

Catch (Exception .net)

{ /* 讲求理性观点的工程师社区 */ }

jQuery Mobile

开发跨平台移动应用

最新的 jQuery Mobile 框架
基于 HTML5
兼容几十种最流行移动平台
内容同步更新发布

陆明 编著



前言

jQuery Mobile 并不是一个特别高深的学问，却是有用而富于技巧性的一种技术。通过这种技术可以快速开发出来能在 **iPhone, iPad, Android, Windows Phone** 等多种平台上使用的移动应用。在工程师和创业成本越发成为紧俏，网络基础设施越来越便宜的今天，这可能是一种值得探讨与尝试的技术方向。

我并没有特别才华适合在某个艰深而富有智慧的领域将知识与经验来奉献给和我一样年轻的工程师们。不是不想，而是深知自己知识和能力的浅薄。所以选择一些有趣而或许有用的技术，将自己粗陋的理解和实践也杂糅其中。希望能以此砖头引得众多智慧头脑过来批判指导，对内容的批判也是对我能力的提升。

当我起笔到草稿完成的几周时间，我注意到 **jQuery Mobile** 已经在西方有了很多印刷出版物，并在相关技术领域图书排行榜中处于 **Top** 排行榜中。西方移动应用工程师们也正在对这个技术展开热烈讨论。国内却也有了很多专门的技术博客文章和论坛，就其中使用方法和应用场景进行专业的讨论。只是，系统化的整理和使用中文将这个技术介绍出来，似乎还不多见到。作为一种新产生的技术，我想到十几年前当我开始学习 **Delphi 1.0** 和 **Microsoft C++** 时候的场景，那个时候 **Visual Studio** 还没有现在这么发达，技术资料也很少。作为年轻人的我，如果能有一些可用而易用的技术资料，或许能节省我很多摸索的时间。也基于此段经历，我用一些笨功夫将 **jQuery Mobile** 技术中自认为可能会比较有价值的信息整理出来，写成这个东西。如果能对于国内移动应用开发或者移动 **Web** 应用社区能带来一点些许帮助，或者对于年轻的工程师们的学习能节省些许时间，于我就是一件很开心的事情了。

写作期间，我曾经在四姑娘山脚下的日隆镇小住。

一天，我骑马近 10 个小时，高原反应和颠簸让我发烧而浑身酸痛。在藏族本地人淳朴、善良和辛劳的帮助之下，我才有幸深入期间欣赏山中景色。不管冷杉林，红杉林或者高原草甸，堰塞湖，都是震撼我既往的经验。这段经历给我很大启示，扩大心胸，放眼长远，平和坚韧，脚踏实地，经历崎岖艰难却能欣赏别一番景色。封面照片摄影于其中一天清晨，太阳尚未升起，白塔旁边群山之上一个孤独的红衣登山者已经在向上攀登。而在雪山之上还有两处登山者的营地，一些人已经在前一天感到与此幕天席地享受雪山高原。这些地方都在大概海拔 4000 米上下。他们的坚持与坚毅，是每次我看到此照片，对我的鼓励。

将这样照片作为准备进入移动应用开发新领域的工程师们共同分享。

jQuery Mobile 是一个发展很快的技术。过去这几个月，几乎每个月都有发展。到今天已经到了 1.0 RC2 的版本，正式版本呼之欲出。

相信未来还会有持续的更新。这本电子书也会在未来不断发展和完善。如果您有对于这部分内容的理解和批判，也希望能指点给我。我很愿意将有价值的经验融入到未来的内容中，以丰富和完善，并给未来工程师们更多的有价值的资料。

CatchException.NET 创始人

陆明

2011 年 11 月 06 日

小经验

CatchException.NET 是一个旨在帮助工程师提升技术能力，分享专业知识，建立专业领域人脉与口碑的技术社区。

您在技术开发与实践领域，如果有所体验、感悟、探讨、思辨可以通过这个平台与工程师们进行互动交流。

CatchException.NET 也会经常举办线下技术沙龙和交流活动。您可以将感兴趣的技术沙龙话题发送到 webmaster@catchexception.net 于我们分享。

目录

第 1 章 jQuery Mobile 概述	1
1.1. 本地应用还是 Web 应用	1
1.2. jQuery Mobile 兼容的移动平台	3
1.3. 其他的 Web 应用开发框架	4
1.3.1. Sencha Touch	4
1.3.2. iUI	5
1.3.3. jQTouch	5
1.3.4. iWebKit	5
1.4. 不适合的场景	6
第 2 章 框架初探	7
2.1. jQuery Mobile 框架组成	7
2.2. 第一个程序	8
2.2.1. 开发前的准备	8
2.2.2. Hello World!	9
2.3. jQuery Mobile 与 HTML5	12
第 3 章 页面与对话框	14
3.1. 单页与多页模版	14
3.2. 页面标题	18
3.3. 链接到内部页面或外部页面	19
3.4. 对话框建立, 关闭	21
3.5. 切换方式美化	23
3.6. 页面风格美化	24

第 4 章 页面布局与内容呈现	28
4.1. 基本页面布局	28
4.2. 分栏与网格布局	33
4.3. 集成自定义 CSS	40
第 5 章 按钮	41
5.1. jQuery Mobile 的按钮	41
5.2. 内置按钮样式	43
5.3. 按钮图标	44
5.4. 按钮组	46
5.5. 选项、方法和事件	47
5.5.1. 选项	48
5.5.2. 方法	49
5.5.3. 事件	49
第 6 章 导航工具栏	51
6.1. 导航工具栏	51
6.2. 头部工具栏	52
6.3. 脚注工具栏	54
6.4. 导航工具条	57
6.5. 导航栏位置	60
6.6. 导航内容高亮	60
第 7 章 列表视图	62
7.1. 基本列表	62
7.2. 嵌套列表	64
7.3. 数字列表	68
7.4. 分立按钮列表	70
7.5. 缩略图列表	72
7.6. 列表图标与气泡提示	73

7.7. 检索列表内容	76
7.8. 列表性能测试	78
第 8 章 表单	79
8.1. HTML5 表单	79
8.2. 表单提交	80
8.3. jQuery Mobile 表单样式	80
8.4. 表单风格	81
8.5. 浏览器原生样式	82
8.6. 输入框	82
8.7. 开关按钮	83
8.8. 无线按钮	84
8.9. 复选按钮	86
8.10. 滑动条	87
8.11. 选择菜单	88
8.12. 表单元素禁用	90
8.13. 隐藏标签	92
8.14. 选项、方法和事件	94
8.14.1. 选项	95
8.14.2. 方法	95
8.14.3. 事件	95
附录 A. 答疑资源	97
文档	97
问答	97
沙龙	97
讲座	98
附录 B. 升级计划	99

第 1 章 jQuery Mobile 概述

jQuery Mobile 是一套基于 jQuery 的移动应用开发框架。开发工程师可以将应用托管在 Web 服务器上，智能手机或者平板电脑访问使用 jQuery Mobile 所开发的应用的时候，尽管是基于浏览器访问，从用户界面和操作体验上却和通常的本地应用程序非常接近。

jQuery Mobile 开发者投入很大精力在支持更多的兼容的移动设备浏览器，良好的运行速度与统一而灵活的用户体验。一处开发，随处运行，是这种移动应用开发框架的共同特点。

在这一章我们将会了解，

- jQuery 的基本概念
- jQuery, jQuery Mobile 和 jQuery UI 之间的关系。
- 使用 Web 应用方式开发移动应用的原因和好处
- jQuery Mobile 对于不同移动设备的支持情况
- 与其他移动应用开发框架比较，jQuery Mobile 的优势
- 不适合 Web 应用的场景

在使用 jQuery Mobile 之前，很多开发工程师可能已经使用 jQuery 或 jQuery UI 进行网站应用开发。jQuery Mobile 与 jQuery UI 都是基于 jQuery 框架提供相应的界面和解决方案。不同的是，jQuery Mobile 是面向移动平台，而 jQuery UI 是面向桌面应用的。虽然一些 jQuery UI 应用也可以运行于 Android 或者 iPhone 手机上，但 jQuery Mobile 从界面到操作方式更多面向移动应用进行优化，所以用户体验更好。

通常熟悉 jQuery 或 jQuery UI 的工程师可以非常方便的基于已有经验而将应用迁移到 jQuery Mobile 上来。值得指出的是，在桌面 Web 应用开发 JavaScript 库中，jQuery 是当前市场占有率最高的框架。可以预见，大量软件工程师将可以通过 jQuery Mobile 迅速开发高质量跨平台移动应用。

1.1. 本地应用还是 Web 应用

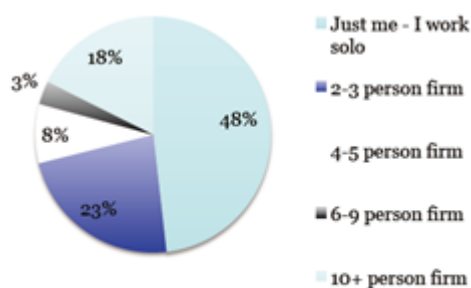
Web 应用开发是一种有趣的尝试。软件开发团队将不需要在不同移动操作系统和移动设备之间为能兼容而重复进行开发。例如，在 Apple 环境下使用 iOS 的应用，在 Android 平台下使用 Java 又重复开发一遍，而这些应用如果需要运行在 Windows Phone 环境下，则又需要

使用 C# 或者 EVC 再开发一遍。

传统移动应用开发中，即便在一种操作系统之下，不同的应用的兼容性也很难保证，例如，一个 Android 应用在 HTC G7 手机下开发完成，在部署到 G6 手机上的时候，还需要根据不同分辨率进行调试。当然，这种情况在 Web 应用开发过程中，虽然也会发生，却相对简单。例如在 jQuery Mobile 开发中，可以通过定义 CSS3 的 Media Queries 进行不同分辨率下的界面设置。

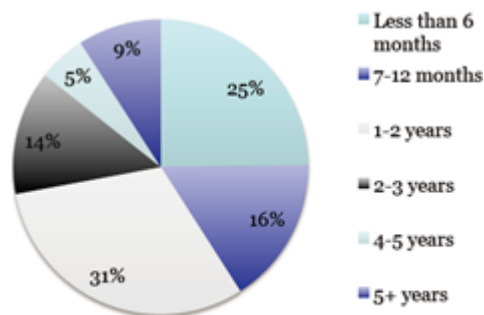
在最近 Distimo 所进行的移动产业市场调查中，可以看到这样的现象，

- 大部分移动应用开发的技术团队工程师数量在 10 人之内。



N=284

- 大部分技术团队从业人员的工作经验少于 5 年。



N=286

这样的创业团队可能也意味着，创业资金不足，既有移动应用开发沉淀比较有限。那么一种低成本的应用开发和快速交付方式，对于移动平台创业团队就具有明显的价值。设想一个移动应用需要覆盖 iPhone, Android 和 Windows Phone 三个平台。则开发团队需要在 Windows 和 Mac 操作系统下分别搭建开发环境并开展开发活动，还需要独立的硬件测试环境来进行应用测试。

假如通过 Web 应用开发的方式，则会简单很多。一处开发，随处运行。

1.2. jQuery Mobile 兼容的移动平台

jQuery Mobile 对于移动应用平台的支持，分为四种不同级别，

- A 级
支持全部界面特效，包括基于 Ajax 的页面动画切换效果。
- B 级
支持用户界面特效，但是不支持基于 Ajax 的页面动画切换效果。
- C 级
能运行，也可以支持基本的 HTML 效果。但是比较高级的 HTML 特效就不能很好支持。
- D 级
可能运行，但是并不能得到 jQuery Mobile 官方支持。

通常我们希望 jQuery Mobile 可以支持目标客户人群的移动设备在 A 级。

各种移动设备支持 jQuery Mobile RC1 版本情况如下，

- A 级
 - Apple iOS 3.2-5.0 beta
 - Android 2.1-2.3
 - Android 3.0 Honeycomb
 - Windows Phone 7
 - Blackberry 6.0
 - Blackberry 7
 - Blackberry Playbook
 - Palm WebOS (1.4-2.0)
 - Palm WebOS 3.0
 - Firebox Mobile (Beta)
 - Opera Mobile 11.0
 - Meego 1.2
 - Kindle 3
 - Chrome Desktop 11-13
 - Firefox Desktop 3.6-4.0
 - Internet Explorer 7-9
 - Opera Desktop 10-11
- B 级
 - Blackberry 5.0

小经验

开发移动应用却没有足够的主流移动测试平台怎么办？如果你恰巧在北京，可以选择在车库咖啡厅租用他们的手机进行测试。车库咖啡是一个咖啡厅，也是一个为早期创业者提供的开放式办公环境。你在这里买一杯咖啡就可以工作一整天。他们提供大多数主流手机以进行测试，每小时的租金只有几块钱。

如果要过去，最好提前电话确认，这里经常会举办会议和沙龙活动。

车库咖啡

地址：北京市海淀区海淀图书城步行街鼎鑫宾馆2层。

电话：010-82627127

- Opera Mini (5.0-6.0)
- Windows Phone 6.5
- Nokia Symbian^3
- C 级
 - Blackberry 4.x
 - 其他更早版本的智能手机
- D 级
 - Samsung Bada

从上面的清单可以看到，当前主流的移动平台 Apple iOS 3.2+，Android 2.1+，Windows Phone 7 和 Blackberry 7 都对于 jQuery Mobile 具有 A 级良好的兼容性支持。尽管如此，对于大多数应用在面对客户进行分发和部署之前，建议还是对主流目标用户的移动平台进行兼容性测试，以保证应用部署和运行正常。

1.3. 其他的 Web 应用开发框架

1.3.1. Sencha Touch

Sencha Touch 是 jQuery Mobile 之外的另一个流行的移动平台开发框架。

Sencha Touch 继承了 ExtJS 的血统，可以看成 ExtJS 在移动平台上的实现。工程师可以通过统一的 Sencha Touch 框架实现跨平台移动应用开发。

Sencha Touch 具有比较明显的优势，

- 界面样式标准、美观，开发团队可以集中于应用功能实现，而对于美工方面的依赖较小。
- 具有 ExtJS 环境下开发经验的工程师，学习这种框架比较容易，并易于将桌面应用迁移到 Sencha Touch 框架之下。
- 对于能够支持的移动平台，实现动画，Ajax，界面风格比较统一。
- 基于 Sencha Touch 的 Sencha Chart 可以提供丰富的图表制作能力，并能在移动平台上良好运行。这对于企业信息化厂商具有很大吸引力。

同样，Sencha Touch 也继承了 ExtJS 的局限性，

- 在封闭的 JavaScript 框架下完成应用开发，虽然支持对于其他 JavaScript 的扩展，但是定制整体而言还是比较困难的。

- 全部 JavaScript 开发，调试成本比较高。搜索引擎友好性也比较有限。

另一个需要指出的局限性在于 Sencha Touch 支持的移动平台相对有限。所以如果要面对公众用户开展应用开发和部署，选择 Sencha Touch 就要相对谨慎。

相信未来国内很多企业信息系统开发商很可能会基于既有 Sencha ExtJS 沉淀，而将应用迁移到 Sencha Touch 的移动应用上来。

1.3.2. iUI

iUI 也是一种基于 HTML 的 Web 移动应用框架，包括，JavaScript, CSS, 图片。旨在为 iPhone 和与 iPhone 相兼容的设备开发移动应用。

iUI 的主要特性为，

- 基于标准的 HTML 代码建立具有 iPhone 风格的导航菜单。
- 很少甚至不需要 JavaScript 就可以建立基于 HTML 的应用界面。
- 美化过的移动应用的用户体验具有 iPhone 应用风格。
- 适应移动手机应用的界面设计要求。

1.3.3. jqTouch

jqTouch 也是一个基于 jQuery 的 Web 移动应用开发框架，大概在 2009 年前后诞生。对于 Android, iPhone 和基于 Webkit 核心的桌面浏览器支持较好。

只是 jqTouch 在纷繁的移动应用平台和移动浏览器框架的环境下，兼容性能力比较弱。所以并不建议支持 jqTouch 进行跨平台移动应用开发，除非本身就是面对特定用户群体的。

1.3.4. iWebKit

iWebKit 是一套通过 HTML 和 JavaScript 开发建立 iPhone 和 iPod Touch 移动应用的框架。

为何选择 jQuery Mobile

jQuery Mobile 很有可能成为未来跨平台移动应用的主流开发工具，尽管直到今天 jQuery Mobile 依然还是 Release Candidate 1 版本。

通过与其他跨平台移动应用比较，我们可以看到 jQuery Mobile 具有如下优势，

- 框架轻量，适合移动应用场景

- 优化的移动应用界面风格
- 移动设备和浏览器厂商支持广泛
- 从手机到平板电脑支持领域广泛
- jQuery 社区的开发工程师和技术支持资源丰富
- 框架开放，并允许进行扩展，如开发基于 jQuery Mobile 的第三方插件

1.4. 不适合的场景

也有一些应用场景并不适合于 jQuery Mobile，通常是与硬件集成相对紧密的功能。例如，要开发一个手机的手电筒功能，这个硬件设备就是 HTML 所无法控制的。

当然，大多数移动应用并不与硬件具有特别的交互，这些场景通常是适合 jQuery Mobile 的。而一些传统上与硬件设备需要进行交互的功能，例如 GPS，也因为移动设备的地理功能已经集成到 HTML5 之中，所以事实上在 jQuery Mobile 框架应用中使用并没有太大问题。

第 2 章 框架初探

通过 jQuery Mobile 开发 Web 移动应用和开发面向桌面的 Web 应用，其实差别并不大。对于当前的工程师而言，熟悉程序的框架结构，了解基本的 HTML5 程序样式，通常就能很快开发出来基本的移动应用程序来。

2.1. jQuery Mobile 框架组成

jQuery Mobile 是一套以 jQuery 为核心的 JavaScript 框架。和广泛应用的 jQuery UI 非常类似，jQuery Mobile 框架中也需要包含这样几个部分，

- jQuery JavaScript 文件
- jQuery Mobile JavaScript 文件
- jQuery Mobile 层叠样式单
- jQuery Mobile 图形资源库

在使用 jQuery Mobile 进行移动应用开发的时候，需要能够使页面访问到上面的资源。

最近的 jQuery Mobile 下载页面位于 <http://jquerymobile.com/download/>。开发者可以从这里下载 jQuery Mobile。在 jQuery Mobile 下载资源中并没有包括 jQuery JavaScript 库。下载 jQuery 可以访问，http://docs.jquery.com/Downloading_jQuery。

小经验

在 jQuery 和 jQuery Mobile 下载页面中会看到有 xxx.min.js 或者 xxx.min.css 文件。例如，jquery-1.6.4.min.js, jquery.mobile-1.0rc2.min.js, jquery.mobile-1.0rc2.min.js。同时也可以看到同样的目录下会存放不包含 .min 的文件。这个两种文件的区别在于包含 min 的文件对 JavaScript 或者 CSS 进行了压缩，删除了不需要的回车，空格等内容，以使得下载体积更小。

如果只是作为调用，而不是调试，则建议部署包含 .min 的文件。以提升移动设备访问内容的速度，减少流量消耗。

jQuery Mobile 推荐几种方法在页面中调用 jQuery 和 jQuery Mobile JavaScript，

1. 下载 JavaScript 和 CSS 并部署在服务器本地。
2. 在页面中访问 code.jquery.com 中托管的代码资源。

通常建议将 JavaScript 和 CSS 下载到本地服务器，页面调用本地服务器的资源。通常国内的移动用户连接国外互联网资源的速度都比较慢。如果将 JavaScript 和 CSS 部署在国内的服

