**[jQuery Mobile 入门教程](http://www.cnblogs.com/filod/archive/2011/08/15/2135399.html)**

原作：[Getting started with jQuery Mobile](http://www.adobe.com/devnet/dreamweaver/articles/getting-started-with-jquery-mobile.html" \t "_blank) —— [Matthew David](http://www.adobe.com/devnet/author_bios/matthew_david.html)

翻译：[filod](http://www.filod.net)

译文：<http://blog.filod.net/jquerymobile-2/295.html>

转载声明：请注明原作、翻译以及译文链接。

你每天都会对着它讲话，和它玩游戏，用它看新闻——没错，它就是你裤兜里的智能手机。android，黑莓还是iphone？为了让你清楚意识到究竟哪些才算是智能手机，我在下面总结了一个智能手机系统/设备的列表：

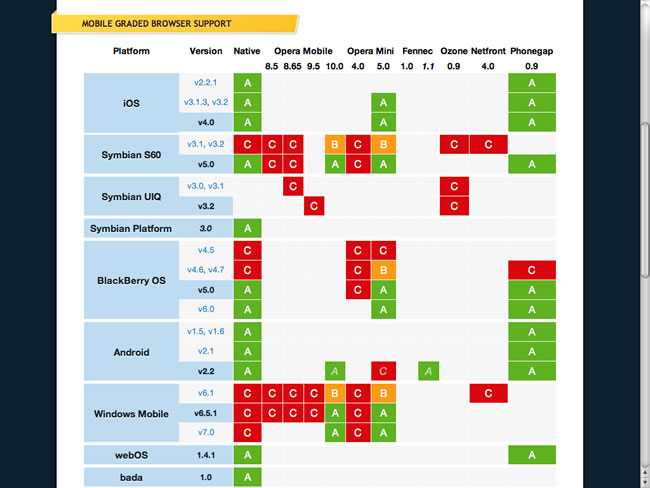
* Apple iPhone/iPod Touch
* Google Android
* RIM BlackBerry/Playbook OS
* Nokia Symbian (我承认它该退休了，可是在世界范围内它仍然拥有10亿以上的用户）
* HP/Palm WebOS
* Microsoft Window Phone 7

没错，这令人眼花缭乱的众多智能手机系统对HTML标准支离破碎的支持，直接导致了处理这些系统浏览器的兼容性成为了你最大的挑战。举例来说，苹果的iphone对HTML5标准有着很好的支持，然而Symbian和Microsoft Phone系列却几乎不支持HTML5的任何特性（译注：准确的说，是windows mobile系列，目前的windows phone 7对HTML5支持很好）。jQuery社区为了在解决兼容性和高效创建移动web站点做了大量努力，并开发了jQueryMobile这个移动框架。而本文旨在为读者介绍仍在alpha版本阶段的jQueryMobile框架（译注：目前jQueryMobile已经为beta2版本）的开发基础知识和技巧。

**用jQueryMobile来开发网站**

Gartner study(link)的数据表明在未来将会有越来越多的用户通过手机或者平板电脑访问你的网站，到2013年，将会有18亿的移动设备用户，对你的web设计团队来说开发移动站点已经不能再仅仅是“计划”而已了，你必须将你的移动站点迅速变为“现实”才行。

在美国，IOS和Android设备占了很大比重，而美国以外的地方主要是Nokia的设备占统治地位。jQueryMobile为了在尽可能多的设备上运行，承诺将支持绝大部分移动设备/系统。在早些时候（2010年9月），jQuery的作者John Resig透露了一张jQueryMobile对各种设备/系统的支持表格（见下）。对于每个系统来说，jQueryMobile将他们的支持情况分为A（好），B（一般），C（基本）三个等级。



由于移动设备市场的变幻莫测，这张图表也会迅速变化（一个很好的例子就是目前Nokia正在与Microsoft合作生产基于Windows Phone 7的设备——而在jQueryMobile中WP7的支持等级为A）。

jQueryMobile始终贯彻***渐进增强***的设计观念来满足你的开发需要。jQuerymobile的核心能使得基本的HTML标签在所有的浏览器中生效，在此基础之上，再对网页的行为和效果进行增强，这意味着你的网页在等级较高的浏览器中能获得非常优秀的体验，而在较差的浏览器也能正常的使用。

