

Jquery Mobile 综述

快速上手

开始使用

jQuery Mobile 提供了一系列针对触摸操作友好的 ui 组件和基于 ajax 的导航系统来支持动画的页面转场。构建一个自己的 jQuery Mobile 页面是非常容易的：

基本的页面的模板

打开你最喜欢的文本编辑器，把下面的页面模板代码粘贴进去，保存然后用浏览器打开。你现在也是移动开发者了！

在 head 里，viewport 的 **meta 标签将屏幕的宽度设置为了与设备的宽度相同**，并且从 Cdn 引入了 jQuery, jQuery Mobile 和 jQuery Mobile 的主题样式表。jQuery Mobile 1.1 需要工作于 1.6.4 或 1.7.1 版本的 jQuery 框架之上。

在 body 之中，用 data-role="page" 来标记的一个 div 容器，用来定义一个 page，页面，data-role="header" 来标记的 div 容器用来定义页面的头部栏，data-role="content" 来标记的 div 容器用来定义页面的内容区域，这些用 HTML5 的 data- 属性标记的标签会

通过 jQuery Mobile 框架自动将基本的标记增强为部件。

HTML 代码:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
    <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
    <script
src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></scrip
t>
  </head>
  <body>
    <div data-role="page">
      <div data-role="header">
        <h1>My Title</h1>
      </div><!-- /header -->

      <div data-role="content">
        <p>Hello world</p>
      </div><!-- /content -->
    </div><!-- /page -->
  </body>
</html>
```

加入你自己的内容

在 **content** 容器中，你可以加入你自己的标准的 **HTML** 元素 **<h1>**,****,**<p>**等，你可以自己写样式来创建自定义的布局，但是注意要在 jQuery Mobile 的样式表之后加入自己的样式。

列表

jQuery Mobile 包含了一系列不同的列表，可以用 **data role="listview"** 标记的 **ul** 标记来定义。可以添加

`data-inset="true"`的属性来使列表看起来是内嵌的。还可以加入 `data-filter="true"`属性加入一个动态的搜索过滤条。

HTML 代码:

```
<ul data-role="listview" data-inset="true" data-filter="true">
    <li><a href="#">Acura</a></li>
    <li><a href="#">Audi</a></li>
    <li><a href="#">BMW</a></li>
    <li><a href="#">Cadillac</a></li>
    <li><a href="#">Ferrari</a></li>
</ul>
```

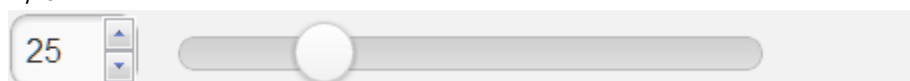


滑动条

jquery mobile 框架包含了一整套表单元素会自动将表单增强为易于触摸的样式。这个滑动条是用 html5 新的 `input` 类型“range”写的，不需要 `data-role` 属性。务必把他放到 `form` 标签内，并且与 `label` 元素相关联。

html 代码:

```
<form>
    <label for="slider-0">Input slider:</label>
    <input type="range" name="slider" id="slider-0" value="25" min="0" max="100" />
</form>
```

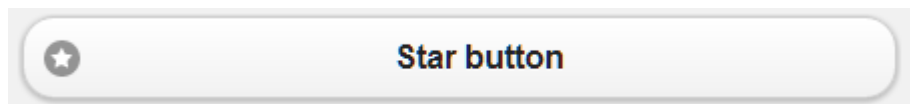


按钮

有几种办法可以做按钮，先让 **a** 标签成为按钮，只需要在 **a** 标签添加 **data-role="button"** 属性就可以了。通过 **data-icon** 属性可以给按钮添加图标，并且通过 **data-iconpos** 属性控制图标的位置。

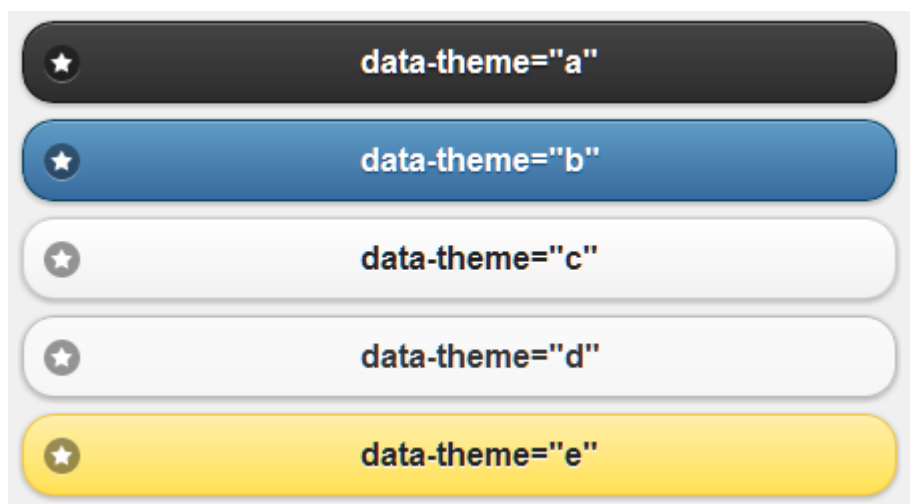
html 代码:

```
<a href="#" data-role="button" data-icon="star">Star  
button</a>
```



玩转主题样式

jQuery Mobile 拥有一套健壮的主题样式系统支持最多 26 套（因为英文字母只有 26 个）工具栏，内容和按钮的主题，叫做 "swatch".（斯沃琪手表...）.只需要给页面的任意组件添加 **data-theme="e"** 的属性，就可以将之设计为黄色的主题样式。



如果你想构建自己的主题样式，请使用 [**ThemeRoller\(主题编辑器\)**](#)吧。

再构建点什么

这仅仅是你用 jQuery Mobile 轻松构建 WEBAPP 的开始。想要学习更多，请访问页面的链接，给页面添加转场和创建对话框。使用 data-attribute 属性尝试其他的 data- attributes。

关键特性

- 1 构建于 **jQuery** 的核心之上。使之兼容于 jq 的语法，对于开发人员有最易的开发曲线
- 2 兼容于所有的主流移动手机，平板电脑，电子书,和 **pc**，：iOS, Android, Blackberry, Palm WebOS, Nokia/Symbian, Windows Phone 7, MeeGo, Opera Mobile/Mini, Firefox Mobile, Kindle, Nook, 和所有的现代浏览器。
- 3 轻量级 压缩后只 12k，对图片的依赖程度非常低，保证了速度
- 4 模块化的架构可以根据你的独特的需求用来构建最优化的应用
- 5 页面和行为均基于 **html5** 标记的驱动进行配，开发效率高，对脚本的需求小
- 5 渐进增强使所有的移动设备，平板电脑和 **pc** 电脑都支持核心的内容和方法。而对于新的移动平台，则可以展现像安装在设备中的应用程序一样出色的富媒体和交互的浏览体验
- 6 弹性的设计可以使同样的代码在智能设备上和桌面的屏幕上都自动缩放适应。
- 7 强大的 **ajax** 驱动的导航系统在保持后退按钮，收藏夹和干净的地址栏的同时完成页面转场。
- 8 优秀的可访问性 一些特性比如 **WAI-ARIA** 也包含在内，以确保页面也可以在一些屏幕阅读器(比如苹果的 VoiceOver)或者其他手持设备中正常工作。
- 9 支持触摸和鼠标事件增加了触摸屏设备支持的触摸，鼠标，和基于光标的输入方法的 API

10 统一的 UI 组件增加了触摸屏设备支持的触摸, 鼠标, 和基于光标的输入方法的 API

11 强大的主题样式框和主题编辑器能很容易的进行高度个性化和品牌化的的界面定制

可访问性

Jquery Mobile 是基于标准的, 系统化的 html 构建的, 使得页面能够在最广泛范围的设备上得到支持。对于 A 级的浏览器, 许多 Jquery Mobile 组件, 比"焦点管理", "键盘导航"等都能支持, 其他可以详细参加 W3C 的 WAI-ARIA 说明。

通过运用这些技术, 我致力于使得通过 Jquery Mobile 开发的 web 产品拥有最好的可访问性, 对于伤残人士, 例如盲人, 也可以用读屏软件, 例如 iphone 的 voiceover, 使用。

我们现在正在改进可访问性, 我们的目标是在 1.0 版本的时候, 使 Jquery mobile 的所有控件都拥有全部的可访问性。

支持的平台

到了正式版, jQuery Mobile 已经覆盖了所有的目标平台。即, 我们已经支持绝大多数的常用现代桌面电脑, 智能设备, 平板电脑和电子书平台, 并且一些怪异的手机和古老的浏览器也都支持, 因为我们渐进增强的尝试。我们为我们的努力而达到的最大化的可用性而感到自豪。

我们根据目标在过去的一年间制作了一个浏览器分 3 级支持的表格, 并且根据实机测试改进了这个表。

视觉表现的好坏主要依赖于设备的浏览器对 css 的渲染能力, 所以不是所有的 A 级的浏览器都能都达到像素级的完美但是表现绝对是正常的。

A-grade

支持全部的增强的体验, 包括基于 ajax 的动画的页面转场

- Apple iOS 3.2-5.0 - Tested on the original iPad (4.3 / 5.0), iPad 2 (4.3), original iPhone (3.1), iPhone 3 (3.2), 3GS (4.3), and 4 (4.3 / 5.0)
- Android 2.1-2.3 - Tested on the HTC Incredible (2.2), original Droid (2.2), Nook Color (2.2), HTC Aria (2.1), Google Nexus S (2.3). Functional on 1.5 & 1.6 but performance may be sluggish, tested on Google G1 (1.5)
- Android Honeycomb- Tested on the Samsung Galaxy Tab 10.1
- Windows Phone 7 - Tested on the HTC 7 Surround
- Blackberry 6.0 - Tested on the Torch 9800 and Style 9670
- Blackberry 7 - Tested on BlackBerry® Torch 9810
- Blackberry Playbook - Tested on PlayBook version 1.0.1 / 1.0.5
- Palm WebOS (1.4-2.0) - Tested on the Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0)
- Palm WebOS 3.0 - Tested on HP TouchPad

- Firebox Mobile (Beta) - Tested on Android 2.2
- Opera Mobile 11.0: Tested on the iPhone 3GS and 4 (5.0/6.0), Android 2.2 (5.0/6.0), Windows Mobile 6.5 (5.0)
- Meego 1.2 NEW - Tested on Nokia 950
- Chrome Desktop 11-13 - Tested on OS X 10.6.7 and Windows 7
- Firefox Desktop 3.6-4.0 - Tested on OS X 10.6.7 and Windows 7
- Internet Explorer 7-9 - Tested on Windows XP, Vista and 7 (minor CSS issues)
- Opera Desktop 10-11 - Tested on OS X 10.6.7 and Windows 7

B-grade

增强的体验,但是不支持 ajax 的导航

- Blackberry 5.0: Tested on the Storm 2 9550, Bold 9770
- Opera Mini (5.0-6.0) - Tested on iOS 3.2/4.3
- Nokia Symbian^3 - Tested on Nokia N8 (Symbian^3), C7 (Symbian^3), also works on N97 (Symbian^1)

C-grade

没有增强的体验，但是依然可用

- Blackberry 4.x - Tested on the Curve 8330
- Windows Mobile - Tested on the HTC Leo (WinMo 5.2)
- 所有古老的智能平台 - 所有不支持 css 的 media queries 的设备，的表现都为 C 级

不正式支持

可能支持，但是没有经过彻底的测试或修 BUG

- Samsung Bada - 本项目目前还没有测试或模拟，但是现在看起来支持的还不错。

组件

Jquery Mobile 支持包含历史回退按钮的自动 ajax 读取外部页面的功能，一组页面转场动画，以及将页面显示为对话框的简易工具。

页面结构

Jquery Mobile 中的"页面"结构已经优化为支持单个的页面，或者在一个页面内嵌入的多个“**page**”。

使用这种模型的目的是使 **WEB** 开发人员使用最佳实践创建 **WEB** 网站，使得普通的链接不需要任何特殊配置就可以工作，而且能创造出富媒体的像本地应用一样通过标准的 **http** 无法创造出的用户体验。

移动网站页面结构

Jquery Mobile 网站必须使用 **HTML5** 文档声明开始，使得网站能够适用 **HTML5** 的特性（不支持 **HTML5** 的旧浏览器会安全地静默忽略 **HTML5** 的文档声明和一些自定义属性）。在 **head** 标签内需要引用 **Jquery**, **Jquery Mobile** 和 **mobile** 的主题 **CSS** 文件，我们推荐链接到 **jQuery** 的 **CDN** 上：

HTML 代码:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.css" />
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.5.min.js"></script>
  <script type="text/javascript"
src="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.js"></script>
</head>
<body>
...
</body>
</html>
```

Viewport 的 meta 标签

注意在内容之前在 HEAD 标签内有一个 viewport 的 meta 标签设定了浏览器怎样展示页面的缩放等级和范围。如果没有设置的话，许多移动设备的浏览器会使用一个“虚拟的”页面宽度，大约是 900PX 宽，使得桌面的网站工作的也很好，但是这样屏幕看起来是缩小的而且太宽了。通过设置 **viewport** 属性为 **content="width=device-width, initial-scale=1**，会使宽度会被设定为设备的屏幕宽度。

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

这个设置没有禁用用户缩放页面的权限，使得可访问性非常好。但是在 iOS 设备上有一个小问题，就是旋转屏幕的时候，这个 viewport 的设定没有设置宽度，在未来的版本我们希望能修正。如果有需求的话，你可以设定 viewport 的其他的属性来禁用缩放。

在 body 内：pages

在 <body> 标签内，移动网站的每一个视图或者“page”都要通过 **data-role="page"** 属性的标签来定义。

```
<div data-role="page">
    ...
</div>
```

在 page 容器内，任何有效的 html 标签都可以使用，但是对于 JQuery Mobile 的典型页面来说，**page** 容器的直接子结点应该为使

用**"data-role"**标记属性为**"header"**、**"content"**和**"footer"**的 3 个容器。

```
<div data-role="page">
  <div data-role="header">...</div>
  <div data-role="content">...</div>
  <div data-role="footer">...</div>
</div>
```