务器，则可以很大程度加快移动设备的访问和呈现速度。

如果希望直接使用 jQuery 所托管的代码资源，则只需要在页面 head 标记中加入这一段话就可以了，

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0rc2/jquery.mobile-1.0rc2.min.css" />
<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script src="http://code.jquery.com/mobile/1.0rc2/jquery.mobile-1.0rc2.min.js"></script>
```

如果希望阅读其他 jQuery Mobile 资料可以参阅 jQuery Mobile 官方网站，
<http://jquerymobile.com/>

2.2. 第一个程序

如同所有软件开发教程一样，我们第一个范例也是 Hello world。

2.2.1. 开发前的准备

开发 jQuery Mobile 应用可以使用市场常见的，或者你所熟悉的 HTML 编辑工具，例如，记事本，EditPlus，Adobe Dreamweaver，Aptana Studio 等。对于初学者而言，建议使用支持 HTML5 的开发工具。

虽然说是移动应用，其核心其实是通过 HTML 页面实现的。所以调试 jQuery Mobile 应用的方式和调试桌面应用的方式基本上是一致的。工程师可以首先在支持 HTML5 的桌面浏览器上调试通过，然后再进行移动设备的兼容性测试。由于 HTML5 的支持已经比较标准化，通常兼容性测试是比较容易通过的。

在进行调试的时候，推荐使用安装 FireBug 插件的 Firefox 来帮助调试。关于

小经验

关于哪些 HTML5 即插即用编辑器的选择，可以参阅这篇文章《[有哪些 HTML5 编辑器比较方便使用?](#)》

FireBug 的安装方法，可以参考《[如何安装 FireBug 进行 HTML, JavaScript 和 CSS 调试?](#)》

尽管 jQuery Mobile 可以支持高版本的 Internet Explorer 浏览器，但是需要注意一些 Internet Explorer 的外壳程序将可能会影响 jQuery Mobile 正常使用。如果不想有更多麻烦，建议还是使用 Firefox 或者 Google Chrome 浏览器。

2.2.2. Hello World!

一个基本的 jQuery Mobile 应用程序结构如下所示。在这个程序中，包含这样几个部分，HTML 代码框架，jQuery 和 jQuery Mobile 框架 JavaScript 库。

基本的 jQuery Mobile 程序结构

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
</body>
</html>
```

在上面的框架中，包含这样几部分内容，

1. <!DOCTYPE html> 是 HTML5 新的标记定义，表示一个 HTML5 文档的开始。<!DOCTYPE html>，在任何使用 jQuery Mobile 实现的页面中，必须以此作为开始。
2. <html><title><body> 这些标记是继承自 HTML4 标准的，表示 HTML 文档的各个部分。
3. <head> 标记中包含将会被引用的 CSS 和 JavaScript，对于 jQuery 和 jQuery Mobile 的引用也定义在这里。
4. <meta charset=" utf-8" > 表示页面内容的字符集，用于多语言显示。

基于上面的代码框架，就可以填充内容到这个页面 body 之中了。

在填充页面内容的时候，通常是基于这样的框架，

jQuery Mobile 页面内容程序结构

```
<section id=" page1" data-role=" page" >
  <header data-role=" header" >
```

```
</header>  
<div class=" content" data-role=" content" >  
</div>  
<footer data-role=" footer" >  
</footer>  
</section>
```

在这段代码中，定义了这样几部分内容，

1. <section data-role=" page" >，这部分告诉 jQuery Mobile 框架，下面的内容将在一个页面中显示。
2. <header>，缺省将显示在浏览器上的标题栏。
3. <div class=" content" data-role=" content" >，表示其中内容为移动应用界面的正文部分。
4. <footer>，表示为移动应用界面中的页脚部分。

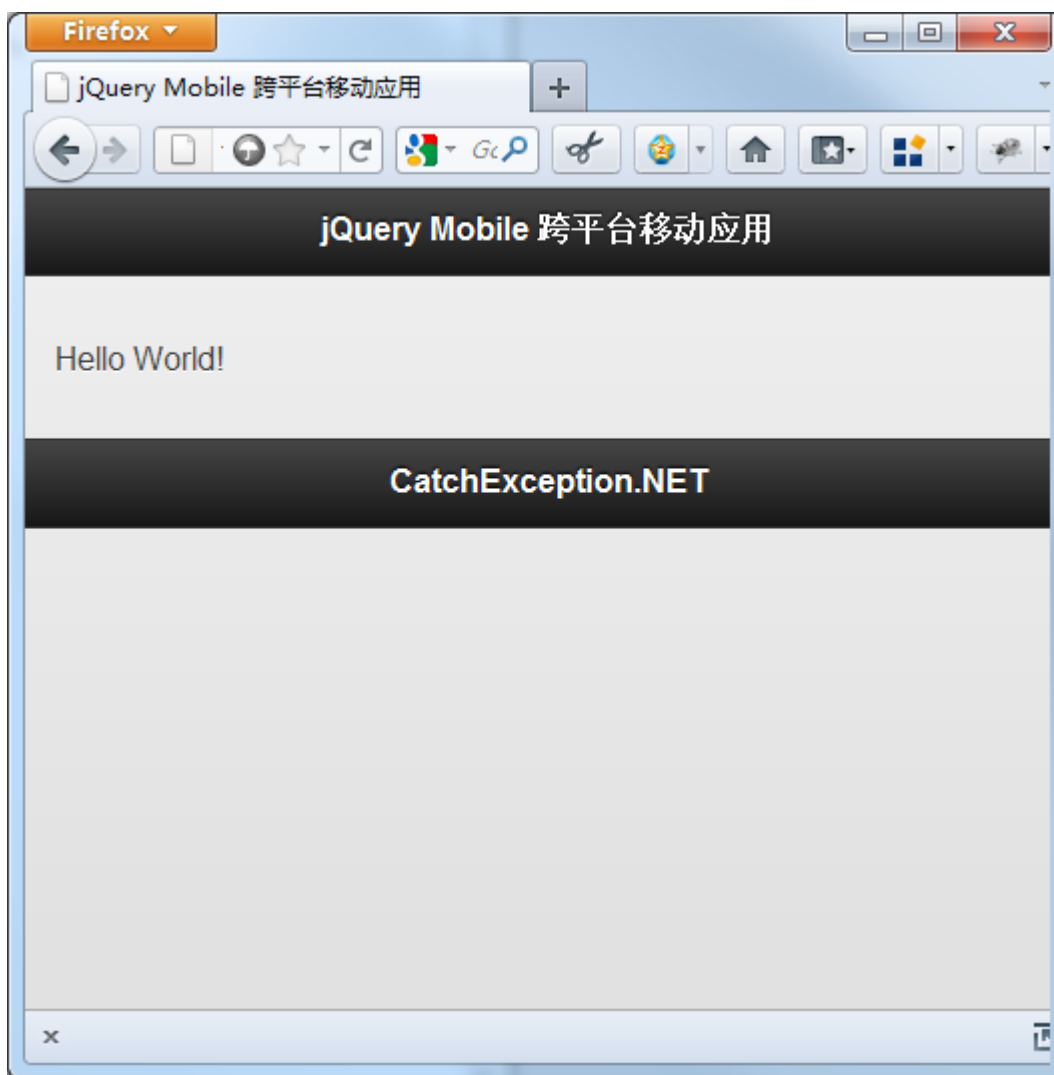
我们在页面内容程序中，填入所要展现的 Hello World! 内容。然后，将前面 HTML 代码以及 jQuery Mobile 库引用的代码合在一起，于是得到了下面的代码。

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset=" utf-8" >  
<title> 跨平台移动应用 </title>  
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css" />  
<script src=" ../js/jquery-1.6.4.min.js" ></script>  
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>  
<meta http-equiv=" Content-Type" content=" text/html; charset=utf-8" >  
</head>  
<body>  
<section id=" page1" data-role=" page" >  
<header data-role=" header" >  
<h1>jQuery Mobile 跨平台移动应用 </h1>  
</header>  
<div class=" content" data-role=" content" >
```

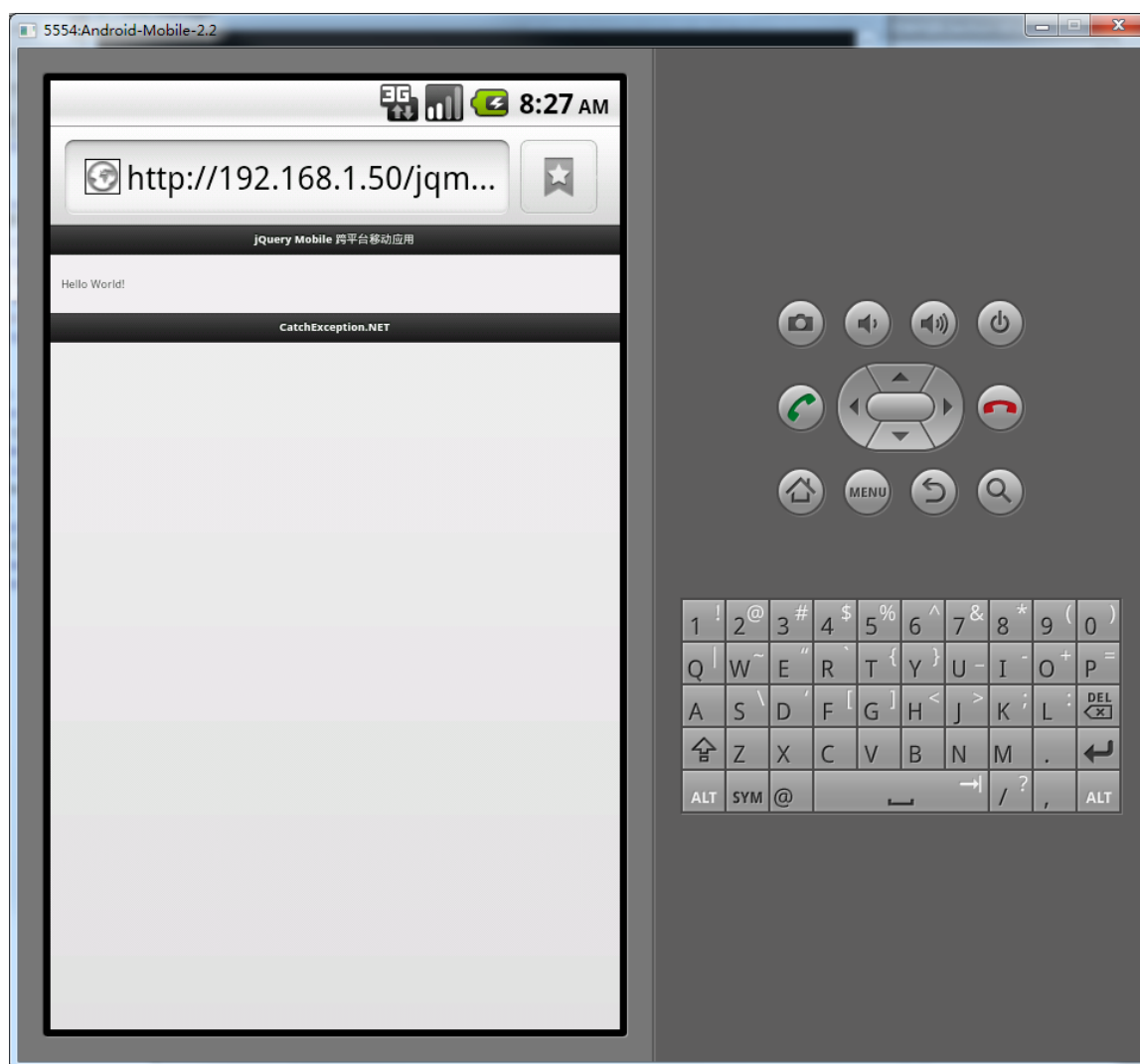


```
<p>Hello World!</p>  
</div>  
<footer data-role=" footer" >  
  <h1>CatchException.NET</h1>  
</footer>  
</section>  
</body>  
</html>
```

在 FireFox 7.0 中，界面呈现如下，



在 Android Mobile 2.2 模拟器中呈现界面效果如下，



和很多 Hello World! 程序一样，第一个程序总是带有那么一些兴奋。尽管并不完美，却是看到的第一个跨平台移动应用的样子。后面，我们将会一起看到更多通过 jQuery Mobile 实现移动应用开发的方法。

2.3. jQuery Mobile 与 HTML5

虽然早期范例代码和很多网站中，jQuery Mobile 是可以通过定义 DIV 的方式与 HTML4 页面集成。还是建议使用 HTML5 来完成 jQuery Mobile 框架。一来 HTML5 所书写的框架更加简洁，二来一些 HTML5 新特性对于移动应用开发非常有用。

这样的新特性包括，

- 简化而实用的语义标签。
- 增强的表单功能。
- 原生视频和音频支持
- 绘图 API
- Web Sockets API
- 离线 Web 应用
- Web Storage
- Web Worker 多线程
- 基于地理位置的 Geolocation API

很多应用将会对移动应用开发产生长远影响，例如传统的 HTML4 Cookie 只能保存 4KB 数据，而通过 Web Storage 将可以存储 5MB 甚至更多的数据。在基于 Geolocation API 的应用中，SoLoMo 应用的开发将不需要涉及复杂的基于 GPS 或移动基站的位置定位，而通过浏览器将可以大大简化之前的操作。

第 3 章 页面与对话框

jQuery Mobile 框架的页面就相当于传统本地移动应用程序的界面，了解页面和对话框开发技术，就可以很快定义基本的软件框架出来。

jQuery Mobile 的页面和对话框开发通过基于定义 HTML 和 CSS 实现，非常简单。基于这个框架开发应用，对于初学者甚至比传统的桌面 Web 应用开发还要更简单。

在这一章中，我们将接触到，

- 单页与多页模板
- 页面标题
- 链接到内部页面或外部页面
- 页面切换方式美化
- 对话框建立与关闭
- 页面风格美化

3.1. 单页与多页模版

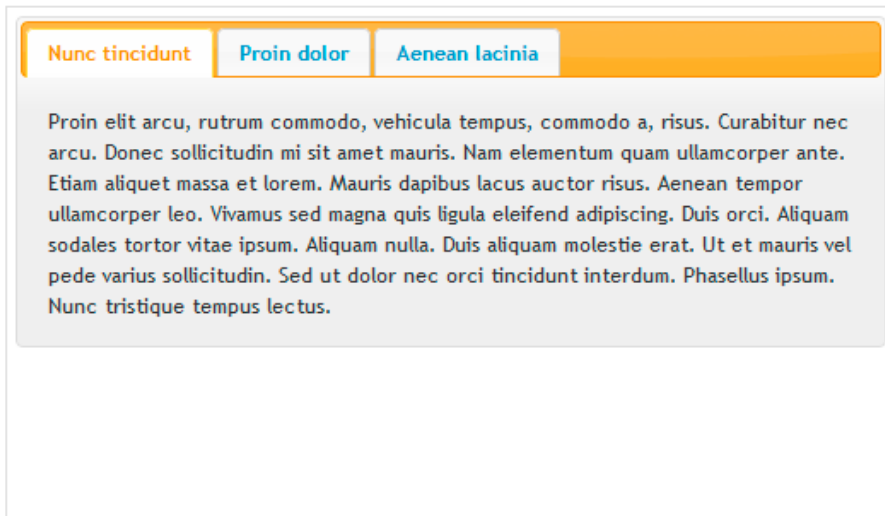
通常在面向桌面浏览器开发一个网页的时候，一个页面就是一个网站。好像我们在上一章所看到的 Hello World 程序，这是一种典型的单页模版页面。简单的说，单页模板是，一个应用界面就通过一个网页文件实现。

在一些页面中，例如一个网页中包括多个选项卡。通常为了节省用户打开各个选项卡的时间，网页设计师会在 CSS 方面下一些功夫，从界面看来是打开了新的选项卡，而实际上这个选项卡的内容已经从服务器下载到本地浏览器了。关于在桌面浏览器中显示多选项卡，可以参见 jQuery UI Tabs 组件，<http://jqueryui.com/demos/tabs/>。

有别于桌面计算机连接互联网，移动互联网当使用 3G 网络通过移动基站连接到服务器获得页面内容的时候，网络通讯环境会大大复杂，而移动网络性能是比较不稳定的。传输的速度收到手机连接到基站的速度，基站自身带宽品质，以及移动设备运行过程中从一个移动基站

Tabs

Theme: UI lightness



Examples

[Default functionality](#)[Content via Ajax](#)[Open on mouseover](#)[Collapse content](#)[Sortable](#)[Simple manipulation](#)[Tabs below content](#)[Cookie persistence](#)

Toggle sections open/closed on mouseover with the `event` option. The default value for event is "click."

```
event: 'mouseover'
```

切换到另一个移动基站过程中，对于网络性能的影响。这也是，为什么我们乘坐地铁的时候打开网页通常速度都会比较慢的原因。虽然通过 Wifi 能保证在一定范围内的通讯品质，但并不是随处都能有免费而具有加密保护的 Wifi 热点的。这样的环境下，移动 Web 应用就需要能够针对这样的网络环境进行专门优化设计。

在 jQuery Mobile 框架下，多页模版在加快小网站或具有关联性页面的加载速度，改善用户体验的作用更进一步被突出。用户在下载第一个页面的时候，同时也下载了相关的其他页面。当用户打开这样的页面的时候，不再需要重新从服务器下载新的页面，而只需要将“隐藏”在背后的页面调出来就好了。

多页模板页面范例，

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
```

```
<body>
<!-- 这里是多页显示的第一页 -->
<section id=" page_MultiPageDemo" data-role=" page" >
  <header data-role=" header" >
    <h1> 多页面模板演示 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p> 这是一个“多页面模板演示”。 <br/>
    请点击下面的“关于我们”页面，以跳转页面。 </p>
    点击这里，进入 <a href=" #page_aboutus" > 关于我们 </a> 页面。 </div>
  <footer data-role=" footer" > CatchException.NET </footer>
</section>
<!-- 这里是多页显示的第二页，关于我们 -->
<section id=" page_aboutus" data-role=" page" >
  <header data-role=" header" >
    <h1> 关于我们 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p>CatchException.NET 是帮助工程师分享工作中所遇到的各种各样技术问题，寻求解决方案，并帮助建立现实社交网络的工程师社区。 </p>
    <p>CatchException.NET 的口号是，讲求理性观点的工程师社区。 </p>
    <br/>
    点击这里，进入 <a href=" #page_MultiPageDemo" > 演示 </a> 页面。 </div>
  <footer data-role=" footer" > CatchException.NET </footer>
</section>
</body>
</html>
```

这是一个通过多页模板生成的页面。包括两个基本页面，“主页面”点击之后进入到“关于我们”页面。

第二个页面是一个独立的 <section>，设置 data-role 为 page。这样点击打开是一个关于我们的页面。如果设置为 dialog，则点击关于我们按钮进入的第二个页面会变为对话框样式。在第二个页面设置 section 的 id 为 page_aboutus。此时在需要调用的位置加入超链接指向到锚链 #page_aboutus，就可以通过点击而打开第二个页面了。

使用浏览器打开上面的页面，看到“多页面模板演示”界面。



点击“关于我们”，跳转到下面的页面。同样，点击“进入演示页面”链接，回到前面的页面。



3.2. 页面标题

在前面的演示中，不管单页模版或者多页模板，在页面代码的 head 中 title 部分都设置为“跨平台移动应用”。但是在打开的浏览器中，却发现这个标题并没有被显示而取而代之的是在当前界面打开的页面 section 中定义在 <header> 中的文字内容。这样的好处在多页模板的显示方式下，不同页面的浏览器标题都是不同的。这样，用户体验可以被加强。

