



作者简介

Christopher Allen著名iPhone开发技术专家。最大的iPhone Web开发社区iPhoneWebDev.com创办人，iPhoneDevCamp技术大会创办人。他还是TLS安全协议的作者之一。

编辑推荐

“本书对iPhone Web开发的阐述是别处找不到的，非常珍贵。对SDK的探讨也定能使程序员大呼过瘾。”

——Mobile orchard社区

“唯一必备的iPhone开发书籍。”

——Rama Krishna Vavilala，资深技术专家。《ASP.NET AJAX实战》的作者

对于程序员而言，iPhone是一个近乎完美的目标平台。它不仅有优雅的SDK，对Web程序也提供了很好的支持。

本书是一部畅销著作。除了讲述Objective—C、苹果公司标准工具(如Dashcode、Xcode和Interface Builder)和SDK编程(包括UI和各种特性,如加速计、GPS、地址簿、SQLite等)之外。还用很大的篇幅全面探讨了iPhone开发的另一个重要领域——使用WebKit、iUI和Canvas等技术开发iPhone Web应用程序。丰富的实战示例。更使本书成为iPhone开发人员不可或缺的利器。

本书简介

本书全面探讨了iPhone平台的两种编程方式——Web开发和SDK编程。全书结合示例对这两种编程方式的基本流程、基本原理和基本原则给出了详细而通俗的讲解。在Web开发方面,分别介绍了三个iPhone Web库,即WebKit、iUI和Canvas,并讨论了Web开发环境Dashcode,最后阐述Web应用程序的调试。在SDK开发方面,详细描述其各种组件和功能,包括Xcode、Interface Builder、视图控制器、用户交互、SDK工具包、加速计和GPS、媒体、图形等。

