

写给大忙人看的 *Swift*

[美] Matt Henderson Dave Wood 著

周智勋 杨宏焱 龙达 曹磊 译



SWIFT
for the Really Impatient

写给大忙人看的 *Swift*

[美] Matt Henderson Dave Wood 著

周智勋 杨宏焱 龙达 曹磊 译



Swift

for the Really Impatient

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书全面介绍了苹果公司新推出的 Swift 编程语言。主要内容包括 Swift 语法简介, Swift 应用程序中常见的对象模型、可空类型、泛型、函数和闭包的使用方法, Swift 代码的安全性, 如何在 Swift 中使用 Objective-C 和苹果平台已有的框架, 如何利用 Swift 及其新功能来创建一个应用程序等。本书作者 Matt Henderson 及 Dave Wood 是 iOS 和 OS X 开发者的领军人物, 他们运用简练的语言介绍了 Swift 的相关知识。书中含有大量实战代码和课后习题, 通过理论和实践相结合的方式, 帮助读者迅速掌握 Swift 的主要功能和用法。

本书适合各种层次的 Swift 学习者和开发者阅读, 包括拥有 Objective-C 开发经验的开发者, 想要用 Swift 编写程序的爱好者等任何对 Swift 感兴趣的人。对于想要快速了解和掌握 Swift 的人来说, 《写给大忙人看的 Swift》是一本不可多得的枕边读物。

Authorized translation from the English language edition, entitled SWIFT FOR THE REALLY IMPATIENT, 9780133960129 by Matt Henderson, Dave Wood, published by Pearson Education, Inc., publishing as Addison-Wesley Professional, Copyright © 2015 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2015.

本书简体中文版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号 图字: 01-2015-3481

图书在版编目 (CIP) 数据

写给大忙人看的 Swift / (美) 亨德森 (Henderson, M.), (美) 伍德 (Wood, D.) 著; 周智勋译. —北京: 电子工业出版社, 2015.6

书名原文: Swift for the Really Impatient

ISBN 978-7-121-26439-9

I. ①写… II. ①亨… ②伍… ③周… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 139183 号

策划编辑: 张春雨

责任编辑: 徐津平

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16

印张: 12.25

字数: 265 千字

版 次: 2015 年 6 月第 1 版

印 次: 2015 年 6 月第 1 次印刷

定 价: 55.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线: (010) 88258888。

推荐序一

我认识周智勋是在 2012 年，当时他还在 DevDiv 开发者社区创业，那时我就感觉这是个翻译狂人。当时我们都是一个名为“真·iOS 开发”的 QQ 群的成员，这个群由国内主要的原创 iOS 博客博主组成，大家会时常分享和讨论一些国外最新的关于 iOS 开发的文章。周智勋除了积极地和我们讨论外，还会在讨论当晚就把那些文章译成中文放到他的博客上。

当然，这些并不能支撑他“翻译狂人”的名号。除了频繁地翻译国外的优秀博客文章外，周智勋还组织翻译了两本优秀的 iOS 开发书籍，分别是 *iOS 5 by tutorials* 和 *iOS 6 Programming Cookbook*。此外，他也组织出版了多本 iOS 相关的图书，包括《iOS 开发实战体验》等。

记得有一次，大伙从 objc.io 上看到一篇不错的英语文章，标题是“Communication Patterns”，文中介绍了 iOS 各种通知的方式，例如，KVO、Notification、Delegate 等。大家都在讨论这个标题如果翻译成中文，应该叫什么。有人说译成“通讯模式”，也有人说译成“通讯机制”，但总是觉得怪怪的，不太容易理解。这个时候周智勋提议译成“消息的传递机制”，大家一下子都服了，觉得翻译得非常贴切，我想这就从某种意义上显示了他的翻译功底吧。

我自己也做过一些翻译工作，但不太喜欢，我更喜欢写原创文章。因为我认为翻译文章大多锻炼的就是翻译能力本身，而写原创文章的话，锻炼的则是知识的整理和加工能力。从我个人的角度出发，我更想做一些能够提升自己知识理解能力

的事情。

所以说，做翻译的人除了爱好之外，更需要有一种分享精神，或者说是利他的奉献精神。正是因为有像周智勋这样的人不求回报地把国外的优秀 iOS 开发类文章源源不断地翻译成中文，才使得广大开发者能够更快地吸收相关的开发知识。

2013 年，周智勋放弃创业，离开北京回到了老家云南，在那个四季如春的城市开始了他的新的工作。云南的互联网工作强度不像北京那么大，周智勋得以继续做他喜欢的翻译工作。能够淡泊名利，安心在云南做 iOS 开发，我想这也与他做翻译以来一直不计回报的性格有关吧。除了翻译之外，周智勋其实还是一个健身爱好者，我常常看到他在朋友圈分享跑步的信息，有时候他还会参加马拉松。