例如，在本章页面展示中，“关于我们”页面的标题就是和 header 中定义的文字“关于我们”一致。

如果多页模板第一个页具有 <header>，而之后的页面没有，那么会怎么样呢？后面的页面会使用前一个页面的标题。假如第一个页面没有 <header>，这个时候页面显示的标题才会显示 HTML 的 head 中的标题内容。

而这样的处理的确容易引起混乱，正确使用方法是在 <section> 中定义 data-title 属性值。

例如，<section id="page_aboutus" data-role="page" data-title=" 关于我们 ">

即便我们将“关于我们”部分的代码删除掉 <header> 部分。在实际使用的时候，浏览器标题依然不会出现错误。

修改之后的“关于我们”页面，删除掉 <header>

```
<section id=" page_aboutus" data-role=" page" data-title=" 关于我们 ">
```

```
<div class=" content" data-role=" content" >
```

```
<p>CatchException.NET 是帮助工程师分享工作中所遇到的各种各样技术问题，寻求解决方案，并帮助建立现实社交网络的工程师社区。</p>
```

```
<p>CatchException.NET 的口号是，讲求理性观点的工程师社区。</p>
```

```
<br/>
```

```
点击这里，进入 <a href=" #page_MultiPageDemo" > 演示 </a> 页面。 </div>
```

```
</div>
```

```
<footer data-role=" footer" > CatchException.NET </footer>
```

```
</section>
```


3.3. 链接到内部页面或外部页面

内部页面是当前页面或者通过锚链指向到的当前页面中的一部分。

外部页面是外部独立页面。如果声明为外部页面，即便是对当前页面的请求也会到服务器重新请求一遍。这个时候相当于刷新当前页面。

链接样式	链接类型
<code></code>	内部链接
<code></code>	内部链接
<code></code>	外部链接
<code></code>	外部链接
<code></code>	外部链接

这个例子展示了不同的内部链接和外部链接的实现方式。

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
<section id=" page_ExternalPageDemo" data-role=" page" >
  <header data-role=" header" >
    <h1> 外部页面链接演示 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p> 内部链接跳转到 <a href=" #page_aboutus" > 关于我们 </a> </p>
    <p> 外部链接跳转到 <a href=" #page_aboutus" rel=" external" > 关于我们 </
a><br/>
    外部链接跳转到 <a href=" http://www.catchexception.net" >CatchException.NET </
a><br/>
    外部链接跳转到 <a href=" http://www.catchexception.net" target=" _
blank" >CatchException.NET </a><br/>
  </div>
</body>
</html>
```

```
</div>
<footer data-role=" footer" > CatchException.NET </footer>
</section>
<section id=" page_aboutus" data-role=" page" >
  <header data-role=" header" >
    <h1> 关于我们 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p>CatchException.NET 是帮助工程师分享工作中所遇到的各种各样技术问题，寻求解决方案，并帮助建立现实社交网络的工程师社区。 </p>
    <p>CatchException.NET 的口号是，讲求理性观点的工程师社区。 </p>
    <br/>
    点击这里，进入 <a href=" #page_ExternalPageDemo" > 演示 </a> 页面。 </div>
  </div>
  <footer data-role=" footer" > CatchException.NET </footer>
</section>
</body>
</html>
```



3.4. 对话框建立，关闭

对话框可以看成一种特别的页面样式。还以之前的“关于我们”为例，只需要替换关于我们 section 的 data-role 从 page 到 dialog 就可以实现对话框效果。

为了更加接近真实的对话框样式，将之前例子中的“返回”链接调整为按钮样式。

调整之后的“关于我们”对话框代码为，

```
<section id=" page_aboutus" data-role=" dialog" >
```

```
  <header data-role=" header" data-theme=" e" >
```

```
    <h1> 关于我们 </h1>
```

```
  </header>
```


```
  <div class=" content" data-role=" content" data-theme=" e" >
```

```
    <p>CatchException.NET 是帮助工程师分享工作中所遇到的各种各样技术问题，寻求解决方案，并帮助建立现实社交网络的工程师社区。 </p>
```

```
    <p>CatchException.NET 的口号是，讲求理性观点的工程师社区。 </p>
```

```
<br/>  
<a href=" PageTransition.html" data-role=" button" data-theme=" b" data-  
rel=" back" > 返回 </a> </div>  
</div>  
</section>
```



关闭对话框可以使用 jQuery 选择器实现，也可以点击对话框左上方图标，。

通过程序实现对话框关闭的代码如下，

```
$( '.ui-dialog' ).dialog( 'close' );
```

3.5. 切换方式美化

不管页面或者对话框，在呈现的时候都可以设定其切换方式，以改善用户体验。只需要在链接中声明 data-transition 属性为期望的切换方式。

当前 jQuery Mobile 支持的页面切换方式包括 6 种。

切换方式	data-transition 属性值
横向幻灯方式	slide
自上向下幻灯方式	slideup
自下向上幻灯方式	slidedown
中央弹出	pop
淡入淡出	fade
旋转弹出	flip

值得注意的是，旋转弹出在 Android 某些版本中可能不能很好支持。因为这个美化需要使用到 3D CSS，而很多 Android 尚未支持。

这部分代码片段显示了各种切换到“关于我们”界面的样式。

```
<section id=" page_PageTransition" data-role=" page" >
  <header data-role=" header" >
    <h1> 页面切换方式演示 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p> 这段演示将会呈现不同的页面切换方式。 </p>
    <p>
      <a href=" #page_aboutus" data-transition=" slide" > 横向 </a> 缺省的切换方式
    <br/>
      <a href=" #page_aboutus" data-transition=" slideup" > 自下向上 </a><br/>
      <a href=" #page_aboutus" data-transition=" slidedown" > 自上向下 </a><br/>
      <a href=" #page_aboutus" data-transition=" pop" > 中央跳出 </a><br/>
      <a href=" #page_aboutus" data-transition=" fade" > 淡入淡出 </a><br/>
      <a href=" #page_aboutus" data-transition=" flip" > 旋转弹出 </a> 可惜，Android
    <br/>
    不支持这种切换 <br/>
  </div>
  <footer data-role=" footer" > CatchException.NET </footer>
</section>
```

页面切换方式演示

这段演示将会呈现不同的页面切换方式。

[横向](#) 缺省的切换方式

[自下向上](#)

[自上向下](#)

[中央跳出](#)

[淡入淡出](#)

[旋转弹出](#) 可惜，Android不支持这种切换

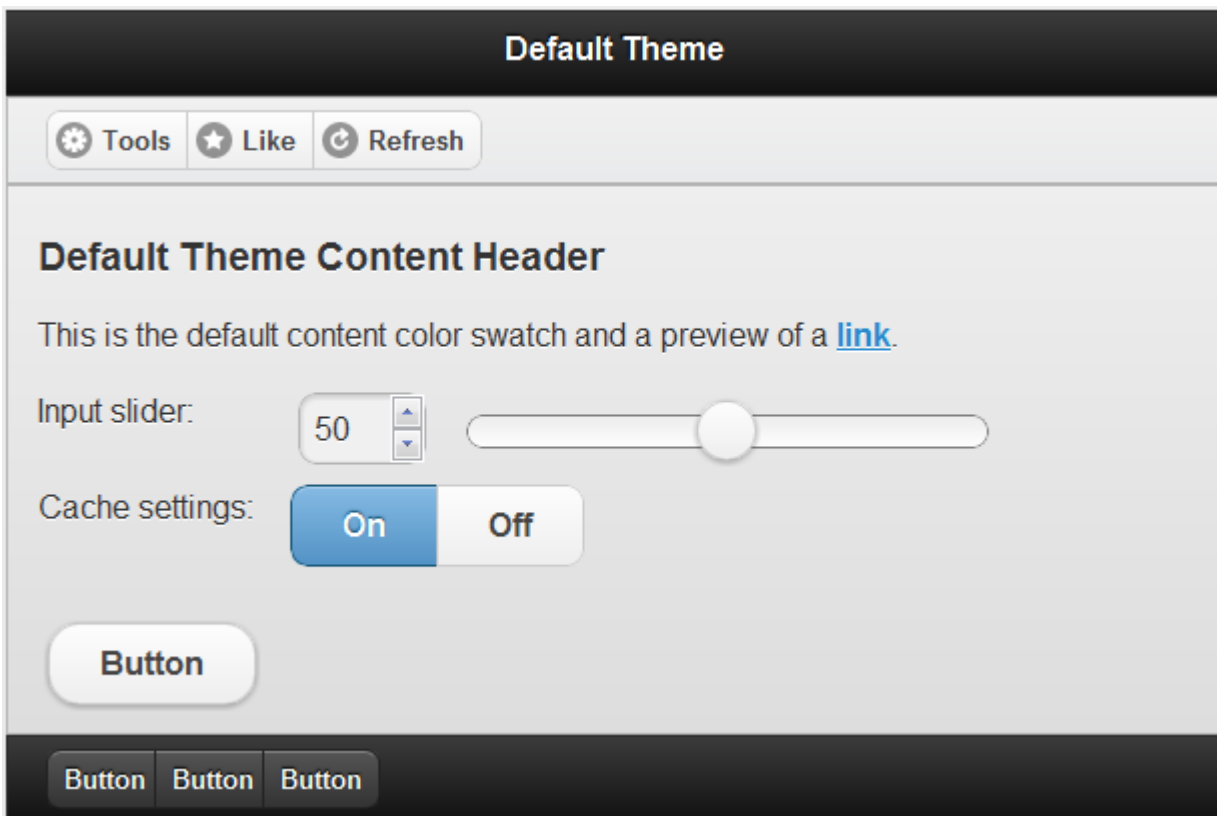
CatchException.NET

3.6. 页面风格美化

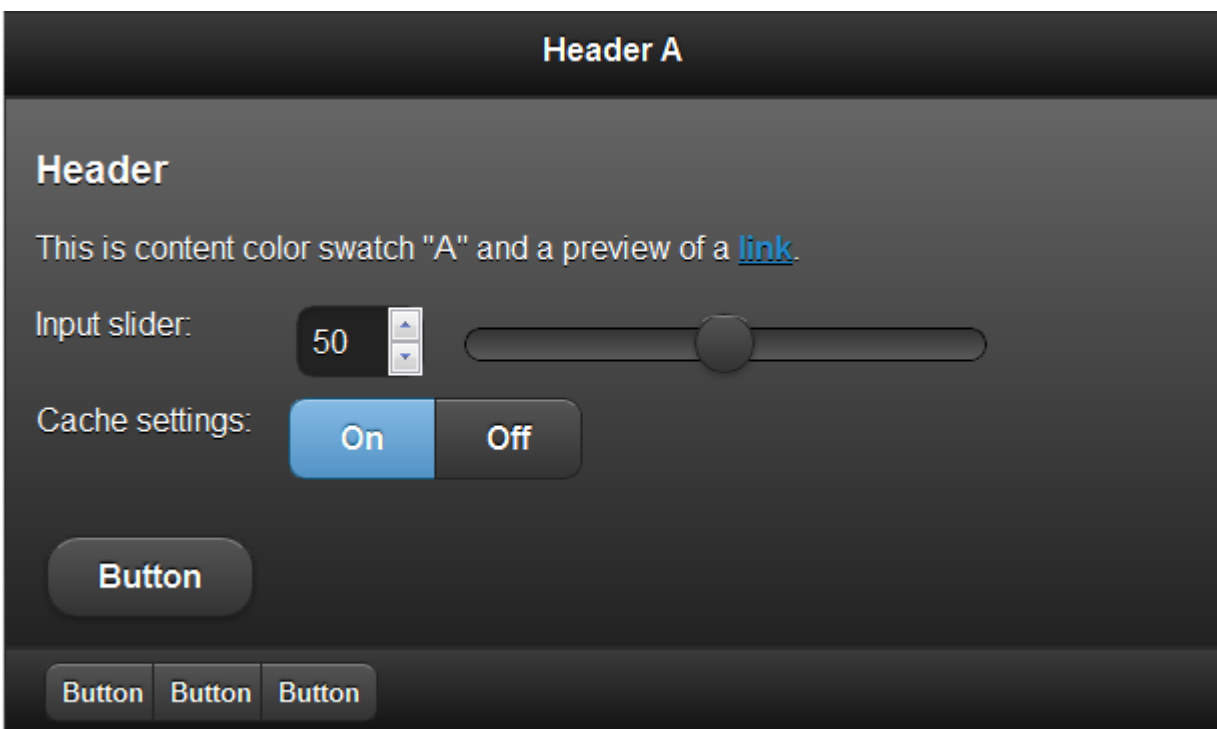
jQuery Mobile 拥有 6 种已经预设好的配色方案。在使用不同配色方案的时候，只需要在需要控制的部分设置 data-theme 属性，就可以进行颜色设置。例如，本章对话框范例程序中分别在 header 和 content 部分声明为界面风格 e。