一个完整的独立页面实例

集合在一起后，这就是一个标准的 Jquery Mobile 页面的样板了。

html 代码：

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.4.3.min.js"></script>
    <script type="text/javascript"
src="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.js"></script>
  </head>
  <body>
    <div data-role="page">
      <div data-role="header">
        <h1>Page Title</h1>
      </div><!-- /header -->
      <div data-role="content">
        <p>Page content goes here.</p>
      </div><!-- /content -->
      <div data-role="footer">
        <h4>Page Footer</h4>
      </div><!-- /footer -->
    </div><!-- /page -->
  </body>
</html>
```

外部页面的链接

Jquery Mobile 会自动构建 AJAX 驱动的站点和应用。

默认情况下，当你点击了一个指向外部页面（比如：`products.html`）的链接，JQmobile 的分析链接地址，然后产生一个 ajax 请求（Hajax），显示一个读取中的提示框。

如果 AJAX 请求成功，新页面的内容会添加到 DOM 中，所有 MOBILE 组件都会自动初始化，所以新的页面会通过页面转场动画看到。

如果 AJAX 请求失败，JQmobile 会显示一个小错误提示框（默认的主题为 E），然后过一会就消失了，不会影响你继续浏览。

多个页面的模板结构

一个独立的 html 文件可以包含多个“pages”，在加载的时候这些 `data-role="page"` 的 div 会一起堆栈加载。**每一个 page 区块需要由一个独特的 ID(`id="foo"`)标记**，通过给内部链接地址定义为 (`href="#foo"`)，互相跳转。当连接被点击时，JQmobile 会在文档内寻找带有 ID 的 page 容器，然后通过页面转场显示。

这是一个包含 2 个“page”的站点的示例，通过每个页面的 ID 进行连接。注意每个 page 容器的 ID 属性只是用来做内部连接导航的，如果每一个 PAGE 都是一个独立的 HTML 文件的话，那这个 ID 属性就可有可无了。

<body>

```

<!-- Start of first page -->
<div data-role="page" id="foo">
  <div data-role="header">
    <h1>Foo</h1>
  </div><!-- /header -->
  <div data-role="content">
    <p>I'm first in the source order so I'm shown as the page.</p>
    <p>View internal page called <a href="#bar">bar</a></p>
  </div><!-- /content -->
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div><!-- /footer -->
</div><!-- /page -->
<!-- Start of second page -->
<div data-role="page" id="bar">
  <div data-role="header">
    <h1>Bar</h1>
  </div><!-- /header -->
  <div data-role="content">
    <p>I'm first in the source order so I'm shown as the page.</p>
    <p><a href="#foo">Back to foo</a></p>
  </div><!-- /content -->
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div><!-- /footer -->
</div><!-- /page -->
</body>

```

请注意：我们对于所有 AJAX 的"page"使用的是哈希值来跟踪导航历史，所以现在还不可以通过外部页面的一个 ID 指向那一个锚点，因为 JQuery Mobile 会寻找带有该 ID 的"page",而不是滚屏到带有该 ID 的内容上。

一些约定，不是要求

尽管上述的页面结构是通过 jQuery Mobile 构建移动网站的标准格式，但是 jqm 框架是一个灵活的框架。page, header, content, and footer 和 data-role 元素都是可选的。 page 容器用来进行页

面初始化，但是现在的版本对于单独的 **page** 文档也不是必须的。对于一个自定义布局的页面，所有的这些结构元素都可以没有，但是基于 **AJAX** 的导航和组件都必须在标准的结构中才能工作。如果页面没有 **page** 容器，**jqm** 会自动插入一个，因为需要这样一个容器来管理页面，但是基本的标记现在可以写的非常简单了。

注意：在多 **PAGE** 的结构中，你在页面必须要有多个 **page** 标签的容器。

后退链接

如果你添加了 **data-rel="back"** 属性给某个链接，那对于该链接的任何点击行为，都是后退的行为，会无视链接的 **herf**，后退到浏览器历史的上一个地址。当想链接到一个命名好的页面，或者要关闭对话框时特别有用。使用这一特性时要注意：务必请使用一个有意义的 **herf** 值实际地指向要引用的地址（因为这样也会让 **GRADE-C** 的浏览器也支持这个按钮）。而且如果你只是要看到一个翻转的页面转场而不是真正的回到上一个历史记录地址，你就要使用 **data-direction="reverse"** 属性，而不是后退链接。

重定向以及链接到目录

当连接到的是一个目录地址（比如 **href="typesofcats/"** 而不是 **href="typesofcats/index.html"**），你必须在地址后面加一个斜杠（**"/"**）。因为 **Jquery Mobile** 假定最后一个斜杠后的部分为文件名，

当接下来引用到的页面生成基地址的会的时候会删除它。

但是，你可以通过已经指定好的 **data-url** 地址来返回到上一个 **page** 容器，就可以绕过这一个问题。**Jquery Mobile** 会使用那个 **data-url** 值来更新 **url**，来替代过去请求的那个页面。这样你也允许你通过返回 **url** 的更改来作为重定向的结果。例如：你想通过 **post** 方式提交表单到 **/login.html**，但是提交成功以后返回一个 **url** 为 **/account**。这个工具就可以让你在这种情况下控制历史记录堆栈。以下是一个实例：

有一个链接是指向 **"docs-links-urltest/index.html"**，该链接是指向目录的索引页，返回的页面会使用 **"docs/pages/docs-links-urltest/"** 更新哈希值，这是通过代码里的 **data-url** 值完成的。请注意 **data-url** 值会替换整个哈希值，由你来把他替换成当刷新或者下层链接时 **URL** 发出的请求能解析正确的页面。

更多技术细节请参见导航模型 **Ajax, hashes and history**

Ajax 的导航的标题

当你的 **jQuery Mobile** 的网站第一个页面加载后，然后点击一个连接或者提交一个表单后，**jQuery Mobile** 会使用 **ajax** 转到请求的页面的内容。在 **DOM** 中拥有 2 个或多个 **pages**，会触发动画的页面转场，但是这种处理方法的一个弊端就是页面的标题始终为了修复这个，**jQuery Mobile** 会自动通过 **ajax** 解析页面的 **title** 属性

`data-title` 属性。

多个 page 的标题

在多个 `page` 的文档中，我们还是遵守相似的约定，但是因为所有的 `page` 都共用一个标题，我们给每个 `page` 容器添加了一个 `data-title` 属性，这样在多 `page` 页面中，就可以手动的定义标题。

HTML 文档的标题会自动设置为现在活动页面的 `data-title` 属性值。

```
<div data-role="page" id="foo" data-title="Page Foo">
</div><!-- /page -->
```

链接的格式

Jquery Mobile 支持所有标准的 html 格式，为了让体验更加流畅,Jquery Mobile 会把同一个域下的所有指向页面的链接都会自动转为 `ajax` 请求，并且用动画的转场效果实现切换

指向别的域的链接或者是含有 `rel="external"`, `data-ajax="false"` 属性的链接以及含有 `target` 属性的链接不会通过 `ajax` 加载，页面会整体被刷新。如果一个 html 文档内含有多个 `pages` 容器，可以通过 `pages` 容器设置 `id` 属性，然后将链接指向 `id`。

所有其他类型的链接别如：`mailto:`以及 `tel:` 不会被 Jquery Mobile 影响到。更多细节请参见链接导航模型

页面转场

Jquery Mobile 框架内置了一套基于 `css` 的页面转场效果，你可

以给任何通过 ajax 对象或页面添加。

给链接添加 **data-transition** 属性，可以设定自定义的页面转场效果

```
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="fade" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="pop" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="flip" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="turn" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="flow" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="slidefade" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="slide" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="slideup" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="slidedown" data-inline="true">dialog</a>
<a href="page-transitions-dialog.html" data-role="button"
data-rel="dialog" data-transition="none" data-inline="true">dialog</a>
```

只看到了淡入淡出的效果？ 翻转效果在大多数的安卓系统上表现的都不太好，因为安卓的系统没有 **css3D** 渲染能力。不幸的是，他不会忽略这个这个翻转的效果，而是使页面乱转。所以我们建议少用这个效果。

给链接或者表单的提交添加转场效果

默认情况下，Jquery Mobile 应用的是从淡入淡出的转场效果。如果想要自定义转场特效，需要给链接添加“**data-transition**”属性

```
<a href="index.html" data-transition="pop">I'll pop</a>
```

当后退按钮按下时，框架会自动使用“后退”转场效果退到上一个页面，如果想强制给转场应用“后退”的转场效果，可以链接添加 **data-direction="reverse"**

页面转场的全局设置

如果你想给全局设置一个不一样的默认转场效果，可以设置 **defaultPageTransition**。如果想把对话框的 popup 默认效果设置为一个不一样的话，则需要设置 **defaultDialogTransition**。

浏览器的支持和表现

jquery mobile 中所有的动画都是通过 css 的关键帧动画技术实现的，并且包含了 **-webkit** 的前缀来针对 iOS, Blackberry, Android, Safari 和 Chrome 浏览器，以及 **-moz** 前缀来针对火狐浏览器，是否支持 css 的关键帧动画技术取决于浏览器的版本和 移动设备的硬件，如果不支持的话，转场会安全地退化为无动画的效果，

为不支持 3D 转场的情况定义平稳退化的转场效果

默认情况下，除了淡入淡出外所有转场都需要 3D 支持。如果设备不支持的话，转场会安全地退化为淡入淡出效果。我们这样做是想主动地排除像 Android 2.x 那样表现不佳的平台，并确保他们仍然有一个流畅的体验。但是 Android 3.0 支持 3D 变换，但仍然表现不佳（闪屏？），所以我们不能保证每个浏览器都表现良好。

所以可以给每种转场效果都设置一个备选的转场，以备浏览器

不支持的情况，默认情况下指定的为“fade”，例如，下面的代码可以将 **slideout** 的转场设置为“无”

```
$.mobile.transitionFallbacks.slideout = "none"
```

设置转场的最大宽度

默认情况下，当窗口宽度大于一定值的时候，转场效果可能会被关闭。因为转场效果会消耗系统资源，所以屏幕太宽的时候表现可能会不佳。通过 **\$.mobile.maxTransitionWidth** 可以设置这个宽度，默认情况下值为 **false**。它也接受一个数字，代表能够发生转场效果的最大宽度。

创建自定义的基于 css 的转场效果

[jQuery Mobile 允许自定义页面转场，具体请查看](#)

创建自定义的基于 css 的转场效果

如果想要创建一个自定义的 **css** 转场效果，选择一个与转场的 **DOM** 对应的 **class** 选择符，比如：“**slide**”，然后使用 **csstransitions** 和关键帧动画定义自己的“**in**”和“**out**”的 **css** 规则。

css 代码

```
.slide.in {
  -webkit-transform: translateX(0);
  -moz-transform: translateX(0);
  -webkit-animation-name: slideinfromright;
  -moz-animation-name: slideinfromright;
}

.slide.out {
  -webkit-transform: translateX(-100%);
```

```

    -moz-transform: translateX(-100%);
    -webkit-animation-name: slideouttoleft;
    -moz-animation-name: slideouttoleft;
}
@-webkit-keyframes slideinfromright {
    from { -webkit-transform: translateX(100%); }
    to { -webkit-transform: translateX(0); }
}

@-webkit-keyframes slideouttoleft {
    from { -webkit-transform: translateX(0); }
    to { -webkit-transform: translateX(-100%); }
}

@-moz-keyframes slideinfromright {
    from { -moz-transform: translateX(100%); }
    to { -moz-transform: translateX(0); }
}

@-moz-keyframes slideouttoleft {
    from { -moz-transform: translateX(0); }
    to { -moz-transform: translateX(-100%); }
}

```

在页面转场过程中，jQuery Mobile 会将转场的 class 名放入转场的“来自”和“到达”的两个页面中。然后将"out"class 放入“来自”页面中，"out"class 放入“到达”页面中。

如果你的页面转场支持后退，你需要使用.reverse 类，加到"in"和"out"类中

css 代码

```

.slide.in.reverse {
    -webkit-transform: translateX(0);
    -moz-transform: translateX(0);
    -webkit-animation-name: slideinfromleft;
    -moz-animation-name: slideinfromleft;
}
.slide.out.reverse {
    -webkit-transform: translateX(100%);
    -moz-transform: translateX(100%);
    -webkit-animation-name: slideouttoright;
}

```

```

    -moz-animation-name: slideouttoright;
}
@-webkit-keyframes slideinfromleft {
    from { -webkit-transform: translateX(-100%); }
    to { -webkit-transform: translateX(0); }
}
@-webkit-keyframes slideouttoright {
    from { -webkit-transform: translateX(0); }
    to { -webkit-transform: translateX(100%); }
}

@-moz-keyframes slideinfromleft {
    from { -moz-transform: translateX(-100%); }
    to { -moz-transform: translateX(0); }
}
@-moz-keyframes slideouttoright {
    from { -moz-transform: translateX(0); }
    to { -moz-transform: translateX(100%); }
}

```

写好 **css** 规则之后，只需要给要导航的链接的 **data-transition** 属性指定为自己的 **css** 名字就可以了。

html 代码

```
<a href="#page2" data-transition="slide">Page 2</a>
```

当用户点击此链接时，jQuery Mobile 会调用你自己写 **css** 转场。上面的 **css** 规则中都没有定义 **css animation** 的缓动或缓冲，是因为 jQuery Mobile 的 **css** 中默认定义了 **.in**, **.out** 的 **css** 缓动

当用户点击此链接时，jQuery Mobile 会调用你自己写 **css** 转场。上面的 **css** 规则中都没有定义 **css animation** 的缓动或缓冲，是因为 jQuery Mobile 的 **css** 中默认定义了 **.in**, **.out** 的 **css** 缓动

css 代码

```

.in {
    -webkit-animation-timing-function: ease-out;
    -webkit-animation-duration: 350ms;
}

```

```
-moz-animation-timing-function: ease-out;
-moz-animation-duration: 350ms;
}
.out {
-webkit-animation-timing-function: ease-in;
-webkit-animation-duration: 225ms;
-moz-animation-timing-function: ease-in;
-moz-animation-duration: 225;
}
```

对话框

创建对话框

通过给指向页面的链接增加 **data-rel="dialog"** 的属性，可以把任何指向的页面表现对对话框。当应用了对话框的属性之后，**qjmobile** 框架会给新页面增加圆角，页面周围增加 **margin**，以及深色的背景，使得对话框浮在页面之上

HTML 代码:

```
<a href="foo.html" data-rel="dialog">Open dialog</a>
```

对话框的转场

因为 **Jquery Mobile** 里对话框也是一个标准的"page",所以他会以默认的 **slide** 转场效果打开。而且像其他的页面一样，你也可以通过给链接添加 **data-transition** 的属性指定对话框的转场效果。为了让对话框看起来效果更好，我们推荐"pop", "slideup", "flip" 三种转场效果。

HTML 代码:

```
<a href="foo.html" data-rel="dialog" data-transition="pop">Open dialog</a>
```

关闭对话框

对话框内的任何链接被点击时，**Jquery Mobile** 会自动关闭对话框，用回退效果转场到指向该对话框的页面。要在对话框内创建一个取消"按钮，只需要把该链接的 **href** 指向打开该对话框的按钮然后给链接加上 **data-rel="back"** 属性。这样制作的后退链接在不支持 **js** 的设备上也同样管用。

对于 **js** 生成的链接，可以把链接的 **href** 设为 **"#"**，添加 **data-rel="back"** 属性。也可以手动地调用对话框的 **close()** 方法关闭对话框，比如？`$('.ui-dialog').dialog('close');`

设置关闭按钮的文字

就像页面插件一样，你可以设置程序或属性两种方法设置对话框关闭按钮的文字。程序的设置可以通过更改 **mobileinit** 绑定的事件，然后？`$.mobile.dialog.prototype.options.closeBtnText` 设置一个字符串。或者给标记代码增加 **data-close-btn-text** 属性来设置文字

历史和后退按钮的行为

因为对话框是典型地用来给页面起支持作用的，所以 **Jquery Mobile** 不会在历史记录的哈希值里包含对话框。这就意味着当点击对话框后退按钮时，他不会出现在你的浏览器历史记录里。比如说，

如果你在一个页面上，点击了一个按钮打开了一个对话框，然后关闭了这个对话框，到了另一个页面。这时如果你点击浏览器的后退按钮，那你会回到第一个页面，而不是对话框

工具栏

工具栏基础

工具栏在此是指在移动网站和应用中的头部，尾部和内容中的工具条。所以 **Jquery Mobile** 提供了一套标准的工具和导航栏的工具，可以在绝大多数情况下直接使用

工具栏综述

在 **Jquery Mobile** 中，有两种标准的工具栏：**头部栏**和**尾部栏**

头部栏的作用为网站的标题，通常是移动网站页面的第一个元素，一般包括页面的标题文字和最多两个按钮

尾部栏通常是移动网站页面的最后一个元素，在内容和作用上比头部栏更自由一些，但一般也要包含文字和按钮

在头部栏或尾部栏里放置一个水平的导航栏或选项卡栏的做法是很普遍的，所以 **Jquery Mobile** 包含导航栏组件，即把无序列表 **ul** 转化成水平的按钮栏，使用也非常方便

工具栏定位的设置

在页面中设置头部栏和尾部栏的位置定位有几种方法。默认情

况下，工具栏的定位的属性为**"inline"**.在这种模式下，头部栏和尾部栏通过 **html** 自动的文档流放置，保证了他们能在所有的设备上可见，而不需要依靠 **css** 和 **js** 的定位的支持

固定的定位模式可以使工具条在页面处于固定的位置，而不需要通过 **js** 设置。工具条处于他们在页面自然的位置上，就像 **inline** 模式一样，但是当他被滚动出屏幕之外时，**Jquery Mobile** 会自动通过动画使滚动条重新出现在屏幕的顶部或底部

任何时候，点击屏幕会切换固定定位模式的工具条的显示：当工具条消失时点击屏幕会让他出现，再点击则会让它消失。这样用户就有选择在最大化浏览时要不要隐藏工具栏， 要给工具栏设置固定的定位模式，只需给工具栏的容器加**"data-position="fixed"**的属性即可

全屏的定位模式与固定的定位模式基本相同，但是当他被滚动出屏幕之外时，不会自动重新显示，除非点击屏幕。这对于图片或视频类有提升代入感的应用是非常有用的，当浏览时你想全屏都显示内容，而工具栏可以通过点击屏幕呼出。注意这种模式下工具栏会遮住页面内容，所以最好用在比较特殊的场合下

头部栏

头部栏结构

头部栏是处于页面顶部的工具栏，通常包含页面标题文字，文

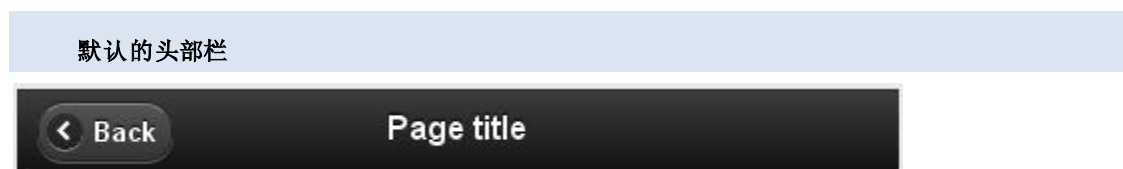
字左边和/或右边可以放置几个可选的按钮用作导航操作

标题文字一般用 **h1** 标签，但是从 **h1-h6** 都是可以的，这样可以使结构有弹性。比如说，一个页面内包含了多个"page"标记的页面，这样可以给主"page"的标题文字用 **h1** 标签，次级"page"的标题文字用 **h2** 标签。所有的头部默认下在样式上都是相同的，保持的外观的一致性

```
<div data-role="header">
  <h1>Page Title</h1>
</div>
```

默认的头部的特性

头部栏的主题样式默认情况下为"**a**"(黑色)，但是你可以很轻松的设置主题样式



看见 **back** 按钮了吗 JQuery Mobile 会自动给每个页面的头部栏生成一"back"按钮，来简化把按钮放进头部栏的过程。如果你不想"back"按钮放进头部栏，你可以自己加一个需要的按钮或者给头部栏添加"**data-backbtn="false"**"属性

添加按钮

在标准的头部栏的设置下，标题文字两边各有一个可放置按钮的位置。每一个按钮通常都是都是 **a**,但是任何有效的按钮标记的元

素都可以。为了节省空间，工具栏里的按钮都是内联按钮，所以按钮的宽度只容纳 icon 和里面的文字

创建自定义的后退按钮

给 **a** 标签增加 **data-rel="back"** 的属性，任何链接都可以样式化为后退按钮，行为上为后退到上一个历史记录的页面，无视该链接的 **href** 地址。这对于要链接回到一个命名好的页面的情况是很有用的但是注意请给链接设定一个有意义的 **href** 地址，指向实际要连接的页面，使得 **C** 级浏览器也能够使用这个按钮而且要注意如果你只是需要一个后退的页面转场效果，而不是真正的后退到上一个历史记录的页面，应该用 **data-direction="reverse"** 的属性，而不是用 **data-rel="back"** 属性的按钮

按钮默认的定位

头部的按钮是头部栏容器的直接子节点，第一个链接定位于头部栏左边，第二个链接放在右边，在这个例子中，根据两个链接在源代码中的位置，取消在左边，保存在右边

```
<div data-role="header" data-position="inline">
  <a href="index.html" data-icon="delete">Cancel</a>
  <h1>Edit Contact</h1>
  <a href="index.html" data-icon="check">Save</a>
</div>
```

按钮默认的定位



按钮会自动应用他们的父容器的主题样式，所以应用了**"a"**主题

样式的头部栏里的按钮也会应用"a"主题样式，我们通过给按钮增加 **data-theme** 的属性并设置，可以使按钮看起来有所区别

```
<div data-role="header" data-position="inline">
  <a href="index.html" data-icon="delete">Cancel</a>
  <h1>Edit Contact</h1>
  <a href="index.html" data-icon="check" data-theme="b">Save</a>
</div>
```

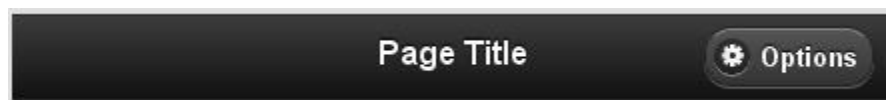


通过 **class** 控制按钮的定位

按钮的位置可以通过 **class** 设置，而不依赖他们在源代码中的顺序。如果你想把唯一一个按钮放在右边，这时就非常有用。两个控制的类为 **ui-btn-right** 和 **ui-btn-left**

在这个例子中，我们要把头部栏唯一一个按钮放于右边，首先给头部栏增加 **data-backbtn="false"** 属性来阻止头部栏自动生成后退按钮的行为，然后给自己的按钮增加 **ui-btn-right** 的 **class**

```
<div data-role="header" data-position="inline" data-backbtn="false">
<h1>Page Title</h1>
<a href="index.html" data-icon="gear" class="ui-btn-right">Options</a>
</div>
```



自定义头退按钮的文字

如果你想设置后退按钮的文字，需要给页面 **?page** 元素增加 **data-back-btn-text=" 文字 "** 的属性，或者通过程序来实现 **?\$.mobile.page.prototype.options.backBtnText = "文字";**

如果你想配置后退按钮的主题样式，使用 `$.mobile.page.prototype.options.backBtnTheme = "a"` 如果你要用这段程序，需要在 `mobileinit` 时间的处理程序中使用

自定义头部栏的配置

如果你想创建一个不遵循默认配置的头部栏，在 `header` 容器里用 `div` 包裹你的自定义内容就好，`Jquery Mobile` 不会自动生成按钮，所以你可以给你的头部栏自定义你的样式

尾部栏

尾部栏结构

尾部栏除了使用的 `data-role` 的属性与头部栏不同之外，基本的结构与头部栏是相同的

```
<div data-role="footer">
  <h4>Footer content</h4>
</div>
```

尾部工具栏默认下应用的主题样式为"`a`"(黑色)，但是你可以很轻松的设置主题样式

默认的尾部栏

Footer content

尾部栏的设置与头部栏基本相同。最大的不同是，尾部栏被设定为结构尽量简单，所以 `Jquery Mobile` 不会自动给尾部栏生成按钮

添加按钮

给尾部栏添加任何有效的按钮标记的元素都会生成按钮。为了节省空间，工具栏里的按钮都是内联按钮，所以按钮的宽度只容纳 icon 和里面的文字

默认情况下，工具栏内部容纳组件与导航时是不留 padding 的。

如果要给工具栏增加 padding, 请增加一个 ui-bar 的 class.

```
<div data-role="footer" class="ui-bar">
  <a href="index.html" data-role="button" data-icon="delete">Remove</a>
  <a href="index.html" data-role="button" data-icon="plus">Add</a>
  <a href="index.html" data-role="button" data-icon="arrow-u">Up</a>
  <a href="index.html" data-role="button" data-icon="arrow-d">Down</a>
</div>
```

要想把几个按钮打包成一个按钮组，则需要把这些按钮用一个容器包裹，并给该容器增加 data-role="controlgroup" 和 data-type="horizontal" 属性

```
<div data-role="controlgroup" data-type="horizontal">
```

尾部的成组按钮



增加表单元素

表单元素和其他元素也可以添加进工具栏中。这是一个把 select 添加到尾部栏的实例：



持续的尾部栏

有些情况下你需要一个尾部栏为全局导航元素，希望页面转场时尾部栏也固定并显示。创建一个持续的尾部栏可以做到这一点

给尾部栏添加 **data-id** 属性，并且在所有关联的页面的尾部栏设定同样的 **data-id** 的值，就可以使尾部栏在页面转场时也固定并显示，例如给当前页面和目标页面的尾部栏添加 **id="myfooter"** 属性，Jquery Mobile 会在页面转场动画的时候保持尾部栏固定 请注意：这个效果只有在头部栏和尾部栏设定为固定的定位模式（**data-position="fixed"**）时在才有用，这样他们在页面转场时才不被隐藏

导航栏

简易的导航栏

Jquery Mobile 提供了一个基本的导航栏组件，每一行可以最多放 5 个按钮，通常在顶部或者底部

导航栏的代码为一个 **ul** 列表，被一个容器包裹，这个容器需要有 **data-role="navbar"** 属性。要设定某一个链接为活动（**selected**）状态，给链接增加 **class="ui-btn-active"** 即可。我们给尾部栏设置了一个导航栏，把 **one** 项设置为活动状态

```
<div data-role="footer">
  <div data-role="navbar">
    <ul>
      <li><a href="a.html" class="ui-btn-active">One</a></li>
      <li><a href="b.html">Two</a></li>
```

```
</ul>
</div><!-- /navbar -->
</div><!-- /footer -->
```

尾部工具栏默认下应用的主题样式为"a"(黑色),但是你可以很轻松的设置主题样式

导航栏内每项的宽度都被设定为相同的,所以按钮的宽度为浏览器宽度 $1/2$.新增加一项的话,每项的宽度自动匹配为 $1/3$,以此类推。如果导航栏多于 5 项,导航栏自动表现为多行

两个按钮的导航栏

三个按钮的导航栏



五个按钮的导航栏

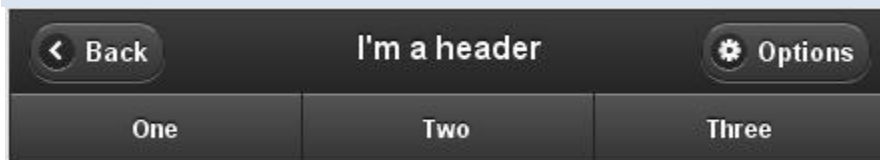


多余 5 项的导航

头部的导航栏

如果要给页面头部增加一个导航栏,你也可以保留头部栏的页面标题和按钮。只需要把导航栏容器放进头部栏容器内

头部的导航栏



导航栏的图标

给导航栏的列表项链接增加 **data-icon** 属性，可以给链接设置一个标准的移动网站的图标。给列表项链接增加 **data-iconpos="top"** 属性，可以给链接的图标设置位置为在文字上方

导航栏的图标

使用第三方的图标

你可以可以把任何你喜欢的第三方的 **icon** 组库加入到自己的项目中。只需要在 **css** 中自定义 **icon** 的地址和位置就可以了。下面一个实例为使用了 **Glyphish icon** 组的导航栏

导航栏的图标

Icons by Joseph Wain / glyphish.com. Licensed under the Creative Commons Attribution 3.0 United States License.

给导航栏主题样式

给导航栏设置 **data-theme** 属性,, 可以给导航栏设置主题样式

导航栏的主题



固定定位的工具栏会在滚屏后重新出现

这个页面是一个使用 **Jquery Mobile** 设定了固定定位的头部栏和尾部栏的实例。页面的文档流是正常的，让我们能够正常的布局文档，而不用通过脚本写。头部栏和尾部栏在文档流中的正常位置，但是当他们被滚动出屏幕后，你可以通过点击屏幕，使他们重新出现。再次点击或者滚动屏幕会使他们重新出现在文档流中（在顶部或底部）

要给头部栏或底部栏设置这样的行为，需要给工具栏的容器增加 **data-position="fixed"** 属性