本书适合所有iPhone开发人员学习参考。

目录

第一部分 iPhone编程简介

第1章 iPhone简介

1.1 iPhone核心规范

1.1.1 iPhone的输入及输出规范

1.1.2 iPhone网络规范

1.1.3 iPhone浏览器规范

1.1.4 iPhone的其他硬件特性

1.2 iPhone在行业中的比较

1.2.1 物理比较

1.2.2 具有竞争力的因特网浏览

1.2.3 移动Web标准

1.2.4 其他创新

1.3 iPhone的独特之处

1.4 理解iPhone输入和输出

1.4.1 输出和iPhone视区

1.4.2 输出和方向

1.4.3 输入和iPhone定位机制

1.5 小结

第2章 Web还是SDK

2.1 比较两种编程风格

2.2 核心思想:编程的连续性

2.3 优点和缺点

2.3.1 Web开发

2.3.2 SDK开发

2.3.3 程序平台

2.4 独立iPhone开发

2.4.1 Web开发模型

2.4.2 SDK开发模式

2.5 集成iPhone开发

2.5.1 镜像开发

2.5.2 混合开发

2.5.3 客户机-服务器开发

2.5.4 关于集成的最后想法

2.6 小结

第二部分 为iPhone设计网页

第3章 为iPhone重新开发网页

3.1 iPhone视区

3.1.1 更改整站视区

3.1.2 更改局部视区

3.1.3 视区属性和常量

3.2 实现“iPhone友好的”网页

3.2.1 避免缺少iPhone功能

3.2.2 创建链接

3.2.3 Web开发中的良好习惯

3.2.4 解决常见问题

3.3 实现“iPhone优化的”网页

3.3.1 通过USER_AGENT检测iPhone

3.3.2 通过CSS检测iPhone

3.3.3 使用CSS进行优化

3.4 使用iPhone chrome

3.4.1 三个栏

3.4.2 Web剪贴

3.5 捕获iPhone事件

3.5.1 单指触摸

3.5.2 两指手势

3.6 重新显示网页

3.6.1 Gmail iPhone页面

3.6.2 Facebook iPhone页面

3.7 支持非iPhone用户

3.8 小结

第4章 高级WebKit和文本Web应用程序

4.1 WebKit简介

4.1.1 新的HTML元素

4.1.2 新的CSS元素

4.2 CSS变换、过渡和动画

4.2.1 变换函数

4.2.2 过渡函数

4.2.3 动画函数

4.3 WebKit数据库

4.3.1 加载数据库

4.3.2 运行事务

4.3.3 一个示例数据库

4.4 调整chrome

4.5 识别触摸和手势

4.5.1 访问事件

4.5.2 转换事件

4.5.3 访问触摸

4.5.4 访问手势

4.6 识别方向

4.7 即将推出的特性：CSS渐变和蒙版

- 4.7.1 CSS渐变
- 4.7.2 CSS蒙版
- 4.7.3 Canvas替代方法

4.8 小结

第5章 使用iUI开发Web应用程序

5.1 创建你自己的iPhone UI

- 5.1.1 图形界面
- 5.1.2 iPhone数据范型
- 5.1.3 其他iPhone设计元素

5.2 准备iUI

5.3 使用iUI进行开发

- 5.3.1 iUI工具栏
- 5.3.2 iUI列表
- 5.3.3 iUI对话框
- 5.3.4 iUI搜索支持Ajax
- 5.3.5 iUI面板和行
- 5.3.6 iUI按钮
- 5.3.7 iUI属性

5.4 创建iUI后端

5.5 其他iUI技巧

- 5.5.1 组织代码
- 5.5.2 改善数据代码清单
- 5.5.3 压缩iUI
- 5.5.4 选择不同的外观

5.6 集成iUI与其他库

- 5.6.1 结合jQuery与iUI
- 5.6.2 结合iUI与WebKit

5.7 小结

第6章 使用Canvas开发Web应用程序

6.1 准备Canvas

- 6.1.1 启用Canvas
- 6.1.2 确保兼容性
- 6.1.3 合并功能

6.2 绘制路径

- 6.2.1 基本路径命令
- 6.2.2 曲线命令

6.3 绘制形状

- 6.3.1 绘制矩形
- 6.3.2 编写形状函数

6.4 创建样式：颜色、渐变和线条

- 6.4.1 颜色样式
- 6.4.2 渐变样式
- 6.4.3 线条样式

6.5 修改合成与剪切

- 6.5.1 全局变量
- 6.5.2 剪切路径

6.6 变换和恢复

- 6.6.1 变换

- 6.6.2 状态栈
- 6.7 整合图像、模式和文本
 - 6.7.1 图像命令
 - 6.7.2 模式命令
 - 6.7.3 文本命令
- 6.8 合并功能
- 6.9 应用动画
- 6.10 小结
- 第7章 使用Dashcode构建Web应用程序
 - 7.1 Dashcode简介
 - 7.1.1 开始Dashcode项目
 - 7.1.2 Dashcode剖析
 - 7.1.3 运行Dashcode项目
 - 7.1.4 部署Dashcode项目
 - 7.2 编写Dashcode程序
 - 7.2.1 使用库部件
 - 7.2.2 添加动作按钮
 - 7.2.3 使用基于列表的Browser模板
 - 7.2.4 使用stackLayout部件
 - 7.2.5 探究Dashcode的其余方面
 - 7.3 集成Dashcode与已有库
 - 7.3.1 集成Dashcode与Webkit
 - 7.3.2 集成Dashcode与iUI
 - 7.3.3 集成Dashcode与Canvas
 - 7.3.4 更深的集成
 - 7.4 小结
- 第8章 调试iPhone网页
 - 8.1 在本地使用Apache
 - 8.2 使用桌面浏览器进行调试
 - 8.2.1 使用Safari
 - 8.2.2 使用Firefox
 - 8.2.3 使用iPhone仿真器
 - 8.3 调试iPhone
 - 8.3.1 使用iPhone调试工具
 - 8.3.2 使用bookmarklet
 - 8.4 分析iPhone
 - 8.5 小结
- 第9章 适用于Web开发人员的SDK编程
 - 9.1 C语言概念简介
 - 9.1.1 声明和类型
 - 9.1.2 内存管理和指针
 - 9.1.3 文件结构和指令
 - 9.1.4 编译
 - 9.1.5 其他元素
 - 9.2 面向对象编程简介
 - 9.2.1 对象和类
 - 9.2.2 消息传递
 - 9.3 模型-视图-控制器(MVC)模式

9.4 小结

第三部分 SDK基本原理

第10章 Objective-C和iPhone OS

10.1 下载SDK

10.1.1 安装SDK

10.1.2 SDK剖析

10.2 Objective-C介绍

10.2.1 概述

10.2.2 消息

10.2.3 类定义

10.2.4 属性

10.2.5 其他编译器指令

10.2.6 类别和协议

10.2.7 Objective-C总结

10.3 iPhone OS介绍

10.3.1 iPhone OS剖析

10.3.2 iPhone对象的层次结构

10.3.3 窗口和视图

10.4 iPhone OS的方法

10.4.1 对象创建

10.4.2 内存管理

10.4.3 事件响应

10.4.4 生命周期管理

10.5 小结

第11章 使用Xcode

11.1 Xcode简介

11.1.1 剖析Xcode

11.1.2 在Xcode中编译和执行

11.2 在Xcode中创建第一个项目：Hello, World!