```
<header data-role=" header" data-theme=" e" >
```

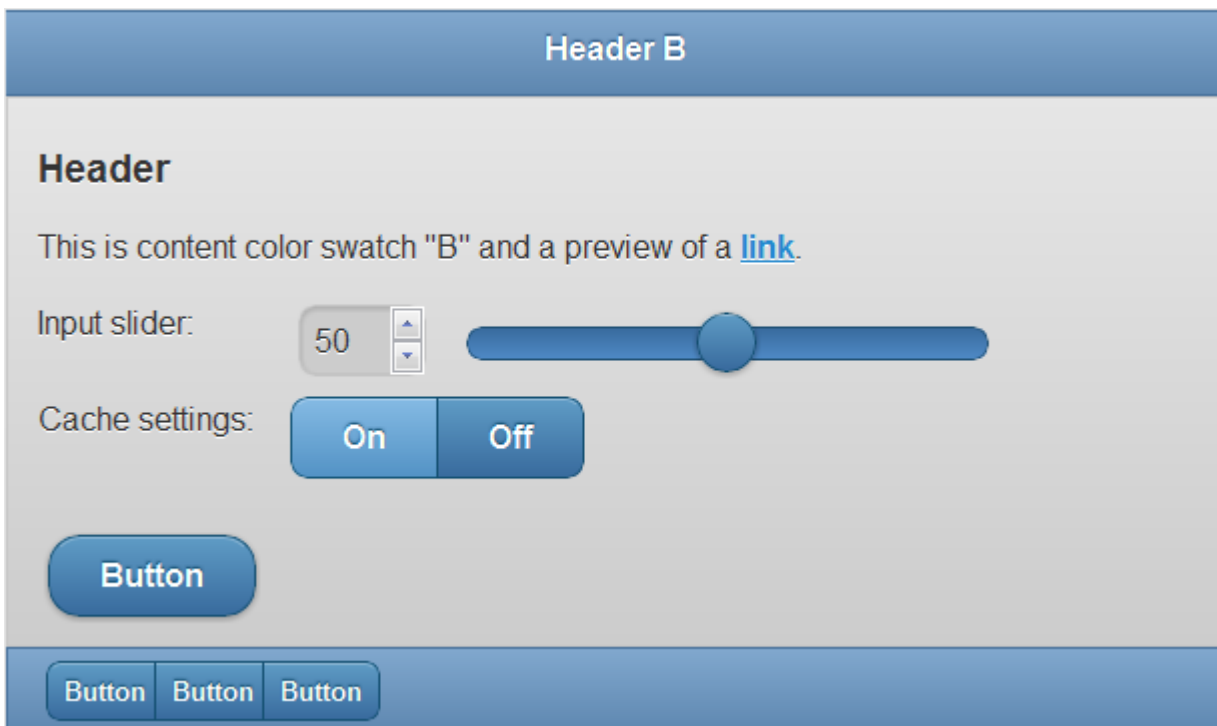
各种不同界面风格样式如下。



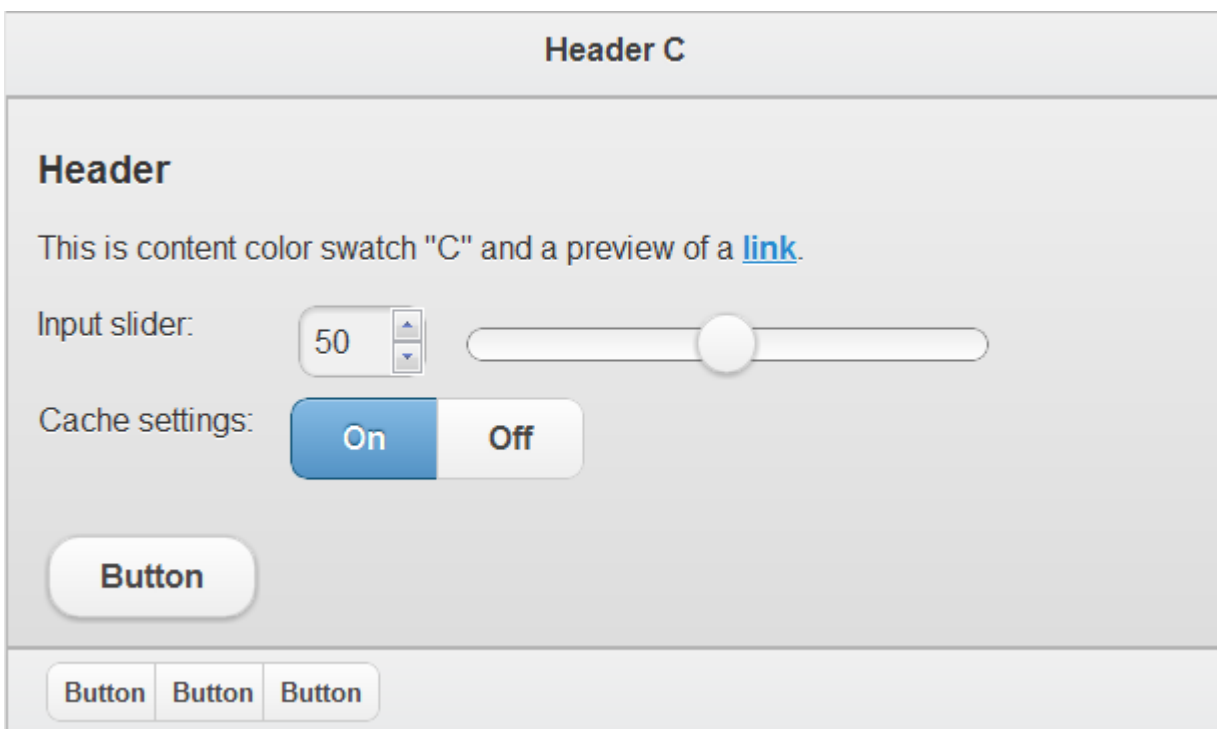
缺省界面风格



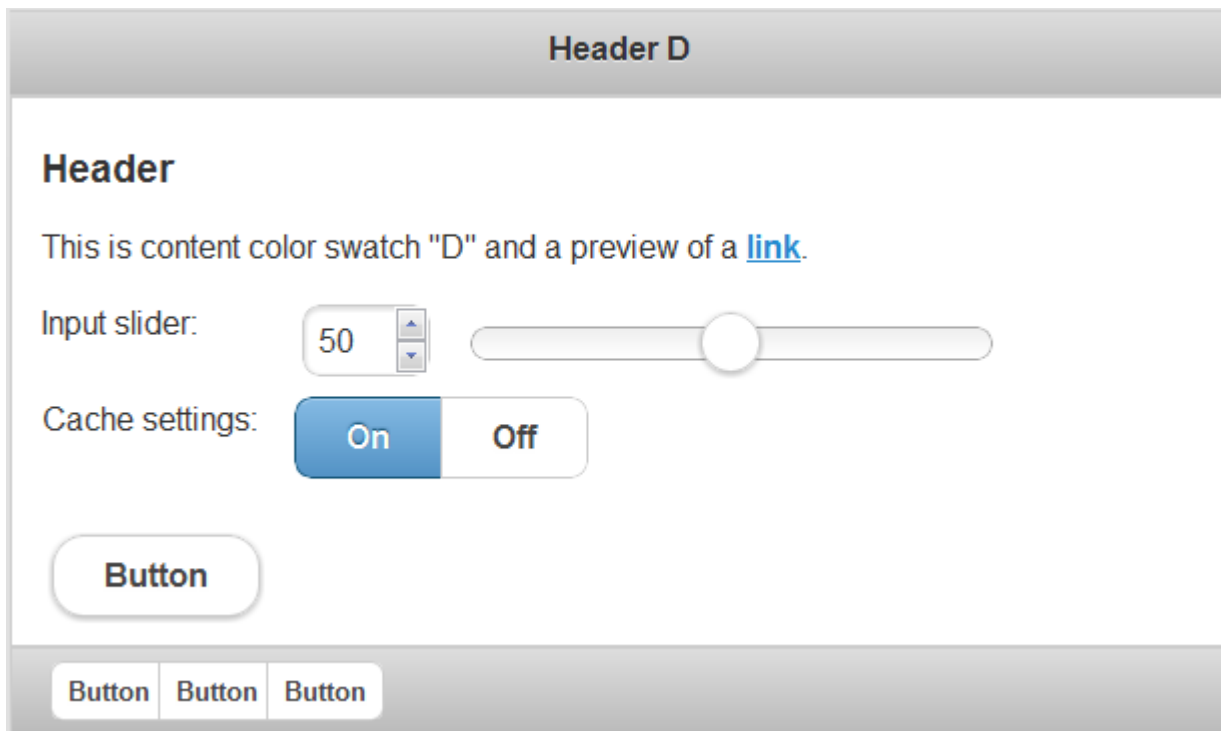
界面风格 A



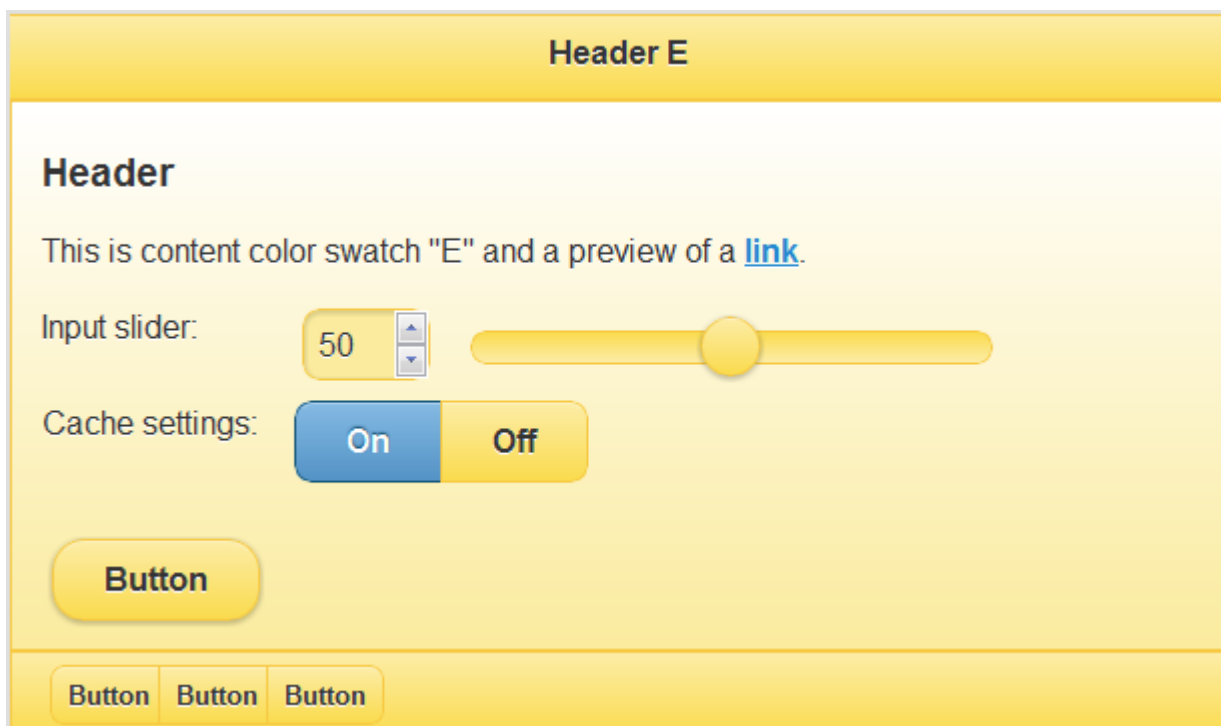
界面风格 B



界面风格 C



界面风格 D



界面风格 E

第 4 章 页面布局与内容呈现

移动应用的界面通常受到屏幕尺寸限制，尤其手机最为明显。而移动设备顾名思义，在移动中使用，自然没有桌面电脑可以两只手敲击键盘，滑动鼠标那么方便便捷。这些就为移动应用程序的布局与用户体验设计提出了要求。

在这一章中，我们将接触到，

- 基本页面布局
- 分栏布局与网格布局
- 集成自定义 CSS

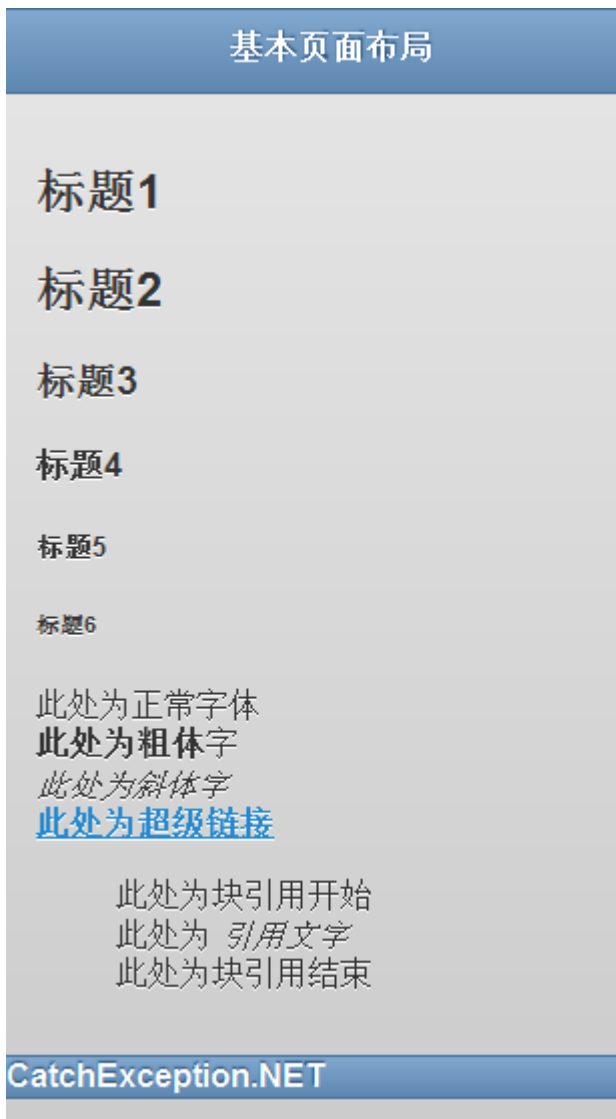
4.1. 基本页面布局

在之前章节的范例中，各个范例的界面都是一栏的。也就是各个页面元素基本都是自上而下顺序排列的。在基于 jQuery Mobile 开发的应用中，HTML4 页面标记几乎都可以得到良好支持。

下面的代码以顺序显示了主要的 HTML4 标记，并可以通过之后的界面来看到不同标记的内容所呈现的效果。

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
```

```
<section id=" page_LayoutBasic" data-role=" page" data-title=" 基本页面布局 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 基本页面布局 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <h1> 标题 1</h1>  
    <h2> 标题 2</h2>  
    <h3> 标题 3</h3>  
    <h4> 标题 4</h4>  
    <h5> 标题 5</h5>  
    <h6> 标题 6</h6>  
    <p> 此处为正常字体 <br/>  
    <strong> 此处为粗体 </strong> 字 <br/>  
    <em> 此处为斜体字 </em><br/>  
    <a href=" #" > 此处为超级链接 </a><br/>  
    <blockquote> 此处为块引用开始 <br/>  
    此处为 <cite> 引用文字 </cite><br/>  
    此处为块引用结束 </blockquote>  
  </p>  
</div>  
  <footer data-role=" footer" data-theme=" b" > CatchException.NET </footer>  
</section>  
</body>  
</html>
```



下面的代码展示了项目符号列表，数字编号列表和表格在界面中的呈现效果。

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
```

```
<section id=" page_LayoutTable" data-role=" page" data-title=" 基本页面布局 " data-theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 基本页面布局 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p> 项目符号列表
    <ul>
      <li> 项目符号列表 1</li>
      <li> 项目符号列表 2</li>
      <li> 项目符号列表 3</li>
    </ul>
    </p>
    <hr/>
    <p> 数字编号列表
    <ol>
      <li> 数字编号列表 1</li>
      <li> 数字编号列表 2</li>
      <li> 数字编号列表 3</li>
    </ol>
    </p>
    <hr/>
    <p>
    <table>
      <caption>
        表格标题
      </caption>
      <thead>
        <tr>
          <th scope=" col" > 书名 </th>
          <th scope=" col" > 作者 </th>
          <th scope=" col" > 年代 </th>
        </tr>
      </thead>
      <tfoot>
```

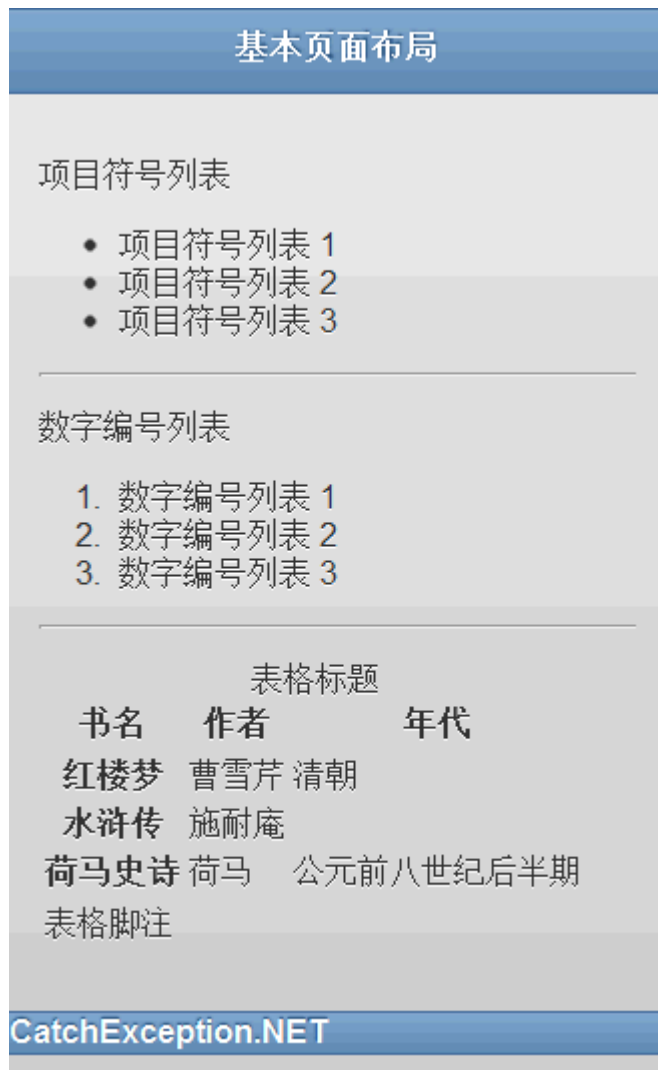
```
<tr>
  <td colspan=" 5" > 表格脚注 </td>
</tr>
</tfoot>
<tbody>
  <tr>
    <th scope=" row" > 红楼梦 </th>
    <td> 曹雪芹 </td>
    <td> 清朝 </td>
  </tr>
  <tr>
    <th scope=" row" > 水浒传 </th>
    <td> 施耐庵 </td>
    <td></td>
  </tr>
  <tr>
    <th scope=" row" > 荷马史诗 </th>
    <td> 荷马 </td>
    <td> 公元前八世纪后半期 </td>
  </tr>
</tbody>
</table>
</p>
</div>
<footer data-role=" footer" data-theme=" b" > CatchException.NET </footer>
</section>
</body>
</html>
```

在上面的代码中，可以看到关于表格的定义方式与传统的表格定义方式有些区别。这部分表格定义是由 HTML5 的方式进行的定义，

- caption 标记表示表格标题
- thead 表示表格列名称
- tfoot 表示表格脚注
- tbody 表示表格内容。这部分相当于传统 HTML4 中 tr 和 td 标记中所包含的内容。

有关于 HTML5 表格的具体定义，可以参照 HTML5 相关参考资料。

下图展示了前面各种列表和表格基于 jQuery Mobile 的输出样式。



4.2. 分栏与网格布局

虽然通过表格方式可以实现多行多列的内容呈现，但是代码毕竟不够简洁。另外，通过表格方式辅助内容排版还存在这样两个可能的弊端，

- 呈现和渲染表格的速度在移动设备中通常比较耗时和消耗资源
- 通过表格进行排版的内容，搜索引擎友好型通常不怎么好

jQuery Mobile 通过支持分栏和网格布局，提供了简单而有效的界面排版方式。这种方式对于之前通过表格的方式进行排版，有不错的应对。

jQuery Mobile 的分栏格式通常来说具有两类标记嵌套而成，

- Ui-grid-a，表示每栏宽度为移动设备屏幕的 50%

- Ui-block-a, 表示栏目所处的顺序。

```
<div class=" ui-grid-a" >
  <div class=" ui-block-a" >
    <p> 第一栏 </p>
  </div>
  <div class=" ui-block-b" >
    <p> 第二栏 </p>
  </div>
</div>
```

如果要设置不同的栏宽度, 可以选择 ui-grid-b, ui-grid-c... 最多可以到 ui-grid-e。

同样, 对于各个栏目的位次排列, 每个块的顺序也可以通过设置, a, b, c, d, e 的顺序进行排列。如果顺序没有按照 a 到 e 的顺序进行排列, 则可能会出现换行的情况。而如果一行容纳不下足够的块, 也会出现多行显示。这样就呈现网格的效果了。

下面的例子中, 我们将看到不同分栏数量的现实效果。

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
<style>
.content div div p {
    background-color: #aaa;
    height: 50px;
    border: 1px solid #fff;
    margin: 0px 2px;
}
</style>
</head>
<body>
```

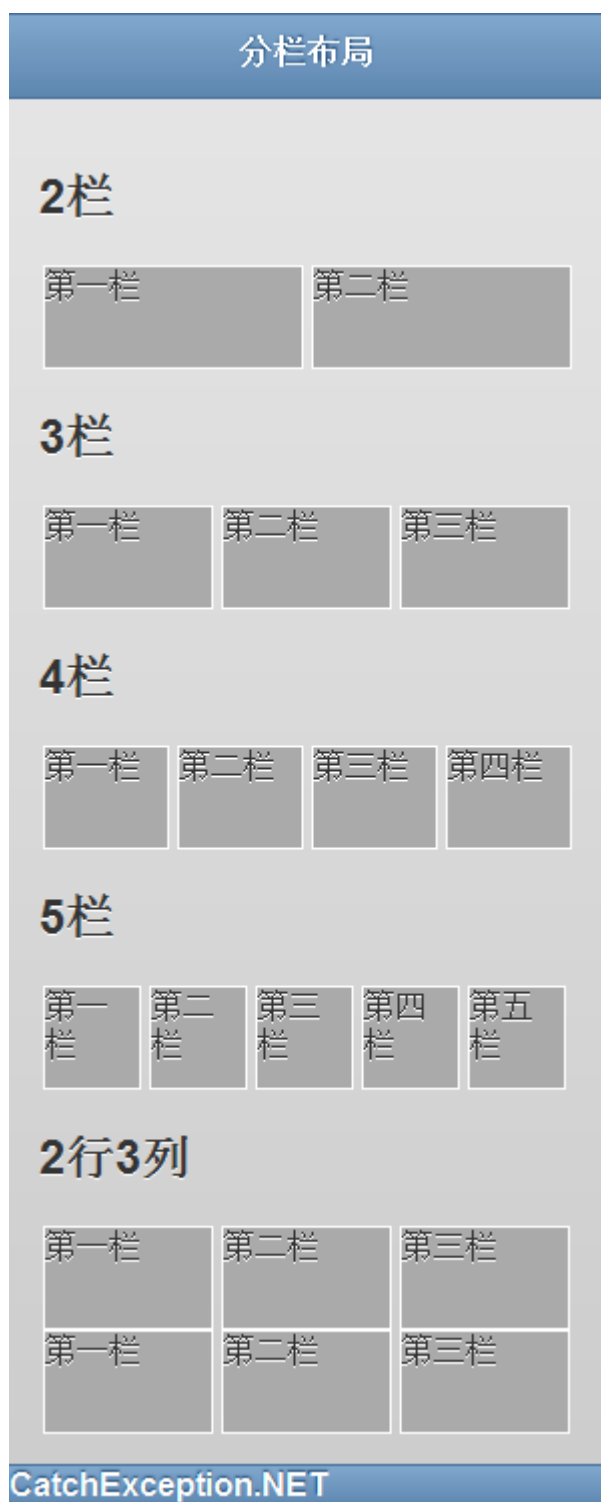


```
<section id=" page_LayoutColumn" data-role=" page" data-title=" 分栏布局 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 分栏布局 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <h2>2 栏 </h2>  
    <div class=" ui-grid-a" >  
      <div class=" ui-block-a" >  
        <p> 第一栏 </p>  
      </div>  
      <div class=" ui-block-b" >  
        <p> 第二栏 </p>  
      </div>  
    </div>  
    <h2>3 栏 </h2>  
    <div class=" ui-grid-b" >  
      <div class=" ui-block-a" >  
        <p> 第一栏 </p>  
      </div>  
      <div class=" ui-block-b" >  
        <p> 第二栏 </p>  
      </div>  
      <div class=" ui-block-c" >  
        <p> 第三栏 </p>  
      </div>  
    </div>  
    <h2>4 栏 </h2>  
    <div class=" ui-grid-c" >  
      <div class=" ui-block-a" >  
        <p> 第一栏 </p>  
      </div>  
      <div class=" ui-block-b" >  
        <p> 第二栏 </p>  
      </div>
```

```
<div class=" ui-block-c" >
  <p> 第三栏 </p>
</div>
<div class=" ui-block-d" >
  <p> 第四栏 </p>
</div>
</div>
<h2>5 栏 </h2>
<div class=" ui-grid-d" >
  <div class=" ui-block-a" >
    <p> 第一栏 </p>
  </div>
  <div class=" ui-block-b" >
    <p> 第二栏 </p>
  </div>
  <div class=" ui-block-c" >
    <p> 第三栏 </p>
  </div>
  <div class=" ui-block-d" >
    <p> 第四栏 </p>
  </div>
  <div class=" ui-block-e" >
    <p> 第五栏 </p>
  </div>
</div>
<h2>2 行 3 列 </h2>
<div class=" ui-grid-b" >
  <div class=" ui-block-a" >
    <p> 第一栏 </p>
  </div>
  <div class=" ui-block-b" >
    <p> 第二栏 </p>
  </div>
  <div class=" ui-block-c" >
    <p> 第三栏 </p>
  </div>
</div>
```

```
</div>  
<div class=" ui-block-a" >  
  <p> 第一栏 </p>  
</div>  
<div class=" ui-block-b" >  
  <p> 第二栏 </p>  
</div>  
<div class=" ui-block-c" >  
  <p> 第三栏 </p>  
</div>  
</div>  
</div>  
<div>  
<footer data-role=" footer" data-theme=" b" > CatchException.NET </footer>  
</section>  
</body>  
</html>
```