```
<div data-role="header" data-position="fixed">  
  <h1>Fixed toolbars</h1>  
</div>
```

全屏的工具栏

这个页面展示“全屏”工具栏模式。这个工具栏一般用在比较特殊的场合，比如想要把内容撑满全屏，然后头部栏和尾部栏随着点击页面而出现和消失-在图片，照片和视频浏览器经常需要这样

要打开工具栏的这种特性，给页面的 **"page"** 增加 **data-fullscreen="true"** 属性，然后给头部工具栏和尾部栏的容器增加 **data-position="fixed"** 属性

注意在这种模式下工具栏会遮住页面的内容，所以当工具栏可见时不是全部内容都能够看见的

持续的工具栏

这个页面是一个持续的尾部导航栏的示例。在页面的底部，你会看到一个持续的水平导航栏。点击任何链接，你会看到页面会转场到另一个，但是尾部栏保持了固定。因 2 个页面都是用了同样的 `data-id` 属性

给导航栏主题样式

头部栏和尾部栏都默认都设“a”主题样式（黑色），因为这两个工具栏在视觉上为页面最首要的

给头部栏和尾部栏设置主题样式

给头部栏和尾部栏设置 `data-theme` 属性，并指定不同的字母（a, b, c 等）。下面一个例子是把头部栏的主题样式设“b”(蓝色)

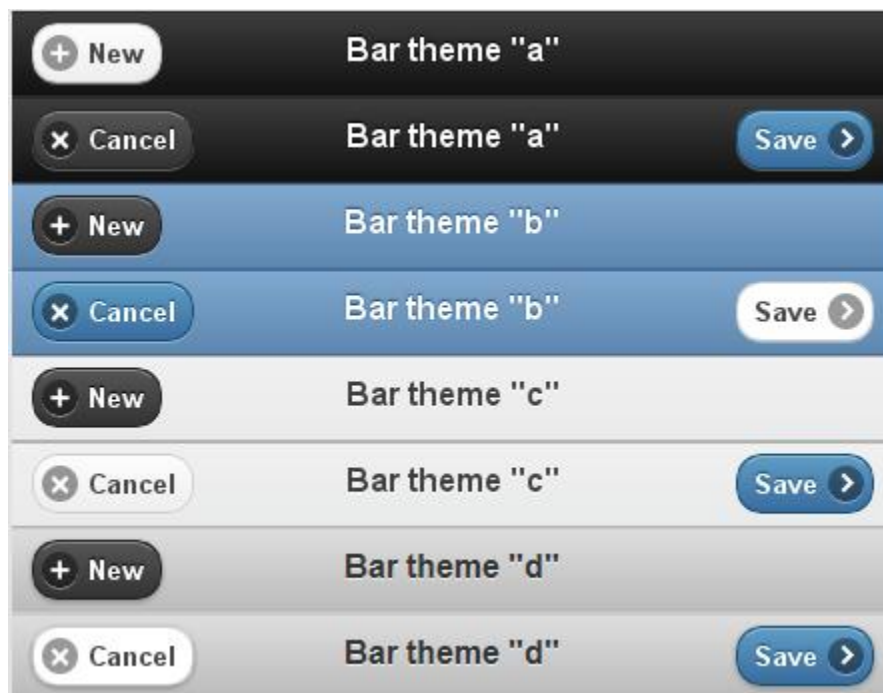
```
<div data-role="header" data-theme="b">
  <h1>Page Title</h1>
</div>
```

给导航栏的按钮设置主题样式

头部栏里的任何加入的任何链接会自动设为与此工具栏相同的主题样式。给链接增加 `data-theme` 属性并设置，可以让按钮的主题样式与工具栏的不相一致。例如：如果我们给头部栏设定的主题样式为“c”(浅灰)，里面的两个按钮的主题样式默认也会为“c”，如果我们想让按钮有所区别，可以给按钮设置 `data-theme=b`(蓝色)

```
<a href="add-user.php" data-theme="b">Save</a>
```

不同主题的头部栏



不同主题的尾部栏



按钮

按钮综述

综述 [选项方法事件](#)

按钮是由标准的 **HTML** 的 **a** 标签和 **input** 元素写成，然后 **Jquery Mobile** 会让他们变得更吸引人而且易于触摸使用。查阅 **data- attribute reference** 来了解你可以给按钮添加的属性。

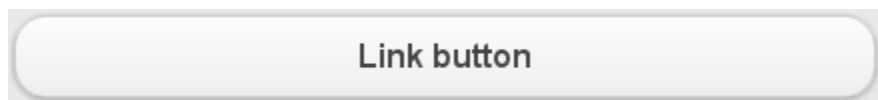
将链接样式化为按钮

在 **page** 元素的主要 **block** 内，可以通过给任意链接加 **data-role="button"** 的属性样式化为按钮。**Jquery Mobile** 会给链接加一些必要的 **class** 来把他表现为按钮。

例如：

```
<a href="index.html" data-role="button">Link button</a>
```

就会生成一个基于连接的按钮：



在 **page** 元素的主要 **block** 内，可以通过给任意链接加 **data-role="button"** 的属性样式化为按钮。**Jquery Mobile** 会给链接加一些必要的 **class** 来把他表现为按钮。

表单按钮

为了容易的样式化按钮，**Jquery Mobile** 自动把 **type** 为 **submit**, **reset**, **button** 或 **image** 的按钮元素或输入元素样式化为按钮，所以没有必要增加 **data-role="button"** 的属性

基于表单(**form-based**)的按钮的原始按钮(**input**)是隐藏的，但是依然保留其标记。当一个按钮的点击事件触发时，也会在原始的表单按钮上触发点击事件。

下面的选项适用于所有的按钮

\$.mobile.changePage

布尔值

默认: **true**

此参数为 **true** 的话则按钮会应用主题的圆角。此选项也可以通过 **data-corners="false"** 的属性设置

`$('#a').buttonMarkup({ corners: "false" });`

icon

字符串

默认: **null**

从图标组给按钮添加图标。此选项也可以通过 **data-corners="false"** 的属性设置

`$('#a').buttonMarkup({ icon: "star" });`

iconpos

字符串

默认: **"left"**

定义按钮的图标的位置。可能的值是: **left, right, none, notext**. 使用 **notext** 值的话按钮按钮只显示图标, 没有文字。此选项也可以通过 **data-iconpos="left"** 的属性设置

`$('#a').buttonMarkup({ iconpos: "right" });`

iconshadow

布尔值

默认: **true**

此参数为 **true** 的话则按钮会应用主题的圆角。此选项也可以通过 **data-iconshadow="false"** 的属性设置

`$('#a').buttonMarkup({ iconshadow: "false" });`

inline

布尔值

默认: **null (false)**

设为 **true** 的话, 会使按钮为内联的样式, 这样按钮的宽度就由按钮内的文字来决定。默认情况下此项为 **null (false)**, 所以按钮会的宽度会撑满, 不管里面有多少文字。可以使用的值是 **:true, false**。此选项也可以通过 **data-inline="true"** 的属性设置

```
$('#a').buttonMarkup({ inline: "true" });
```

shadow

布尔值

默认: **true**

设为 **true** 的话, 会使按钮有阴影。此选项也可以通过 **data-shadow="false"** 的属性设置

```
$('#a').buttonMarkup({ shadow: "false" });
```

theme

字符串

默认: **null**, 从父容器继承

给按钮的所有实例设定主题样式。接受从 **a-z** 的一个单个字母来映射相应的主题样式。默认情况下会继承父容器的主题样式。此选项也可以通过 **data-theme="a"** 的属性设置

```
$('#a').buttonMarkup({ theme: "a" });
```

下面的选项只适用于**表单的按钮**, 由框架自动初始化

initSelector

css 选择符

默认: **"button, [type='button'], [type='submit'], [type='reset'], [type='image']"**

此选项用来定义被初始化为表单按钮的选择符 (通过元素类型, 数据规则等)

要改变被初始化的元素, 需要给 **mobileinit event** 事件绑定这个选项

```
$( document ).bind( "mobileinit", function(){  
    $.mobile.button.prototype.options.initSelector = ".myButtons";  
});
```

下面的方法只适用于表单中的按钮。基于链接的按钮没有相关联的方法。

此参数为 `true` 的话则按钮会应用主题的圆角。此选项也可以通过 `data-corners="false"` 的属性设置
`$( 'a' ).buttonMarkup( { corners: "false" } );`

enable

是一个被 **disabled** 的表单按钮可用

```
$( '[type='submit']' ).button( 'enable' );
```

disable

禁用一个表单按钮

```
$( '[type='submit']' ).button( 'disable' );
```

refresh

更新一个表单按钮

如果你通过 `js` 更新了一个表单按钮，你必须再对他通过 `refresh` 方法更新他的视觉样式。

```
$( '[type='submit']' ).button( 'refresh' );
```

通过 `jquery mobile` 的 虚拟事件 或者绑定标准的 `js` 事件，例如 `change`, `focus`, `blur` 等, 可以给 `a`, `input`, 或者是 `button` 元素直接绑定事件。

```
$( ".myButton" ).bind( "click", function(event, ui) {  
    ...
```

```
});
```

enable

使一个被 **disabled** 的表单按钮可用

```
$('[type="submit"]').button('enable');
```

表单按钮拥有如下的自定义事件：

create

在一个表单元素被创建是触发

```
$('[type="submit"]').button({  
create: function(event, ui) { ... }  
});
```

按钮的图标

给按钮添加图标

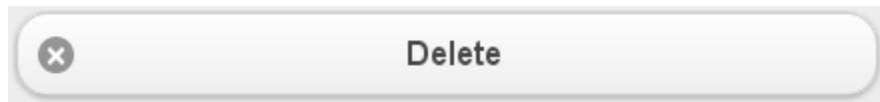
Jquery Mobile 框架包含了一组最常用的移动应用程序所需的图标，为了减少下载的大小，Jquery Mobile 包含的是的白色的图标 **sprite** 图片，并自动在图标后添加一个半透明的黑圈以确保在任何背景色下图片都能够清晰显示。

给链接添加 **data-icon** 属性，可以添加按钮的图标

html 代码

```
<a href="index.html" data-role="button" data-icon="delete">Delete</a>
```

带有图标的按钮：



图标

`data-icon` 属性可以被用来创建如下所示的图标

左箭头 `data-icon="arrow-l"`

右箭头 `data-icon="arrow-r"`

上箭头 `data-icon="arrow-u"`

下箭头 `data-icon="arrow-d"`

删除 `data-icon="delete"`

添加 `data-icon="Plus"`

减少 `data-icon="minus"`

检查 `data-icon="Check"`

齿轮 `data-icon="gear"`

前进 `data-icon="Forward"`

后退 `data-icon="Back"`

网格 `data-icon="Grid"`

五角星 `data-icon="Star"`

警告 `data-icon="Alert"`

信息 data-icon="info"

首页 data-icon="home"

搜索 data-icon="Search"

图标组

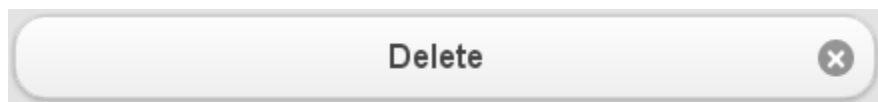
默认情况下，所有按钮图标出现在按钮的文本的左侧。

可以通过 data-iconpos="top" / "bottom" 属性来覆盖此默认

```
<a href="index.html" data-role="button"
```

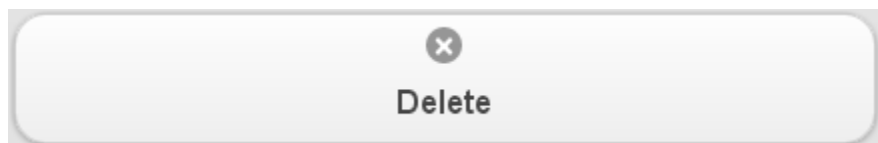
```
data-icon="delete" data-iconpos="right">Delete</a>
```

一个图标在右边的按钮：



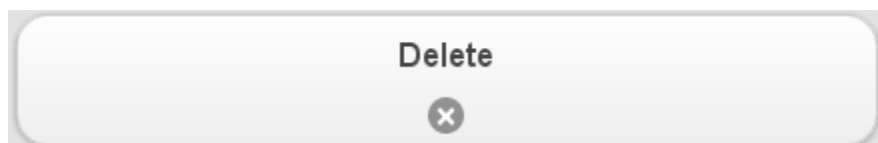
也可以用 data-iconpos="top" 创建图标在文本上方的按钮

一个图标在文字上方的按钮：



也可以用 data-iconpos="bottom" 创建图标在文本下方的按钮

一个图标在文字下方的按钮：



你可以通过 data-iconpos="notext" 创建一个只有图标的按钮。

button 插件会在屏幕上隐藏文本，但是会把文本作为 **title** 属性作为 screen readers 的内容和支持小提示的浏览器， 例如，把前例中的 data-iconpos="right" 替换为 data-iconpos="notext"