繁杂的移动浏览器们带来了巨大的挑战。一方面某些浏览器（例如Android的原生浏览器，Safari移动版等）都基于WebKit的一个变种版本（WebKit是一个web渲染引擎，Google Chrome桌面版，Apple的Safari都采用了该引擎。WebKit并不知道当前运行的网络的好坏，操作系统是什么甚至屏幕滚动到了哪里，为了让WebKit知道这些情况，操作系统/浏览器/设备厂商都需要基于WebKit来构建自己的浏览器程序），并带有丰富的特性。另一方面Nokia的Symbian和Windows Mobile 6（及更早的版本）这些胡乱支持标准的设备又占有相当大的市场。雪上加霜的是，WebKit在不同移动设备中还有不同的版本。所以渐进增强的基本目标就是你的内容能够在任何的设备中都能够正常“显示”。

**jQueryMobile使用入门：**

使用jQueryMobile的第一步，先创建一个html页面，并在head标签中加入以下内容：

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a4.1/jquery.mobile-1.0a4.1.min.css">  
 <script src="http://code.jquery.com/jquery-1.5.min.js"></script>  
 <script src="http://code.jquery.com/mobile/1.0a4.1/jquery.mobile-1.0a4.1.min.js"></script>

正如你在代码中看到的，jQueryMobile是jQuery的一个扩展。目前来说，压缩后的jQuerymobile仅12Kb。

上面的代码均来自jQuery的CDN服务器，css文件中也包含必需的图片链接，如果你需要在你自己的服务器上运行，可以下载下面的文件解压缩后部署到你的服务器上：

[Zip File: jquery.mobile-1.0b2.zip](http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.zip)

在创建第一个jQueryMobile页面时你需要创建三块基本内容，下面的推荐模版展示了这一过程，你可以在未来的项目中使用它：

<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">  
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a4.1/jquery.mobile-1.0a4.1.min.css">  
<script src="http://code.jquery.com/jquery-1.5.min.js"></script>  
<script src="http://code.jquery.com/mobile/1.0a4.1/jquery.mobile-1.0a4.1.min.js"></script>  
<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><!-- /header --></div><!-- /page -->



在模版中有些地方值得我们注意。首先是DIV元素的使用，我们知道，既然HTML5在移动设备中如此流行，为什么不使用更加新的HEADER, ARTICLE, SECTION, FOOTER元素呢？这是因为较老的智能手机浏览器无法明白新的HTML5元素。在某些情况下，例如windows phone上老版本的IE会出现一个bug使得无法加载页面的css。而DIV元素却被广泛支持。

此时你已经可以保存你的网页并在浏览器中查看了，这些代码同样可以在桌面浏览器中正常工作。我推荐你使用Chrome来进行本地测试。要在真实环境测试，你就需要相应移动设备了。

**使用超链接**

普通网页和移动网页的一个巨大的不同便是屏幕中呈现内容的数量多寡上。虽然你可以在你的iPhone上加载纽约时报的主页，但你需要缩放它才能顺利阅读上面的内容。这样的体验并不好，而更好的解决方案是减少那些杂乱的内容，只在屏幕上显示你需要显示的内容。

如果是传统的网站，你可能会创建一些包含少量内容的子页面，而当你使用jQueryMobile时，你最好在页面中包含“微量”的内容，这样才会更有效率。

在上面例子中你已经看到了如何利用模版来创建一个页面。现在让我们更进一步，来创建内容的“page”。jQueryMobile中的一个“page”结构一般使用一个DIV来组织。现在你可以使用上面的模板来创建一个包含四个跳转到其他页面的链接的导航页面：

<!-- Start of first page -->  
<div data-role="page" id="menu">  
 <div data-role="header">  
 <h1>Menu</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
 <p>What vehicles do you like?</p>   
 <p><a href="#Cars">Cars</a></p>  
 <p><a href="#Trains">Trains</a></p>  
 <p><a href="#Planes">Planes</a></p>   
 </div><!-- /content -->  
 <div data-role="footer">  
 <h4>Page Footer</h4>  
 </div><!-- /footer -->  
</div><!-- /page -->