上述代码呈现效果如下，



这部分示例代码中，最后一段两行三列的代码实现方式就可以看成是一个网格布局。在分栏和网格布局的结构下，通常是没有背景颜色和边框颜色的。为了能够看到分栏的各个部分的效果，在 HTML 开始的部分进行了 CSS 定义，设置了背景颜色，边框颜色以及边框之间的间距。

下面的代码是结合应用场景的具体事例。在一个界面中，一段提示语言之后，有两个并排的按钮。点击不同的按钮将执行不同的操作。而两个并排的按钮被放置于两个独立的分栏中。

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
<section id=" page_LayoutColumn" data-role=" page" data-title=" 分栏布局 " data-
theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 分栏布局 </h1>
  </header>
  <div class=" content" data-role=" content" > 这是一个分栏布局的示例。在下方有两
个按钮，“确认”和“取消”分别放置于不同的分栏之内。
    <div class=" ui-grid-a" >
      <div class=" ui-block-a" > <a href=" PageTransition.html" data-role=" button"
data-theme=" b" data-rel=" back" > 关闭 </a> </div>
      <div class=" ui-block-b" > <a href=" PageTransition.html" data-role=" button"
data-theme=" d" data-rel=" back" > 取消 </a> </div>
    </div>
  </div>
  </div>
  <footer data-role=" footer" data-theme=" b" > CatchException.NET </footer>
</section>
</body>
</html>
```

这段代码呈现样式如下，



4.3. 集成自定义 CSS

虽然 jQuery Mobile 提供了几种系统缺省的 Themes 界面定义风格，并可以通过简单的 `data-theme` 属性指定为不同风格。在实际使用过程中依然可能会涉及到自定义样式，例如，行间距，字体，字号，颜色，间距等。这样的标记可以使用自定义的 CSS 来实现，就好象传统的网页设计一样。只是如果要选择 CSS3 的一些升级的功能，可能要检查一下不同移动平台浏览器的支持能力，例如 CSS3 中对于 3D 的支持，在 Android 的多个操作系统中是不能支持的。

对于界面的定义，建议通过独立的 CSS 文件实现，如果可能，也可以将 CSS 缓存到本地，这样将有助于减少页面下载速度，提升用户体验效果。

第 5 章 按钮

我们在之前的章节中曾经使用到按钮，这是一种典型的页面元素。在 jQuery Mobile 技术中，按钮是一种比较有趣的实现。有别于传统 HTML 语言对于按钮的实现，除传统 button 标签之外，jQuery Mobile 还支持通过定义超链接 `<a>` 来实现具有页面链接和跳转功能的按钮功能。我们将专门探讨不同的按钮功能、样式和实现方式。

在这一章中，我们将接触到，

- jQuery Mobile 的按钮
- 内置按钮样式
- 按钮图标
- 按钮组

5.1. jQuery Mobile 的按钮

按钮是移动应用中最常见的页面元素。通常按钮功能主要是两种，

- 页面或者功能跳转
- 提交表单内容

所以在 jQuery Mobile 定义中，按钮也有两种不同的书写方式，按钮外形的超级链接和表单按钮。

两种按钮的样例比较可以参阅下面这段代码，

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
```

```
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
<section id=" page_LayoutButton" data-role=" page" data-title=" 基本按钮样式 "
data-theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 基本按钮样式 </h1>
  </header>
  <div class=" content" data-role=" content" > <a href=" #" data-role=" button"
data-theme=" b" > 链接按钮 </a>
    <button data-theme=" b" > 表单按钮 </button>
  </div>
</section>
</body>
</html>
```

在这段代码中，姑且对于传统 HTML Input 类型的按钮称之为“表单按钮”，而对于超级链接的按钮样式称之为“链接按钮”。显然他们的功能是不同的。而从下图中看到，两种按钮的样式却是一致的。



除去上面基于链接的按钮和通过 button 标签声明的按钮，在传统 HTML4 的方式下声明的按钮，在 jQuery Mobile 中也可以被显示为按钮，包括，如下声明方式

```
<input type=" button" />
```

```
<input type=" submit" />
```

```
<input type=" reset" />
```

```
<input type=" image" />
```

而这样的声明的按钮功能和传统 HTML4 的按钮功能也是一致的。

5.2. 内置按钮样式

在前面一个样例中，按钮宽度几乎占到整个移动设备屏幕的宽度。而很多情况下，按钮是不需要占用这么大尺寸的。如果能占用空间相对有效，则方便手指操作，提升用户体验。针对这种场景，jQuery Mobile 提出了一种内置按钮（inline button）样式的声明方式。

只需要在之前的声明上添加 data-inline 属性，则按钮宽度自然按照文字尺寸进行安排。而按钮也按照自左向右的顺序依次排列。

在前述样例基础上，对于按钮部分添加 data-inline 属性 true，则更新代码段如下，

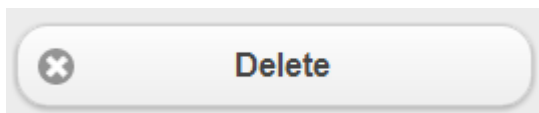
```
<section id=" page_LayoutButton" data-role=" page" data-title=" 内置按钮样式 "
data-theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 内置按钮样式 </h1>
  </header>
  <div class=" content" data-role=" content" > <a href=" #" data-role=" button"
data-theme=" b" data-inline=" true" > 链接按钮 </a>
  <button data-theme=" b" data-inline=" true" > 表单按钮 </button>
  </div>
</section>
```

其显示效果为，



5.3. 按钮图标

jQuery Mobile 在进行按钮设计的时候，缺省定义了一套 data-icon 参数，基于这套参数可以指定常用的按钮图标。例如，` Delete `，则会呈现如下效果，



data-icon 可以指定不同的参数，参数不同，所呈现的按钮图标样式也会不同。

左箭头	data-icon = "arrow-l"	
右箭头	data-icon="arrow-r"	
向上箭头	data-icon="arrow-u"	
向下箭头	data-icon="arrow-d"	
删除	data-icon="delete"	
加号	data-icon="plus"	
减号	data-icon="minus"	
检查	data-icon="check"	
齿轮	data-icon="gear"	

刷新		<code>data-icon="refresh"</code>
向前		<code>data-icon="forward"</code>
向后		<code>data-icon="back"</code>
网格		<code>data-icon="grid"</code>
星型		<code>data-icon="star"</code>
警告		<code>data-icon="alert"</code>
信息		<code>data-icon="info"</code>
主页		<code>data-icon="home"</code>
查询		<code>data-icon="search"</code>

此外，设置 `data-iconpos` 属性也可以设置图标的位置。

值	图标位于按钮的位置
left	左侧（缺省）
right	右侧
top	上方正中
bottom	下方正中
notext	只有图标，而没有文字

缺省图标位于左侧，所以如果需要图标在缺省位置，则可以不用设置 `data-iconpos` 属性。

如果只是设置图标，而不希望包括任何文字，则可以添加参数 `data-iconpos` 为 `notext`。例如，`Delete `，则呈现效果如下图所示，



5.4. 按钮组

按钮组，顾名思义，一组按钮在一起。jQuery Mobile 统一对一组样式一致的按钮进行样式定义和渲染。

按钮组分为横向按钮组和纵向按钮组。

在纵向按钮组中，声明按钮组的 div 标签，添加属性 data-role="controlgroup"。缺省为纵向按钮组，如果希望呈现为横向按钮组，则添加 data-type="horizontal" 属性。

下面的范例代码分别呈现了纵向按钮组和横向按钮组。

```
<section id=" page_LayoutButton" data-role=" page" data-title=" 按钮组样式 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 按钮组样式 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 如下内容为纵向按钮组  
    <div data-role=" controlgroup" > <a href=" #" data-role=" button" data-  
theme=" b" > 链接按钮 </a>  
      <button data-theme=" b" > 表单按钮 </button>  
    </div>  
    <hr/>  
    </p>  
    <p> 如下内容为横向按钮组  
    <div data-role=" controlgroup" data-type=" horizontal" > <a href=" #" data-  
role=" button" data-theme=" b" > 链接按钮 </a>  
      <button data-theme=" b" > 表单按钮 </button>  
    </div>  
    </p>  
  </div>  
</section>
```

呈现效果如下图所示，



5.5. 选项、方法和事件

在进行页面编程的时候，按钮被点击之后所执行的行为以及对于事件的捕获，在 jQuery Mobile 中有专门的定义。这些属性、方法和事件都是基于 jQuery 实现的，方便工程师进行灵活的前端功能定制。

5.5.1. 选项

这部分选项和之前程序范例标签中的属性从功能和实现效果是相同的。

不同之处在于，之前范例标签中的属性值是在 HTML 标签中声明的，而选项则是在 JavaScript 代码中，经由 jQuery 选择器定位到某个按钮而进行的选项的赋值。从命名来说，通常选项的名称就是所要操作的语义，例如，icon，而属性的命名则会添加“data-”作为前缀，例如，data-icon。

作为属性，定义一个按钮的图标是这样实现的。

```
<button data-icon=" plus" />
```

作为选项，在 JavaScript 中，选项是这样使用的，

```
$( "button" ).buttonMarkup({ icon: "plus" });  
jQuery( "button" ).buttonMarkup({ icon: "plus" });
```

在下面的选项列表中，是按照在 JavaScript 中使用的选项名称来标识的。如果要在 HTML5 的标签中使用，对应的属性名称，请参见最后的“标签属性”一栏。

选项名称:	corners
选项类型:	Boolean
缺省值:	True
功能:	定义按钮为圆角按钮则设置为 true，否则直角按钮为 false.
标签属性:	data-corners
选项名称:	icon
选项类型:	string
缺省值:	null
功能:	定义按钮图标，缺省为没有图标
标签属性:	data-icon
选项名称:	iconpos
选项类型:	string
缺省值:	left
功能:	按钮图标的位置，缺省为居左。也可以自定义位置， right – 居右 up – 上方居中 bottom – 下方居中
标签属性:	data-iconpos
选项名称:	iconshadow
选项类型:	boolean
缺省值:	true
功能:	按钮图标阴影状态。 true – 有阴影 false – 无阴影
标签属性:	data-iconshadow
选项名称:	inline
选项类型:	boolean
缺省值:	false/null

功能:	内嵌按钮样式。 false – 非内嵌（内联）按钮。按钮与移动设备屏幕等宽。缺省值。 true - 内嵌按钮。
标签属性:	data-inline
选项名称:	shadow
选项类型:	boolean
缺省值:	true
功能:	按钮阴影状态。 true – 有阴影 false – 无阴影
标签属性:	data-shadow
选项名称:	theme
选项类型:	string
缺省值:	null, 继承自上一级
功能:	缺省的按钮风格是 null, 表示按钮风格继承自上一级 div 或者 section。如果上一级也没有定义风格, 则使用缺省风格。 风格定义自 a 到 e, 定义 5 中不同配色样式。
标签属性:	data-theme
选项名称:	initSelector
选项类型:	CSS 选择器字符串
缺省值:	button, [type='button'], [type='submit'], [type='reset'], [type='image']
功能:	用于指定 CSS 选择器。
标签属性:	无

5.5.2. 方法

jQuery Mobile 的按钮方法通过 jQuery 选择器获得某个或者某一类按钮实例之后, 执行相应方法内容。

方法名称:	enable
功能:	按钮使能
示例:	<code>\$('[type= submit']').button('enable');</code>
方法名称:	disable
功能:	按钮禁用
示例:	<code>\$('[type= submit']').button('disable');</code>
方法名称:	refresh
功能:	按钮刷新
示例:	<code>\$('[type= submit']').button('refresh');</code>

5.5.3. 事件

jQuery Mobile 的按钮事件, 主要是 create 事件, 在创建按钮对象的时候触发这个事件。虽然继承自 jQuery, 但是 jQuery Mobile 的事件管理和 jQuery 的事件管理略微有些差异。建议使用 jQuery Mobile 的事件管理而不是 jQuery 的。

代码原型,

```
$( “.selector” ).button({  
  create: function(event, ui) { ... }  
});
```


第 6 章 导航工具栏

导航工具栏是移动应用开发中使用较多的技术。方便的导航设计，有助于用户方便的拇指操作。

在这一章中，我们将接触到，

- 导航工具栏
- 头部工具栏
- 脚注工具栏
- 导航栏位置

6.1. 导航工具栏

有如传统页面所呈现，导航工具栏有助于用户直接的从一个应用界面跳转到另一个应用界面中。

通常导航工具栏主要分为头部导航工具栏（Header Bar），脚注导航工具栏（Footer Bar）和导航栏（Navbar）。脚注导航工具栏，Footer Bar，翻译可能并不准确，这里是指处于屏幕最下方的导航工具栏。

有别于传统桌面应用，脚注导航工具栏比头部工具栏的实用性更强。这是由于不论左手或者右手操作移动设备，脚注导航工具栏距离手指，特别是拇指的距离更近。所以一般移动应用中，更多的功能会放在脚注的位置。也可能是基于这样的用户行为方式，通常头部导航工具栏中除了包含导航按钮，还可能会包含页面的标题。而脚注导航工具栏的功能就更聚焦在不同的导航按钮上了。

导航工具栏的核心价值与在于方便的引导用户到指定的功能界面，所以从某种角度来说，导航工具栏包含两个核心



This page demonstrates the "fullscreen" toolbar mode. This toolbar treatment is used in special cases where you want the content to fill the whole screen, and you want the header and footer toolbars to appear and disappear when the page is clicked responsively — a common scenario for photo, image or video viewers.

To enable this toolbar feature type, you apply a `data-fullscreen="true"` attribute to the `div` contain that has the attribute `data-role="page"`, and the `data-position="fixed"` attribute to both the header and footer `div` elements. Fullscreen Fi...

概念,

- 合适位置放置头部或者脚注导航工具栏
- 合适的导航按钮, 布局与样式

这也是我们在此部分讨论导航工具栏的重点所在。

6.2. 头部工具栏

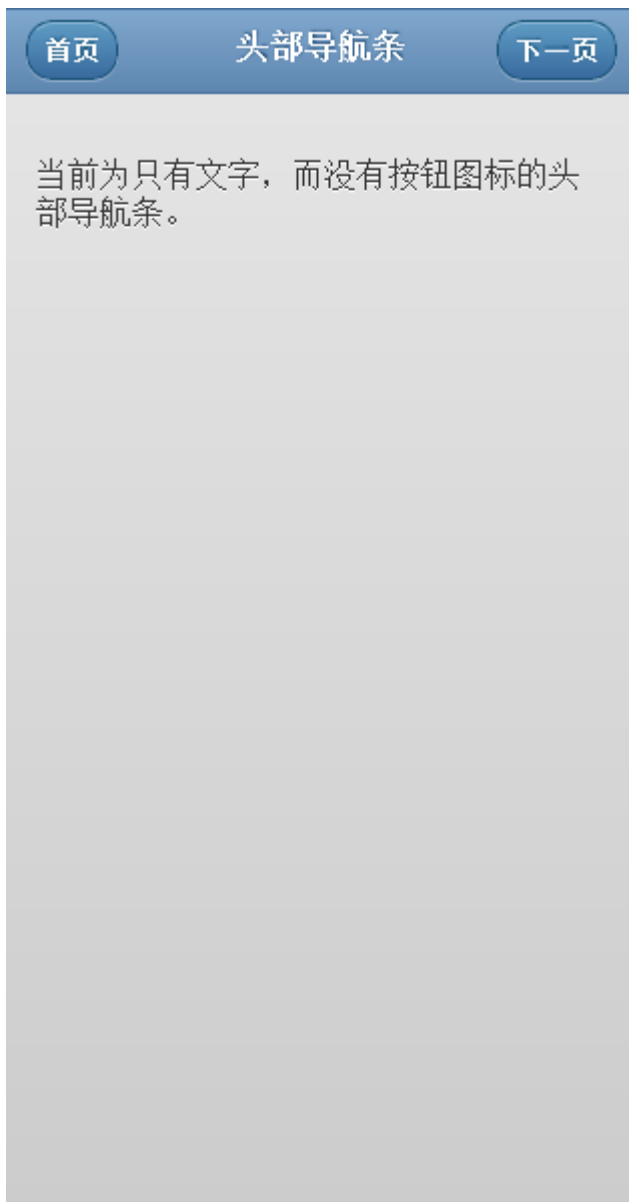
头部工具栏是位于界面上方的导航栏。

我们之前的很多页面 `<header>` 中定义的内容就可以看成是没有导航按钮的工具栏。

基于前一章关于按钮的加入方式, 在 `<header>` 中的标题文字前后添加按钮, 将实现最常见的头部导航样式。

```
<section id=" HeaderBar" data-role=" page" data-title=" 头 部 导 航 条 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" > <a href=" #MainPage" > 首页 </  
a>  
  <h1> 头部导航条 </h1>  
  <a href=" #HeaderBarWithICON" > 下一页 </a> </header>  
<div class=" content" data-role=" content" >  
  <p> 当前为只有文字, 而没有按钮图标的头部导航条。 </p>  
</div>  
</section>
```

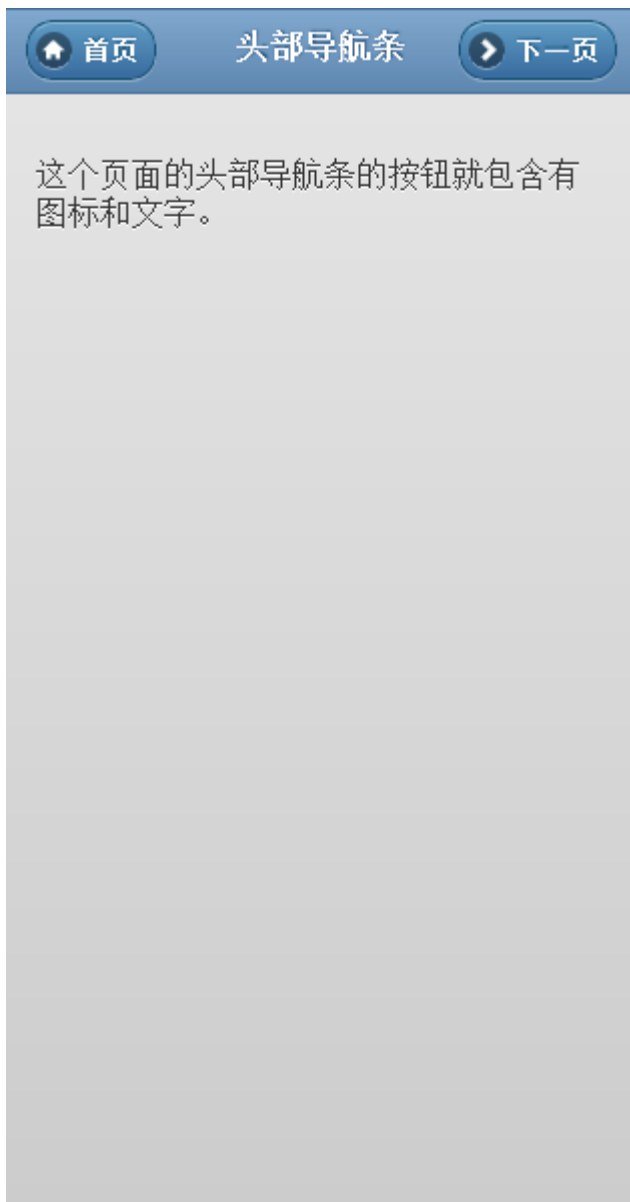
界面呈现效果如下,



如果要在按钮上添加图标，不管是 jQuery Mobile 框架定义图标样式或者用户自定义图标样式。实现方式和前一章所涉及的按钮定义方式是一样的。

```
<section id=" HeaderBarWithICON" data-role=" page" data-title=" 头部导航条 " data-theme=" b" >
  <header data-role=" header" data-theme=" b" > <a href=" #MainPage" data-icon=" home" > 首页 </a>
    <h1> 头部导航条 </h1>
    <a href=" #HeaderBar" data-icon=" arrow-r" > 下一页 </a> </header>
  <div class=" content" data-role=" content" >
    <p> 这个页面的头部导航条的按钮就包含有图标和文字。 </p>
  </div>
</section>
```

