽度，为了在高度方向也能充满其父容器，可以通过配置 heightStyle 为 fill。HeightStyle 可以使用的值如下：

- "auto"：所有 Panel 使用最高的那个 Panel 的高度。
- "fill"：根据父容器的高度来填充。
- "content"：每个 Panel 的高度取决于其内容。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>

  <style>
    #accordion-resizer {
      padding: 10px;
      width: 350px;
      height: 300px;
    }
  </style>
  <script>
    $(function () {
      $("#accordion").accordion({
        heightStyle: "fill"
      });
    });
    $(function () {
      $("#accordion-resizer").resizable({
        minHeight: 140,
        minWidth: 200,
        resize: function () {
          $("#accordion").accordion("refresh");
        }
      });
    });
  </script>
</head>
<body>
  <div id="accordion-resizer">
    <div id="accordion">
      <div>Panel 1</div>
      <div>Panel 2</div>
      <div>Panel 3</div>
    </div>
  </div>
</body>
</html>
```

```

</script>
</head>
<body>
  <h3 class="docs">Resize the outer container:</h3>

  <div id="accordion-resizer" class="ui-widget-content">
    <div id="accordion">
      <h3>Section 1</h3>
      <div>
        <p>Mauris mauris ante, blandit et,
          ultrices a, suscipit eget,
          quam. Integer ut neque. Vivamus nisi metus,
          molestie vel, gravida in,
          condimentum sit amet, nunc. Nam a nibh.
          Donec suscipit eros. Nam mi.
          Proin viverra leo ut odio.
          Curabitur malesuada.
          Vestibulum a velit eu ante
          scelerisque vulputate.</p>
      </div>
      <h3>Section 2</h3>
      <div>
        <p>Sed non urna. Donec et ante.
          Phasellus eu ligula.
          Vestibulum sit amet purus.
          Vivamus hendrerit,
          dolor at aliquet laoreet,
          mauris turpis porttitor velit,
          faucibus interdum tellus
          libero ac justo.
          Vivamus non quam.
          In suscipit faucibus urna. </p>
      </div>
      <h3>Section 3</h3>
      <div>
        <p>Nam enim risus, molestie et,
          porta ac,
          aliquam ac, risus. Quisque lobortis.
          Phasellus pellentesque purus in massa.
          Aenean in pede. Phasellus
          ac libero ac tellus
          pellentesque semper.
          Sed ac felis. Sed commodo,
          magna quis lacinia ornare,
          quam ante aliquam nisi,

```

```
        eu iaculis leo purus
        venenatis dui. </p>
    <ul>
        <li>List item one</li>
        <li>List item two</li>
        <li>List item three</li>
    </ul>
</div>
<h3>Section 4</h3>
<div>
    <p>Cras dictum. Pellentesque
        habitant morbi
        tristique senectus et netus
        et malesuada
        fames ac turpis egestas.
        Vestibulum
        ante ipsum primis in faucibus
        orci luctus
        et ultrices posuere cubilia
        Curae;
        Aenean lacinia mauris
        vel est. </p>
    <p>Suspendisse eu nisl. Nullam ut libero.
        Integer dignissim consequat lectus.
        Class aptent taciti sociosqu ad litora
        torquent per conubia nostra,
        per inceptos himenaeos. </p>
</div>
</div>
</div>
</body>
</html>
```

Resize the outer container:

▼ **Section 1**

Mauris mauris ante, blandit et,
ultrices a, suscipit eget, quam.
Integer ut neque. Vivamus nisi
metus, molestie vel, gravida in,
condimentum sit amet, nunc.
Nam a nibh. Donec suscipit
eros. Nam mi. Proin viverra leo.

▶ **Section 2**

▶ **Section 3**

▶ **Section 4**

滑鼠移动式自动打开内容

预设情况下，打开 Accordion 某个部分内容，是点击该部分标题，如果实现移动滑鼠到该部分自动展开内容，可以设置 event 属性，例如：

```
$(function () {  
    $("#accordion").accordion({  
        event: "mouseover"  
    });  
});
```


支持调整顺序

使用 `sortable` 方法允许重新调整 Accordion 每个部分的顺序，可以通过拖放的方法调整顺序：

[查看源代码列印帮助](#)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <style>
    /* IE has layout issues when sorting (see #5413) */
    .group {
      zoom: 1;
    }
  </style>
  <script>
    $(function () {
      $("#accordion").accordion({
        header: "> div > h3",
        collapsible: true
      }).sortable({
        axis: "y",
        handle: "h3",
        stop: function (event, ui) {
          // IE doesn't register the blur when sorting
          // so trigger focusout handlers to remove .ui-state-focus
          ui.item.children("h3").triggerHandler("focusout");
        }
      });
    });
  </script>
</head>
<body>
  <div id="accordion">
    <div class="group">
      <h3>Section 1</h3>

      <div>
```

```
        <p>Description 1</p>
    </div>
</div>
<div class="group">
    <h3>Section 2</h3>
    <div>
        <p>Description 2</p>
    </div>
</div>
<div class="group">
    <h3>Section 3</h3>

    <div>
        <p>Description 3</p>
        <ul>
            <li>List item one</li>
            <li>List item two</li>
            <li>List item three</li>
        </ul>
    </div>
</div>
<div class="group">
    <h3>Section 4</h3>

    <div>
        <p>Description 4</p>
    </div>
</div>
</div>
</body>
</html>
```



注意，这裡使用 Header 属性，标题部分由 div ,h3 共同构成。



22



jQuery UI Autocomplete 示例（一）



AutoComplete 在获取焦点后，随着用户键入的内容，可以在预订的数据源中查找和已输入的内容相匹配的内容列表供用户选择。

这可以用作之前输入过的内容也可以用作自动填充相关内容，比如根据城市名，自动填充邮编等。你可以使用本地数据源或是远程数据源，本地数据一般使用小数据集合，比如包含50条记录的通讯录，远程数据源一般为保护大量记录的数据库。

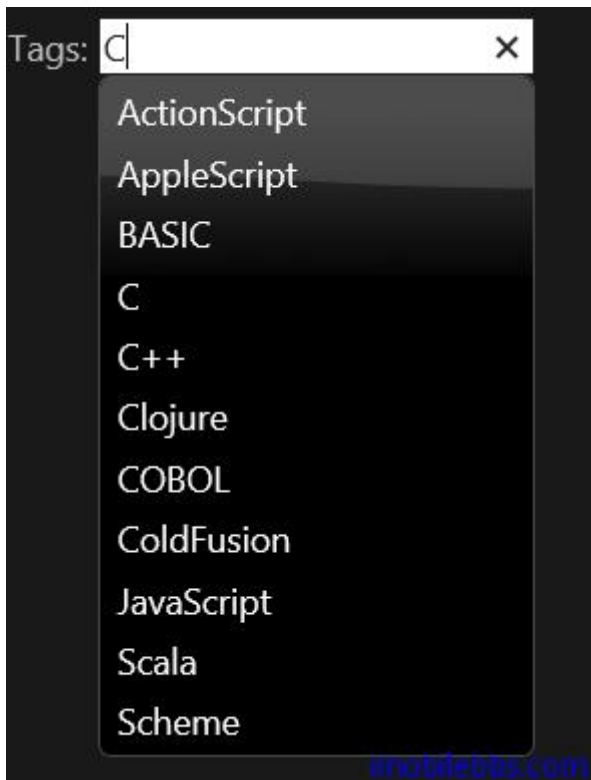
基本用法

本例为使用 AutoComplete 的基本用法，通过本地数据源（数组）定义一组语言列表，用户输入字母后，包含该字母的语言会作为列表显示出来：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>

  <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>
    <input type="text" value="tags" />
  </div>
</body>
</html>
```

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



语调支持

某些语言支持语调字符，比如 Jörn 中的 ö，希望在输入 o 时，也能显示包含 ö 的内容，AutoComplete 支持使用 function 来定义 Source 属性：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>

  <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)
          }));
        }
      });
    });
  </script>
</head>
<body>
  <div>
    <input type="text" value="Jörn Zaefferer" />
  </div>
</body>
</html>
```



```
        || matcher.test(normalize(value));  
    }));  
    }  
    });  
    });  
</script>  
</head>  
<body>  
    <div class="ui-widget">  
        <form>  
            <label for="developer">Developer: </label>  
            <input id="developer" />  
        </form>  
    </div>  
</body>  
</html>
```



分类支持

本例是提供简单扩展 AutoComplete 组件实现了一个自定义的 custom.catcomplete UI 组件以支持 AutoComplete 的分类支持。自定义组件有兴趣的可以参见 jQuery 的 Widget Factory。一般无需自定义 UI 组件，如果需要，可以在网站查找是否有人已经实现你需要的 UI 组件，实在不行才自定义 UI 组件，使用 Widget Factory 自定义组件的方法并不十分直观（这是因为 JavaScript 使用了面向“原型”的面向对象方法，而非通常的使用类的面向对象方法）。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>

  <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>
```

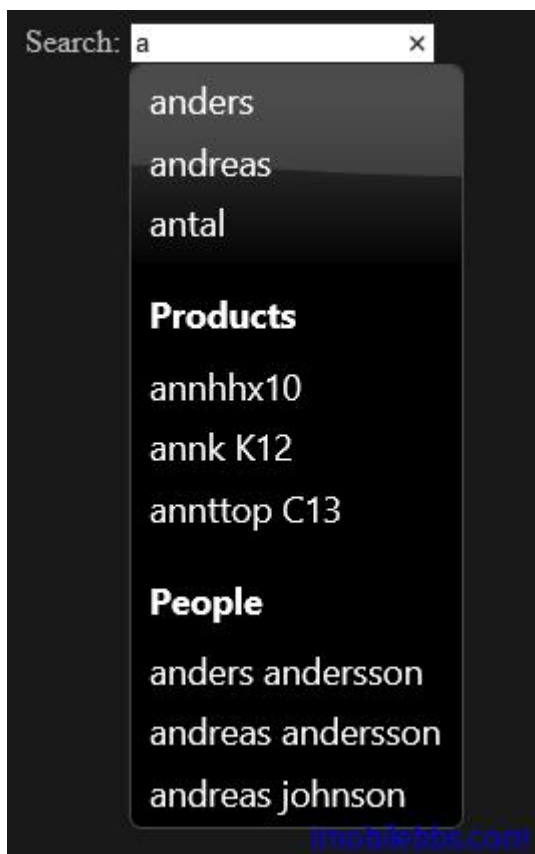
```