由于我也经常写文章，所以认识了一些出版社的老师。当电子工业出版社的刘芸老师找我推荐 iOS 图书 *Swift for the Really Impatient* 的译者时，我的脑海里蹦出来的第一个名字就是周智勋。于是，在我的引荐下，周智勋和他的几位同事很快就接手了这本书的翻译工作。

几个月后，周智勋发给了我该书的译文：《写给大忙人看的 Swift》。在仔细阅读译文之后，我感觉自己做了一个非常正确的决定。整本书的翻译质量非常高，苹果公司的官方 Swift 文档长达千页，但这本书以不到 200 页的篇幅，就将 Swift 的重要知识进行了浓缩介绍，非常适合新手在没有压力的情况下，用相对短的时间，对 Swift 的整体知识进行入门级的了解。

《写给大忙人看的 Swift》是一本高质量的 Swift 语言入门书籍，感谢电子工业出版社的引进，以及我的好友周智勋及其同伴的辛苦翻译，也祝愿本书能帮助更多人进入到 iOS 的编程世界里来！

《iOS 开发进阶》作者 唐巧

2015 年夏

推荐序二

乔帮主过世后，我再也没有追过凌晨一点档的发布会，总是第二天中午再下载官方的 Podcast 来看，2014 年也不例外。用了八年 Mac，大部分软件新功能以及新的硬件升级对我来说都已没什么感觉。但这次的视频播放到最后十几分钟时我调高了音量，因为视频里出乎意料地介绍起了一种新的编程语言——Swift。Optional 能简化原本用 Objective-C 写起来成堆的判断语句，Playground 直接显示变量的值，并且能绘制值的变化曲线。在看完视频以及社交圈大家对 Swift 的各种褒扬后，我恨不得立刻把现有的 iOS 项目都用 Swift 重写一遍。

但是，我并没有这么多时间和精力去全身心地投入到 Swift 的学习和开发中。甚至连官方的教程“The Swift Programming Language”我都只是在地铁里粗略地看了一半。正巧那时 IDEO 为 Quartz Composer 做了一套叫作 Avocado 的插件包，我用它实现了一个弹动 UITextField 的 placeholder 的交互原型，便动手用 Swift 来实现它的原生版本。我习惯了这种通过实战来快速入门的学习方式，想尽可能地涵盖原来 Objective-C 中的一些模式，以便区分两种语言的开发方式。于是很快就有了 LTBouncyPlaceholder 这个开源项目，戴着“尝鲜 Swift”这个光环，居然有网友专门写了一篇博客分析这个项目的技术实现。

接着没过几天，我在 GitHub 上看到了 TOMSMorphingLabel 这个实现文字变换动画的项目。这个动画最早出现在 iOS 的 Messages 应用输入文字时键盘上的 Quick Type 提示框中，也在 WWDC 2014 上有几秒钟的镜头。但是，总觉得用

NSAttributedString 实现的动画效果哪里不对劲。于是，我做了一个包含 UITextField 的测试 App，打开 iOS 模拟器，按下“⌘+T”组合键打开 Slow Animations 开关，反复按动键盘，观察 Quick Type 提示文字的变化。看了二十分钟后，我认清了动画的每一个细节，我确信自己能比 TOMSMorphingLabel 做得更好。在花一周时间不断试验各种参数后，完全使用 Swift 编写的 LTMorphingLabel 开源了。虽然我没有使用太多 Swift 的语法特性，也没有像 TOMSMorphingLabel 那样使用现成的 NSAttributedString 来快速地实现动画，而是自己计算每一帧每个字符的位置，但这并不影响动画效果的呈现。而正因为如此，我能控制每一个细节，我的动画几乎和原生的一样完美。所以当 Dave Verwer 在他每周一期的“iOS Dev Weekly”中提到 TOMSMorphingLabel 时，我骄傲地在 Twitter 上把我的项目地址发给他。不过他的回复让我很意外，他说他更喜欢 TOMSMorphingLabel 的效果。当时我真想回复：“你看我这是用 Swift 写的啊，怎么也得加十分吧，而且动画更优雅，节奏更自然”；但又想到 Mark Bauerlein 说的“一个人成熟的标志之一就是，明白每天发生在自己身上的 99% 的事情对于别人而言根本毫无意义”，我想既然我能做一个文字变换的动画效果，我还能做更多。半年后，我仍然没能耐心地看完“The Swift Programming Language”，但这时 LTMorphingLabel 已经有了七个文字变换动画。其中三个使用 CAEmitterLayer 实现粒子喷发的效果。我甚至尝试用 SpriteKit 实现了更好的粒子效果，但这对于一个 UILabel 的子类来说，显然成本太高。那段时间 GitHub Trending 上总能看到 LTMorphingLabel，偶尔能排到 Swift 语言分类的第一。直到年底，我把业余的精力投入到了一个新的 Swift 项目——VPN On。