上面这段代码中第一个div非常重要，它包含有一个id和一个data-role属性：

data-role="page" id="menu"

data-role定义了这个div是一个“page”，page这个术语稍微有点让人误解，“page”这里其实指的是一个可视面或者在屏幕里未隐藏的HTML代码部分，而不是指的一个单独的页面（或者说单独的HTML文件）。data-role="page" 意味着jQueryMobile会根据div元素在屏幕中构建可视内容。而id属性允许你通过a标签链接到该page，或者其他page。

上面创建的导航页是我们在浏览器看到的第一个页面，接下来我们再添加三个“page”，他们有不同的id：Cars， Planes，Trains。

<div data-role="page" id="Cars">  
 <div data-role="header">  
 <h1>Cars</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
 <p>We can add a list of cars</p>  
 </div><!-- /content -->  
 <div data-role="footer">  
 <h4>Page Footer</h4>  
 </div><!-- /footer -->  
</div><!-- /page -->  
<!-- Start of third page -->  
<div data-role="page" id="Trains">  
 <div data-role="header">  
 <h1>Trains</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
 <p>We can add a list of trains</p>  
 </div><!-- /content -->  
 <div data-role="footer">  
 <h4>Page Footer</h4>  
 </div><!-- /footer -->  
</div><!-- /page -->  
<!-- Start of fourth page -->  
<div data-role="page" id="Planes">  
 <div data-role="header">  
 <h1>Planes</h1>  
</div><!-- /header -->  
 <div data-role="content">   
 <p>We can add a list of planes</p>  
 </div><!-- /content -->  
 <div data-role="footer">  
 <h4>Page Footer</h4>  
</div><!-- /footer -->  
</div><!-- /page -->

现在，在你的Android或者IOS设备里测试一下，当你加载好页面后你会发现这三件事情：

* 导航页出现在屏幕中（你可以上下滚动一下，并没有别的东西出现）
* 当你点击一个链接时，会动画切换到另一个页面。
* 新页面的顶部会自动出现一个“back”按钮（译注：jQueryMobile目前版本默认已经取消了这一功能）

其实这些div元素预先会加载并缓存到你的浏览器中，因此“页面”间的切换会非常流畅。

在同一个HTML页面创建多个在屏幕上显示的“页面”使得你可以大大减少页面加载的次数，但同时也会导致许多移动设备运行缓慢。jQueryMobile将页面所有的链接跳转都视作Ajax调用，这样可以充分利用CSS的过渡效果，当你想要链接到你自己的web程序之外的某些链接时，你可以这样编写你的链接代码：

<a href="http://www.madinc.co" rel="external">madinc.co</a>

如上所示，仅需要为a标签添加 rel="external"属性即可。然而jQueryMobile对于（同域的）外部链接并不是简单地跳转完事儿，相比于其他移动框架它更进了一步，因为他对（同域的）所有链接都采用Ajax调用方式，从而实现漂亮的转场效果。基于此你可以将你的网页内容分离到许多页面来创建更大型的项目。

**使用组件**

链接和页面只是移动网页设计中一个很小的部分，APP程序（比如采用Object-C，C#，java等在Android或者IOS设备上创建的本地应用程序）快速增长的同时产生了丰富的控件和组件（例如菜单栏，列表等控件）使得开发者可以很方便地创建复杂的应用程序，这正是许多移动网页开发者第二个巨大的挑战——因为在原生的HTML里面并没有这些控件或者组件。

针对这一问题，jQueryMobile当前正在创建一组非常有用的组件。以下是在alpha版本中已经发布的组件：

* Pages（页面）
* Dialog boxes（对话框）
* Toolbars（工具栏）
* Buttons（按钮）
* Content formatting（内容格式化）
* Form elements（表单）
* List views（列表）