$(function () {
    var data = [
        { label: "anders", category: "" },
        { label: "andreas", category: "" },
        { label: "antal", category: "" },
        { label: "annhxx10", 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">Search: </label>
    <input id="search" />
</body>
</html>

```

其中 `custom.catcomplete` 为自定义的 UI 组件，因此 `$("#search").catcomplete` 使用 `catcomplete`，而非 `autocomplete`。





23



jQuery UI Autocomplete 示例（二）



如果一個輸入框可以接受多個輸入項，比如選擇你喜歡的語言，以逗號隔開，這是也可以使用 AutoComplete 為每個輸入項提供輸入提示。不過此時除了設置數據源外，還需要添加一些事件處理。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>

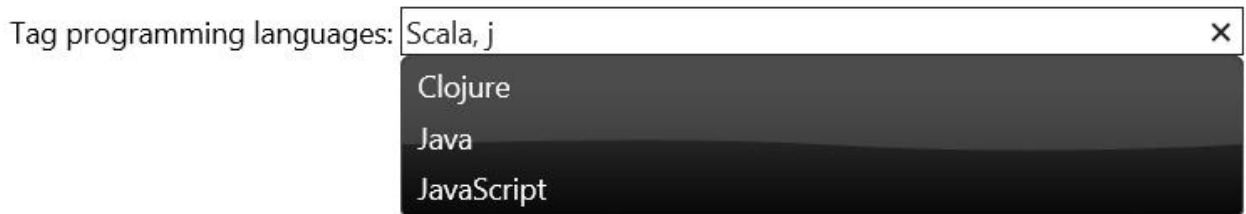
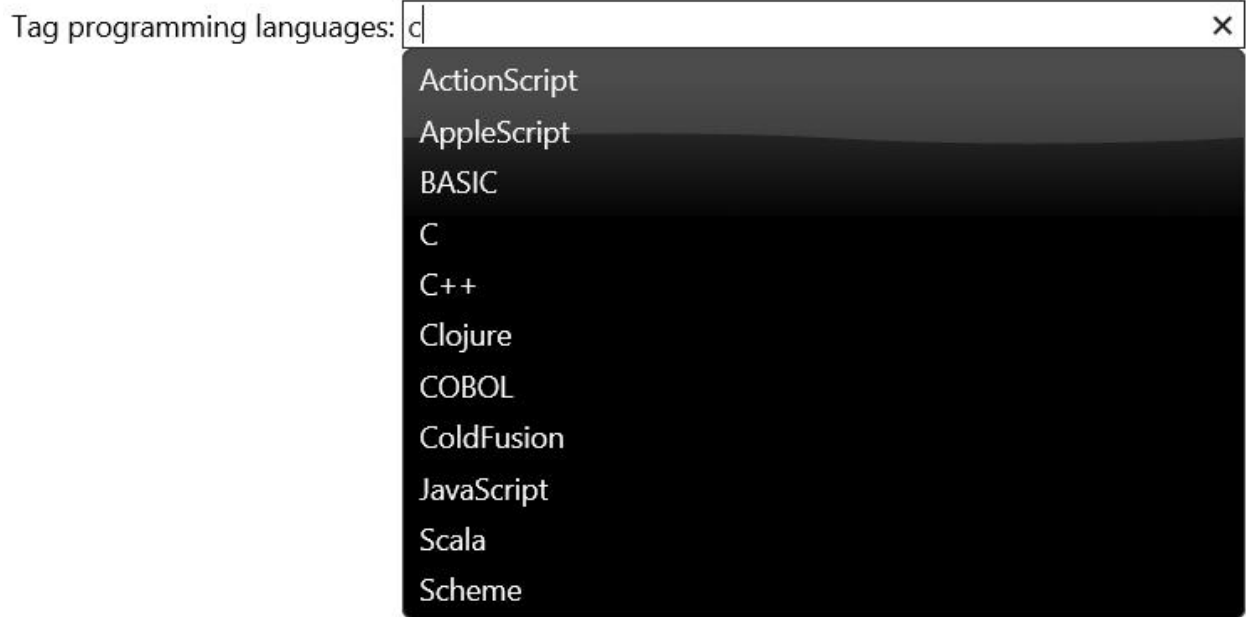
  <script>
    $(function () {
      var availableTags = [
        "ActionScript",
        "AppleScript",
        "Asp",
        "BASIC",
        "C",
        "C++",
        "Clojure",
        "COBOL",
        "ColdFusion",
        "Erlang",
        "Fortran",
        "Groovy",
        "Haskell",
        "Java",
        "JavaScript",
        "Lisp",
        "Perl",
        "PHP",
        "Python",
        "Ruby",
        "Scala",
        "Scheme"
      ];
      function split(val) {
        return val.split(/,\s*/);
      }
      function extractLast(term) {
        return split(term).pop();
      }
    });
```

```

$( "#tags" )
// don't navigate away from the field on tab
//when selecting an item
    .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 ) {
            // delegate back to autocomplete,
            // but extract the last term
            response( $.ui.autocomplete.filter(
                availableTags, extractLast( request.term ) ) );
        },
        focus: function () {
            // prevent value inserted on focus
            return false;
        },
        select: function ( event, ui ) {
            var terms = split( this.value );
            // remove the current input
            terms.pop();
            // add the selected item
            terms.push( ui.item.value );
            // add placeholder to get the
            //comma-and-space at the end
            terms.push( "" );
            this.value = terms.join( ", " );
            return false;
        }
    });

});
</script>
</head>
<body>
    <div class="ui-widget">
        <label for="tags">Tag programming languages: </label>
        <input id="tags" size="50" />
    </div>
</body>
</html>

```





24



jQuery UI Autocomplete 示例（三）



AutoComplete 在用戶輸入時，可以根據用戶輸入給出提示。其數據源可以來自本地，也可以使用遠程數據源，AutoComplete 的 DataSource 可以使用一個 Function，本篇中的 Function，我們使用 JSONP 查詢。

當數據源為 Function 時，其函數定義為：

```
Function( Object request, Function response( Object data) )
```

其中 request 為請求，它含有一個 term 屬性，為當前用戶輸入。

response 為一回調函數，它有一個參數 data，用來提示用戶的數據，這個值理應根據輸入的 term 進行過濾。

本例使用由 [genames](#) 提供的 Web Service。可以查詢世界的城市名稱。使用了 jQuery 提供的.ajax 函數來訪問這個 Web Service。

```
$( "#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.countryName,
            value: item.name
          };
        }));
      }
    });
  },
});
```

Your city:

Result:

- Perth, Western Australia, Australia
- Perth, Scotland, United Kingdom
- Perth, Ontario, Canada
- Perth, New York, United States
- Perth, North Dakota, United States
- Perth, Delaware, United States
- Perth, Indiana, United States
- Perth, Tasmania, Australia
- Perth, Mahaica-Berbice, Guyana
- Perth, North-West, South Africa
- Perth, Manchester, Jamaica
- Perth, Kansas, United States

imobilebbs.com



25



jQuery UI Button 示例（一）



jQuery Button 组件可以增强表单 (Form) 中的 Buttons, Inputs 和 Anchor 元素, 使其具有按钮显示风格, 能够正确对鼠标滑动做出反应。

基本用法

下例显示把表单中的 button, input 和 anchor 元素都变为按钮风格的 jQuery Button 组件。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("input[type=submit], a, button")
        .button()
        .click(function (event) {
          event.preventDefault();
        });
    });
  </script>
</head>
<body>

  <button>A button element</button>

  <input type="submit" value="A submit button" />

  <a href="#">An anchor</a>

</body>
</html>
```

A button element

A submit button

An anchor

imobilebbs.com

checkboxes

除了支持基本的按钮外，jQuery Button 组件可以把类型为 checkbox 的 input 元素变为按钮，这种按钮可以有两种状态，原态和按下状态。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("input[type=submit], a, button")
        .button()
        .click(function (event) {
          event.preventDefault();
        });
    });
  </script>
</head>
<body>
  <button>A button element</button>
  <input type="submit" value="A submit button" />
  <a href="#">An anchor</a>
</body>
</html>
```



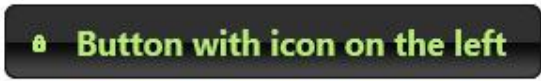
显示图标

按钮也可以添加图标，可以支持多个图标，分别使用 `primary` 和 `secondary` 来指明。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("button:first").button({
        icons: {
          primary: "ui-icon-locked"
        },
        text: false
      }).next().button({
        icons: {
          primary: "ui-icon-locked"
        }
      }).next().button({
        icons: {
          primary: "ui-icon-gear",
          secondary: "ui-icon-triangle-1-s"
        }
      }).next().button({
        icons: {
          primary: "ui-icon-gear",
          secondary: "ui-icon-triangle-1-s"
        },
        text: false
      });
    });
  </script>
</head>
<body>
  <button>Button with icon only</button>
  <button>Button with icon on the left</button>
  <button>Button with two icons</button>
  <button>Button with two icons and no text</button>
```



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



imobilebbs.com

Radio 单选钮

同样, jQuery 也把 type 为 radio 的一组 Radio 按钮构成一组单选钮, 使用 .buttonset 将多个单选钮定义为一个组, 其中只有一个可以是选中状态。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#radio").buttonset();
    });
  </script>
</head>
<body>

<form>
  <div id="radio">
    <input type="radio" id="radio1" name="radio" />
    <label for="radio1">Choice 1</label>
    <input type="radio" id="radio2" name="radio" checked="checked" />
    <label for="radio2">Choice 2</label>
    <input type="radio" id="radio3" name="radio" />
    <label for="radio3">Choice 3</label>
  </div>
</form>
</body>
</html>
```