11.2.1 理解main.m

11.2.2 理解应用程序委托

11.2.3 编写“Hello, World!”

11.3 在Xcode中新建类

11.3.1 新类简介

11.3.2 头文件

11.3.3 源代码文件

11.3.4 链接

11.4 Xcode的其他功能

11.4.1 在Xcode中添加框架

11.4.2 在Xcode中使用其他模板

11.4.3 Xcode提示和技巧

11.5 小结

第12章 使用Interface Builder

12.1 Interface Builder介绍

12.1.1 Interface Builder剖析

12.1.2 在Interface Builder中模拟

12.2 在Interface Builder中创建第一个项目：图片和网页

12.2.1 新建对象

- 12.2.2 操作图形对象
- 12.2.3 使用Inspector窗口
- 12.2.4 使用图片
- 12.3 在Interface Builder中建立连接
 - 12.3.1 声明IBOutlet
 - 12.3.2 连接对象
 - 12.3.3 使用IBOutlet编写代码
- 12.4 其他Interface Builder功能
 - 12.4.1 建立其他连接
 - 12.4.2 创建外部对象
 - 12.4.3 初始化Interface Builder对象
 - 12.4.4 访问.xib文件
 - 12.4.5 新建.xib文件
- 12.5 小结
- 第13章 创建基本视图控制器
 - 13.1 视图控制器家族
 - 13.2 基本的视图控制器
 - 13.2.1 视图控制器的分析图
 - 13.2.2 创建视图控制器
 - 13.2.3 构建视图控制器界面
 - 13.2.4 使用视图控制器
 - 13.3 表视图控制器
 - 13.3.1 表视图控制器的分析图
 - 13.3.2 创建表视图控制器
 - 13.3.3 构建表界面
 - 13.3.4 使用表视图控制器
 - 13.4 小结
- 第14章 监控事件和动作
 - 14.1 事件简介
 - 14.1.1 响应者链
 - 14.1.2 触摸和事件
 - 14.2 触摸的例子：事件报告程序
 - 14.2.1 在Interface Builder中构建应用程序
 - 14.2.2 为触摸准备视图
 - 14.2.3 控制事件
 - 14.3 其他事件功能
 - 14.3.1 规则化事件
 - 14.3.2 其他事件方法和属性
 - 14.4 动作简介
 - 14.4.1 UIControl对象
 - 14.4.2 控件事件和动作
 - 14.4.3 addTarget:action:forControlEvents:方法
 - 14.5 向应用程序添加按钮
 - 14.5.1 使用addTarget:action:forControlEvents:
 - 14.5.2 使用IBAction
 - 14.6 其他动作功能
 - 14.6.1 UITextField
 - 14.6.2 UISlider