```
<a href="index.html" data-role="button" data-icon="delete" data-iconpos="notext">Delete</a>
```

一个只有图标的按钮



自定义图标

要使用自定义图标，指定一个唯一的 **data-icon** 值（比如 data-icon="myapp-email"）？Jquery Mobile 的 button 插件会生成一个 class 值添加上去，该值由 ui-icon-与 data-icon 的值组合而成（ui-icon-myapp-email ），然后在 css 中指定这个类的背景图片地址。为了保持视觉效果的一致，请使用 png-8 格式的 8*18 的透明图标

图标和地址

在白色图标后的半透明的黑色圆圈确保了在任何背景色下图片都能够清晰显示，也使它能很好的工作在 Jquery Mobile 主题系统中。以下是一些在不同主题样式下图标按钮的例子

"A"主题下的图标按钮



"B"主题下的图标按钮

"C"主题下的图标按钮

内联的按钮

默认情况下，`body` 里的所有按钮都会被样式化为块级元素，撑大到与页面等宽

但是，如果你想让按钮外观紧凑，宽度只符合里面的文字和 `icon`，那就给按钮添加 `data-inline="true"` 的属性

内联的按钮：



如果你有多个按钮想在一行横排，可以在按钮上增加 `data-inline="true"` 的属性，这样就可以把按钮样式化为最小宽度，并且浮动，使他们在同一行横排

```
<div data-inline="true">
  <a href="index.html" data-role="button">Cancel</a>
  <a href="index.html" data-role="button" data-theme="b">Save</a>
</div>
```

内联的按钮组



如果你想多个按钮在通过一行横排并且容器自动撑大到与页面同宽，可以使用 内容分栏 把多个普通的块级按钮 放入同一行

内

组按钮

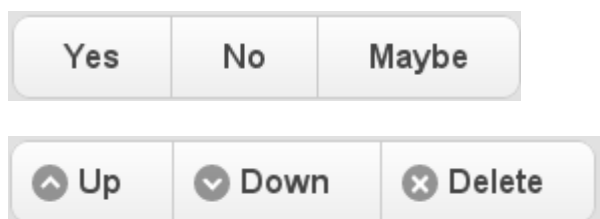
有时候，你想把一组按钮放进一个单独的容器内，使他们看起来像一个独立的导航部件。你可以把一组按钮包裹在一个容器内，然后给该容器添加 `data-role="controlgroup"` 属性，Jquery Mobile 会创建一个垂直的按钮组，删除掉按钮间的 `margin` 和阴影，然后只在第一个按钮和最后一个按钮产生圆角，使他们看起来是一组按钮

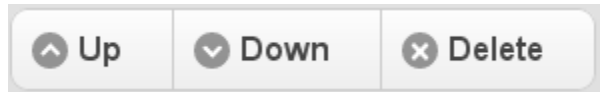
```
<div data-role="controlgroup">
  <a href="index.html" data-role="button">Yes</a>
  <a href="index.html" data-role="button">No</a>
  <a href="index.html" data-role="button">Maybe</a>
</div>
```

按钮组：

默认情况下，组按钮表现为垂直列表，如果给容器添加 `data-type="horizontal"` 的属性，则可以转换为水平按钮的列表，按钮会横向一个挨着一个地排列，容器会自动撑大到适应内容（所以要注意横排情况下按钮不要太多，按钮的字也不要太多）

水平的组按钮 **Horizontal grouped buttons:**





给按钮主题样式

Jquery Mobile 内建了几套样式系统，给你定义样式时多种选择，在容器内添加一个按钮后，它就会自动应用他的容器使用的样式系统，使得按钮和页面看起来协调统一，像变色龙

所以放进一个样式主题为 **a** 系统（黑色主题）容器的按钮会自动应用 **a** 系统样式。如下为实例

a 为黑，**b** 为灰底蓝，**c** 为灰底灰，**d** 白底白色，**e** 黄底黄色

A 主题

B 主题

B swatch

Button

C 主题

C swatch

Button

D 主题

E 主题

E swatch

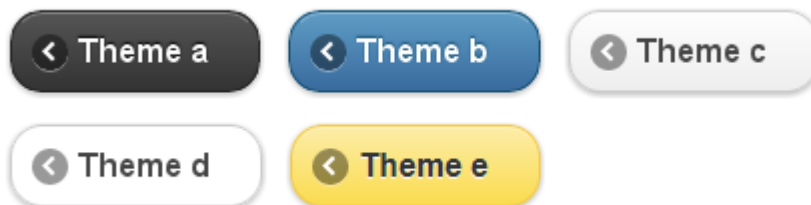
Button

分配系统样式

给按钮增加 `data-theme="字母"` 属性，可以手动的给按钮添加样式，使得按钮不一定非要与父容器的样式相符

```
<a href="index.html" data-role="button" data-theme="a">Theme a</a>
```

5 个通过 **data-theme** 设定不同主题的按钮

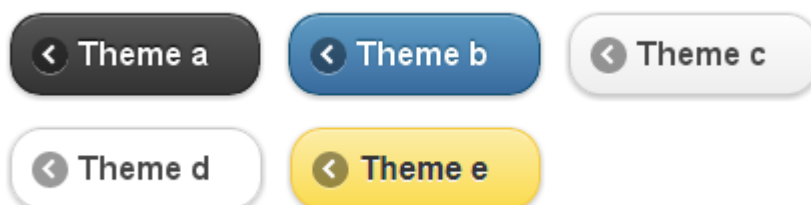


A 主题下的不同主题的按钮

B 主题下的不同主题的按钮

C 主题下的不同主题的按钮

D 主题下的不同主题的按钮



E 主题下的不同主题的按钮

内容格式化

基本的 html 样式

在 Jquery Mobile 页面的内容是完全开放的，但是 Jquery Mobile 框架提供了一些有用的工具及组件，比如可折叠的面板，多列网格布局等，方便地为移动设备格式化你的内容

html 格式化

在 Jquery Mobile 中通过默认方法给内容添加样式是很简单的。我们的目标是让浏览器的默认渲染优先进行，然后我们加了一点点小小的 padding 让页面看起来更有可读性，然后应用主题样式系统来分配字体和颜色

采用熟手优先原则给与了设计者和开发者一个干净的空间工作，而不是和一大堆复杂的样式代码战斗。

默认 HTML 标记样式

默认情况下，Jquery Mobile 的主题样式为标准的 html 元素使用标准的 html 样式，字号，例如 header, p, block quotes, a, ul, ol, dl, dt

网格布局

因为屏幕通常都比较窄，所以使用多栏布局的方法在移动设备上不是推荐的方法。但是总有时候你会想要把一些小的元素并排放

置（比如按钮，或导航标签）

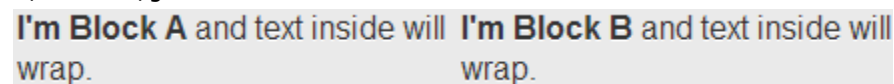
Jquery Mobile 框架提供了一种简单的方法构建基于 **css** 的分栏布局，叫做 **ui-grid**

Jquery Mobile 提供了两种预设的配置布局：两列布局（**class** 含有 **ui-grid-a**）和三列布局（**class** 含有 **ui-grid-b**）——几乎可满足需要列布局的任何情况。网格是 **100%**宽的，不可见（没有背景或边框），也没有 **padding** 和 **margin**，所以它们不会影响内部元素的样式

两栏布局

要构建两栏的布局（**50/50%**），先构建一个父容器，添加一个 **class** 名字为：**ui-grid-a**，内部设置两个字容器，分别给第一个子容器添加 **class: ui-block-a**，第二个子容器添加 **class: ui-block-b**

```
<div class="ui-grid-a">
  <div class="ui-block-a"><strong>I'm Block A</strong> and text inside will wrap</div>
  <div class="ui-block-b"><strong>I'm Block B</strong> and text inside will wrap</div>
</div><!-- /grid-a -->
```



如上所见，默认的两栏没有样式，并行排列。分栏的 **class** 可以应用到任何类型的容器上。下一个实例中，我们给表单的 **fieldset** 添加 **class="ui-grid-a"**，然后给两个 **button** 的所在的子容器添 **class="ui-block-a"**和 **class="ui-block-b"**。使两个容器各自 **50%**的宽



在下面的区块中，我们增加了两个 class,增加 ui-bar 的 class 给默认的 bar padding，增?ui-bar-e 的 class 应用背景渐变以及工具栏的主题 e 的字体样式。然后在每个网格的标签内增加 style="height:120px"的属性来设置高度

三栏布局

另一种布局的方式是三栏布局，给父容器添加 "class="ui-grid-b"，然后分别给三个字容器添加 "class="ui-block-a"，"class="ui-block-b"，"class="ui-block-c"

```
<div class="ui-grid-b">
  <div class="ui-block-a">Block A</div>
  <div class="ui-block-b">Block B</div>
  <div class="ui-block-c">Block C</div>
</div><!-- /grid-a -->
```

就会生成一个 33/33/33% 的分栏布局

3 列网格布局里面是按钮的示例

以此类推：如果是 4 栏布局，则给父容器添加 class="ui-grid-c" (2 栏为 a, 3 栏为 b, 4 栏为 c, 5 栏为 d。。。。)，子容器分别添加 class="ui-block-a"，"class="ui-block-b" "class="ui-block-c".....

多行的网格布局

网格化布局也适用于多栏布局的方式，比如，如果你指定了一列布局的父容器，里面有 9 个子容器，则会包裹为 3 行，每行 3 个。可以给每行的第一个容器设置为 `class=ui-block-a` 来清除浮动，这样 9 个子容器的 `class` 应为：`class=ui-block-(a,b,c,a,b,c,a,b,c)` 重复



可折叠的区块

要创建一个可折叠的区块，先创建一个容器，然后给容器添加 `data-role="collapsible"` 属性

容器内直接的标题 (`h1-h6`) 子结点，Jquery Mobile 会将之表现

为可点击的按钮，并在左侧添加一个"+"按钮，表示是可以展开的

在头部后面你可以添加任何想要折叠的 **html** 标记。框架会自动把这些标记包裹在一个容器里用以折叠或显

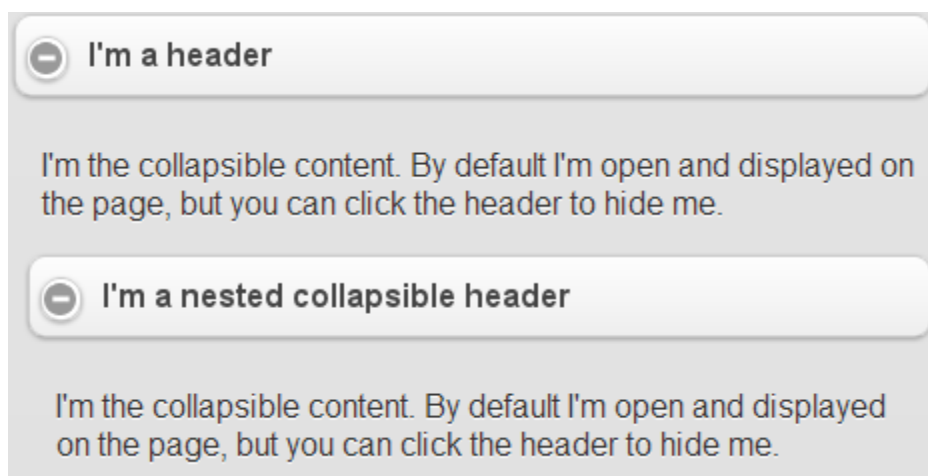
```
<div data-role="collapsible">
  <h3>I'm a header</h3>
  <p>I'm the collapsible content. By default I'm open and displayed on the page, but you can click the
header to hide me.</p>
</div>
```

如上所示：默认情况下，可折叠容器是展开的，可以通过通过点击头部收缩。给折叠的容器添加 **data-collapsed="true"**的属性，可以设为默认收缩

```
<div data-role="collapsible" data-collapsed="true">
```

可折叠的内容采用了精简的样式，我们仅仅在内容和标题间添加了一些 **margin**，标题则采用它所在容器的默认主题

嵌套的折叠



折叠组

通过给父容器添加 `data-role="collapsible-set"` 属性，然后每一个子容器 `data-role="collapsible"` 属性，可以让容器展开时，其他容器被折叠的效果，类似手风琴组件

折叠组标记

折叠组的标记和单个的折叠区域的标记的开头是一样的。将若干可折叠区域用一个容器包裹，再给此容器增加 **`data-role="collapsible-set"`** 属性，框架会自动将这些可折叠的部件组合成为一个视觉上成组的部件，使他们看上去像手风琴，并且在同一个时间只会会有一个容器是展开的。

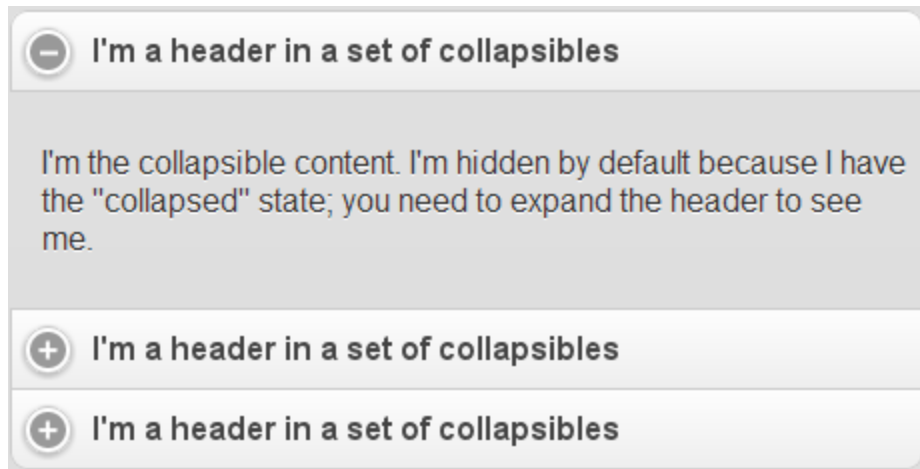
默认情况下，手风琴中所有的部件都是收缩起来的。。如果想设置某个部件是打开的，可以给这个部件的标题容器添加 **`data-collapsed="false"`** 属性。

```
<div data-role="collapsible-set">
    <div data-role="collapsible" data-collapsed="false">
        <h3>Section 1</h3>
        <p>I'm the collapsible set content for section B.</p>
    </div>

    <div data-role="collapsible">
        <h3>Section 2</h3>
        <p>I'm the collapsible set content for section B.</p>
    </div>
</div>
```

如上所示：默认情况下，可折叠容器是展开的，可以通过通过

点击头部收缩。给折叠的容器添加 `data-collapsed="true"` 的属性，可以设为默认收缩



给折叠组主题样式

普通的 **data-theme** 属性可以加在手风琴组上来设定主题样式。如果想让手风琴组的标题单独设计主题样式，可以添加 **data-content-theme** 属性。