26



jQuery UI Button 示例（二）



本例为使用 jQuery 的一个实用的例子，显示媒体播放器的控制条。其中按钮的图标使用 jQuery 库自带的 CSS 定义的一些图标（比如 ui-icon-seek-end 等）。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <style>
    #toolbar {
      padding: 4px;
      display: inline-block;
    }
    /* support: IE7 */
    * + html #toolbar {
      display: inline;
    }
  </style>
  <script>
    $(function () {
      $("#beginning").button({
        text: false,
        icons: {
          primary: "ui-icon-seek-start"
        }
      });
      $("#rewind").button({
        text: false,
        icons: {
          primary: "ui-icon-seek-prev"
        }
      });
      $("#play").button({
        text: false,
        icons: {
          primary: "ui-icon-play"
        }
      })
      .click(function () {
        var options;
        if ($(this).text() === "play") {
          options = {
```

```

        label: "pause",
        icons: {
            primary: "ui-icon-pause"
        }
    };
} else {
    options = {
        label: "play",
        icons: {
            primary: "ui-icon-play"
        }
    };
}

$(this).button("option", options);
});

$("#stop").button({
    text: false,
    icons: {
        primary: "ui-icon-stop"
    }
});

.click(function () {
    $("#play").button("option", {
        label: "play",
        icons: {
            primary: "ui-icon-play"
        }
    });
});

$("#forward").button({
    text: false,
    icons: {
        primary: "ui-icon-seek-next"
    }
});

$("#end").button({
    text: false,
    icons: {
        primary: "ui-icon-seek-end"
    }
});

$("#shuffle").button();
$("#repeat").buttonset();
});
</script>

```

```

</head>
<body>

<div id="toolbar" class="ui-widget-header ui-corner-all">
  <button id="beginning">go to beginning</button>
  <button id="rewind">rewind</button>
  <button id="play">play</button>
  <button id="stop">stop</button>
  <button id="forward">fast forward</button>
  <button id="end">go to end</button>

  <input type="checkbox" id="shuffle" />
  <label for="shuffle">Shuffle</label>

  <span id="repeat">
    <input type="radio" id="repeat0" name="repeat"
      checked="checked" />
    <label for="repeat0">No Repeat</label>
    <input type="radio" id="repeat1" name="repeat" />
    <label for="repeat1">Once</label>
    <input type="radio" id="repeatall" name="repeat" />
    <label for="repeatall">All</label>
  </span>
</div>
</body>
</html>

```





27



jQuery UI Datepicker 示例（一）



jQuery UI 提供的 Datepicker 是一个有着非常灵活配置的选择日期的 UI 组件，你可以配置显示日期的格式，语言，限制所能选择的日期范围，添加按钮等。

基本用法如下：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
</script>
  <script>
    $(function () {
      $("#datepicker").datepicker();
    });
  </script>
</head>
<body>

<p>Date: <input type="text" id="datepicker" /></p>
</body>
</html>
```

只需要给选择的 HTML 元素调用其构造函数即可。



使用动画效果

显示和隐藏 DatePicker 组件支持配置使用不同的动画效果，下列使用一个列表框来选择 DatePicker 可以支持的动画效果。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#datepicker").datepicker();
      $("#anim").change(function () {
        $("#datepicker").datepicker("option",
          "showAnim", $(this).val());
      });
    });
  </script>
</head>
<body>

  <p>Date:
    <input type="text" id="datepicker" size="30" /></p>

  <p>
    Animations:<br />
    <select id="anim">
      <option value="show">Show (default)</option>
      <option value="slideDown">Slide down</option>
      <option value="fadeIn">Fade in</option>
      <option value="blind">Blind (UI Effect)</option>
      <option value="bounce">Bounce (UI Effect)</option>
      <option value="clip">Clip (UI Effect)</option>
      <option value="drop">Drop (UI Effect)</option>
      <option value="fold">Fold (UI Effect)</option>
      <option value="slide">Slide (UI Effect)</option>
      <option value="">None</option>
    </select>
  </p>
```

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



28



jQuery UI Datepicker 示例（二）



显示月份和年份选择下拉框

changeMonth 和 changeYear 选择可以打开和关闭月份和年份下拉选择框。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#datepicker").datepicker({
        changeMonth: true,
        changeYear: true
      });
    });
  </script>
</head>
<body>
  <p>
    Date:
    <input type="text" id="datepicker" />
  </p>
</body>
</html>
```


显示多个月份

选项 `numberOfMonths` 可以用控制同时显示几个月份：

```
$(function () {  
    $("#datepicker").datepicker({  
        numberOfMonths: 3,  
        showButtonPanel: true  
    });  
});
```

Date:





29



jQuery UI Datepicker 示例（三）





第 29 章 格式化日期



可以通過日期格式重新定義 Datepicker 顯示日期時的格式。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#datepicker").datepicker();
      $("#format").change(function () {
        $("#datepicker").datepicker("option",
          "dateFormat", $(this).val());
      });
    });
  </script>
</head>
<body>

  <p>Date:
    <input type="text" id="datepicker" size="30" /></p>
  <p>
    Format options:<br />
    <select id="format">
      <option value="mm/dd/yy">Default - mm/dd/yy</option>
      <option value="yy-mm-dd">ISO 8601 - yy-mm-dd</option>
      <option value="d M, y">Short - d M, y</option>
      <option value="d MM, y">Medium - d MM, y</option>
      <option value="DD, d MM, yy">Full - DD, d MM, yy</option>
      <option value="'day' d 'of' MM 'in the year' yy">
        With text - 'day' d 'of' MM 'in the year' yy
      </option>
    </select>
  </p>
</body>
</html>
```

Date:

Format options:



本地化支持

Datepicker 預設使用英語顯示，可以通過配置修改顯示語言。

如果需要支持不同語言，可以添加，如：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script src="scripts/jquery.ui.datepicker-ar.js"></script>
  <script src="scripts/jquery.ui.datepicker-fr.js"></script>
  <script src="scripts/jquery.ui.datepicker-he.js"></script>
  <script src="scripts/jquery.ui.datepicker-zh-TW.js"></script>

  <script>
    $(function () {
      $.datepicker.setDefaults($.datepicker.regional[""]);
      $("#datepicker").datepicker($.datepicker.regional["fr"]);
      $("#locale").change(function () {
        $("#datepicker").datepicker("option",
          $.datepicker.regional[$(this).val()]);
      });
    });
  </script>
</head>
<body>
  <p>
    Date:
    <input type="text" id="datepicker" />&nbsp;
    <select id="locale">
      <option value="ar">Arabic ((????</option>
      <option value="zh-TW">Chinese Traditional (繁體中文)</option>
      <option value="fr" selected="selected">French (Français)</option>
      <option value="he">Hebrew ((????</option>
    </select>
  </p>
</body>
</html>
```



如果需要添加自定義的語言，比如簡體中文，可以打開 `jquery.ui.datepicker-zh-TW.js` 看看，

```
/* Chinese initialisation for the jQuery UI date picker plugin. */
/* Written by Ressel (ressol@gmail.com). */
jQuery(function($) {
    $.datepicker.regional['zh-TW'] = {
        closeText: '關閉',
        prevText: '<#x3C;上月',
        nextText: '下月>#x3E;',
        currentText: '今天',
        monthNames: ['一月', '二月', '三月', '四月', '五月', '六月',
            '七月', '八月', '九月', '十月', '十一月', '十二月'],
        monthNamesShort: ['一月', '二月', '三月', '四月', '五月', '六月',
            '七月', '八月', '九月', '十月', '十一月', '十二月'],
        dayNames: ['星期日', '星期一', '星期二', '星期三', '星期四', '星期五', '星期六'],
        dayNamesShort: ['周日', '周一', '周二', '周三', '周四', '周五', '周六'],
        dayNamesMin: ['日', '一', '二', '三', '四', '五', '六'],
        weekHeader: '周',
        dateFormat: 'yy/mm/dd',
        firstDay: 1,
        isRTL: false,
        showMonthAfterYear: true,
        yearSuffix: '年' };
    $.datepicker.setDefaults($.datepicker.regional['zh-TW']);
});
```

只要把 `zh-TW` 改成 `zh-CN`，把其中的繁體「關閉」改成「關閉」 保存為 `jquery.ui.datepicker-zh-CN.js`，然後使用 `zh-CN` 作為區域選項即可。



30



jQuery UI Datepicker 示例（四）



DatePicker 支持使用另外的按钮控制日期选择界面显示，可以通过自定义的图标来显示这个按钮。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#datepicker").datepicker({
        showOn: "button",
        buttonImage: "images/calendar.gif",
        buttonImageOnly: true
      });
    });
  </script>
</head>
<body>

<p>Date: <input type="text" id="datepicker" /></p>

</body>
</html>
```

buttonImage 设置按钮显示图标。





31



jQuery UI Datepicker 示例（五）



设置可以选择的日期范围

有时希望用户在给定的日期内选择，比如预约会议的时间，只能在当天开始的一个月带10天以内。这时可以通过配置 `minDate` 和 `maxDate` 来设置，如果 `minDate` 或 `maxDate` 没有配置，表示没有最小日期或最大日期的限制。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#datepicker").datepicker({
        minDate: 0,
        maxDate: "+1M +10D"
      });
    });
  </script>
</head>
<body>

<p>Date: <input type="text" id="datepicker" /></p>

</body>
</html>
```