虽然在这个项目开始之前，我听到不少苹果公司对于 Today Widget 的功能有多么严格的限制的说法，在和朋友讨论我想做的这个 Today Widget 的想法后，所有人都对我说不要做了，Today Widget 里的内容应该和 Today 密切相关，苹果公司不可能让我上架这个功能性的 App。但我的想法很简单，这么实用的功能，苹果公司不让上架的话，大不了将其开源，让所有有开发账号的人都能自己编译。于是我忍受着 Xcode 的 SourceKit 反复蹦出的报错提示，还有那头孩子不肯乖乖睡觉的吵闹声，经过几个夜晚的努力，硬是挤出了 VPN On 的第一版。在把这个版本提交到 iTunes Connect 后，我果断地直接开源了整个项目。两周后，它意外地上架了。我发现开源并不会对销量产生太大的影响，上架后头三天带来 427 美元净收入。有一些人告诉我，他们已经成功编译了开源的版本，但仍然去 App Store 买了一份表示支持。

五个月后的现在，LTMorphingLabel 和 VPN On 成为 GitHub 上星数较高的两个 Swift 开源项目，被列在各种 Swift 相关的 awesome list 中。不过，从 Swift 发布到现在近一年过去，我还是没能看完“The Swift Programming Language”。一是因为各种忙，二是一旦有闲情就又想着做点什么，没能沉下心来看书。所以我觉得应该有一本书，能帮助学习 Swift 语言的人快速入门，用尽可能短的篇幅介绍语言特性及其使用方法。在读过这本《写给大忙人看的 Swift》之后，我发现它就是我想要的书，所以推荐像我一样没太多时间看书又想学 Swift 的朋友阅读这本书。

iOS VPN On 作者 汤圣昱
2015 年夏

译者序

苹果公司在 WWDC 2014 上向全世界发布了一门新的编程语言——Swift。苹果公司开发者官网是这样阐述 Swift 的：“Swift 是一门新的具有创造性的编程语言，它可用于 Cocoa 和 Cocoa Touch。利用 Swift 编写的代码充满了交互性和趣味性，它的语法简洁并富有表现力，开发出的应用程序运行速度快如闪电。Swift 已经准备好帮你开发下一个 iOS 和 OS X 项目了——当然也可以在当前的项目中使用——因为 Swift 代码可以与 Objective-C 并驾齐驱。”

有的开发者在感叹，苹果公司的 iOS SDK 一年升级一个大版本就够开发者折腾了，如果不是持续跟进，或许早已被 WWDC 中推出的新技术淹没了。iOS 7 的扁平化风格对当时应用程序的界面设计产生了非常重大的影响，只不过如今扁平化风格已经大行其道了。现在，又推出了 Swift，这对大多数开发者来说是个不小的挑战。

作为开发者的我，其实也最担心变化，或许厂商一个不经意的变动，会产生一系列的蝴蝶效应。还记得 iOS 7 的企业级部署中要求使用 HTTPS，这样的变化虽说很简单，但是真正由开发到生产环境使用的整个过程耗费了我太多的时间。不过，我又每年期待着苹果公司能在 WWDC 上推出新的东西。苹果公司每次发布的 iPhone、iPad 等都对世界产生着重大的影响。或许苹果公司的使命就是改变世界，让世界变得更好，作为开发者的我们只需要拥抱变化即可。

今天的 Swift 不亚于当年微软推出的 C#，据 Redmond 排名统计显示，Swift 已从 2014 年的第 68 名，上升至目前（2015 年第一季度）的第 22 名。许多开发者也快速跟进了 Swift 编程语言开发，并且许多流行的第三方开源方案都提供支持

Swift 版本。各类相关参考书籍和开发者社区都表现出积极的姿态。

机缘巧合，今年年初，经好友唐巧（唐巧不仅是《iOS 开发进阶》的作者，他还是 iOS 开发界最具影响力的一名开发者，目前在北京一家在线教育创业的明星公司猿题库修行）介绍与电子工业出版社的刘芸老师取得联系，经过几次沟通敲定翻译的相关事宜。因为是一本快速入门的书，《写给大忙人看的 Swift》的内容虽不多，本来预计 2015 年 3 月就完成此书的翻译，无奈由于各种原因，到 6 月才翻完，也算是好事多磨吧。