```
<div data-role="collapsible-set" data-theme="c" data-content-theme="d">
```

给每个折叠部件主题样式

如果想给组内的每个部件不同的主题样式，可以给每个部件单独添加 **data-theme** 和 **data-content-theme** 。

给内容主题样式

表单元素

所有的表单元素都是由标准的 **html** 元素控制的,然后又得到增强,使他们更吸引人并且容易使用。在不支持 **Jquery Mobile** 的浏览器下仍然是可用的,因为它们都是基于原生的 **html** 元素.

表单基础

Jquery Mobile 提供了一套完整的, 适合触摸操作的表单元素, 他们都是基于原生的 **html** 元素

表单结构

所有的表单都应该被包裹在一个 **form** 标签内, 这个标签应该指定好 **action** 和 **method** 属性用来控制与服务器传送数据的方法

html 代码

```
<form action="form.php" method="post"> ... </form>
```

标记的规约

在 **Jquery Mobile** 中组织表单时, 大多数创建普通的 **HTTP post** , **get** 方式的表单是需要遵守的规约依然是适用的? 但是有一点要注意, **form** 的 **id** 属性不仅需要在该页面内唯一? 也需要在整个网站的所有页面中是唯一的

这是因为 **Jquery Mobile** 的单页面内导航的机制使得多个不同 "page" 可以同时在 DOM 中出现，所以你必须给表单使用不同的 id 属性，以保证在每个 DOM 中的表单的 ID 都是不同的。（当然，请务必给内部的 label 元素合适的使用 for 属性）

表单元素的自动初始化

默认情况下，**Jquery Mobile** 会自动把原生的表单元素增强为适合触摸操作的组件。这是它通过标签名寻找表单元素，然后对他们执行 **Jquery Mobile** 插件的方法内部实现的，所以比如说，select 元素被找到后，**Jquery Mobile** 通过用 selectmenu 插件进行初始化，而一个属性为 type="checkbox" 的 input 元素会被 checkboxradio 插件来增强。初始化完毕后，你可以用他们的 **Jquery UI** 的组件的方法通过程序进一步使用或设定他们的增强功能。（查看 [查找可用的方法](#)）

防止表单元素被自动初始化

如果你需要某表单元素不被 Jquery Mobile 处理，只需要给这个元素增加 data-role="none" 属性

html 代码

```
<label for="foo">
  <select name="foo" id="foo" data-role="none">
    <option value="a">A</option>
    <option value="b">B</option>
    <option value="c">C</option>
  </select>
```

或者，如果你不想增加上述的属性，可以设置页面插件?keepNative 选项（默认情况下为[`data-role="none"`]），来自定义用来防止初始化的选择器。请务必在 `mobileinit` 事件的处理程序中设置这个选项，能让第一个页面以及随后被加载的页面应用此设置

html 代码

```
$(document).bind('mobileinit',function(){  
$.mobile.page.prototype.options.keepNative = "select, input.foo, textarea.bar";  
});
```

`select` 元素情况比较特殊。如果 `select` 元素在表单当中，上例会阻止针对页面中 `select` 元素的所有的增强行为。如果你想保持菜单本身的外观与表现，行为，同时又想通过 **Jquery Mobile** 增强 `select` 的按钮的视觉外观，你可以在 `mobileinit` 的回调函数中 `.mobile.nativeSelectMenu` 设为 `true` 作为全局设置或者在每个元素上逐个使用 `data-native="true"` 进行设置

动态的表单布局

在 **Jquery Mobile** 中，所有的表单元素都被设计成弹性宽度以适应不同移动设备的屏幕宽度。在 **Jquery Mobile** 中内建的一个优化就是根据屏幕宽度的不同，`label` 和表单元素的宽度是不同的

如果屏幕宽度相对窄（小于 `480px`），`label` 元素会被样式化为块级元素，使他们能够置于表单元素上方，节省水平空间

在比较宽的屏幕上，**label** 和表单元素会被样式化为两列的网格布局的一行中，充分利用页面的空间

表单内区域容器

我们建议把表单内的每一个 **label/**表单元素对用 **div** 或 **fieldset** 容器包裹，然后增加 **data-role="fieldcontain"** 属性，以改善标签和表单元素在宽屏设备中的样式。**Jquery Mobile** 会自动在容器底部添加一条细边框作为分隔线，使得 **label/**表单元素对在快速扫视时看起来对齐

html 代码

```
<div data-role="fieldcontain">
...label/input code goes here...
</div>
```

刷新表单元素

每一个表单组件都有刷新方法。以下是实例

html 代码

复选按钮

```
$("input[type='checkbox']").attr("checked",true).checkboxradio("refresh");
```

单选按钮组：

```
$("input[type='radio']").attr("checked",true).checkboxradio("refresh");
```

选择列表：

```
var myselect = $("#select#foo");
myselect[0].selectedIndex = 3;
myselect.selectmenu("refresh");
```

滑动条

```
$("#input[type=range]").val(60).slider("refresh");
```

开关 (they use slider):

```
var myswitch = $("#select#bar");  
myswitch[0].selectedIndex = 1;  
myswitch.slider("refresh");
```

表单元素示例

此页面包含了通过渐进增强驱动的表单元素。原生的表单元素有时被隐藏了，但他们的 **value** 值被保存了，所以表单可以正确的提交

不支持自定义组件的浏览器也能使用它们，因为所有表单组件都基于原生的 **html** 表单元素.

文本输入框 **Text inputs**

文本输入域 **Textarea:**

搜索输入框 **Search inputs**

滑动条 **Slider**

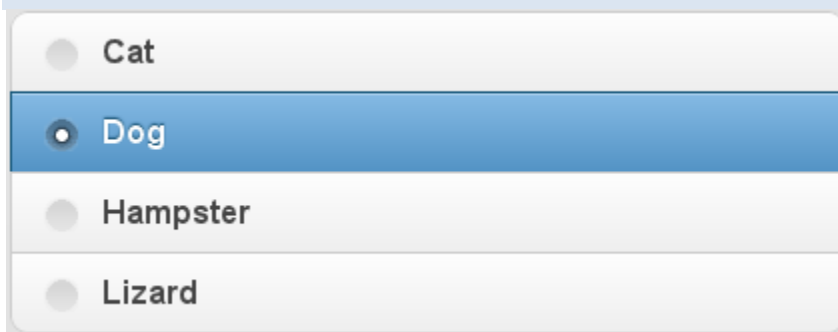
开关 **Flip toggle switches**

复选按钮 checkbox

文字设置



单选按钮组 radiobox



列表按钮?select

按钮 button



Text 输入框

文本输入框和文本输入域

[综述](#) [选项方法事件](#)

文本输入框和文本输入域是使用标准的 `html` 标记的, 然后 `Jquery Mobile` 会让他们变得更吸引人而且易于触摸使用。查阅 `data- attribute reference` 来了解你可以给文本输入框添加的属性。

文本输入框

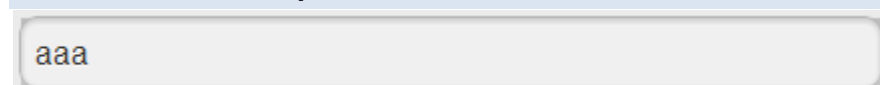
要使用标准字母数字的输入框，给 `input` 增加 `type="text"` 属性。注意要把 `label` 的 `for` 属性设为 `input` 的 `id` 值，使他们能够在语义上相关联。如果在页面内不想看到 `label` 的话可以隐藏 `label`。

HTML 代码:

```
<label for="name">Text Input:</label>
<input type="text" name="name" id="name" value="" />
```

这样就创建了一个 **Text** 输入框。默认的风格是宽度为父容器的 **100%**，`label` 在另一行上。

文本输入框 Text Input:



你也可以用 `div` 容器包裹它们，并给他设定 `data-role="fieldcontain"` 属性，使他们在 一个大的表单里在视觉上是成组的。

更多的文本输入类型

在 `jquery mobile` 中，你可以使用现存的和新的 **HTML5** 输入类型比如 **password**, **email**, **tel**, **number** 和更多的类型。有一些类型会在不同的浏览器被渲染成不同的样式-比如 **Chrome** 会将 **range** 输入框渲染成滑动条-所以我们通过把类型转为 **text** 来标准化他们的外观（目前只作用于 **range** 和 **search** 元素）。你可以用 `page` 插件的选项来配置那些被降级为 **text** 的输入框的表现。使用这些特殊类型的输入框的好处是，在智能手机上不同的输入框对应的是不同的触摸键盘。

给 `input` 设置 `type="password"` 属性，可以设置为密码框，注意要把 `label` 的 `for` 属性设为 `input` 的 `id` 值，使他们能够在语义上相关联，并且要用 `div` 容器包裹它们，并给他设定 `data-role="fieldcontain"` 属性。

HTML 代码:

```
<div data-role="fieldcontain">
  <label for="password">Password Input:</label>
  <input type="password" name="password" id="password" value="" />
</div>
```

文本域

对于多行输入可以使用 `textarea` 元素。`Jquery Mobile` 框架会自动加大文本域的高

度防止出现在移动设备中很难使用的滚动条的出现。

注意要把 **label** 的 **for** 属性设为 **input** 的 **id** 值，使他们能够在语义上相关联，并且要用 **div** 容器包裹它们，并给他设定 **data-role="fieldcontain"** 属性

HTML 代码:

```
<div data-role="fieldcontain">
    <label for="textarea">Textarea:</label>
    <textarea cols="40" rows="8" name="textarea" id="textarea"></textarea>
</div>
```

文本域 Textareas

文本域

这个插件会在包含有 **text** 输入框的任何页面自动初始化，不需要标记 **data-role** 属性，但是如果需要的话你可以在选择器上调用 **textinput** 插件。

HTML 代码:

```
$('input').textinput();
```

被降级的输入框类型

jQuery Mobile 降级了几种 HTML5 的几种输入类型为 **text** 型或者 **number** 型。比如 **range** 类型的输入框会被增强为自己定义的划动条，他们的类型会被设置为 **number**，在划动条旁边会提供一个 **number** 输入框使之可用。而 **SEARCH** 输入框在我们给输入值添加主题后添加自己会被降级为 **text**。

page 插件包含了一系列输入类型，当他们被设定为 **TRUE** 时意味着他们会被降级成为 **text** 类型，**false** 意味着不处理他们，或者设为一个字符串比如 **"number"**，意味着他们会被转为该类型。

你也可以通过 **page** 插件的 **degradeInputs** 选项来配置那些被改变的输入类型，即通过外部的 **\$.mobile.page.prototype.options.degradeInputs** 来操作，它包含的属性有 **color**, **date**, **datetime**, **"datetime-local"**, **email**, **month**, **number**, **range**, **search**, **tel**, **time**, **url**, 和 **week**。请确保把这个配置放于 **mobileinit** 绑定的事件处理的函数内，使得他能在页面加载的第一时间起作用。

Search 输入框

搜索输入框

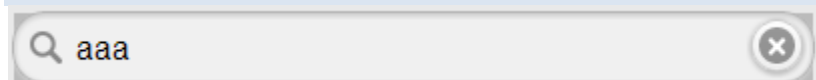
搜索输入框是一个新兴的 **html** 元素，外观为圆角，当你输入文字后右边会出现一个叉的图标，点击则会清除你输入的内容。给 **input** 增加 **type="search"** 属性来定义

注意要把 **label** 的 **for** 属性设为 **input** 的 **id** 值，使他们能够在语义上相关联，并且要用 **div** 容器包裹它们，并给他设定 **data-role="fieldcontain"** 属性

HTML 代码:

```
<div data-role="fieldcontain">
  <label for="search">Search Input:</label>
  <input type="search" name="password" id="search" value="" />
</div>
```

搜索输入框 Search inputs



滑杆

滑动条

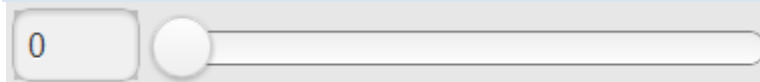
给 **input** 的设置一个新的 **HTML5** 属性为 **type="range"**，可以给页面添加滑动条组件，可以指定它的 **value** 值（当前值），**min** 和 **max** 属性的值配置滑动条。**Jquery Mobile** 会解析这些属性来配置滑动条。当你滑动滑动条时，**input** 会随之更新数值，反之亦然，使你能够很简单的在表单里提交数值。注意要把 **label** 的 **for** 属性设为 **input** 的 **id** 值，使他们能够在语义上相关联，并且要用 **div** 容

器包裹它们，并给他设定 `data-role="fieldcontain"` 属性

HTML 代码:

```
<div data-role="fieldcontain">
    <label for="slider">Input slider:</label>
    <input type="range" name="slider" id="slider" value="0" min="0"
max="100" />
</div>
```

滑动条 Slider



设置 `min` 和 `max` 属性的值你可以配置滑动条上下能取到的值，而 `value` 值是用来指定滑动条初始的位置和 `input` 框内的值

滑动杆同样对键盘有响应。右箭头，上箭头，`Page Up` 键可以用来增加当前值，左箭头，下箭头，`Page Down` 键则减少当前值。`Home` 键和 `End` 键则可以分别调到滑动条的最小值和最大值

刷新滑动条

如果你想通过 `js` 手动控制滑动条，务必调用 `refresh` 方法刷新滑动条的样式

```
$("input[type=range]").val(60).slider("refresh");
```

开关

开关在移动设备上是一个常用的 `ui` 元素，用来二元的切换开/关或者输入 `true/false` 类型的数据。你可以像滑动框一样拖动开关，或者点击开关任意一半进行操作

创建一个只有 2 个 `option` 的选择菜单就可以构造一个开关了。

第一个 **option** 会被样式化为 开, 第二个 **option** 会被样式化为 关, 所以请注意代码书写顺序。注意要把 **label** 的 **for** 属性设为 **input** 的 **id** 值, 使他们能够在语义上相关联, 并且要用 **div** 容器包裹它们, 并给他设定 **data-role="fieldcontain"** 属性。

HTML 代码:

```
<div data-role="fieldcontain">
    <label for="slider">Select slider:</label>
    <select name="slider" id="slider" data-role="slider">
        <option value="off">Off</option>
        <option value="on">On</option>
    </select>
</div>
```

开?Flip toggle switches

刷新开关

如果你想通过 **js** 手动控制开关, 务必调用 **refresh** 方法刷新样式