可以看到小于当天的日期变灰且无法选择。

设置日期范围

可以使用两个 DatePicker 配合使用，用户可以选择一个开始日期和一个终止日期。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#from").datepicker({
        defaultDate: "+1w",
        changeMonth: true,
        numberOfMonths: 3,
        onClose: function (selectedDate) {
          $("#to").datepicker("option", "minDate", selectedDate);
        }
      });
      $("#to").datepicker({
        defaultDate: "+1w",
        changeMonth: true,
        numberOfMonths: 3,
        onClose: function (selectedDate) {
          $("#from").datepicker("option", "maxDate", selectedDate);
        }
      });
    });
  </script>
</head>
<body>

  <label for="from">From</label>
  <input type="text" id="from" name="from" />
  <label for="to">to</label>
  <input type="text" id="to" name="to" />

</body>
</html>
```

From to

| Mar 2013 | | | | | | | April 2013 | | | | | | | May 2013 | | | | | | |
|----------|----|----|----|----|----|----|------------|----|----|----|----|----|----|----------|----|----|----|----|----|----|
| Su | Mo | Tu | We | Th | Fr | Sa | Su | Mo | Tu | We | Th | Fr | Sa | Su | Mo | Tu | We | Th | Fr | Sa |
| | | | | | 1 | 2 | | 1 | 2 | 3 | 4 | 5 | 6 | | | | 1 | 2 | 3 | 4 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 28 | 29 | 30 | | | | | 26 | 27 | 28 | 29 | 30 | 31 | |
| 31 | | | | | | | | | | | | | | | | | | | | |

moxie8.com



32



jQuery UI Dialog 示例（一）



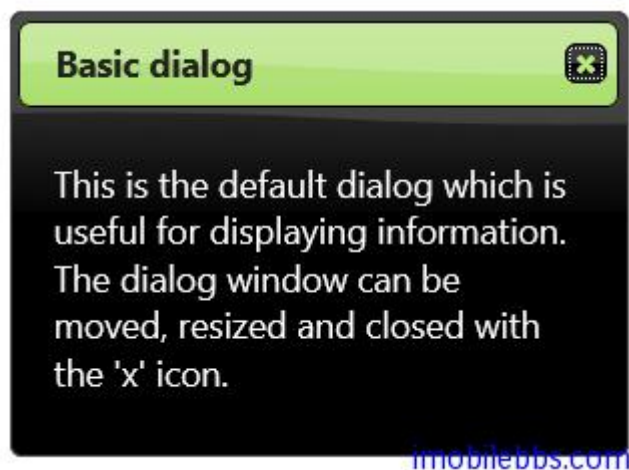
jQuery Dialog 组件用来显示对话框，模式或非模式的。

基本用法

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#dialog").dialog();
    });
  </script>
</head>
<body>

<div id="dialog" title="Basic dialog">
  <p>This is the default dialog which
    is useful for displaying information.
    The dialog window can be moved,
    resized and closed with the 'x' icon.</p>
</div>

</body>
</html>
```



对话框的缺省显示有“X”关闭按钮，可以缩放，移动。

动画显示效果

可以为对话框显示和关闭添加动画效果，如果不希望对话框一开始就显示（这可能是大部分情况，在点击按钮或是某个事件发生后再显示对话框）可以通过配置 `autoOpen=false` 来设置。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#dialog").dialog({
        autoOpen: false,
        show: {
          effect: "blind",
          duration: 1000
        },
        hide: {
          effect: "explode",
          duration: 1000
        }
      });

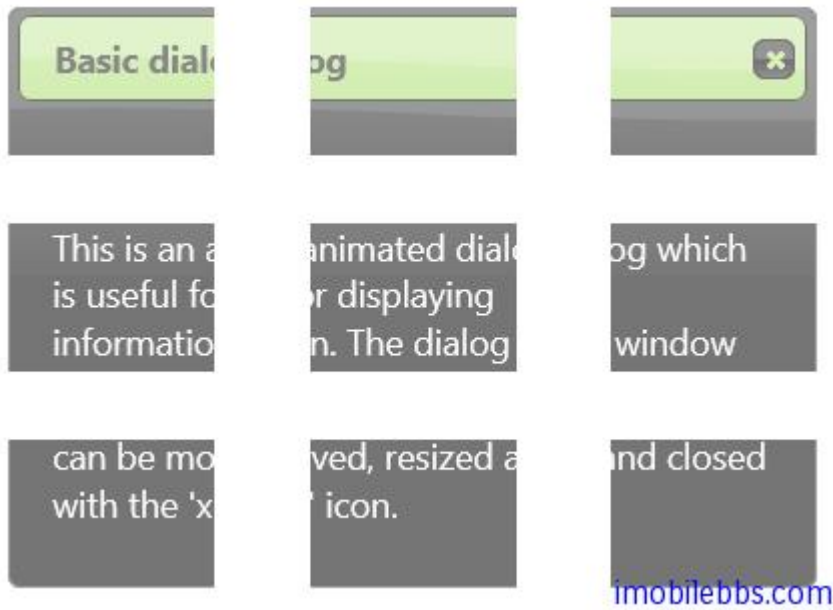
      $("#opener").click(function () {
        $("#dialog").dialog("open");
      });
    });
  </script>
</head>
<body>

<div id="dialog" title="Basic dialog">
  <p>This is an animated dialog which is useful
    for displaying information.
    The dialog window can be moved,
    resized and closed with the 'x' icon.</p>
</div>

<button id="opener">Open Dialog</button>
```



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



show 和 hide 支持的动画效果，后面再专门介绍，这些效果同时使用与其它方面，为 jQuery 支持的通用的动态效果。



33



jQuery UI Dialog 示例（二）



模式对话框

如想对话框显示为模式的，可以通过配置 `modal: true` 来设置。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#dialog-modal").dialog({
        height: 140,
        modal: true
      });
    });
  </script>
</head>
<body>

  <div id="dialog-modal" title="Basic modal dialog">
    <p>
      Adding the modal overlay screen makes the dialog
      look more prominent because it dims out the page content.
    </p>
  </div>

  <p>
    Sed vel diam id libero <a href="http://example.com">rutrum convallis</a>.
    Donec aliquet leo vel magna. Phasellus rhoncus faucibus ante.
    Etiam bibendum, enim faucibus aliquet rhoncus,
    arcu felis ultricies neque, sit amet auctor elit eros a lectus.
  </p>
</body>
</html>
```

Sed vel diam id libero [rutrum convallis](#). Donec aliquet leo vel magna. Phasellus rhoncus faucibus ante. Etiam bibendum, enim faucibus aliquet rhoncus, arcu felis ultricies neque, sit amet auctor elit eros a lectus.

Basic modal dialog



Adding the modal overlay screen makes the dialog look more prominent because it dims out the page content.

添加确认和取消按钮

可以通过配置主按钮，和从按钮的方法来为对话框添加按钮，此时如果需要把”X”从右上角去掉，可以通过 CSS 来实现，具体可以参考下例：使用 dialogClass: “no-close”

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <style>
    .no-close .ui-dialog-titlebar-close {
      display: none;
    }
  </style>
  <script>
    $(function () {
      $("#dialog-confirm").dialog({
        dialogClass: "no-close",
        resizable: false,
        width: 400,
        height: 250,
        modal: true,
        buttons: {
          "Delete all items": function () {
            $(this).dialog("close");
          },
          Cancel: function () {
            $(this).dialog("close");
          }
        }
      });
    });
  </script>
</head>
<body>

  <div id="dialog-confirm" title="Empty the recycle bin?">
    <p>
```

```

        <span class="ui-icon ui-icon-alert"
            style="float: left; margin: 0 7px 20px 0;"></span>
        These items will be permanently deleted
        and cannot be recovered. Are you sure?
    </p>
</div>

<p>
    Sed vel diam id libero <a href="http://example.com">rutrum convallis</a>.
    Donec aliquet leo vel magna. Phasellus rhoncus faucibus ante.
    Etiam bibendum, enim faucibus aliquet rhoncus,
    arcu felis ultricies neque,
    sit amet auctor elit eros a lectus.
</p>

</body>
</html>

```

Sed vel diam id libero [rutrum convallis](#). Donec aliquet leo vel magna. Phasellus rhoncus faucibus ante. Etiam bibendum, enim faucibus aliquet rhoncus, arcu felis ultricies neque, sit amet auctor elit eros a lectus.





34



jQuery UI Dialog 示例（三）



本篇使用 Dialog 构造一个比较实用的对话框，它可以内嵌一个表单用来接受用户输入，每个输入框支持数据校验，部分使用正则表达式来检验。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <style>
    body {
      font-size: 62.5%;
    }

    label, input {
      display: block;
    }

    input.text {
      margin-bottom: 12px;
      width: 95%;
      padding: .4em;
    }

    fieldset {
      padding: 0;
      border: 0;
      margin-top: 25px;
    }

    h1 {
      font-size: 1.2em;
      margin: .6em 0;
    }

    div#users-contain {
      width: 350px;
      margin: 20px 0;
    }

    div#users-contain table {
      margin: 1em 0;
      border-collapse: collapse;
```



```

        width: 100%;
    }

    div#users-contain table td, div#users-contain table th {
        border: 1px solid #eee;
        padding: .6em 10px;
        text-align: left;
    }

    .ui-dialog .ui-state-error {
        padding: .3em;
    }

    .validateTips {
        border: 1px solid transparent;
        padding: 0.3em;
    }
</style>
<!-- ReSharper disable ExpressionIsAlwaysConst -->
<script>
    $(function () {
        var name = $("#name"),
            email = $("#email"),
            password = $("#password"),
            allFields = $([]).add(name).add(email).add(password),
            tips = $(".validateTips");

        function updateTips(t) {
            tips
                .text(t)
                .addClass("ui-state-highlight");
            setTimeout(function () {
                tips.removeClass("ui-state-highlight", 1500);
            }, 500);
        }

        function checkLength(o, n, min, max) {
            if (o.val().length > max || o.val().length < min) {
                o.addClass("ui-state-error");
                updateTips("Length of " + n + " must be between " +
                    min + " and " + max + ".");
                return false;
            } else {
                return true;
            }
        }
    });