14.6.3 TextField/Slider组合

14.6.4 创建动作很容易

14.6.5 使用动作

14.7 通知简介

14.8 小结

第15章 创建高级视图控制器

15.1 选项卡栏视图控制器

15.1.1 选项卡栏控制器分析

15.1.2 创建选项卡栏控制器

15.1.3 构建选项卡栏界面

15.1.4 使用选项卡栏控制器

15.2 导航控制器

15.2.1 导航控制器分析

15.2.2 创建导航控制器

15.2.3 构建导航控制器

15.2.4 使用导航控制器

15.3 使用翻转控制器

15.4 模式视图控制器

15.5 小结

第四部分 使用SDK工具包编程

第16章 处理数据

16.1 接收用户动作

16.2 管理用户首选项

16.2.1 创建自己的首选项

16.2.2 使用系统设置

16.3 打开文件

16.3.1 访问软件包

16.3.2 访问其他目录

16.3.3 操纵文件

16.3.4 Filesaver：一个UITextView示例

16.4 使用SQLite

16.4.1 设置SQLite数据库

16.4.2 访问SQLite

16.4.3 访问SQLite数据库

16.4.4 通过数据库构建导航菜单

16.4.5 本例的扩展

16.5 访问地址簿

16.5.1 框架概述

16.5.2 访问地址簿属性

16.5.3 查询地址簿

16.5.4 使用地址簿UI

16.6 小结

第17章 定位：加速计和位置

17.1 加速计和方向

17.1.1 orientation属性

17.1.2 方向通知

17.2 加速计和移动

17.2.1 访问UIAccelerometer

- 17.2.2 解析UIAcceleration
- 17.2.3 查看重力
- 17.2.4 查看移动
- 17.2.5 识别简单的加速计移动
- 17.3 加速计和手势
- 17.4 关于Core Location
 - 17.4.1 位置类
 - 17.4.2 使用位置和距离的示例
 - 17.4.3 使用海拔的示例
 - 17.4.4 Core Location和互联网
- 17.5 小结
- 第18章 媒体：图像和声音
 - 18.1 图像介绍
 - 18.1.1 加载UIImage
 - 18.1.2 绘制UIImageView
 - 18.1.3 在UIKit中修改图像
 - 18.2 利用Core Graphics绘制简单图像
 - 18.3 访问照片
 - 18.3.1 使用图像选取器
 - 18.3.2 拍照
 - 18.3.3 保存到相册
 - 18.4 拼合：一个图像例子
 - 18.4.1 拼合视图控制器
 - 18.4.2 拼合临时图像视图
 - 18.4.3 拼合视图
 - 18.4.4 扩展这个例子
 - 18.5 使用Media Player框架
 - 18.5.1 媒体播放器类
 - 18.5.2 音量视图
 - 18.5.3 更好地集成媒体播放器
 - 18.6 手动播放声音
 - 18.6.1 播放简单声音
 - 18.6.2 振动iPhone
 - 18.6.3 播放复杂的声音
 - 18.6.4 其他音频框架
 - 18.7 小结
- 第19章 绘制图形
 - 19.1 Quartz 2D简介
 - 19.2 Quartz上下文
 - 19.2.1 在UIView上绘制图形
 - 19.2.2 在位图上绘制图形
 - 19.3 绘制路径
 - 19.3.1 结束路径
 - 19.3.2 创建可重用路径
 - 19.3.3 绘制矩形
 - 19.4 设置图形状态
 - 19.4.1 设置颜色
 - 19.4.2 变换

- 19.4.3 设置裁剪路径
- 19.4.4 其他设置
- 19.4.5 管理状态
- 19.5 Quartz中的高级绘图功能
 - 19.5.1 绘制渐变
 - 19.5.2 绘制图像
 - 19.5.3 绘制文字
 - 19.5.4 未介绍的内容
- 19.6 示例：在图片上绘图
 - 19.6.1 photodraw视图控制器
 - 19.6.2 photodraw视图
 - 19.6.3 扩展示例
- 19.7 Core Animation简介
 - 19.7.1 Core Animation基础
 - 19.7.2 Core Animation入门
 - 19.7.3 绘制简单的隐式动画
 - 19.7.4 绘制简单的显式动画
- 19.8 OpenGL简介
- 19.9 小结
- 第20章 Web：Web视图和因特网协议
 - 20.1 因特网的层次结构
 - 20.2 低层次联网
 - 20.3 使用URL
 - 20.3.1 创建NSURL
 - 20.3.2 构建NSURLRequest
 - 20.3.3 手动操作HTML数据
 - 20.4 使用UIWebView
 - 20.4.1 调用Web视图
 - 20.4.2 管理Web视图委托
 - 20.4.3 缩略图：一个Web视图例子
 - 20.4.4 Google Maps：一个Core Location示例
 - 20.5 解析XML
 - 20.5.1 启动NSXMLParser
 - 20.5.2 充当委托
 - 20.5.3 构建示例RSS阅读器：一个XML示例
 - 20.5.4 海拔高度重读：一个Core Location示例
 - 20.6 提交给Web
 - 20.6.1 手动提交
 - 20.6.2 提交表单
 - 20.7 访问社会网络
 - 20.7.1 使用Web协议
 - 20.7.2 使用TouchJSON
 - 20.8 小结
- 附录A iPhone OS类参考
- 附录B 外部资源和参考资料
- 附录C 发布你的SDK程序