只要你懂的一点点的HTML，你会发现添加这些组件并不困难，接下来我们看几个例子。

**为页面添加header和footer**

上面的模版已经为你展示了如何轻松地创建工具栏（header，footer）。我们知道，在不同尺寸的屏幕上创建自适应工具栏通常都是非常难的活儿。而现在，jQueryMobile让你能够非常容易地创建一个带有一个居中标题和两个按钮并且自适应任何屏幕尺寸的header：

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



代码中a标签的顺序决定了按钮显示的位置。以上代码在几乎所有的设备中都能取得一致的显示效果。

header和footer同样可以被自定义为你想要的样式，比如改造一个导航条：你可以添加一些按钮在footer里面，从而导航到某一页面的不同部分：

<pre class="brush:html;gutter:false;">  
<div data-role="navbar">  
 <ul>  
 <li><a href="#nav1" class="ui-btn-active">One</a></li>  
 <li><a href="#nav2">Two</a></li>  
 </ul>  
</div><!-- /navbar -->

从这些代码你可以发现jQueryMobile不过是一些简单的HTML代码，导航条依然是div，而诸如ui-btn-active这个class可以使得你的按钮显示为被选择状态。

当你点击一个按钮并导航到该页面的另一个部分时，jQueryMobile会自动加上一个back按钮，以下的代码结构可以清晰地展现这一过程：

<div data-role="page">  
 <div data-role="header">  
 <h1>Navigation</h1>  
 </div><!-- /header -->  
<div data-role="content">   
Navigation page   
</div><!-- /content -->  
 <div data-role="footer">  
 <div data-role="navbar">  
 <ul>  
 <li><a href="#nav1" class="ui-btn-active">One</a></li>  
 <li><a href="#nav2">Two</a></li>  
 </ul>  
 </div><!-- /navbar -->  
</div><!-- /footer -->  
</div><!-- /page -->  
<div data-role="page" id="nav1">  
 <div data-role="header">  
 <h1>Nav Screen 1</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
Screen for Navigation One </div><!-- /content -->  
 <div data-role="footer">  
<h4>Additional Footer information</h4>  
 </div><!-- /footer -->  
</div><!-- /page -->  
<div data-role="page" id="nav2">  
 <div data-role="header">  
 <h1>Nav Screen 2</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
Screen for Navigation Two  
 </div><!-- /content -->  
<div data-role="footer">  
<h4>Additional Footer information</h4>  
</div><!-- /footer -->  
</div><!-- /page -->

**创建位置固定的header和footer**

工具条固定在屏幕上方或者下方是一个很常见的界面设计，你可以为footer或者header添加 data-position="fixed"来实现这一点。以下代码会强制footer/header固定在下方/上方：

<div data-role="header" data-position="fixed">  
 <h1>Navigation</h1>  
 </div><!-- /header -->  
 <div data-role="content" >  
 <ul data-role="listview" data-dividertheme="d" style="margin-top: 0;">  
 <li data-role="list-divider">Royal Family</li>  
 <li><a href="#nav1">Henry VIII</a></li>  
 <li><a href="#nav1">George V</a></li>  
 <li><a href="#nav1">Prince of Wales</a></li>  
 <li><a href="#nav1">Elizabeth I</a></li>  
 <li><a href="#nav1">Elizabeth II</a></li>  
 <li data-role="list-divider">Prime Miniseters</li>  
 <li><a href="#nav2">Winston Churchill</a></li>  
 <li><a href="#nav2">Tony Blare</a></li>  
 <li><a href="#nav2">David Cameron</a></li>  
 </ul>  
 </div><!-- /content -->  
<div data-role="footer" data-position="fixed">  
 <div data-role="navbar">  
 <ul>  
 <li><a href="#nav1" class="ui-btn-active">Royals</a></li>  
 <li><a href="#nav2">Leaders</a></li>  
 </ul>  
 </div><!-- /navbar -->  
</div><!-- /footer -->  
</div><!-- /page -->  
<div data-role="page" id="nav1" data-position="fixed">  
 <div data-role="header">  
 <h1>Royal Family</h1>  
 </div><!-- /header -->  
<div data-role="content">  
<p>Members and relatives of the British Royal Family historically represented the monarch in various places throughout the British Empire, sometimes for extended periods as viceroys, or for specific ceremonies or events. Today, they often perform ceremonial and social duties throughout the United Kingdom and abroad on behalf of the UK, but, aside from the monarch, have no constitutional role in the affairs of government. This is the same for the other realms of the Commonwealth though the family there acts on behalf of, is funded by, and represents the sovereign of that particular state, and not the United Kingdom.</P>  
</div><!-- /content -->  
 <div data-role="footer" data-position="fixed">  
 <h4>Royal Family</h4>  
 </div><!-- /header -->  
</div><!-- /page -->  
<div data-role="page" id="nav2" data-position="fixed">  
 <div data-role="header">  
 <h1>Prime Ministers</h1>  
 </div><!-- /header -->  
<div data-role="content">  
The Prime Minister of the United Kingdom of Great Britain and Northern Ireland is the Head of Her Majesty's Government in the United Kingdom. The Prime Minister and Cabinet (consisting of all the most senior ministers, who are government department heads) are collectively accountable for their policies and actions to the Sovereign, to Parliament, to their political party and ultimately to the electorate. The current Prime Minister, David Cameron, was appointed on 11 May 2010.</div><!-- /content -->  
<div data-role="footer" data-position="fixed">  
 <h4>Prime Minister</h4>  
 </div><!-- /header -->  
 </div><!-- /page -->