《写给大忙人看的 Swift》主要内容包括 Swift 语法简介，Swift 应用程序中常见的对象模型、可空类型、泛型、函数和闭包的使用方法，Swift 代码的安全性，如何在 Swift 中使用 Objective-C 和苹果平台已有的框架，如何利用 Swift 及其新功能来创建一个应用程序等。本书通过简练的语言介绍了 Swift 的基础知识，书中包含了大量的实战代码，通过理论和每章最后给出的题目相结合的方式，帮读者迅速掌握 Swift 的主要功能和用法。此书非常适合目前正在使用 Objective-C 编写程序的开发者，希望用 Swift 开始编写程序的开发者，以及对 Swift 感兴趣的开发者阅读。

在此书的翻译过程中，刘芸老师给了我很多的支持，使得我能够将注意力集中在书籍的翻译上，在此特别感谢这位专业的编辑。

我的同事杨宏焱（翻译第 4、6、7、8 章）、龙达（翻译第 3 章）和曹磊（翻译第 5 章）为此书的翻译做了大量的工作，并且在我的日常生活和工作中，他们也给予了无微不至的关怀，在此特别感谢。

此外，要感谢我的好友唐巧以及 Swift 先驱者之一汤圣罡，他们在对此书通读校对的过程中展现出了认真负责的态度，帮助我们排除了译文中存在的错误。

最后，要感谢我的妻子王琼梅，是她让我每天能投入大量的业余时间在本书的翻译工作中，并且为我提供可口的饭菜。

也正是因为得到了他们的热心帮助，此书才得以与读者见面。

如果你无意间看到了这本书，那说明这是一种缘分，真心希望本书对你起到些许作用。如果你发现译文有误，可以发邮件（BeyondVincent@gmail.com）给我，也可以通过新浪微博（BeyondVincent）告诉我，在此感激不尽。

人真是一种奇怪的精灵，总是在徘徊与纠结中绘制着自己的人生轨迹。

周智勋
2015 年 5 月

前言

2008 年 3 月，苹果公司发布了取得巨大成功的 iPhone 对应的 SDK。该 SDK 和随后几个月里发布的 App Store 引起了大量开发者的关注。许多开发者都是首次接触苹果公司的平台，他们惊奇地发现整个开发工具包是基于——一门非主流的问世仅 25 年的 Objective-C 编程语言开发的，而不是非常流行的编程语言，例如 C++ 或 Java。

而 Mac 开发者以及苹果公司的粉丝并没有特别惊讶。在 1996 年苹果公司收购 NeXT 后不久，Objective-C 就成为开发 Mac 应用程序的主要编程语言。苹果并没有选择像 C++ 或 Java 这种通用的被广泛用于多种平台以及各种编程任务的编程语言，而是选择了一门在其可控范围的编程语言，来构建 OS X 和随后的 iOS。Objective-C 发展得非常缓慢，苹果公司只专注于其操作系统的 GUI 应用程序开发。

Objective-C 和 NeXT 框架多年来通过各种方式被用于构建应用程序，虽然作为商业公司 NeXT 从来没有特别成功过，但是 NeXT 的应用程序开发工具非常受欢迎，因为它能够比市面上其他工具更快地开发出应用程序。

但是 Objective-C 已经有三十多年的历史，就技术而言，这是很长的时间了。尽管 Objective-C 在某些方面得到了改善，但它确实不够与时俱进了。在过去几十年里，编程语言和编译器已经演变了很多，许多开发者极力要求苹果公司选用一门更具“现代化”特征的编程语言来开发 iOS 和 OS X 应用程序。

在 2014 年的苹果电脑全球研发者大会（WWDC）中，苹果公司发布了一门叫

作 Swift 的新型现代化编程语言，这几乎让每个人都感到惊讶。

实际上，苹果公司已经为此秘密研发了 4 年，Swift 是一门与众不同的编程语言。用与运行 Objective-C 一样的时间，Swift 可以运行出许多 Objective-C 中是没有的有趣的功能，还能够使用已有的所有组成 iOS 和 OS X SDK 的框架。

但是，Swift 确实与众不同。它看起来不同，用起来感受也不同。你甚至觉得必须要用“想象非凡”来形容 Swift 才合适。

幸运的是，你已经获得了正确的书籍来帮助你实现想象非凡。Matt 和 Dave 为此做出了非常出色的工作，他们将通过本书引导你学习 Swift 语言，可能 Swift 会使你感到稍微有点吓人。同时也会引导你该如何做，以及为什么那样做，帮助你避免在学习过程中遇到太多陷阱，以便让你快速掌握 Swift 开发。