具有图标按钮的头部导航条呈现效果如下，



同样，按钮样式也可以通过属性 data-theme 进行设置。

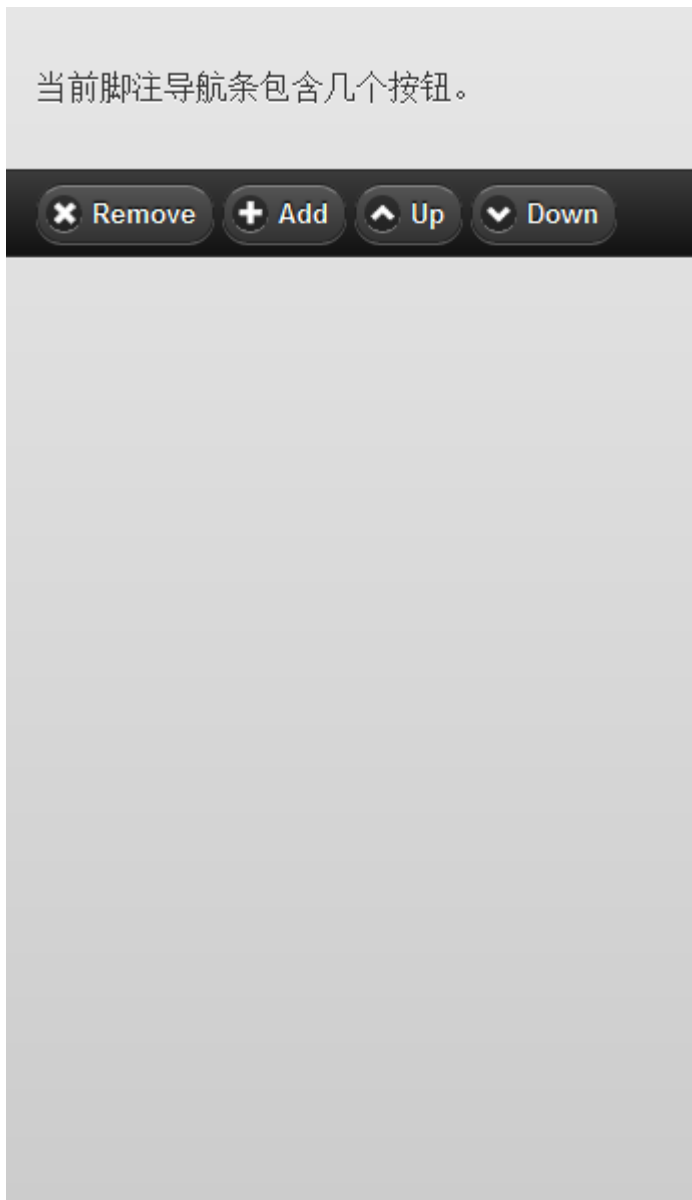
6.3. 脚注工具栏

脚注工具栏能呈现的效果较头部工具栏更加丰富一些，将需要呈现的页面元素或者表单元素置于 <footer> 中，则可以简单的定义出所需要的脚注工具栏。

这里范例是包含四个按钮的脚注工具栏。

```
<div data-role=" footer" class=" ui-bar" >
```

```
<a href="#" #FooterBar2" data-role=" button" data-icon=" delete" >Remove</a>  
<a href="#" #FooterBar2" data-role=" button" data-icon=" plus" >Add</a>  
<a href="#" #FooterBar2" data-role=" button" data-icon=" arrow-u" >Up</a>  
<a href="#" #FooterBar2" data-role=" button" data-icon=" arrow-d" >Down</a>  
</div>  
</div>
```

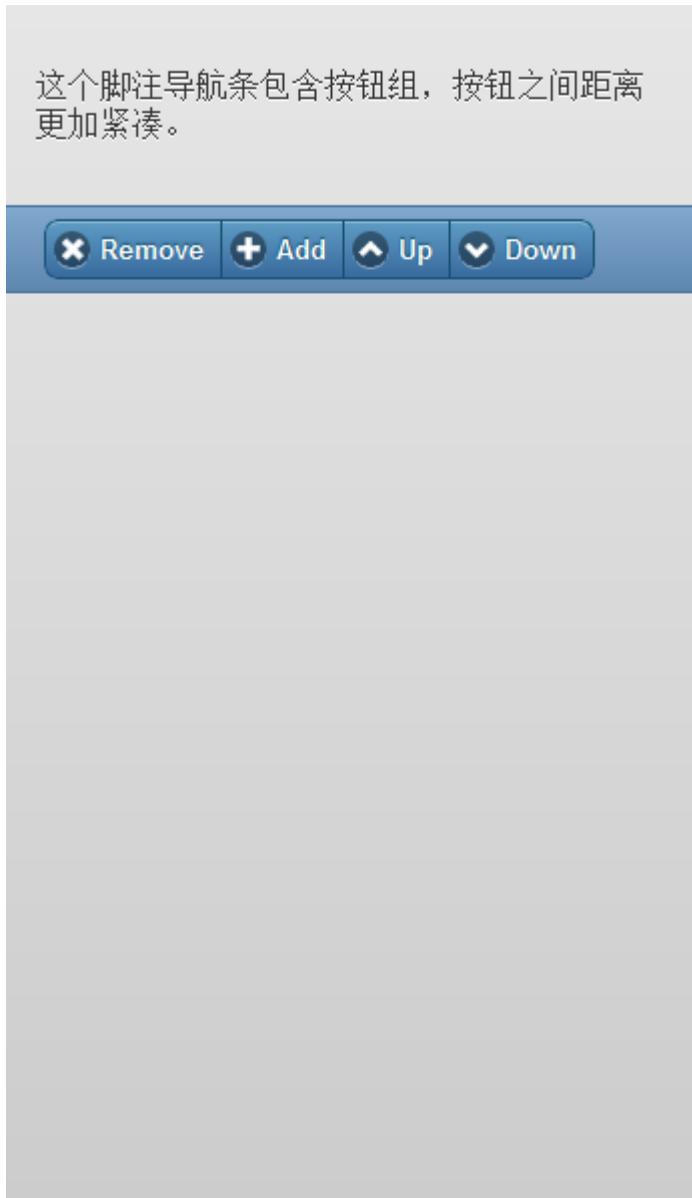


独立的按钮，每个按钮之间存在一定的空间，在移动设备显示面积有限的环境下，可以通过按钮组的方式进行呈现。这样按钮和按钮之间的空间将不再存在，各个按钮靠在一起。

```
<div data-role=" footer" class=" ui-bar" data-theme=' b' >  
<div data-role=" controlgroup" data-type=" horizontal" >  
  <a href="#" #FooterBar2" data-role=" button" data-icon=" delete" >Remove</a>
```

```
<a href="#" #FooterBar2" data-role=" button" data-icon=" plus" >Add</a>  
<a href="#" #FooterBar2" data-role=" button" data-icon=" arrow-u" >Up</a>  
  <a href="#" #FooterBar2" data-role=" button" data-icon=" arrow-d" >Down</a>  
</div>  
</div>  
</div>
```

在下图中按钮之间的空间没有了，而各个按钮形成一个组。外部有一个统一的圆角矩形。



其他按钮、文字或者其他表单元素也可以放置在脚注工具栏，以实现开发者所期望的呈现效果。

```
<div data-role=" footer" class=" ui-bar" >  
  CatchException.NET
```

```
<a href=" http://catchexception.net" data-role=" button" data-  
icon=" Home" >Home</a>  
</div>
```

6.4. 导航工具条

导航工具条可以理解为头部或者脚注工具条在导航上的补充。基于导航工具条用户可以定义灵活的导航操作，而对于 `<header>` 和 `<footer>` 的使用也更加整齐。

导航工具条通过在 `<header>` 或者 `<footer>` 中定义 `<div data-role=" navbar" >` 来实现。其内部通过 `` 标签的嵌套包含各个导航，每行最多 5 个导航，超过 5 个则自动换行显示。

这是一段典型的头部导航工具条代码，

```
<header data-role=" header" data-theme=" b" >  
<h1> 导航工具条 </h1>  
<div data-role=" navbar" >  
  <ul>  
    <li><a href=" #" class=" ui-btn-active" > 第一栏 </a></li>  
    <li><a href=" #" > 第二栏 </a></li>  
    <li><a href=" #" > 第三栏 </a></li>  
    <li><a href=" #" > 第四栏 </a></li>  
  </ul>  
</div>  
</header>
```

如果希望导航按钮包含图标，可以使用 `data-icon` 进行定义。例如，

```
<footer>  
<div data-role=" navbar" >  
  <ul>  
    <li><a href=" #" data-icon=" arrow-l" class=" ui-btn-active" > 向前 </a></li>  
    <li><a href=" #" data-icon=" arrow-r" > 向后 </a></li>  
  </ul>  
</div>  
</footer>
```

同样，也可以通过 data-theme 对样式进行定义。

将头部和脚注导航工具条的代码合并在一起，

```
<!DOCTYPE html>
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
<section id=" MainPage" data-role=" page" data-title=" 导 航 工 具 条 " data-
theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 导航工具条 </h1>
    <div data-role=" navbar" >
      <ul>
        <li><a href=" #" class=" ui-btn-active" > 第一栏 </a></li>
        <li><a href=" #" > 第二栏 </a></li>
        <li><a href=" #" > 第三栏 </a></li>
        <li><a href=" #" > 第四栏 </a></li>
      </ul>
    </div>
  </header>
  <div class=" content" data-role=" content" >
    <p> 头部导航工具条或者脚注导航工具条，都可以通过定义单独的 Navbar 来实现丰富
的导航效果。 <br/>
  </p>
</div>
<footer>
  <div data-role=" navbar" >
```



```
<ul>
  <li><a href="#" data-icon=" arrow-l" class=" ui-btn-active" >向前 </a></li>
  <li><a href="#" data-icon=" arrow-r" >向后 </a></li>
</ul>
</div>
</footer>
</section>
</body>
</html>
```

其呈现效果如下，



6.5. 导航栏位置

导航栏位置有两种不同的定义方式，固定模式和全屏模式。

在固定模式下，页面被点击之后，导航条显示或者隐藏掉。而当页面移动到最上方，则正常页面的最上方的内容会被显示在导航栏的下部。而页面移动到最下方的时候，同样脚注导航栏也不会遮盖掉页面最下方的内容。

在全屏模式下，除非轻点屏幕，否则导航条不会隐藏掉，而这个时候导航条下的内容就被导航条遮盖住了。

定义固定模式的导航栏，代码原型为，

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

定义全屏模式的导航栏，代码原型为，

```
<div data-role="page" data-fullscreen=" true" >
  <div data-role=" header" data-position=" fixed" >
    <h1> 全屏模式导航栏 </h1>
  </div>
</div>
```

要注意，不同于固定模式导航栏定义的位置。在全屏模式下，data-fullscreen 定义的 data-role 为 page。也就是说，全屏的定义是在整个页面上。

6.6. 导航内容高亮

如果需要对导航内容进行高亮显示，则在需要高亮显示的条目上通过定义 class 为 ui-btn-active 进行高亮显示定义。在之前的代码范例中，向前按钮就这样被进行高亮显示的定义。

```
<footer>
<div data-role=" navbar" >
  <ul>
    <li><a href=" #" data-icon=" arrow-l" class=" ui-btn-active" > 向前 </a></li>
    <li><a href=" #" data-icon=" arrow-r" > 向后 </a></li>
  </ul>
</div>
```

</footer>

第 7 章 列表视图

列表视图，List View，是 HTML 语言在桌面上进行的列表优化。在传统的 HTML 桌面应用中，通过列表标签可以实现有层级的无数字列表，数字列表等功能。一些网站也通过对于列表标签进行 CSS 重新定义，而演化出菜单，表格等不同的呈现效果。

在移动应用开发领域，jQuery Mobile 通过

在这一章中，我们将接触到，

- 嵌套列表
- 数字列表
- 只读列表
- 分立按钮列表
- 缩略图
- 列表图标
- 查询列表内容
- 列表性能测试

7.1. 基本列表

在传统的桌面 HTML 开发中，列表视图通过 ``，`` 的嵌套进行标记。如果需要呈现为数字列表，可以使用 `` 取代 ``。

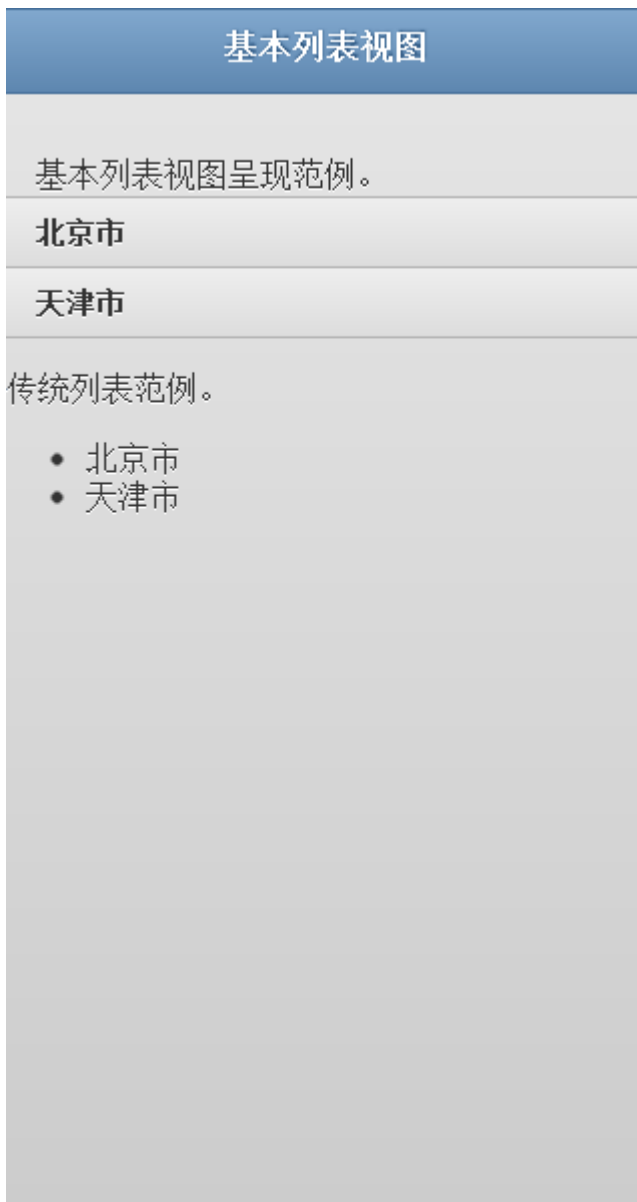
UI, Unordered Lists, 表示非编号列表。OL, Ordered Lists, 表示数字编号列表。Li, Lists, 表示为列表。

在移动应用开发中，jQuery Mobile 定义在传统的 ul 标记中通过声明属性 `data-role` 为 `list-view`，表示为列表视图。

下面的代码片段前半部分呈现的 jQuery Mobile 的列表视图样式。后半部分是传统的列表代码。两部分代码几乎都是一样的，只是前一部分代码有了 `data-role="listview"`，而后面的代码没有这个属性。

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 基本列表视图 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 基本列表视图呈现范例。 </p>  
    <ul data-role=" listview" >  
      <li> 北京市 </li>  
      <li> 天津市 </li>  
    </ul>  
  </div>  
  <p> 传统列表范例。 </p>  
  <ul>  
    <li> 北京市 </li>  
    <li> 天津市 </li>  
  </ul>  
</div>  
</section>
```

差异仅在于 data-role 属性的声明，而呈现效果如下，



有的时候，我们会看到 jQuery Mobile 或其他移动应用开发资料中提到 Read-only Lists 或者 Read-only inset lists，我将此直译为只读列表视图。只读列表视图并没有特别的设置，只要没有下一层列表嵌套，本身也没有链接指向，那么就是只读列表了。样式上，上面的这个例子，就可以看成一种简单的只读列表。

7.2. 嵌套列表

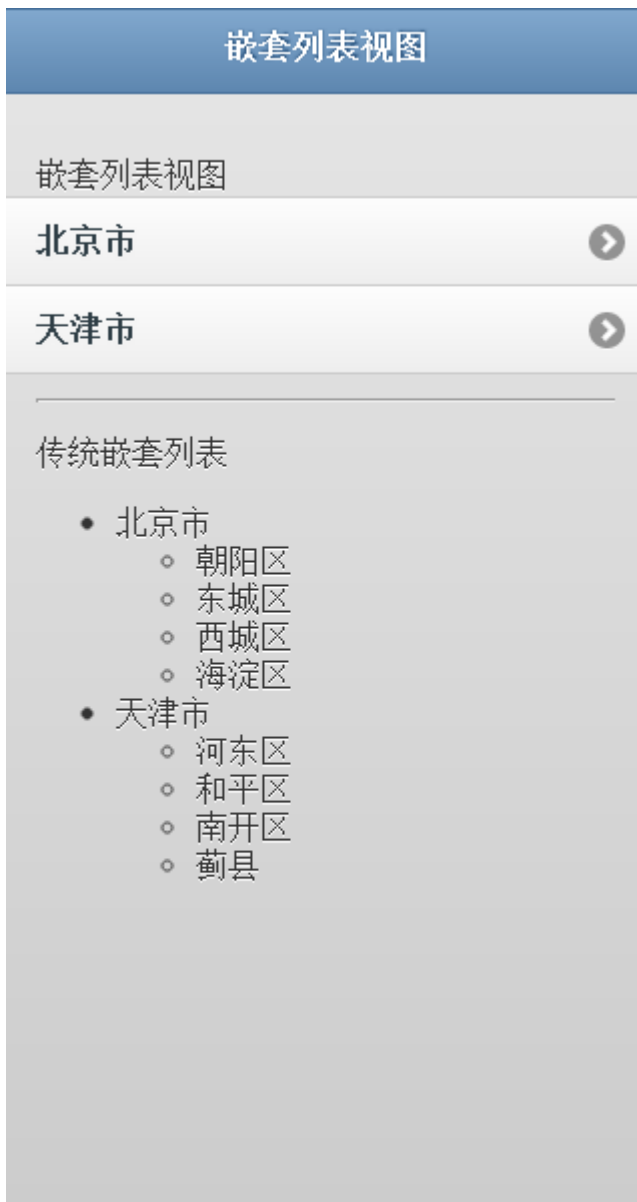
嵌套列表，顾名思义就是一个层列表之下，嵌套多一层的列表。如此一层层嵌套下去。开发嵌套列表的方法，就是在基本列表的基础之上，加入新的一层 `` 或者 `` 嵌套就可以实现。下一级的嵌套列表，将会继承上一级的 theme 等属性设置。

这段代码在之前基本列表视图的基础之上，实现了嵌套列表的功能，

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 嵌套列表视图 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 嵌套列表视图 </p>  
    <ul data-role=" listview" >  
      <li> 北京市  
        <ul>  
          <li> 朝阳区 </li>  
          <li> 东城区 </li>  
          <li> 西城区 </li>  
          <li> 海淀区 </li>  
        </ul>  
      </li>  
      <li> 天津市  
        <ul>  
          <li> 河东区 </li>  
          <li> 和平区 </li>  
          <li> 南开区 </li>  
          <li> 蓟县 </li>  
        </ul>  
      </li>  
    </ul>  
    <br/>  
    <hr/>  
    <p> 传统嵌套列表 </p>  
    <ul>  
      <li> 北京市  
        <ul>  
          <li> 朝阳区 </li>  
          <li> 东城区 </li>  
          <li> 西城区 </li>
```

```
<li> 海淀区 </li>
</ul>
</li>
<li> 天津市
<ul>
<li> 河东区 </li>
<li> 和平区 </li>
<li> 南开区 </li>
<li> 蓟县 </li>
</ul>
</li>
</ul>
</div>
</section>
```