看，现在不使用Objective-C ，你也可以创建一个易用的界面，是不是很不错？

**对话框**

使用data-rel属性可以方便地创建弹出式对话框，以下代码将会以对话框形式加载dialog.html页面：

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

由于对话框在jQueryMobile里面其实和其他page没有任何区别，因此你可以在对话框里添加任何类型的HTML代码，当然你也可以在同一个html文件里创建两个page，一个是主界面：

<div data-role="page">  
 <div data-role="header">  
 <h1>Dialog Box</h1>  
 </div><!-- /header -->  
<div data-role="content">   
 <a href="#dialogPopUp" data-rel="dialog" data-role="button">Open dialog</a>   
</div><!-- /content -->  
<div data-role="footer">  
 <h4>Page Footer</h4>  
</div><!-- /footer -->  
</div><!-- /page -->

另一个是对话框（上面的HREF属性便指向这个对话框page）：

<div data-role="page" id="dialogPopUp">  
 <div data-role="header">  
 <h1>Dialog Title</h1>  
 </div><!-- /header -->  
 <div data-role="content">   
 This is a dialog box   
 </div><!-- /content -->  
<div data-role="footer">  
<h4>Additional Footer information</h4>  
</div><!-- /footer -->  
</div><!-- /page -->

要注意以上代码中footer是可选的，但是你必须包括header，因为如果没有header，将无法显示自动生成的对话框关闭按钮。

**列表**

当你有大量的数据条目需要显示时，列表是一个不错的选择。下面是使用ul时的HTML代码：

<ul>  
 <li><a href="#nav1">Henry VIII</a></li>  
 <li><a href="#nav1">George V</a></li>  
 <li><a href="#nav1">Prince of Wales</a></li>  
 <li><a href="#nav1">Elizabeth I</a></li>  
 <li><a href="#nav1">Elizabeth II</a></li>  
</ul>

　　在jQueryMobile你可以通过data-role="listview"来将一个普通的列表转换成一个非常华丽如同本地APP一般的列表样式：

<ul data-role="listview">  
 <li><a href="#nav1">Henry VIII</a></li>  
 <li><a href="#nav1">George V</a></li>  
 <li><a href="#nav1">Prince of Wales</a></li>  
 <li><a href="#nav1">Elizabeth I</a></li>  
 <li><a href="#nav1">Elizabeth II</a></li>  
</ul>



就是这样，仅需要20个字母，你就搞定了这一切。

除此之外，jQueryMobile还为列表提供了一些增强的选项，以下代码将为列表添加分隔符（divider):