```
var myswitch = $("select#bar");
myswitch[0].selectedIndex = 1;
myswitch.slider("refresh");
```

单选按钮

复选框用来提供一组选项, 可以选中不止一个选项。传统的桌面程序的单选按钮组没有对触摸输入的方式进行优化, 所以在 **Jquery Mobile** 中, **label** 也被样式化为复选按钮, 使按钮更长, 容易点击。并添加了自定义的一组图标来增强视觉反馈

单选按钮组和复选按钮组都是用标准的 `html` 代码写的,但是都被样式化得更容易点击。你所看见的控件其实是覆盖在 `input` 上的 `label` 元素?所以如果图片没有正确加载,你仍然可以正常使用控件。在大多数浏览器里,点击 `label` 会自动触发在 `input` 上的点击,但是我们不得不为部分不支持该特性的移动浏览器人工去触发该点击。在桌面程序里,键盘和屏幕阅读器也可以使用这些控件

要创建一组复选框,为 `input` 添加 `type="checkbox"` 属性和相应的 `label` 即可。注意要把 `label` 的 `for` 属性设为 `input` 的 `id` 值,使他们能够在语义上相关联

因为复选按钮使用 `label` 元素放置 `checkbox` 后,用来显示其文本,我们推荐把复选按钮组用 `fieldset` 容器包裹,并给 `fieldset` 容器内增加一个 `legend` 元素,用来表示该问题的标题

最后,还需将 `fieldset` 包裹在有 `data-role="controlgroup"` 属性的 `div` 里以便于将该组元素和文本框,选择框等其他表单元素同时设置样式?

HTML 代码:

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <legend>Agree to the terms:</legend>
    <input type="checkbox" name="checkbox-1" id="checkbox-1" class="custom" />
    <label for="checkbox-1">I agree</label>
  </fieldset>
</div>
```

垂直成组的复选按钮组

通常情况下，复选按钮组会置于标题下面，Jquery Mobile 会自动移除按钮间的 `margin`，给按钮组上下添加圆角，使得单选按钮组看起来为一个整体

水平的复选按钮组

复选按钮组也可用作复选的水平按钮组，可以同时选择多个按钮。比如说文字编辑器的粗体，斜体，下划线。只需要为 `fieldset` 添加 `data-type="horizontal"` 属性即可

```
<fieldset data-role="controlgroup" data-type="horizontal" data-role="fieldcontain">
```

Jquery Mobile 会自动将标签浮动，并排放置，并隐藏复选按钮的 `icon`，并给左右两边的按钮增加圆角



刷新复选按钮

如果你想通过 `js` 手动控制复选按钮，务必调用 `refresh` 方法刷新样式

```
$("#input[type='checkbox']").attr("checked",true).checkboxradio("refresh");
```

复选按钮

复选框用来提供一组选项，可以选中不止一个选项。传统的桌面程序的单选按钮组没有对触摸输入的方式进行优化，所以在

Jquery Mobile 中，**label** 也被样式化为复选按钮，使按钮更长，容易点击。并添加了自定义的一组图标来增强视觉反馈。

单选按钮组和复选按钮组都是用标准的 **html** 代码写的，但是都被样式化得更容易点击。你所看见的控件其实是覆盖在 **input** 上的 **label** 元素，所以如果图片没有正确加载，你仍然可以正常使用控件。在大多数浏览器里，点击 **label** 会自动触发在 **input** 上的点击，但是我们不得不为部分不支持该特性的移动浏览器人工去触发该点击。在桌面程序里，键盘和屏幕阅读器也可以使用这些控件。

要创建一组复选框，为 **input** 添加 **type="checkbox"** 属性和相应的 **label** 即可。注意要把 **label** 的 **for** 属性设为 **input** 的 **id** 值，使他们能够在语义上相关联。

因为复选按钮使用 **label** 元素放置 **checkbox** 后，用来显示其文本，我们推荐把复选按钮组用 **fieldset** 容器包裹，并给 **fieldset** 容器内增加一个 **legend** 元素，用来表示该问题的标题。

最后，还需将 **fieldset** 包裹在有 **data-role="controlgroup"** 属性的 **div** 里以便于将该组元素和文本框，选择框等其他表单元素同时设置样式。

HTML 代码:

```
<div data-role="fieldcontain">
<fieldset data-role="controlgroup">
<legend>Agree to the terms:</legend>
<input type="checkbox" name="checkbox-1" id="checkbox-1"
```

```
class="custom" />
<label for="checkbox-1">I agree</label>
</fieldset>
</div>
```

垂直成组的复选按钮组

通常情况下，复选按钮组会置于标题下面。Jquery Mobile 会自动移除按钮间的 `margin`，给按钮组上下添加圆角，使得单选按钮组看起来为一个整体。

水平的复选按钮组

复选按钮组也可用作复选的水平按钮组，可以同时选择多个按钮。比如说文字编辑器的粗体,斜体,下划线。只需要为 `fieldset` 添加 `data-type="horizontal"` 属性即可。

```
<fieldset data-role="controlgroup" data-type="horizontal" data-role="fieldcontain">
```

Jquery Mobile 会自动将标签浮动，并排放置，并隐藏复选按钮的 `icon`，并给左右两边的按钮增加圆角。

Refreshing a checkbox

复选按钮组也可用作复选的水平按钮组，可以同时选择多个按钮。比如说文字编辑器的粗体,斜体,下划线。只需要为 `fieldset` 添加 `data-type="horizontal"` 属性即可。

列表表单

选择菜单

选择菜单摒弃了原生的 `select` 元素的样式，原生的 `select` 元素被隐藏，并被一个由 `jquery mobile` 框架自定义样式的按钮和菜单替代。菜单是 **ARIA** 的（即 **Accessible Rich Internet Applications**）并且桌面电脑的键盘也是可访问的。

当被点击时，手机自带的原生的菜单选择器会打开。菜单内某个值被选中后，自定义的选择按钮的值更新为你选择的那一个。

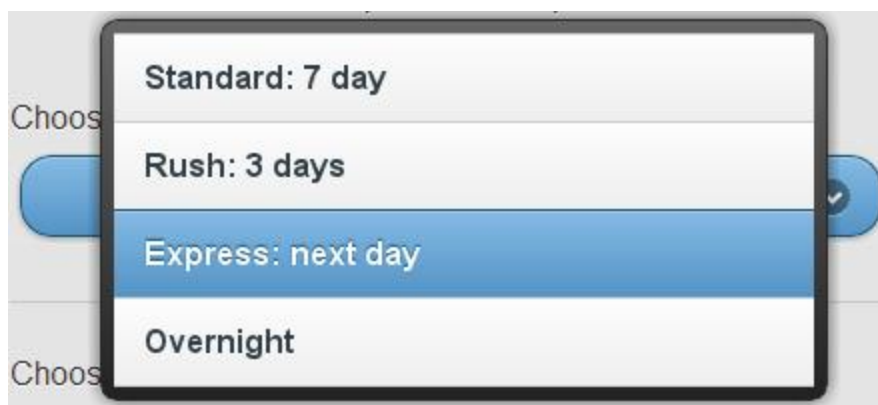
要添加这样的选择菜单组件，使用标准的 `select` 元素和位于其内的一组 `option` 元素。注意要把 `label` 的 `for` 属性设为 `select` 的 `id` 值，使 他们能够在语义上相关联。把它们包裹在 `data-role="fieldcontain"` 的 `div` 里面进行分组。

框架会自动找到所有的 `select` 元素并自动把他们增强为自定义的选择菜单。

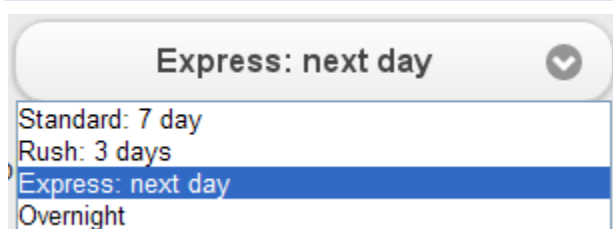
选择菜单 **Select menus**

```
<div data-role="fieldcontain">
  <label for="select-choice-1" class="select">Choose shipping method:</label>
  <select name="select-choice-1" id="select-choice-1">
    <option value="standard">Standard: 7 day</option>
    <option value="rush">Rush: 3 days</option>
    <option value="express">Express: next day</option>
    <option value="overnight">Overnight</option>
  </select>
</div>
```

选择菜单 **Select menus**



就会形成这样的列表菜单



注意：原生的选择菜单不能通过 `multiple="multiple"` 属性进行多项选择，如果你有这个需求，我们建议使用自定义菜单。

使用自定义菜单的选项

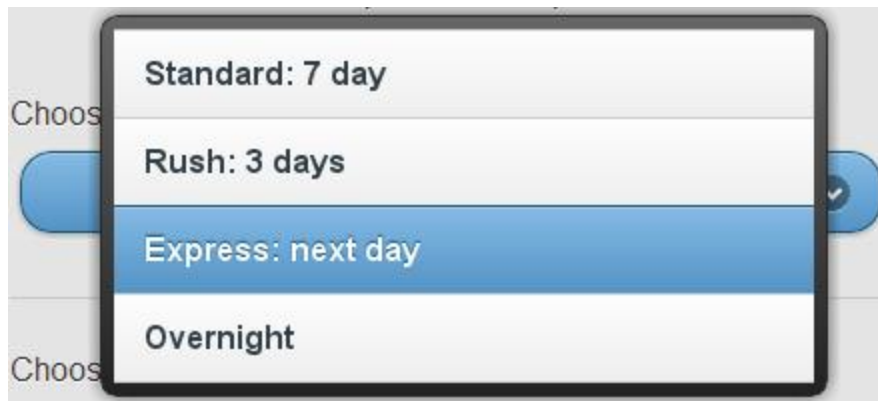
你可以自己选择使用自定义样式的选择菜单，而不是原生的。这样可以给菜单定义主题，使你的网站在视觉和体验上更加出色，而且在跨平台时也能够保持一致性。并且，他提供了对菜单的多项选择，并且修复了某些平台缺失的功能，比如 **Android** 对于 **optgroup** 的支持，并且增加了一种优雅的方式来处理占位符值。（下面会解释）。最后，自定义的菜单在桌面电脑浏览器上视觉上会更出色，因为原生的菜单在桌面电脑上看上去比移动设备上要小一些，那如果你的项目主要面对桌面用户的话，那视觉上就有点奇怪了。

注意，把原生菜单解析为自定义菜单的话会有性能开销。如果你页面里面有很多选择菜单，，或者某个菜单有很多选项，就会影响页面性能，所以我们的建议是不要滥用自定义菜单。

给菜单添加 `data-native-menu="false"` 属性，就可以把菜单转为自定义菜单。或者，在页面的 `mobileinit` 事件的回调函数中将选择菜单的 `nativeMenu` 设为 `false` 也可以达到相同的效果。这将会在全局把所有的选择菜单都默认设为自定义菜单。以下代码必须在 jQuery 加载后且 jQuery Mobile 加载前添加：

```
$(document).bind('mobileinit',function(){  
    $.mobile.selectmenu.prototype.options.nativeMenu = false;  
});
```

如果菜单的选项不多的话，会以小弹出框的形式和动画出现。



如果选择菜单太多的选项而不能显示在屏幕中，框架会自动创建一个新 `"page"`，表现为一个标准的列表，包含了所有的选项。使我们能通过设备原生的滚动条在很长的一个列表内选择所有的选项。页面的标题就是 `label` 的文本。

占位符选项

- 没有值的 **option**（或者一个空值属性）
- 没有文本节点的 **option**
- 带有 **data-placeholder="true"** 属性的 **option**。（使你将带有值

以及文本节点的选项用来作为占位符。）

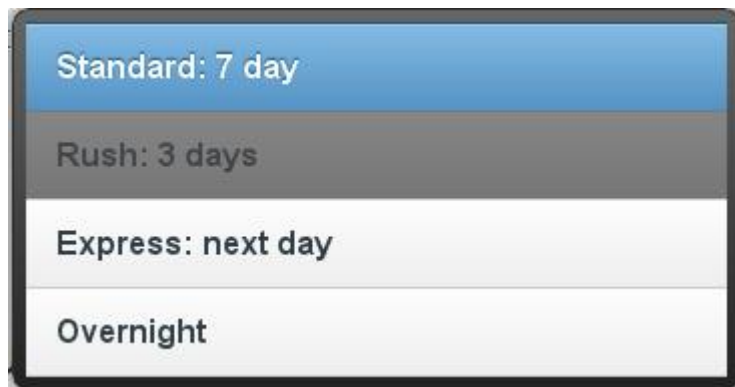
你可以通过插件的 **hidePlaceholderMenuItems** 选项禁用该特性，方法如下：

```
$.mobile.selectmenu.prototype.options.hidePlaceholderMenuItems = false;
```

禁用选项

jQuery Mobile 会把带有 **disabled** 属性的选项禁用并且在样式上标记。

在下面的实例中，第二个选项 "Rush: 3 days" 被设置为禁用：



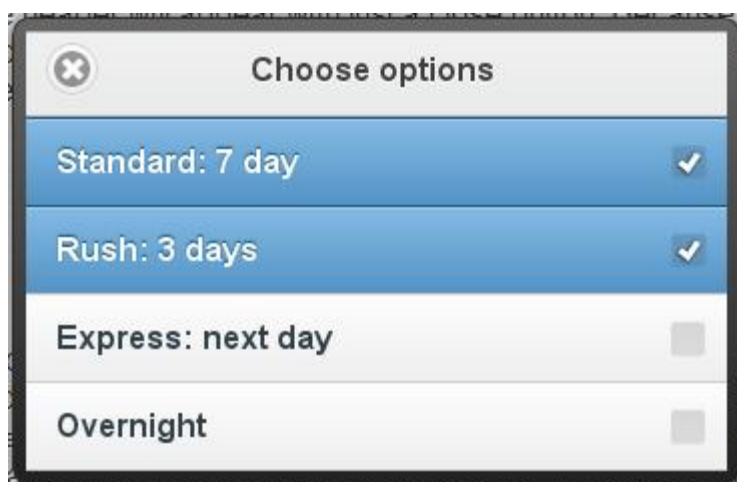
对于选项组的支持

如果一个选择菜单包含 **optgroup** 元素，jQuery Mobile 会根据 **label** 的文本创建一个分割项的选项：

多项选择

如果给你的选单加入 **multiple** 属性，jQuery Mobile 通过一系列的方式来增强元素：

- 菜单里面会创建一个 **header** 元素并在占位符文本旁边显示一个关闭按钮。
- 点击选项时不会关闭弹出菜单。
- 选项旁边会出现类似 **checkbox** 的图标。
- 超过 2 项被选择时会在选择按钮里出现一个表示当前选项数目的气泡。
- 选择的各项文本会在按钮里变成一行用逗号分隔的文本。如果按钮不够宽，多余的部分则会变成省略号。
- 如果没有选项被选择，按钮文本会保持占位符的文本
- 如果占位符元素不存在，默认按钮文本会是空白的，**header** 也仅仅显示一个关闭按钮。因为这不是友好的用户体验， 我们建议你在使用多选菜单时总是指定一个占位符元素。。