```

```

}

function checkRegexp(o, regexp, n) {
    if (!(regexp.test(o.val()))) {
        o.addClass("ui-state-error");
        updateTips(n);
        return false;
    } else {
        return true;
    }
}

$("#dialog-form").dialog({
    autoOpen: false,
    height: 300,
    width: 350,
    modal: true,
    buttons: {
        "Create an account": function () {
            var bValid = true;
            allFields.removeClass("ui-state-error");

            bValid = bValid && checkLength(name, "username", 3, 16);
            bValid = bValid && checkLength(email, "email", 6, 80);
            bValid = bValid && checkLength(password, "password", 5, 16);

            bValid = bValid && checkRegexp(name, /^[a-z]([0-9a-z_])+$/i,
                "Username may consist of a-z, 0-9, " +
                "underscores, begin with a letter.");
            // From jquery.validate.js (by joern),
            // contributed by Scott Gonzalez:
            // http://projects.scottplayground.com/email_address_validation/
            bValid = bValid && checkRegexp(email,
                /^((([a-z]|\d|!#$%&'*\+\-\/=?\^_`{|}~) | [\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF]))+(\.
            bValid = bValid && checkRegexp(password,
                /^[0-9a-zA-Z])+$/,
                "Password field only allow : a-z 0-9");

            if (bValid) {
                $("#users tbody").append("<tr>" +
                    "<td>" + name.val() + "</td>" +
                    "<td>" + email.val() + "</td>" +
                    "<td>" + password.val() + "</td>" +
                    "</tr>");
                $(this).dialog("close");
            }
        }
    }
});

```

```

        }
    },
    Cancel: function () {
        $(this).dialog("close");
    }
},
close: function () {
    allFields.val("").removeClass("ui-state-error");
}
});

$("#create-user")
    .button()
    .click(function () {
        $("#dialog-form").dialog("open");
    });
});
</script>
<!-- ReSharper restore ExpressionIsAlwaysConst -->
</head>
<body>

    <div id="dialog-form" title="Create new user">
        <p class="validateTips">All form fields are required.</p>

        <form>
            <fieldset>
                <label for="name">Name</label>
                <input type="text" name="name" id="name"
                    class="text ui-widget-content ui-corner-all" />
                <label for="email">Email</label>
                <input type="text" name="email" id="email"
                    value="" class="text ui-widget-content ui-corner-all" />
                <label for="password">Password</label>
                <input type="password" name="password"
                    id="password" value=""
                    class="text ui-widget-content ui-corner-all" />
            </fieldset>
        </form>
    </div>

    <div id="users-contain" class="ui-widget">
        <h1>Existing Users:</h1>
        <table id="users" class="ui-widget ui-widget-content">

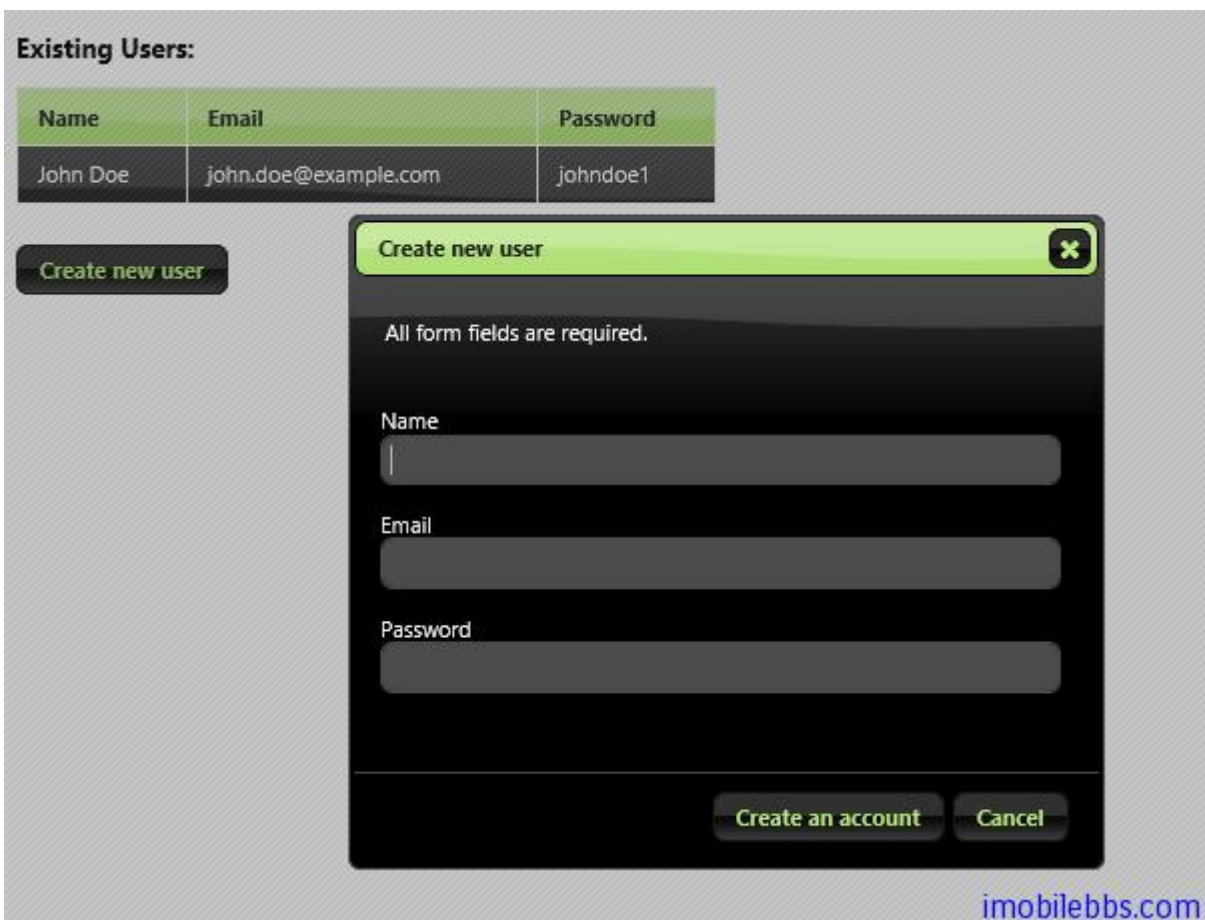
```

```

<thead>
  <tr class="ui-widget-header">
    <th>Name</th>
    <th>Email</th>
    <th>Password</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>johndoe1</td>
  </tr>
</tbody>
</table>
</div>
<button id="create-user">Create new user</button>

</body>
</html>

```





35

jQuery UI Menu 示例



jQuery Menu 组件可以应用到任何具有父/子关系的元素，就其变为菜单，但通常使用 `ul, li`，如果你希望使用除 `ul, li` 之外的元素，可以通过 `menus` 来配置。下例使用缺省的 `ui` 和 `li` 菜单支持选择事件 `select`，因此可以为菜单添加事件处理。

基本用法

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>jQuery UI Demos</title>
    <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
    <script src="scripts/jquery-1.9.1.js"></script>
    <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
    <script>
        $(function () {
            $("#menu").menu({
                select: function(event, ui) {
                    alert(ui.item.context.innerText);
                }
            });
        });
    </script>
    <style>
        .ui-menu {
            width: 150px;
        }
    </style>
</head>
<body>

    <ul id="menu">
        <li class="ui-state-disabled"><a href="#">Aberdeen</a></li>
        <li><a href="#">Ada</a></li>
        <li><a href="#">Adamsville</a></li>
        <li><a href="#">Addyston</a></li>
        <li>
            <a href="#">Delphi</a>
            <ul>
                <li class="ui-state-disabled"><a href="#">Ada</a></li>
                <li><a href="#">Saarland</a></li>
                <li><a href="#">Salzburg</a></li>
            </ul>
        </li>
        <li><a href="#">Saarland</a></li>
        <li>
            <a href="#">Salzburg</a>

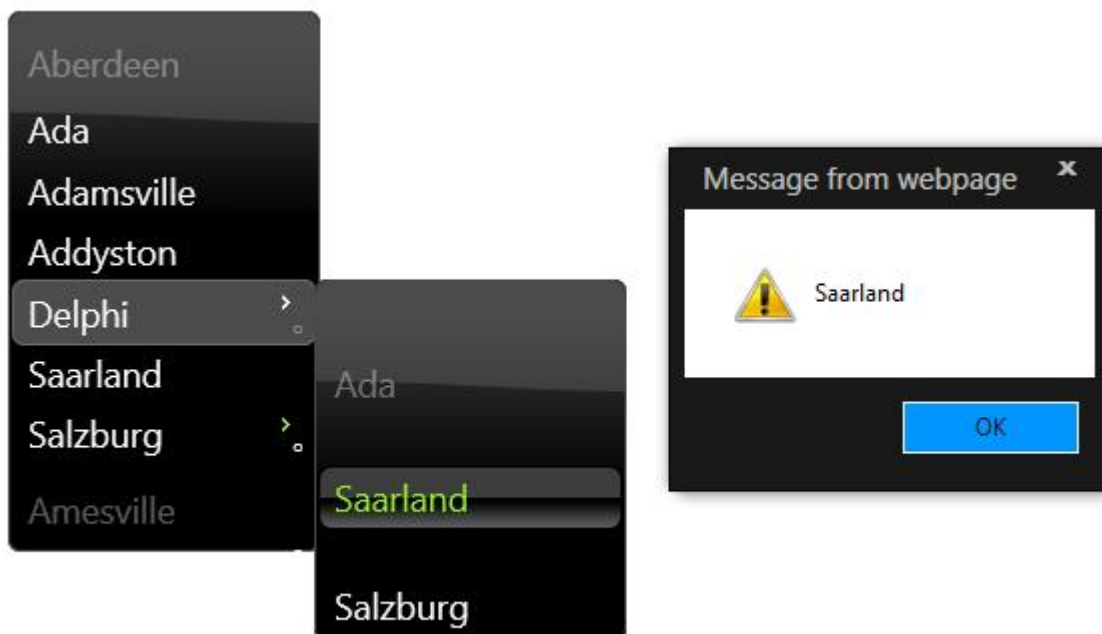
```

```

        <ul>
            <li>
                <a href="#">Delphi</a>
                <ul>
                    <li><a href="#">Ada</a></li>
                    <li><a href="#">Saarland</a></li>
                    <li><a href="#">Salzburg</a></li>
                </ul>
            </li>
            <li>
                <a href="#">Delphi</a>
                <ul>
                    <li><a href="#">Ada</a></li>
                    <li><a href="#">Saarland</a></li>
                    <li><a href="#">Salzburg</a></li>
                </ul>
            </li>
            <li><a href="#">Perch</a></li>
        </ul>
    </li>
    <li class="ui-state-disabled"><a href="#">Amesville</a></li>
</ul>