这个界面呈现了 jQuery Mobile 列表视图和传统的嵌套列表之间的呈现样式差别,



注意，由于在这个列表中，北京市和天津市分别有更下一层的区县信息，所以在每个列表条目的右侧，出现了指向下一级的图标。这个图标在之前的基本列表中是没有的，jQuery Mobile 会根据是否存在下一层列表信息而自动呈现这个图标。

点击“北京市”这个列表信息之后，将会进入到北京市的列表信息中。

北京市
朝阳区
东城区
西城区
海淀区

有别于传统 jQuery Mobile 教程中所见，这里的列表框没有“返回”按钮。缺省是否具有“返回”按钮，是区别 jQuery Mobile 开发版本和后续版本的一个显著标志。当然，这个“返回”按钮是可以通程序去生成的。具体方法可以参见，《jQuery Mobile 无法自动显示返回按钮》(<http://www.catchexception.net/QuestionDetail/257/jquery%20mobile%20%E6%97%A0%E6%B3%95%E8%87%AA%E5%8A%A8%E6%98%BE%E7%A4%BA>) 这篇文章。

7.3. 数字列表

数字列表的主要特点在于，在每个列表项目之前，呈现序号标记。数字从 1 开始，自上而下依次递增。声明数字列表只需要将 HTML 标记定义为 OL (Ordered Lists)，而不再是 UL

(Unordered Lists) 就可以实现了。

这是一段数字列表的代码原型，

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 数字列表视图 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <p> 数字列表 </p>
    <ol data-role=" listview" >
      <li> 北京市
        <ol>
          <li> 朝阳区 </li>
          <li> 东城区 </li>
          <li> 西城区 </li>
          <li> 海淀区 </li>
        </ol>
      </li>
      <li> 天津市
        <ol>
          <li> 河东区 </li>
          <li> 和平区 </li>
          <li> 南开区 </li>
          <li> 蓟县 </li>
        </ol>
      </li>
    </ol>
  </div>
</section>
```

呈现效果如下，



北京市
1. 朝阳区
2. 东城区
3. 西城区
4. 海淀区

注意每个列表项目之前，出现了数字序数的编号了。

7.4. 分立按钮列表

分立按钮列表是 Split Button List 的直译，含义是，前面一部分是链接，后面有一个独立的列，其中是独立的图标按钮。用户点击前面的文字或者图片和点击之后的图标按钮可以触发不同的行为事件。

分立按钮列表呈现样式如下，



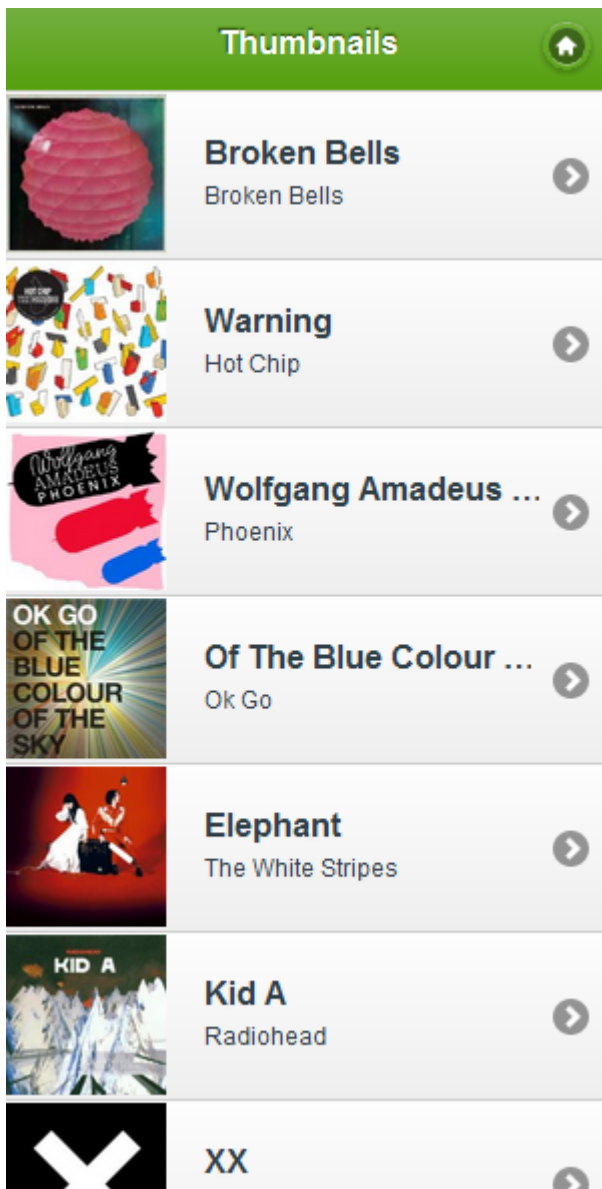
列表部分代码如下，

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 分立按钮列表视图 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 分立按钮列表视图呈现范例。 </p>  
    <ul data-role=" listview" >  
      <li><a href=" #" > 北京市 </a><a href=" #" data-icon=" plus" > 更多 </a> </  
li>  
      <li><a href=" #" > 天津市 </a><a href=" #" data-icon=" plus" > 更多 </a>
```

```
</ul>  
</div>  
</section>
```

7.5. 缩略图列表

缩略图列表是指，在列表项文字的前方包含有一个小的图片缩略图。下图是 jQuery Mobile 文档中关于缩略图列表的样式，



缩略图列表的实现，只需要在列表项目文字之前加入缩略图图片地址就可以了。

在实际生产系统中，设计缩略图还需要注意这样几个细节问题，

- 部署图片在独立域名之下，通过动静分离的手段进行访问，能获得更高的访问速度，也能改善用户体验。
- 图片最好经过压缩，专门为移动应用进行尺寸优化。很多移动用户的网页访问速度并不是很快。
- 辅之以支持本地缓存的技术，能大大改善用户体验。

7.6. 列表图标与气泡提示

在进行列表视图呈现的时候，可以加入对于下一级列表内容的统计数据或者其他相关数据。

在下面的界面，可以看到在界面右侧图标的左侧有了计数气泡。



实现提示气泡，需要在常见的基本列表之上完成两个步骤，

1. 将文字放置于超级链接标签 `<a>` 之中。否则一旦气泡被显示，则列表项文字就不能被显示。例如，
` 列表项文字 `。
2. 在超链的文字之后，添加气泡提示数字，
`123`

上面的界面代码是这样的，

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 数字列表视图 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 数字列表 </p>  
    <ol data-role=" listview" >  
      <li><a href=" #" > 北京市 <span class=" ui-li-count" >4</span></a>  
        <ol>  
          <li> 朝阳区 </li>  
          <li> 东城区 </li>  
          <li> 西城区 </li>  
          <li> 海淀区 </li>  
        </ol>  
      </li>  
      <li><a href=" #" > 天津市 <span class=" ui-li-count" > 天津市区县 </span></a>  
</li>  
    </ol>  
  </div>  
</section>
```

在上面的例子中，第一行为“北京市”后面跟着四个区县统计值。后一行是天津，后面的气泡信息为一段文字描述。

从实际气泡提示中，建议考虑这样几个技巧，

1. 如果部署在手机上，通常受到屏幕尺寸限制，气泡中虽然可以放入文字，但是尺寸可能

会比较大。而放入数字会比较恰当。

2. 不建议使用上面“北京市”的嵌套列表样式。而是用“天津市”的样式,从一个界面点击之后,跳转到后一个界面中。因为嵌套列表和气泡一同使用,则第二页将会把列表项内容和气泡内容显示在一起,而可能产生歧义。就好象下面界面这样,“北京市”的标题旁边出现气泡内容“4”。

也许未来会解决这个问题,但至少在 jQuery Mobile 1.0 RC2 中这个问题一直遗留下来。而从 jQuery Mobile 的文档来看,也没有对于嵌套列表和气泡提示的混合使用介绍。大概暂时这个问题还会遗留。好在使用上面的策略能有所应对。

北京市4
1. 朝阳区
2. 东城区
3. 西城区
4. 海淀区

7.7. 检索列表内容

通常在桌面应用开发过程中，检索列表中的内容会通过 Auto Complete 技术通过 Ajax 访问后台服务实现。这种技术假设，桌面设备具有良好的网络带宽条件，计算能力比较强。虽然在很多移动应用中也能支持 Auto Complete 技术实现智能提示，好像 GIS 查询。这里介绍一种非常接近的技术，只是数据源来自于这一章的主题，列表。

这是一个检索列表的界面，包含一些亚洲的城市。当输入过程中，自动将列表范围缩小到包含有输入文字的范围之内。



如果输入的文字为“粒”，则会出现包含有这个文字的城市“暹粒市”。



实现这样的效果，并不需要特别复杂的开发技术。只需要在列表视图定义的地方声明属性 `data-filter` 为 `true`，例如，

```
<ul data-role="listview" data-filter="true" > ... </ul>
```

上面程序的范例代码片段如下，

```
<section id=" MainPage" data-role=" page" data-title=" 列表视图演示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 查询列表视图 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <p> 查询列表视图内容范例。 </p>
```

```
<ul data-role="listview" data-filter="true" >
  <li>北京市 </li>
  <li>天津市 </li>
  <li>上海市 </li>
  <li>东京市 </li>
  <li>南京市 </li>
  <li>马尼拉市 </li>
  <li>暹粒市 </li>
</ul>
</div>
</section>
```

7.8. 列表性能测试

列表功能是一个比较消耗移动设备计算能力的功能。如果要检验目标客户移动设备的计算能力，jQuery Mobile 提供了一个参考方法，将典型的列表数据以循环方式进行呈现，以检查浏览器显示速度。

jQuery Mobile 列表性能测试的地址在，

<http://jquerymobile.com/test/docs/lists/lists-performance.html>

未来随着 jQuery Mobile 升级，这个测试地址的目录可能 test 会消失掉。所以还是要关注 jQuery Mobile 线上文档的变化情况。

如果自建测试，可以从 10 条列表项，到 20 条，50 条…逐条递增，以测试列表性能状况。另一种策略是，设定一种目标设备机型进行测试。先测试 500 条，如果性能表现不够好，再测试 250，然后再继续折半进行测试，直到定位到某个性能能够接受的区域之中。

在 jQuery Mobile 的测试地址中，给出了 500 条列表项。这个数量即便在台式机上使用 Google Chrome 显示速度也会很慢。所以直接应用这个用例进行移动设备测试，实际价值并不大。

第 8 章 表单

表单是在很早的时候 HTML 中就在提供的一种技术。我在 HTML 开发的时候，最早 96 年的时候，就在 Netscape 的编辑器中见到过这个的身影。当今天我们重新看到表单的时候，一种成熟的技术伴随着 HTML5 和 jQuery Mobile 发生了大变样。虽然功能还是一样的，但是更加简洁，对于开发者更有效率。在这个过程中，jQuery Mobile 又将界面呈现和对于表单元素的控制能力向前提升了一步。

在这一章中，我们将接触到，

- HTML5 表单
- 输入框
- 查询输入框
- 开关按钮，无线按钮和复选按钮
- 滑动条
- 选择菜单
- 表单风格

8.1. HTML5 表单

HTML5 表单虽然已经出现很多年，但是在生产环境大规模使用并不很多见，这主要受到主流互联网浏览器影响。随着移动设备对于 HTML5 支持能力普遍提升以及过去几年支持 HTML5 的桌面浏览器的普及，这种情况有可能在未来会有比较快的改观，特别是在移动计算领域。

小建议：

jQuery Mobile 表单是一个比较核心的功能组成。由于需要介绍一些 HTML5 表单特性，如果之前没有接触过 HTML 表单部分，这部分新内容可能阅读会有些挑战。不如先通过其他资料了解一些 HTML4 的表单内容，再来看这部分内容。

HTML5 表单主要增强了不同标签的呈现效果，增强的表单输入元素有很多，例如电子邮件，电话号码，输入文本，URL，本地不包含时区的时间日期输入框，数字，颜色，日期，时间，月份，按钮等等。这些在 jQuery Mobile 中也有所增强和扩展。

从实现功能来说，HTML5 提供了更多的扩展与元素功能管理能力，例如，通过 placeholder

对输入框进行说明，通过设置 `autocomplete` 来支持或者禁止自动提示功能等等。由于这部分涉及很多 HTML5 的特性，虽然和移动应用开发相关，却并不是这部分的重点。在这个版本的 jQuery Mobile 介绍中，将不进行详细介绍。未来这个学习资料升级过程中，将进行进一步详细介绍。更新内容，可以留意 [CatchException.NET](#) 所发布的版本更新。

从某种角度来说，jQuery Mobile 对于 HTML5 表单具有互补性。HTML5 的表单在提供规范功能方面，较之之前的 HTML4 有了很大的改善。但是这样的改善，并不是在于前端界面效果改善方面。在移动开发领域，jQuery Mobile 通过定义不同表单元素的呈现效果，将 HTML5 的表单元素呈现效果很大的改善。

8.2. 表单提交

和传统的表单提交方式一致，基于 HTML5 的 jQuery Mobile 表单提交方式，也是通过基于 `<form>` 标签的 `post` 或者 `get` 方法实现的。

表单提交的原型化代码是这样的，

```
<form id=" FormID" name=" FormName" method=" post" action=" index.php" >...</form>
```

8.3. jQuery Mobile 表单样式

大多数 jQuery Mobile 表单样式包含这样几个部分，

- 由 `<Form>` 标签包含的表单容器。
- `Fieldcontent`，声明一组表单元素起止位置。
- 所需要呈现的表单元素与定义。

缺省情况下，表单元素的样式将会继承页面或者表单样式。所以在 HTML 代码中，不同于其他页面元素，表单元素的风格即便不进行声明，也是具有 jQuery Mobile 风格的。

关于 jQuery Mobile 各个表单元素所呈现的样式，可以访问范例代码下载资源中，`Section08` 目录下 `FormElement.html` 这个文件。因为界面截屏尺寸有些过长，这一章后续内容中会进行相应部分的截屏。

jQuery Mobile 表单样式的代码框架如下，

```
<!DOCTYPE html>
```

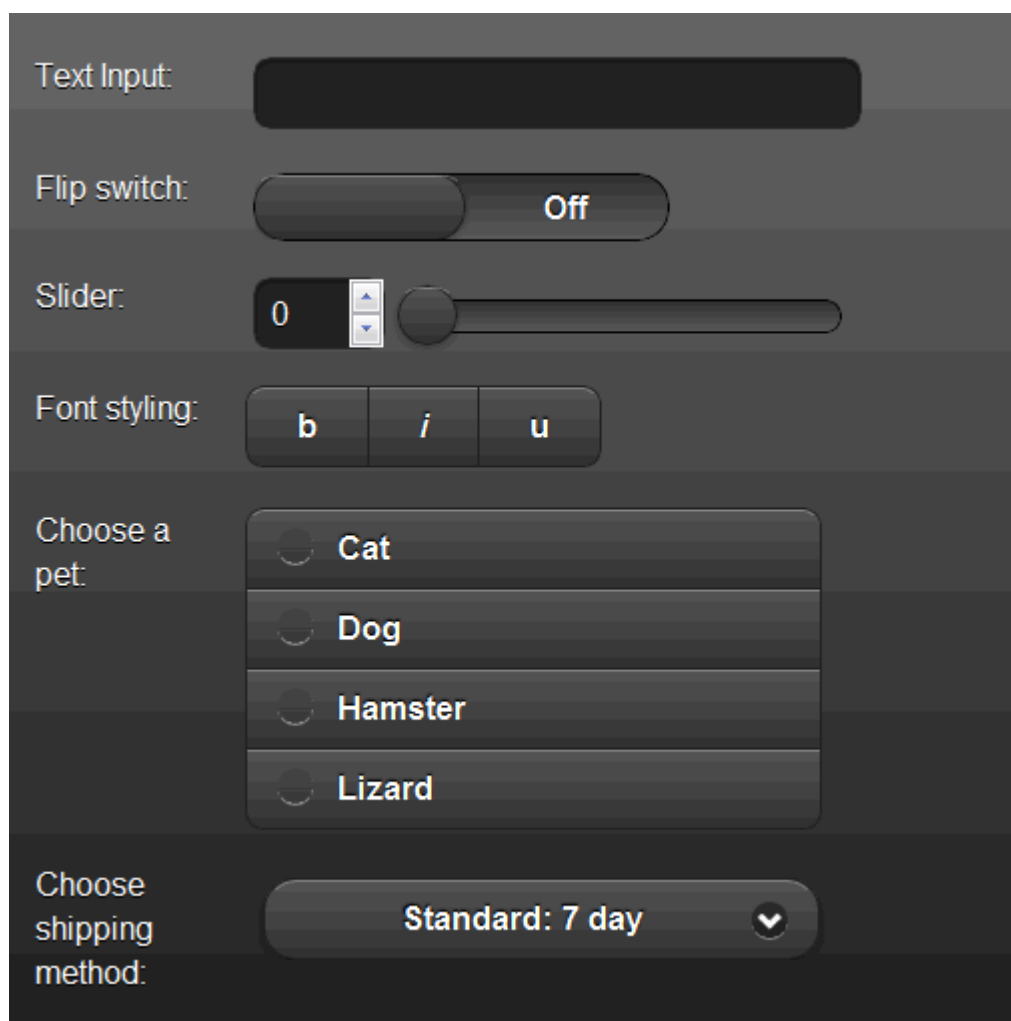
```
<html>
<head>
<meta charset=" utf-8" >
<title> 跨平台移动应用 </title>
<link rel=" stylesheet" href=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.css"
/>
<script src=" ../js/jquery-1.6.4.min.js" ></script>
<script src=" ../js/jquery.mobile-1.0rc2/jquery.mobile-1.0rc2.min.js" ></script>
</head>
<body>
<section id=" MainPage" data-role=" page" data-title=" 表单元素演示" data-
theme=" b" >
  <header data-role=" header" data-theme=" b" >
    <h1> 表单元素演示 </h1>
  </header>
  <div class=" content" data-role=" content" >
    <form action=" #" method=" get" >
      这里插入表单元素的代码。
    </form>
  </div>
</section>
</body>
</html>
```

8.4. 表单风格

如果进行特别声明，则表单样式可以呈现不同系统定义的不同的风格样式。

如果需要元素表单呈现不同的风格样式，需要在元素表单的显示范围上，添加这样的原型化标签，`<div class=" ui-body ui-body-a" >…</div>`。其中 ui-body-a 到 ui-body-e，总共五种不同的界面风格定义，分别由 jQuery Mobile 定义了不同的风格样式。差异主要表现在配色上，如果感兴趣可以根据进行尝试。

下面界面样式是声明为 ui-body-a 的界面风格样式，



8.5. 浏览器原生样式

可以通过声明，以传统 HTML 样式而不是 jQuery Mobile 样式进行呈现。这样只需要在需要在相关表单元素中声明 `data-role="none"` 属性代码。这样这个表单元素就会以原生表单元素样式呈现。例如，`<input type="text" name="name" id="name" value="" data-role="none" />`，就将文本输入框声明为浏览器原生的样式进行显示。

8.6. 输入框

在基于 HTML5 的 jQuery Mobile 框架中，常用的输入框列表如下。

名称	Type	功能
文本输入框	text	传统输入框，实现文本输入
密码输入框	password	输入密码的输入框，相当于 <code><input type="password"/></code> 的功能。

数字输入框	number	能够输入数字的输入框。与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
电子邮件输入框	email	能够输入电子邮件 在部分浏览器，如 Google Chrome，能够从之前输入过的电子邮件中进行匹配并自动提示可能的电子邮件地址。
URL 地址输入框	url	支持输入 URL 地址 在部分浏览器，如 Google Chrome，能够从之前输入过的地址中进行匹配并自动提示可能的电子邮件地址。
电话号码输入框	tel	支持输入电话号码 在部分浏览器输入中，对此进行特别的优化设计
时间输入框	time	支持输入时间 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
日期输入框	data	支持输入日期 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
月份输入框	month	支持输入月份 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
周输入框	week	显示年份，以及相关时间在所在年份的周数。例如 2011-W41，表示 2011 年第 41 周。 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
时间日期输入框	datetime	日期与时间，时区为 UTC 时间 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
本地时间日期输入框	datetime-local	本地时区之下的日期和时间 与 HTML4 不同之处在于，可以通过微调按钮的样式进行上下调整。
色彩输入框	color	支持输入色彩值
查询框	search	具有查询图标的输入框。

输入框部分代码原型如下，