当选项太多以至于菜单会在一个新的页面出现时，占位符文本

会在没有选项是显示在按钮中，**label** 文本会在菜单的 **header** 中显示。这与小对话框式的选择菜单不同的，小对话框式的选择菜单的占位符文本会同时显示在按钮和 **header** 中。

Data 属性的支持

你可以对 **select** 指定 **jqmobile** 的针对 **button** 的 **data-** 属性。
例如：我们给下面的例子添加了 **theme, icon and inline** 属性。

刷新选择菜单

如果你通过 **js** 操作了选择菜单，务必通过 **refresh** 方法来更新他的样式。

```
var myselect = $("#select#foo");  
myselect[0].selectedIndex = 3;  
myselect.selectmenu("refresh");
```

表单主题样式

Jquery Mobile 内建了一套样式主题系统，让你给页面和表单添加样式时有丰富的选择。默认情况下所有的表单元素都会应用与父容器相同的主题样式。是的表单元素融合进他们的布局中。给表单元素单独地应用 **data-theme** 属性，可以指定它的主题样式，使他在布局中凸现出

下面实例中的所有表单的 **html** 都是相同的，并且没有为表单内的元素单独地指定主题样式。表单之间唯一的不同就是给表单父容器通过 **data-theme** 属性指定了不同的主题样式。这表现了表单元素自动应用父容器的主题样式的特性

BODY 为 A 主题

Text Input:

Flip switch: ☐ Off

Slider:

Font styling: ☒ b ☐ i ☐ u

Choose a pet:

- ☐ Cat
- ☒ Dog
- ☐ Hamster
- ☐ Lizard

Choose shipping method:

BODY 为 B 主题

BODY 为 C 主题

Text Input:

Flip switch: ☐ Off

Slider:

Font styling: ☒ b ☐ i ☐ u

Choose a pet:

- ☐ Cat
- ☒ Dog
- ☐ Hamster
- ☐ Lizard

Choose shipping method:

Standard: 7 day

BODY 为 D 主题

Text Input:

Flip switch: ☐ Off

Slider:

Font styling:

Choose a pet:

- ☐ Cat
- ☒ Dog
- ☐ Hamster
- ☐ Lizard

Choose shipping method:

BODY 为 E 主题

表单的提交

ajax 的表单提交

Jquery Mobile 会自动通过 **ajax** 处理表单的提交，并在表单页面和结果页面之间创建一个平滑的转场效果。注意请在 **form** 元素上正确设定 **action** 和 **method** 属性，保证表单的提交。如果没有指定，提交方法默认为 **get**, **action** 默认为当前页的相对路径（通过

`$.mobile.path.get()`方法取得

表单也可以像链接一样指定转场效果的属性，比如 `data-transition="pop"`，和 `data-direction="reverse"`。如果不希望通过 `ajax` 提交表单，可以在全局事件禁用 `ajax`，或给 `form` 设定 `data-ajax="false"` 属性。`target` 属性也可以在 `form` 上设置，表单提交时默认为浏览器的打开规则。。而与链接不同，`rel` 属性不可以在 `form` 上设

非 `ajax` 的处理

要防止表单提交时自动通过 `ajax` 提交，给 `form` 元素增加 `data-ajax="false"` 属性。也可以通过 `ajaxFormsEnabled` 方法在全局控制。

Simple Ajax form example

本页面展示了表单自动通过 `ajax` 处理的提交。下面的表单采用常规（`get` 方式提交）`forms-sample-response.php`。在提交时，`Jquery Mobile` 会确保指定的 `Url` 地址可以通过 `Ajax` 获取，并且适当的处理。请注意，就像普通的 `HTTP` 请求表单提交一样，`Jquery Mobile` 允许通过 `get` 请求成功得到的结果页通过更新 `Url` 的哈希值被添加为收藏页。而就像普通的表单提交一样，`post` 请求不包含查询参数，因此结果页不能被添加为收藏页

列表

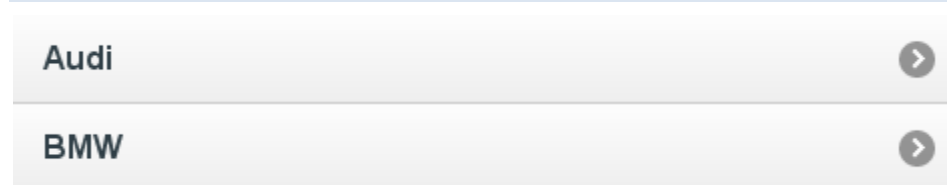
1.基本的带连接的列表

列表的代码为一个含 `data-role="listview"` 属性无序列表 `ul`。
Jquery Mobile 会把所有必要的样式(列表项右出现一个向右箭头，并使列表与屏幕同宽等)应用在列表上，使其成为易于触摸的控件。
当你点击列表项时，**Jquery Mobile** 会触发该列表项里的第一个链接，通过 **ajax** 请求链接的 **URL** 地址，在 **DOM** 中创建一个新的页面并产生页面转场效果。

html 代码

```
<ul data-role="listview">
  <li><a href="index.html">Acura</a></li>
  <li><a href="index.html">Audi</a></li>
</ul>
```

效果图：



2.嵌套的列列表

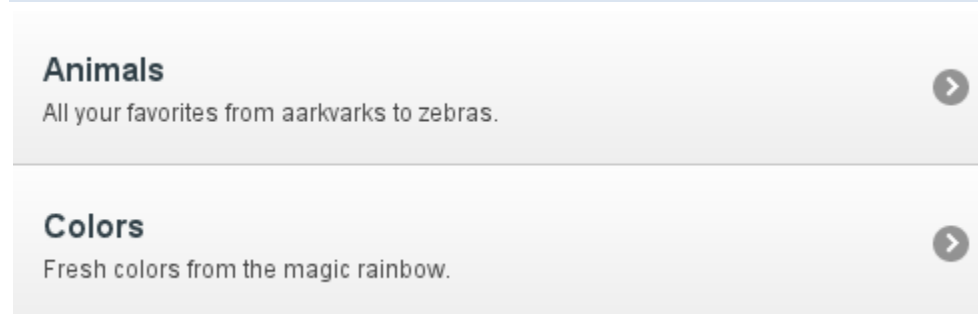
通过“`ul`”“`ol`”的 `li` 中嵌套子 `ul`，`ol` 你可以创建嵌套列表。当一个拥有子列表的列表项被点击时，**Jquery Mobile** 框架会生成一个新的 `ul` 页面充满屏幕，并自动生成一个为父列表项名称为标题的 **header**，以及一个子列表。这个动态生成的嵌套的列表默认的主题

样式“b”(蓝色),提示你是在二级菜单里。列表可以嵌套多层,Jquery Mobile 会自动处理这些链接和页面。

html 代码

```
<ul>
  <li>Pets
    <ul>
      <li><a href="index.html">Canary</a></li>
      <li><a href="index.html">Cat</a></li>
    </ul>
  </li>
  <li>Farm animals
    <ul>
      <li><a href="index.html">Chicken</a></li>
      <li><a href="index.html">Cow</a></li>
    </ul>
  </li>
</ul>
```

效果图:



3.数字排序的列表

通过有序列表 `ol` 可以创建数字排序的列表用来表现顺序序列比如说搜索结果或者电影排行榜时非常有用。当增强效果应用到列表时,Jquery Mobile 优先使用 `css` 的方式给列表添加编号,当浏览器不支持这种方式时,框架会采用 `JavaScript` 将编号写入列表中。

html 代码

```
<ol data-role="listview">
```

```
<li><a href="index.html">The Godfather</a></li>
<li><a href="index.html">Inception</a></li>
</ol>
```

效果图：

4.只读列表

列表也可以用来展示没有交互的条目，通常会是一个内嵌的列表。通过有序或者无序列表都可以创建只读列表，列表项内没有链接即可，Jqmobil 默认将他们的主题样式为“c”白色无渐变色，并把字号设为比可点击的列表项的小，以节省空间。

html 代码

```
<ul data-role="listview" data-inset="true">
  <li>Acura</li>
  <li>Audi</li>
</ul>
```

效果图：



5.拆分的按钮列表

有时每个列表项会有多于一个操作，这时拆分按钮用来提供 2 个独立的可点击的部分：列表项本身和列表项右边的小 icon。要创建这种拆分按钮，在 li 插入第二个链接即可，框架会创建一个竖直的分割线，并把链接样式化为一个只有 icon 的按钮，记得设置

title 属性以保证可访问性。

你可以通过指定 **data-split-icon** 属性来设置位于右边的分隔项的图标（图标详情参见 图标?分隔项的主题样式可以通过 **data-split-theme** 属性来设置。

html 代码

```
<ul data-role="listview" data-split-icon="gear" data-split-theme="d">
  <li>
    
    <h3><a href="index.html">Broken Bells</a></h3>
    <p>Broken Bells</p>
    <a href="lists-split-purchase.html" data-rel="dialog" data-transition="slideup">Purchase album</a>
  </li>
  <li>
    
    <h3><a href="index.html">Warning</a></h3>
    <p>Hot Chip</p>
    <a href="lists-split-purchase.html" data-rel="dialog" data-transition="slideup">Purchase album</a>
  </li>
</ul>
```

效果图：

6.列表分割项

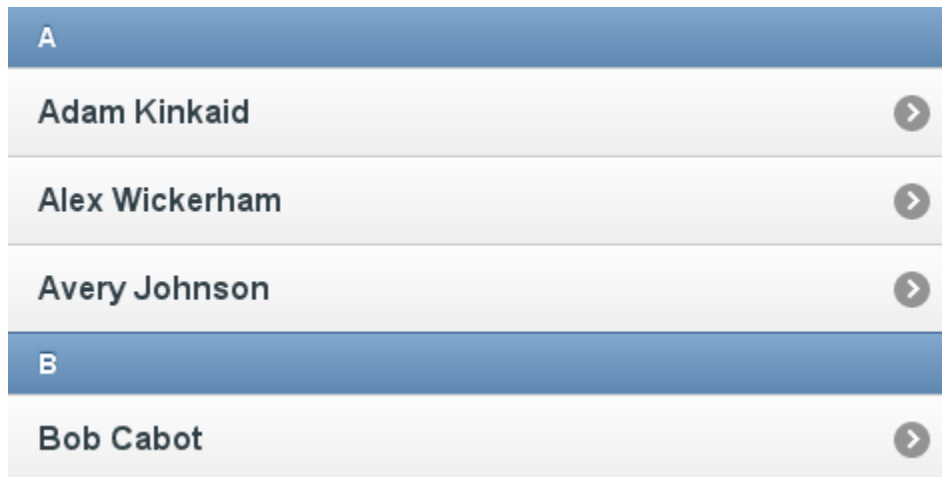
列表项也可以转化为列表分割项，用来组织列表，使列表项成组。给任意列表项添加 **data-role="list-divider"** 属性即可。默认情况下列表项的主题样式为“b”浅灰)，但给列表（**ul** 或 **ol**）添加 **data-dividertHEME** 属性可以设置列表分割项的主题样式。

html 代码

```
<ul data-role="listview">
  <li data-role="list-divider">A</li>
  <li><a href="index.html">Adam Kinkaid</a></li>
```

```
<li><a href="index.html">Alex Wickerham</a></li>
<li><a href="index.html">Avery Johnson</a></li>
<li data-role="list-divider">B</li>
<li><a href="index.html">Bob Cabot</a></li>
</ul>
```

效果图：



7.搜索过滤框

Jquery Mobile 提供了一种非常方便的方式通过在客户端进行的搜索机制过滤列表。要使一个列表可过滤，只需为列表设置 `data-filter="true"` 属性即可。框架会在列表上方增加一个搜索框，当用户在搜索输入框中输入时，Jquery Mobile 会自动过滤掉不含输入字符的列表项。

搜索输入框默认的字符为 "Filter items...".通过设置 `mobileinit` 事件 的 绑 定 程 序 或 者 给 `$.mobile.listview.prototype.options.filterPlaceholder` 选项设置一个字符串，或者给列表设置 `data-filter-placeholder` 属性，可以设置搜索输入框的默认字符。

html 代码

```
<ul data-role="listview" data-filter="true" >
  <li><a href="index.html">Acura</a></li>
  <li><a href="index.html">Audi</a></li>
</ul>
```

效果图：

8.文本格式和计数气泡

Jquery Mobile 支持通过 HTML 语义化的标签来显示列表项中所需常见的文本格式（比如 标题/描述，二级信息，计数等）

- 将数字用一个元素包裹，并添加 **ui-li-count** 的 **class**,放置于列表项内，可以给列表项右侧增加一个计数气泡

；

- 要添加有层次关系的文本可以使用标题来强调，用段落文本来减少强调

；

- 补充信息（比如日期）可以通过包裹在 **class="ui-li-aside"**的容器中来添加到列表项的右侧

；

html 代码

```
<ul data-role="listview">
  <li data-role="list-divider">Friday, October 8, 2010 <span class="ui-li-count">2</span></li>
  <li>
    <h3><a href="index.html">Stephen Weber</a></h3>
    <p><strong>You've been invited to a meeting at Filament Group in Boston, MA</strong></p>
    <p>Hey Stephen, if you're available at 10am tomorrow, we've got a meeting with the JQuery
  </li>
</ul>
```

team.</p>

Sylar-2012-4-27