</body>
</html>

```



添加图标

可以为菜单项添加图标，这是通过 定义不同的 CSS 类来添加的。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#menu").menu();
    });
  </script>
  <style>
    .ui-menu {
      width: 150px;
    }
  </style>
</head>
<body>

  <ul id="menu">
    <li>
      <a href="#">
        <span class="ui-icon ui-icon-disk"></span>Save
      </a>
    </li>
    <li>
      <a href="#">
        <span class="ui-icon ui-icon-zoomin"></span>Zoom In
      </a>
    </li>
    <li>
      <a href="#">
        <span class="ui-icon ui-icon-zoomout"></span>Zoom Out
      </a>
    </li>
    <li class="ui-state-disabled">
      <a href="#">
```

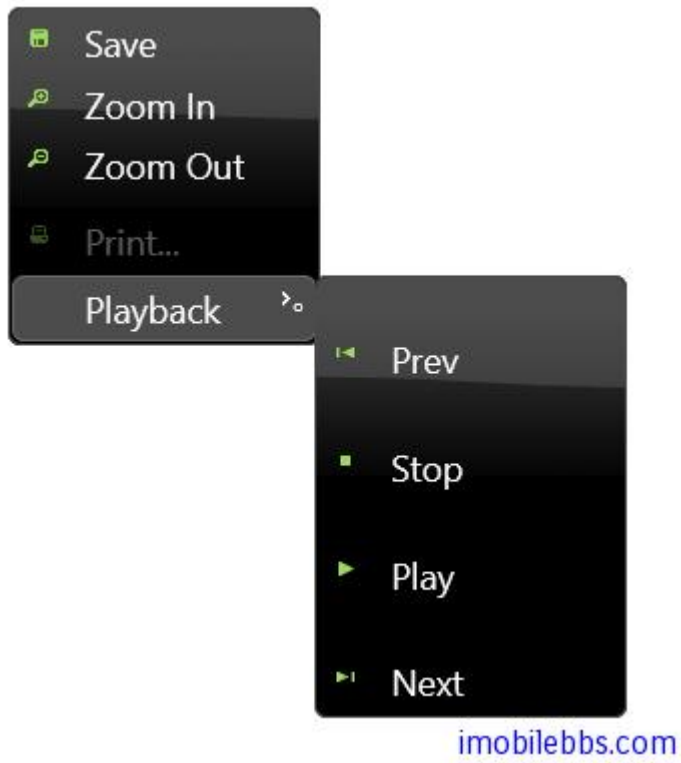
```

        <span class="ui-icon ui-icon-print"></span>
        Print...

    </a></li>
</li>
<a href="#">Playback</a>
<ul>
    <li>
        <a href="#">
            <span class="ui-icon ui-icon-seek-start"></span>Prev
        </a></li>
    <li>
        <a href="#">
            <span class="ui-icon ui-icon-stop"></span>Stop
        </a></li>
    <li>
        <a href="#">
            <span class="ui-icon ui-icon-play"></span>Play
        </a>
    </li>
    <li>
        <a href="#">
            <span class="ui-icon ui-icon-seek-end"></span>Next
        </a>
    </li>
</ul>
</li>
</ul>

</body>
</html>

```





36



jQuery UI Progressbar 示例



前面在介绍 [jQuery 入门教程\(20\): jQuery UI 基本工作过程](#)时简要介绍过 Progressbar 用法。在使用进度条时，如果可以预知进度的大小，可以设置 Max 大小，如果对于一些无法预约时间比如下载文件可以使用“中间状态”的状态条。

基本用法

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#progressbar").progressbar({
        value: 37
      });
    });
  </script>
</head>
<body>

  <div id="progressbar"></div>

</body>
</html>
```

显示进度

可以在显示进度条的同时显示当前的百分比（实际上可以显示任意文字），这是通过两个嵌套的 `div` 元素来实现，本例使用一个定时器来模拟进度条的动态效果。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <style>
    .progress-label {
      float: left;
      margin-left: 50%;
      margin-top: 5px;
      font-weight: bold;
      text-shadow: 1px 1px 0 #fff;
    }
  </style>
  <script>
    $(function () {
      var progressbar = $("#progressbar"),
          progressLabel = $(".progress-label");

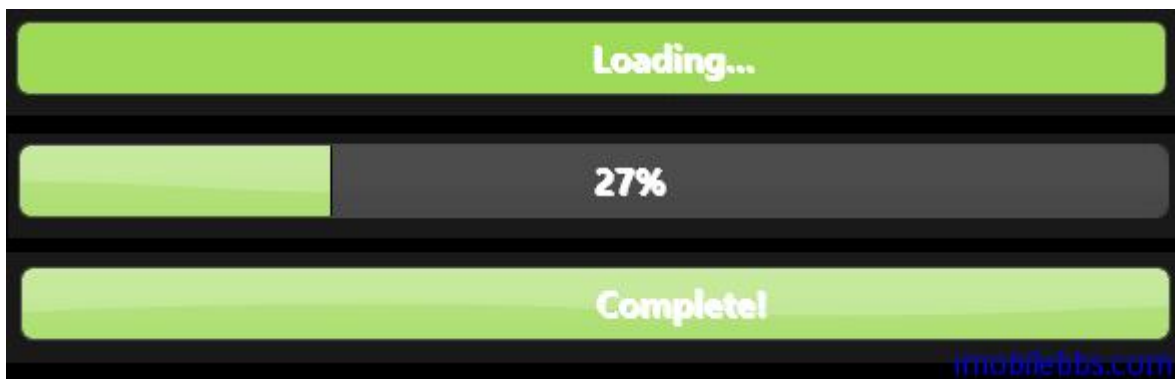
      progressbar.progressbar({
        value: false,
        change: function () {
          progressLabel.text(progressbar.progressbar("value") + "%");
        },
        complete: function () {
          progressLabel.text("Complete!");
        }
      });

      function progress() {
        var val = progressbar.progressbar("value") || 0;

        progressbar.progressbar("value", val + 1);

        if (val < 99) {
```

```
        setTimeout(progress, 100);  
    }  
}  
  
    setTimeout(progress, 3000);  
});  
</script>  
</head>  
<body>  
  
    <div id="progressbar">  
        <div class="progress-label">Loading...</div>  
    </div>  
  
</body>  
</html>
```



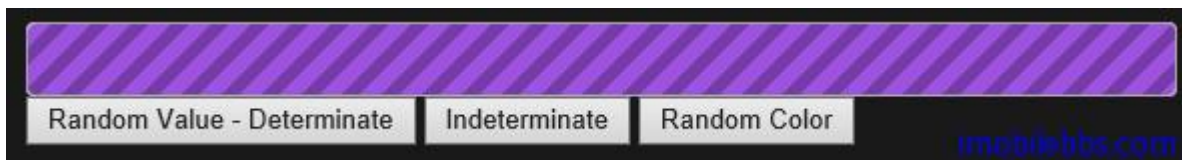
“中间过渡”状态条

可以通过设置 `value=false` 将进度条显示为“过渡”状态的进度条，此外也可以通过配置来修改进度条的颜色

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/base/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#progressbar").progressbar({
        value: false
      });
      $("button").on("click", function (event) {
        var target = $(event.target),
            progressbar = $("#progressbar"),
            progressBarValue
              = progressbar.find(".ui-progressbar-value");

        if (target.is("#numButton")) {
          progressbar.progressbar("option", {
            value: Math.floor(Math.random() * 100)
          });
        } else if (target.is("#colorButton")) {
          progressBarValue.css({
            "background": '#'
              + Math.floor(Math.random()
                * 16777215).toString(16)
          });
        } else if (target.is("#falseButton")) {
          progressbar.progressbar("option",
            "value", false);
        }
      });
    });
  </script>
  <style>
    #progressbar .ui-progressbar-value {
      background-color: #ccc;
    }
  </style>
</html>
```

```
    }  
    </style>  
</head>  
<body>  
  
    <div id="progressbar"></div>  
    <button id="numButton">Random Value - Determinate</button>  
    <button id="falseButton">Indeterminate</button>  
    <button id="colorButton">Random Color</button>  
  
</body>  
</html>
```





37



jQuery UI Slider 示例（一）



jQuery Slider 组件可以把选中的 HTML 元素变成滑动条 UI 控件，滑动条可以支持多个滑块用来设置单个值或一个值域。

基本用法

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#slider").slider();
    });
  </script>
</head>
<body>
  <div id="slider"></div>

</body>
</html>
```