```
<div data-role=" fieldcontain" >  
  <label for=" name" > 文本输入框: </label>  
  <input type=" text" name=" name" id=" name" value=" " />  
</div>
```

这部分代码的实现与效果，可以参见范例代码 Section08 中的相关内容。

8.7. 开关按钮

开关按钮呈现如下的界面效果，点击之后，开关值进行切换。功能类似与无线按钮或者下拉菜单的功能。



开关按钮的代码结构如下，

```
<div data-role=" fieldcontain" >
  <label for=" slider2" > 开关按钮: </label>
  <select name=" slider2" id=" slider2" data-role=" slider" >
    <option value=" off" > 关 </option>
    <option value=" on" > 开 </option>
  </select>
</div>
```

这里虽然叫做开关按钮，但是和按钮有所不同，点击开关按钮会发生状态切换但是并不立刻执行动作。只有当在表单提交的时候，开关按钮的选项数值会被提交到服务器上。所以开关按钮更多的是对界面效果的一种形象化描述。

当然也可以通过 JavaScript 程序对不同开关出发进行响应。

8.8. 无线按钮

jQuery Mobile 无线按钮和其他桌面应用或者移动应用的无线按钮功能是一样的。无线按钮的名称自 90 年代就有了，这个名称并不直观，是从英文 radio 过来的直译。每次点击无线按钮一次，则只能有一个选项被选中。这和复选按钮能够多选形成对比。

界面样式如下，



无线按钮代码结构如下，

```
<div data-role=" fieldcontain" >
  <fieldset data-role=" controlgroup" >
    <legend> 无线按钮 :</legend>
    <input type=" radio" name=" choice-1" value=" choice-1" checked=" checked"
  />
  <label for=" choice-1" > 变态辣 </label>
  <input type=" radio" name=" choice-1" value=" choice-2" />
  <label for=" choice-2" > 很辣 </label>
  <input type=" radio" name=" choice-1" value=" choice-3" />
  <label for=" choice-3" > 中辣 </label>
  <input type=" radio" name=" choice-1" value=" choice-4" />
  <label for=" choice-4" > 微辣 </label>
  <input type=" radio" name=" choice-1" value=" choice-5" />
  <label for=" choice-5" > 一点不辣 </label>
</fieldset>
</div>
```

有的时候，用户界面中单选按钮并不是垂直排列的，也会出现水平排列的情况。就好象这样，



缺省的无线按钮排列为自上而下依次排列。而如果需要无线按钮能水平排列，则需要 在 <fieldset> 标签中声明 data-type=" horizontal" 属性。需要注意的是，水平排列需要针对浏览器分辨率进行设置，一行文字过多而出现折行，则呈现效果可能会变得很糟糕。

水平无线按钮的代码结构如下，

```
<div data-role=" fieldcontain" >
  <fieldset data-role=" controlgroup" data-type=" horizontal" >
    <legend> 水平无线按钮 :</legend>
    <input type=" radio" name=" radio-choice-b" value=" 001" checked=" checked"
  />
```

```
<label for=" radio-choice-c" >变态辣 </label>
<input type=" radio" name=" radio-choice-b" value=" 010" />
<label for=" radio-choice-d" >中辣 </label>
<input type=" radio" name=" radio-choice-b" value=" 100" />
<label for=" radio-choice-e" >微辣 </label>
</fieldset>
</div>
```

8.9. 复选按钮

复选按钮，有的技术文档也称作多选按钮，沿袭了 HTML4 的标签定义格式。



同单选按钮类似，复选按钮也是通过在 `<fieldset>` 标签中声明 `data-type=" horizontal"` 实现从缺省垂直排列转向水平排列。一样需要注意的是，水平排列需要针对浏览器分辨率进行设置，一行文字过多而出现折行，则呈现效果可能会变得很糟糕。

复选按钮的代码结构如下，

```
<div data-role=" fieldcontain" >
  <fieldset data-role=" controlgroup" >
    <legend> 多选按钮: </legend>
    <input type=" checkbox" name=" checkbox-1a" id=" checkbox-1a" />
    <label for=" checkbox-1a" >水饺 </label>
```

```
<input type="checkbox" name="checkbox-2a" id="checkbox-2a" />
<label for="checkbox-2a" > 小笼包 </label>
<input type="checkbox" name="checkbox-3a" id="checkbox-3a" />
<label for="checkbox-3a" > 煎饼果子 </label>
<input type="checkbox" name="checkbox-4a" id="checkbox-4a" />
<label for="checkbox-4a" > 茶汤 </label>
</fieldset>
</div>
```

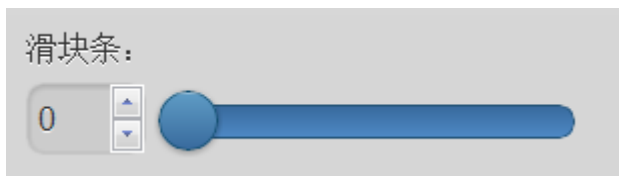
下面代码为水平排列的复选按钮。

```
<div data-role="fieldcontain" >
  <fieldset data-role="controlgroup" data-type="horizontal" >
    <legend> 水平多选按钮 :</legend>
    <input type="checkbox" name="checkbox-6" id="checkbox-6" />
    <label for="checkbox-6" > 清淡 </label>
    <input type="checkbox" name="checkbox-7" id="checkbox-7" />
    <label for="checkbox-7" > 少盐 </label>
    <input type="checkbox" name="checkbox-8" id="checkbox-8" />
    <label for="checkbox-8" > 加辣 </label>
  </fieldset>
</div>
```

8.10. 滑动条

操作滑动条，用户可以直接在一个水平轴上拖动滑块，好像过去我们看到的卖蔬菜的秤上，在秤杆上移动秤砣可以抵达不同的刻度位置而读书后，获得蔬菜重量。和秤杆不同，滑动条可以设置最小值和最大值，这样滑动操作的效率会更高。

通常，每次滑动滑动条，数字会增加 1。如果滑动条数值范围很大，从 0 到 1000，则可能每次滑动条操作，数字会增加的范围比较大。这个时候每次操作的增加可能就不仅仅是 1 了，而如果需要获得精确数字，用户则需要在前方的数字输入框进行重新输入。这并不是一个好的用户体验。



滑动条和数字输入框是两种不同的输入方式。相比之下，如果需要输入某个确定的数字，或者在某个数字之上进行微调，我会倾向于使用数字输入框。而需要设定数字在某个范围的位置上，则使用滑动条更加直观。都是能进行数字输入，要看用户体验和界面版式的设计。当然，数字输入框的输入和微调按钮在滑动条前方也是具有的。所以滑动条会占用更多的移动设备屏幕空间。

滑动条设置最小值属性为 min，最大值属性为 max。代码结构如下，

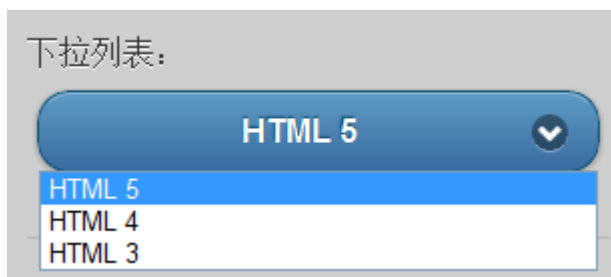
```
<div data-role=" fieldcontain" >  
  <label for=" slider" > 滑块条: </label>  
  <input type=" range" name=" slider" id=" slider" value=" 0" min=" 0"  
max=" 100" />  
</div>
```

8.11. 选择菜单

选择菜单可能有下拉列表方式，或者以浮动窗口的样式所显示的选择菜单。下拉列表和列表菜单只是两种不同的呈现效果，其功能是一样的，甚至如果列表内容比较少，使用无线按钮也可以起到同样的作用。



不同的选择菜单所呈现打开之后的样式也是不同的。下拉列表呈现样式如下，



下拉菜单代码结构如下，

```
<div data-role=" fieldcontain" >
  <label for=" select-choice-1" class=" select" > 下拉列表: </label>
  <select name=" select-choice-1" id=" select-choice-1" >
    <option value=" HTML5" >HTML 5</option>
    <option value=" HTML4" >HTML 4</option>
    <option value=" HTML3" >HTML 3</option>
  </select>
</div>
```

而列表菜单则是这个样子的，



这是列表菜单的代码结构，可以看出，与之前下拉菜单之间的差异在于 <select> 标签中增加了 data-native-menu=" false" 的属性，而其他则是一样的。

```
<div data-role=" fieldcontain" >
  <label for=" select-choice-a" class=" select" > 列表菜单: </label>
  <select name=" select-choice-a" id=" select-choice-a" data-native-menu=" false" >
    <option> 选择你钟爱的语言 </option>
```

```
<option value=" PHP" >PHP</option>  
<option value=" Python" >Python</option>  
<option value=" VB" >Visual Basic</option>  
<option value=" CSharp" >C#</option>  
</select>  
</div>
```

如果用户界面空间有限，而呈现内容很多，则使用下拉列表，文字会紧凑，字号和行间距都比较小，能容纳更多列表内容。如果列表内容不那么多，就可以考虑列表菜单，这种界面美观而依然可以在一屏之内显示比较多的内容。如果就只有三五条内容选择，则可以选择垂直排列的无线按钮。如果就是两三条内容，每条内容文字几个字，则水平排列的无线按钮更能有效的利用用户界面空间，传递尽量多的内容给用户。

8.12. 表单元素禁用

如果要对某个表单元素设置为禁用，则可以通过设置 CSS 为 ui-disabled 实现。当需要重新使这个表单元素可用，可以将 ui-disabled 移除 CSS 就可以了。作为一种 HTML 的表述方式，也可以通过传统的设置 disable 和 enable 来实现表单元素禁用。

由于设置为禁用之后，表单元素的 label 的显示效果会一同发生变化，所以建议使用 jQuery Mobile 的 ui-disabled 来进行设置。

下面的界面是一个被标记 us-disabled 的表单输入元素的样式。



这段表单元素禁用的代码如下，

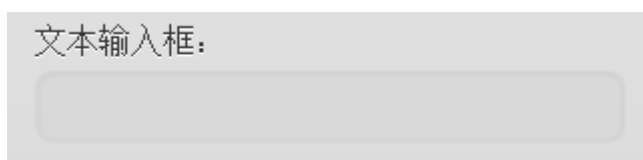
```
<section id=" MainPage" data-role=" page" data-title=" 表 单 元 素 演 示 " data-  
theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 表单元素演示 </h1>  
  </header>  
  <div class=" content" data-role=" content" >  
    <form action=" #" method=" get" >  
      <p> 包含 ui-disabled 定义，设置 Form 元素为无效 </p>  
      <div data-role=" fieldcontain" class=" ui-disabled" >  
        <label for=" name" > 文本输入框: </label>  
        <input type=" text" name=" name" id=" name" value=" " />  
      </div>  
    </div>  
  </form>  
</section>
```

```
</div>
<div data-role=" fieldcontain" class=" ui-disabled" >
  <label for=" search" > 查询输入框: </label>
  <input type=" search" name=" password" id=" search" value=" " />
</div>
</form>
</div>
</section>
```

在阅读这段代码的时候，可以注意到 class=" ui-disabled" 被标记在 data-role=" fieldcontain" 中。如果标记在这个 DIV 容器中的某个表单元素之上，则被禁用就只在这个表单元素之上。例如，将上述代码修改为，

```
<div data-role=" fieldcontain" >
  <label for=" name" > 文本输入框: </label>
  <input type=" text" name=" name" id=" name" value=" " class=" ui-disabled" />
</div>
```

则，Label 标签则不再为禁用状态，但是之后的 Text 文本框依然为禁用状态。



8.13. 隐藏标签

在很多应用场景中，移动应用界面尺寸比较有限，这时候就需要尽量有效的利用可用的界面空间。jQuery Mobile 提供一种隐藏标签的技巧，可以并不在显示 Label 标签，却依然能够指示之后的表单元素的作用。在这种技巧显示的时候，标签不被显示，但是可以在输入框中显示原来标签的功能。

实现隐藏标签需要两个步骤，

1. 在需要隐藏标签的 <div data-role=" filedcontain" > 容器中，加入 class=" ui-hide-label" 标记。
2. 在被隐藏 Label 标签的输入框中，添加 placeholder 属性，并将原有 Label 内容赋值给

placeholder。

Placeholder 不是 jQuery Mobile 新增加的属性，而是 HTML5 的在表单功能上支持的新特性。在使用这个技巧的时候，要注意不是所有表单元素都支持隐藏，例如 TextArea。或许未来可以，但是现在还不行。所以使用的时候要比较小心。

下面的界面是设定 placeholder 之后，包含 ui-hide-label 和没有包含的比较。



隐藏标签的代码原型如下所示，

```
<section id=" MainPage" data-role=" page" data-title=" 表 单 元 素 演 示 " data-theme=" b" >  
  <header data-role=" header" data-theme=" b" >  
    <h1> 表 单 元 素 演 示 </h1>  
  </header>
```

```
<div class=" content" data-role=" content" >
  <form action=" #" method=" get" >
    <p> 包含 ui-hide-label 定义 </p>
    <div data-role=" fieldcontain" class=" ui-hide-label" >
      <label for=" name" > 文本输入框: </label>
      <input type=" text" name=" name" id=" name" value=" " placeholder=" 文
本输入框 "/>
    </div>
    <div data-role=" fieldcontain" class=" ui-hide-label" >
      <label for=" search" > 查询输入框: </label>
      <input type=" search" name=" password" id=" search" value=" " placehold-
er=" 查询输入框 "/>
    </div>
  </form>
</div>
<hr/>
<form action=" #" method=" get" >
  <p> 没有包含 ui-hide-label 定义 </p>
  <div data-role=" fieldcontain" >
    <label for=" name" > 文本输入框: </label>
    <input type=" text" id=" name" value=" " placeholder=" 文本输入框 "/>
  </div>
  <div data-role=" fieldcontain" >
    <label for=" search" > 查询输入框: </label>
    <input type=" search" id=" search" value=" " placeholder=" 查询输入框 "/>
  </div>
</form>
</div>
</section>
```

8.14. 选项、方法和事件

大多数表单元素都具有 theme 和 initSelector 选项。部分表单元素还在此之上，对于选项内容进行了扩展。我们首先来看通用的两个选项内容，然后分别看到不同之处。

8.14.1. 选项

选项名称:	theme
选项类型:	string
缺省值:	null, 继承自上一级
功能:	缺省的按钮风格是 null, 表示按钮风格继承自上一级 div 或者 section。如果上一级也没有定义风格, 则使用缺省风格。 风格定义自 a 到 e, 定义 5 中不同配色样式。
标签属性:	data-theme
选项名称:	initSelector
选项类型:	CSS 选择器字符串
缺省值:	button, [type='button'], [type='submit'], [type='reset'], [type='image']
功能:	用于指定 CSS 选择器。
标签属性:	无

虽然 initSelector 内容和格式大致相同, 在面对不同表单元素的时候, 缺省值却会有些差异。

表单元素	缺省值
输入框与查询框	default: "input[type='text'], input[type='search'], :jqmData(type='search'), input[type='number'], :jqmData(type='number'), input[type='password'], input[type='email'], input[type='url'], input[type='tel'], textarea, input:not([type])"
滑动条、开关按钮、	default: "input[type='range'], :jqmData(type='range'), :jqmData(role='slider')"
无线按钮	无此选项
复选按钮	无此选项
选择菜单	default: "select:not(:jqmData(role='slider'))"

8.14.2. 方法

jQuery Mobile 的表单元素方法通过 jQuery 选择器获得某个或者某一类表单元素实例之后, 执行相应方法内容。

方法名称:	enable
功能:	按钮使能
示例:	<code>\$('.selector').textinput('enable');</code>
方法名称:	disable
功能:	按钮禁用
示例:	<code>\$('.selector').textinput('disable');</code>

8.14.3. 事件

jQuery Mobile 的表单元素事件, 主要是 create 事件, 在创建表单对象的时候触发这个事件。虽然继承自 jQuery, 但是 jQuery Mobile 的事件管理和 jQuery 的事件管理略微有些差异。建议使用 jQuery Mobile 的事件管理而不是 jQuery 的。

代码原型,

```
$( “.selector” ).textinput({  
  create: function(event, ui) { ... }  
});
```

附录 A. 答疑资源

文档

jQuery Mobile 是一种快速发展的技术，在学习的时候建议以 jQuery Mobile 官方网站的文档为准。当前 jQuery Mobile 为 1.0 RC1 版本，所以文档地址位于，<http://jquerymobile.com/demos/1.0rc1/>

不同版本，文档内容会有些差异。通常而言，jQuery Mobile 当前的稳定性已经比较强，从开发者角度来看，文档的变化幅度通常不会很大。

问答

如果在学习过程中存在的问题，可以访问 CatchException.NET 工程师技术社区进行互动问答。也欢迎在学习过程中，将有价值的经验和方法以自问自答的形式提交到 CatchException.NET 社区上来。

沙龙

CatchException.NET 每个月会有若干次自由主题沙龙活动，在茶馆或者咖啡厅举行。沙龙中，潜伏着一些使用 jQuery Mobile 或者 Sencha Touch 等开发框架进行 Web 应用开发的资深工作人员。技术社区的互动也许能对于学习 jQuery Mobile 等开发框架具有一些帮助。

讲座

当前有关于 jQuery Mobile 的中文讲座资源并不是很多。如果发现，请发布到 CatchException.NET 上，分享给其他应用开发工程师吧。

CatchException.NET 会在 2011 年 10 月份组织一期 jQuery Mobile 相关的跨平台移动应用开发技术讲座。如果感兴趣欢迎参加。之后，也会陆续组织其他形式的技术研讨活动。

附录 B. 升级计划

作为一个快速发展的 JavaScript 技术框架, jQuery Mobile 在过去不断快速进行着完善和升级。同样, 作为技术资料, 这套探讨 jQuery Mobile 的电子书, 也将随着软件的升级而不断升级和完善。

更新的电子书内容, 将会发布在 CatchException.NET 网站。你也可以通过新浪微盘等工具进行下载。

后续电子书的升级计划与发布信息可以关注 CatchException.NET 的相关报道, 也可以通过关注 CatchException.NET 的微博信息获得。

CatchException.NET 的微博地址为, <http://weibo.com/2169646091>