<ul data-role="listview" data-dividertheme="d">  
 <li data-role="list-divider">Royal Family</li>  
 <li><a href="#home">Henry VIII</a></li>  
 <li><a href="#home">George V</a></li>  
 <li><a href="#home">Prince of Wales</a></li>  
 <li><a href="#home">Elizabeth I</a></li>  
 <li><a href="#home">Elizabeth II</a></li>  
 <li data-role="list-divider">Prime Ministers</li>  
 <li><a href="#home">Winston Churchill</a></li>  
 <li><a href="#home">Tony Blare</a></li>  
 <li><a href="#home">David Cameron</a></li>  
</ul>



如上面的代码所示，为li元素添加data-role="list-divider"即可实现分隔符的效果。

你还可以为列表添加小气泡来展示一些附加数据（本例展示了英国皇室的统治时间）：

<ul data-role="listview" style="margin-top: 0;">  
 <li><a href="#nav1">Henry VIII <span class="ui-li-count">Reign 37 Years</span></a></li>  
 <li><a href="#nav1">George V <span class="ui-li-count">Reign 25 Years</span></a></li>  
 <li><a href="#nav1">Prince of Wales <span class="ui-li-count">N/A</span></a></li>  
<li><a href="#nav1">Elizabeth I <span class="ui-li-count">Reign 44 Years</span></a></li>  
 <li><a href="#nav1">Elizabeth II<span class="ui-li-count">Reign since 1952</span></a></li>  
</ul>



下面是一个更加复杂的例子，列表中包含了链接，图片，文本：

<ul data-role="listview" style="margin-top: 0;">  
 <li>  
 <img src="http://img.freebase.com/api/trans/image\_thumb/en/henry\_viii\_of\_england?pad=1&errorid=%2Ffreebase%2Fno\_image\_png&maxheight=64&mode=fillcropmid&maxwidth=64" />  
 <h3><a href="index.html">Henry VIII</a></h3>  
 <p>Reign 37 Years</p>  
 <a href="#home">Details</a>  
 </li>  
 <li>  
 <img src="http://www.iwise.com/authorIcons/15/King\_George%20V\_64x64.png" />  
 <h3><a href="index.html">George V</a></h3>  
<p>Reign 25 Years</p>  
 <a href="#home">Details</a>  
 </li>  
 <li>  
 <img src="http://img.freebase.com/api/trans/image\_thumb/en/prince\_of\_wales\_secondary\_school?pad=1&errorid=%2Ffreebase%2Fno\_image\_png&maxheight=64&mode=fillcropmid&maxwidth=64" />  
 <h3><a href="index.html">Prince of Wales</a></h3>  
 <p>Reign N/A</p>  
 <a href="#home">Details</a>  
 </li>  
 <li>  
 <img src="http://www.iwise.com/authorIcons/13846/Elizabeth%20I%20of%20England\_64x64.png" />  
 <h3><a href="index.html">Elizabeth I</a></h3>  
 <p>Reign 44 Years</p>  
 <a href="#home">Details</a>  
 </li>  
 <li>  
 <img src="http://www.iwise.com/authorIcons/9098/Elizabeth%20II\_64x64.png" />  
 <h3><a href="index.html">Elizabeth II</a></h3>  
 <p>Reign Since 1952</p>  
 <a href="#home">Details</a>  
 </li>  
</ul>

在你需要展现富文本时这种列表尤为有用，比如显示一个包含照片，名字，平均分等信息的学生名单。

**部署你的jQueryMobile站点**

当你完成了开发，最后一步当然是让全世界都看到你的工作成果啦~

到现目前为止，jQueryMobile都只是包含了一些HTML，CSS，javascript文件罢了，部署方式与一般的HTML站点无异。用FTP（或者其他你喜欢的方式）上传到你的web服务器就可以了，当然，你要确保你上传了所有用到的文件。

此时已经大功告成，你就可以用你的移动设备访问站点啦！

使用jQueryMobile的目标群是移动设备用户，因此你可以考虑为你的网站创建两个版本，一个为桌面用户准备，另一个则为移动用户（准备比如主站为www.xxxx.com，移动站位m.xxxx.com）。

**接下来呢？**

jQueryMobile当前版本已经做了非常多的工作，如果你想进入移动开发领域那么现在已经无需再等待了，jQueryMobile让一切都来得那么简单。