多个滑块选择值域

Slider 支持使用两个滑块来选择一个值域，通过 `min, max` 指定大范围，`values` 指定当前选择的值域。

```
range: true
min: 0,
max: 500,
values: [ 75, 300 ],
```

完整代码如下：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#slider-range").slider({
        range: true,
        min: 0,
        max: 500,
        values: [75, 300],
        slide: function (event, ui) {
          $("#amount").val("$" + ui.values[0]
            + " - $" + ui.values[1]);
        }
      });
      $("#amount").val("$"
        + $("#slider-range").slider("values", 0) +
        " - $" + $("#slider-range").slider("values", 1));
    });
  </script>
</head>
<body>

  <p>
    <label for="amount">Price range:</label>
    <input type="text" id="amount"
      style="border: 0; color: #f6931f;
```

```
        font-weight: bold;" />  
</p>  
  
<div id="slider-range"></div>  
  
</body>  
</html>
```





38



jQuery UI Slider 示例（二）



前面的 slider 例子 Slider 都是水平顯示的，Slider 也可以顯示成垂直的，這可以通過配置orientation，將其值設為「vertical」。

基本用法

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#slider-vertical").slider({
        orientation: "vertical",
        range: "min",
        min: 0,
        max: 100,
        value: 60,
        slide: function (event, ui) {
          $("#amount").val(ui.value);
        }
      });
      $("#amount").val($("#slider-vertical")
        .slider("value"));
    });
  </script>
</head>
<body>
  <p>
    <label for="amount">Volume:</label>
    <input type="text" id="amount" style="border: 0;
      color: #f6931f; font-weight: bold;" />
  </p>

  <div id="slider-vertical" style="height: 200px;"></div>
</body>
</html>
```

Volume: 60



垂直選擇值域

同樣，Slider 垂直顯示時也可以使用兩個滑塊來選擇值域。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $("#slider-range").slider({
        orientation: "vertical",
        range: true,
        values: [17, 67],
        slide: function (event, ui) {
          $("#amount").val("$" + ui.values[0]
            + " - $" + ui.values[1]);
        }
      });
      $("#amount").val("$" + $("#slider-range")
        .slider("values", 0) +
        " - $" + $("#slider-range").slider("values", 1));
    });
  </script>
</head>
<body>

  <p>
    <label for="amount">Target sales goal (Millions):</label>
    <input type="text" id="amount" style="border: 0;
      color: #f6931f; font-weight: bold;" />
  </p>

  <div id="slider-range" style="height: 250px;"></div>

</body>
</html>
```

Target sales goal (Millions): **\$17 - \$67**



imobilebbs.com



39

jQuery UI Spinner 示例



Spinner 主要用来输入各种格式的数字，可以使用鼠标滚轮，键盘方向键来修改输入值，也支持直接键入数字。支持本地化的输入金额和时间。

基本用法

下面代码显示了 Spinner 的基本用法，设置和取得 Spinner 的当前值。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      var spinner = $("#spinner").spinner();

      $("#disable").click(function () {
        if (spinner.spinner("option", "disabled")) {
          spinner.spinner("enable");
        } else {
          spinner.spinner("disable");
        }
      });

      $("#destroy").click(function () {
        if (spinner.data("ui-spinner")) {
          spinner.spinner("destroy");
        } else {
          spinner.spinner();
        }
      });

      $("#getvalue").click(function () {
        alert(spinner.spinner("value"));
      });

      $("#setvalue").click(function () {
        spinner.spinner("value", 5);
      });

      $("#button").button();
    });
  </script>
</head>
<body>
```



```
<p>
  <label for="spinner">Select a value:</label>
  <input id="spinner" name="value" />
</p>

<p>
  <button id="disable">Toggle disable/enable</button>
  <button id="destroy">Toggle widget</button>
</p>

<p>
  <button id="getvalue">Get value</button>
  <button id="setvalue">Set value to 5</button>
</p>

</body>
</html>
```



显示地图

本例使用两个 Spinner，以步长为0.001 做为经纬度，然后和 Google 地图配合，通过 Spinner 修改地图的中心。

此外为了适应 Google Map API, 需要添加对其引用

代码如下：

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>jQuery UI Demos</title>
    <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
    <script src="scripts/jquery-1.9.1.js"></script>
    <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
    <script src="http://maps.google.com/maps/api/js?sensor=false"></script>
    <script>
        $(function () {
            function latlong() {
                return new window.google.maps.LatLng($("#lat").val(),
                    $("#lng").val());
            }
            function position() {
                map.setCenter(latlong());
            }
            $("#lat, #lng").spinner({
                step: .001,
                change: position,
                stop: position
            });

            var map = new window.google.maps.Map($("#map")[0], {
                zoom: 8,
                center: latlong(),
                mapTypeId: window.google.maps.MapTypeId.ROADMAP
            });
        });
    </script>
    <style>
    #map {
        width:500px;
```

```

    height:500px;
  }
</style>
</head>
<body>

<label for="lat">Latitude</label>
<input id="lat" name="lat" value="44.797" />
<br />
<label for="lng">Longitude</label>
<input id="lng" name="lng" value="-93.278" />

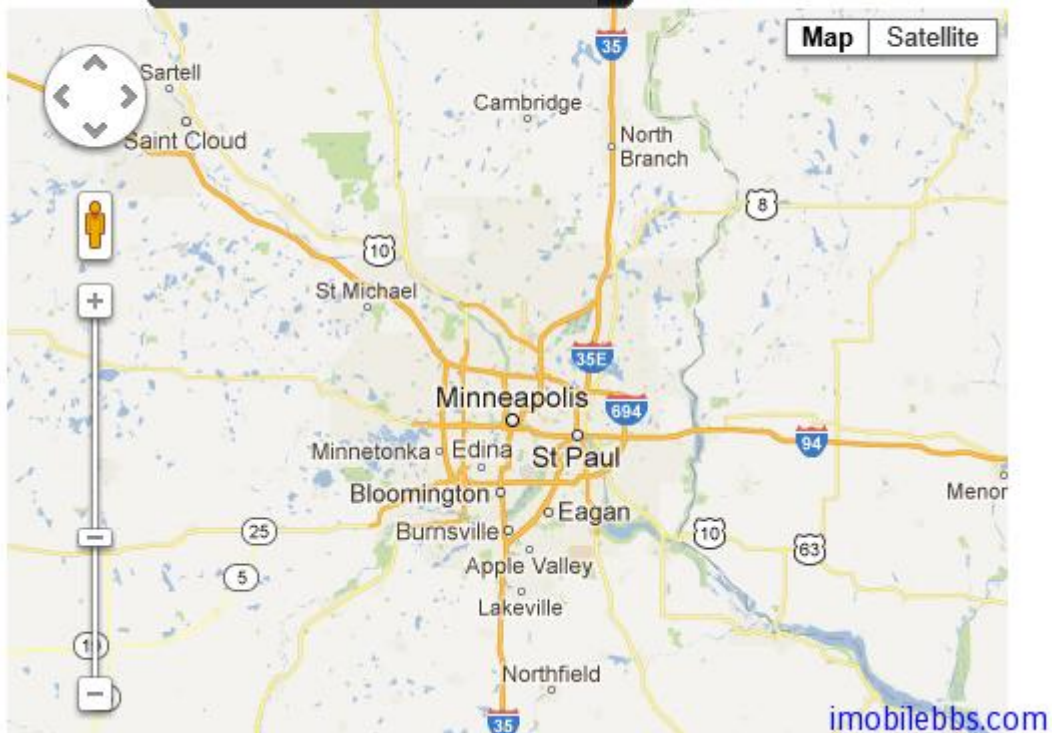
<div id="map"></div>

</body>
</html>

```

Latitude 44.807

Longitude -93.278





40

jQuery UI Tab 示例（一）



Tab 显示多个标签页，每个标签含有一个标签头和一个 Panel（显示标签的内容）。

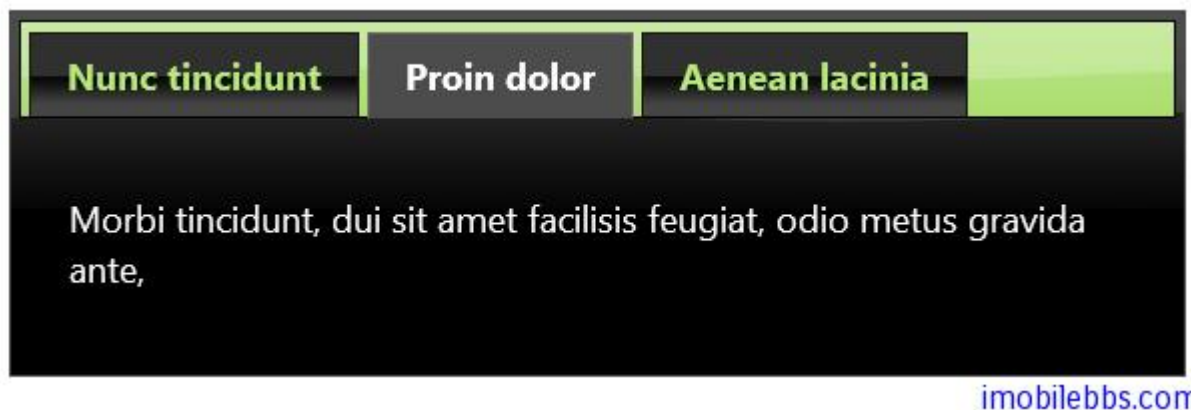
基本用法

首先使用 HTML 定义用来作为 Tab，一般使用列表 (ul ,li) 来定义标签页的标题，每个标题使用 href 链接到每个页面的内容页，比如下例定义了三个标签页：