插图摘要

书摘插图 第一部分 iPhone编程简介

第1章 iPhone简介

20世纪80年代，苹果计算机公司是计算机行业创新的领跑者。苹果公司1984年推出的Macintosh计算机给个人计算和桌面发布领域带来了革命性进步。但到20世纪90年代，苹果公司开始衰落，当时它依赖的是忠实的用户基础和过去的辉煌，而不是前沿技术方面的创新。

1996年，创始人Steve Jobs的强势回归改变了这一切。两年后，他推出了首台糖果色的iMac，这是一种集计算设备、通俗文化和个性魅力于一体的计算机。iMac只是Jobs带来的若干创新中的第一个，随后最著名的可能就是2001年上市的iPod，iPod是便携式设计的杰作。它注重简单和美观的界面，允许用户随身携带数千首歌曲并随时播放它们。但iPod带给公众更多的是一种期待。

2006年，关于苹果公司下一个主要创新的传闻和推测在互联网上闹得沸沸扬扬，传说将是一种类似于iPod的手机，而且最终将被命名为iPhone。苹果公司在21世纪的技术创新和一流的用户设计，让人们们对传说中的iPhone充满了期待。因为它预示了整个手机行业的新前景，以及可大胆改进技术的新方向。

2007年初，苹果公司承认当前正在开发iPhone。苹果公司提供的技术预览已经揭示，iPhone确实是一款新颖且与众不同的手机。人们的热情和兴奋达到了空前的高度。在发布当天，2007年6月29日，人们连夜聚集在苹果商店门前。为成为拥有iPhone的第一批用户，人们在一天内都排着长长的队伍。iPhone是新一代用户友好移动技术的首次应用，也是唯一配得上超智能手机这一称号的手机。

当用户开始试着使用自己新的iPhone时，兴奋之情难于言表。iPhone易于使用，并且提供了许多诱人的附属功能，包括股票行情、天气预报而且随时可访问因特网。销售量足以反映人们的狂热。苹果公司在两天内便售出了27万台iPhone，并让该数字在一个半月之后达到了一百万。现在，也就是距iPhone最初发布一年半之后，人们对iPhone的热情仍然没有降温的趋势。2008年7月11日，苹果公司发布了新的3G iPhone及iPhone软件开发包(SDK)，这预示iPhone未来的成就将不可限量，而iPhone在2009年的销量也将显著增长。3G在三天内的销量达到了一百万。一个新的技术浪潮正向我们袭来，而且它还远未到顶。

但是，究竟是什么技术让iPhone如此风靡，iPhone程序员又应如何利用它们呢？这就是本章要讨论的主题，我们将介绍iPhone的核心规范，并讨论让iPhone开发成为全新体验的6个独特创新。

1.1 iPhone核心规范

iPhone并不仅仅是一款简单的手机或智能手机。iPhone令过去那些支持受限因特网访问和其他功能的智能手机难以望其项背。我们已经说过，它是一款超智能手机。如果说iPod引领了市场的潮流，那么iPhone就是全新的一代设备，并将保持在该领域中的领先地位。苹果公司强劲的品牌认可度和它一如既往的创新追求决定了这一切。

从技术上说，iPhone主要可分为两种相似的版本：2007年的原始发行版和2008年的3G发行版。这两款计算设备的重量分别为133和136克。它们都使用经过降频的620MHz ARM CPU，目的是改进电池性能并减少发热量。它们包含128MB动态RAM(DRAM)，以及4~16GB的闪存。这两款设备之间的主要差异是全球定位系统(GPS)和网络，这方面的话题将随后展开。

从编程角度看，iPhone依托于苹果公司的OS X，而OS X又以Unix为基础。Xcode是该设备本机编程的核心，也是为Macintosh编写代码的开发环境。结合考虑这两个元素，我们看到的是一个成熟的开发和运行时环境：大多数其他手机（可能要把Windows Mobile排除在外）都使用它，而且即将到来的超智能手机技术在数年内也无法与其抗衡。

但是，这些常规的规范只能反映iPhone的冰山一角。通过深入探究iPhone的输入和输出以及它的网络和其他功能，你将了解促使iPhone成为创新计算平台的主要原因。

.....

[下载后 点击此处查看更多内容](#)