```
<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Nunc tincidunt</a></li>
    <li><a href="#tabs-2">Proin dolor</a></li>
    <li><a href="#tabs-3">Aenean lacinia</a></li>
  </ul>
  <div id="tabs-1">
    <p>Proin elit arcu, rutrum commodo, vehicula tempus, </p>
  </div>
  <div id="tabs-2">
    <p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, </p>
  </div>
  <div id="tabs-3">
    <p>Mauris eleifend est et turpis. Duis id erat. </p>
    <p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, </p>
  </div>
</div>
```

然后使用 jQuery 的 tabs() 方法把这部分 Html 元素变为标签页

```
<script>
  $(function () {
    $("#tabs").tabs();
  });
</script>
```





41



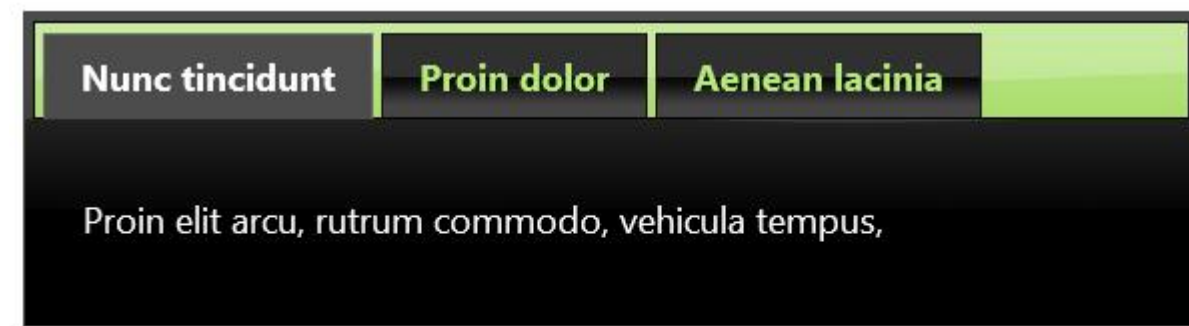
jQuery UI Tab 示例（二）



支持收缩和展开

缺省情况下，标签页是展开的，可以通过设置 `collapsible` 为 `true` 使得标签页支持收缩和展开。

```
<script>
  $(function () {
    $("#tabs").tabs({
      collapsible: true
    });
  });
</script>
```



Ajax 支持

jQuery 的标签页也支持通过 Ajax 方法来加载页面，这是通过设置 href 属性来实现，href 指向其它页面，这是 jQuery 通过 Ajax 方法读取指定页面，如果加载时间过长，则页面显示 Loading（加载中），如果无法加载，也可以通过配置 beforeLoad 设置错误信息。

```
<script>
$(function() {
  $( "#tabs" ).tabs({
    beforeLoad: function( event, ui ) {
      ui.jqXHR.error(function() {
        ui.panel.html(
          "Couldn't load this tab. We'll try to fix this as soon as possible. " +
          "If this wouldn't be a demo." );
      });
    }
  });
});
</script>
</head>
<body>

<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Preloaded</a></li>
    <li><a href="ajax/content1.html">Tab 1</a></li>
    <li><a href="ajax/content2.html">Tab 2</a></li>
    <li><a href="ajax/content3-slow.php">Tab 3 (slow)</a></li>
    <li><a href="ajax/content4-broken.php">Tab 4 (broken)</a></li>
  </ul>
  <div id="tabs-1">
    <p>Proin elit arcu, rutrum commod</p>
  </div>
</div>
```

鼠标移动自动激活页面

可以通过设置 `event:mouseover` 来实现，这样当鼠标移动到某个页面，该页面则自动展开。

```
<script>
$(function() {
    $( "#tabs" ).tabs({
        event: "mouseover"
    });
});
</script>
```



```

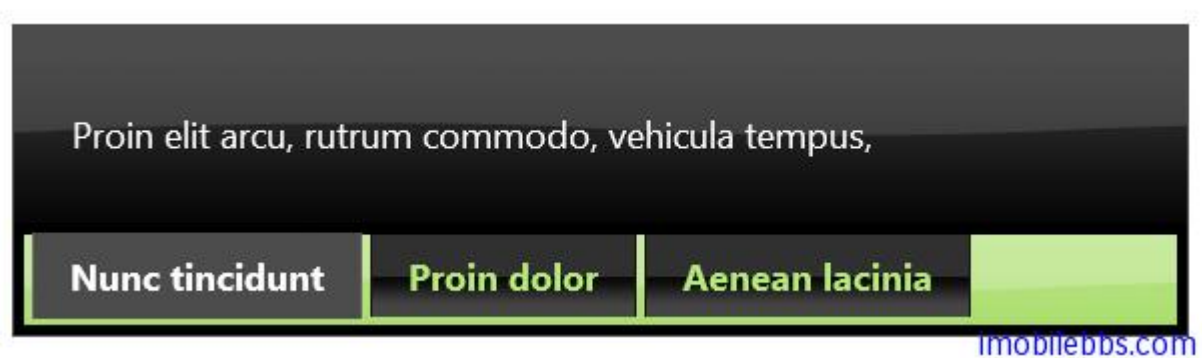
    }

    .tabs-bottom .ui-tabs-nav li.ui-tabs-active {
        margin-top: -1px;
        padding-top: 1px;
    }
</style>
</head>
<body>

<div id="tabs" class="tabs-bottom">
    <ul>
        <li><a href="#tabs-1">Nunc tincidunt</a></li>
        <li><a href="#tabs-2">Proin dolor</a></li>
        <li><a href="#tabs-3">Aenean lacinia</a></li>
    </ul>
    <div id="tabs-1">
        <p>Proin elit arcu, rutrum commodo, vehicula tempus, </p>
    </div>
    <div id="tabs-2">
        <p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, </p>
    </div>
    <div id="tabs-3">
        <p>Mauris eleifend est et turpis. Duis id erat. </p>
        <p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, </p>
    </div>
</div>

</body>
</html>

```


imobilebbs.com

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />

```

```

<title>jQuery UI Demos</title>
<link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
<script src="scripts/jquery-1.9.1.js"></script>
<script src="scripts/jquery-ui-1.10.1.custom.js"></script>
<script>
    $(function () {
        $("#tabs").tabs().addClass("ui-tabs-vertical ui-helper-clearfix");
        $("#tabs li").removeClass("ui-corner-top").addClass("ui-corner-left");
    });
</script>
<style>
    .ui-tabs-vertical {
        width: 55em;

        .ui-tabs-vertical .ui-tabs-nav {
            padding: .2em .1em .2em .2em;
            float: left;
            width: 12em;

            .ui-tabs-vertical .ui-tabs-nav li {
                clear: left;
                width: 100%;
                border-bottom-width: 1px !important;
                border-right-width: 0 !important;
                margin: 0 -1px .2em 0;
            }

            .ui-tabs-vertical .ui-tabs-nav li a {
                display: block;
            }

            .ui-tabs-vertical .ui-tabs-nav li.ui-tabs-active {
                padding-bottom: 0;
                padding-right: .1em;
                border-right-width: 1px;
                border-right-width: 1px;
            }

            .ui-tabs-vertical .ui-tabs-panel {
                padding: 1em;
                float: right;
                width: 40em;
            }
    }

```

```

</style>
</head>
<body>

<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Nunc tincidunt</a></li>
    <li><a href="#tabs-2">Proin dolor</a></li>
    <li><a href="#tabs-3">Aenean lacinia</a></li>
  </ul>
  <div id="tabs-1">
    <p>Proin elit arcu, rutrum commodo, vehicula tempus, </p>
  </div>
  <div id="tabs-2">
    <p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, </p>
  </div>
  <div id="tabs-3">
    <p>Mauris eleifend est et turpis. Duis id erat. </p>
    <p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, </p>
  </div>
</div>

</body>
</html>

```





42

jQuery UI Tooltip 示例



jQuery tooltip 的基本用法，可以把所有元素的 title 属性做为 Tooltip 显示，比如：

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Demos</title>
  <link rel="stylesheet" href="themes/trontastic/jquery-ui.css" />
  <script src="scripts/jquery-1.9.1.js"></script>
  <script src="scripts/jquery-ui-1.10.1.custom.js"></script>
  <script>
    $(function () {
      $(document).tooltip();
    });
  </script>
  <style>
  label {
    display: inline-block;
    width: 5em;
  }
  </style>
</head>
<body>

<p><a href="#" title="That's what this widget is">Tooltips</a> can be attached to any element. When you hover
the element with your mouse, the title attribute is displayed in a little box next to the element, just like a native
<p>But as it's not a native tooltip, it can be styled. Any themes built with
<a href="http://themeroller.com" title="ThemeRoller: jQuery UI's theme builder application">ThemeRoller</a>
will also style tooltips accordingly.</p>
<p>Tooltips are also useful for form elements, to show some additional information in the context of each field.</p>
<p><label for="age">Your age:</label><input id="age" title="We ask for your age only for statistical purposes." /></p>
<p>Hover the field to see the tooltip.</p>

</body>
</html>
```

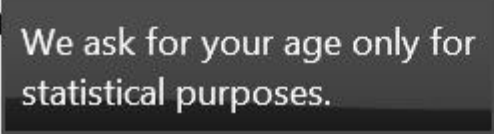

[Tooltips](#) can be attached to any element. When you hover the element with your mouse, the title attribute is displayed in a little box next to the element, just like a native tooltip.

But as it's not a native tooltip, it can be styled. Any themes built with [ThemeRoller](#) will also style tooltips accordingly.

Tooltips are also useful for form elements, to show some additional information in the context of each field.

Your age:

Hover the field



imobilebbs.com

此外，可以通过 CSS 修改缺省 Tooltip 的显示风格，或是自定义 Tooltip 显示内容（比如地图，图像等等），这里就不一一介绍了。

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台



更多信息请访问 

<http://wiki.jikexueyuan.com/project/jquery-tutorial/>