——Jeff LaMarche, *Beginning iPhone Development: Exploring the iPhone SDK* (Apress 出版) 的作者；MarianCraft 的合伙人和创始人

序言

本书简要介绍了苹果公司推出的新的编程语言——Swift，适合目前使用 Objective-C 编写程序的开发者，希望用 Swift 开始编写程序的开发者，以及对 Swift 感兴趣的开发者阅读。

本书的写作风格是“impatient”，模仿 Cay Horstmann 的风格，快速呈现出主题内容，并且以清晰明了的表达方式给开发者提供足够多的信息，让开发者对相关知识点的掌握有立竿见影的效果。书中以简短的代码块来阐述相关知识。这些代码片段旨在快速参考，而不是提供完整的，可以直接使用的示例。

Swift 提供了许多令人兴奋的语言功能，这些功能在 Objective-C 中是没有的，Objective-C 目前是在苹果平台中做开发的主要编程语言。设计这些功能的主旨是让开发者效率更高，并让其代码减少出错概率。使用 Swift，开发者可以利用强类型系统和现代编程语法，很容易地开发出强大的应用程序。在第 1 章“Swift 介绍”和第 2 章“深入学习 Swift 语法”中会把 Swift 语法做一个快速的介绍，基本类型和相关功能对于 Objective-C 开发者来说可能是新概念。后续的章节主要集中在详细介绍这些功能上。

跟 Objective-C 一样，Swift 首先是一门面向对象的编程语言。第 3 章的“对象和类”将介绍在 Swift 应用程序中经常用到的主要对象类型。Swift 中类的使用，与开发者接触过的别的面向对象编程语言一样。另外，Swift 的类型系统有助于避免在其他编程语言中常见的一些错误。

从第 4 章开始，将深入介绍“可空类型”，Objective-C 开发者对这个概念可能不太熟悉。第 4 章主要集中介绍可空类型，这是一个全新的基本概念，在类型级别中，它强制声明的对象要么真正包含一个值，要么就是包含一个 nil 值。这一章开发探索 Swift 注重的代码安全性。第 5 章“泛型”中将介绍泛型，通过泛型，开发者对函数进行抽象，达到代码重用的目的，同时还能保持类型安全。在 Swift 中，函数是第一类类型，也就意味着可以把函数当作参数进行传递，以值的方式返回。在第 6 章“函数和闭包”中将探索 Swift 中关于函数式编程的部分。

学习 Swift 的主要目的是能够使用它开发 iOS 和 Mac OS X 应用程序。第 7 章“使用 Objective-C”主要介绍如何在 Swift 中使用 Objective-C 和苹果平台已有的框架。Swift 设计时，考虑到了 Objective-C，这使得 Swift 与 Objective-C 工程和苹果公司提供的开发应用程序的框架进行集成非常容易，不过在结合使用 C 和 Objective-C 之前，有一些关键的内容需要知道，这些内容将在第 7 章中介绍。

在第 8 章“常用模式”中，你将学习到如何利用 Swift 和相关新功能来创建一个应用程序，所以第 8 章将侧重于实际使用情况。学完第 8 章后，你将知道在开发应用程序过程中，如何处理常见的几种情景。

在每章的最后，会提供练习题，这些题有助于新技能的掌握。建议读者通过做这些练习题来加强阅读效果。如果你在做题的过程中遇到一些困难，你可以访问网址 <http://SwiftForTheReallyImpatient.com/>，我们会把相关解决方案放到这个网站中。在这个网站中，你也可以找到本书的勘误信息，以及其他一些有趣的内容。



注意

如果本书中单行代码过长，无法在单行完整显示，会利用代码连续箭头（➡）来表示代码的续行。

致谢

感谢我们的编辑 Trina MacDonald 和 Betsy Gratner，是他们带领着两个从未出版过书籍的作者，经历所有步骤成功出版本书。感谢我们的审稿员 Rene Cacheaux、Tom Cirtin、Ash Furrow、Kitty Wilson 和 Chris Zahn，经过他们的审稿，使得所有的章节内容读起来非常通畅，也确保我们一路跟随 Swift 测试版不断更新脚步。感谢 MartianCraft 的同事，无论遇到多么晦涩的难题，他们总是给我提供帮助，特别是为本书写了非常精彩的前言的 Jeff LaMarche 和鼓励我们写这本书的 Kyle Richter。最后，要感谢我们的妻子 Clare 和 Sabrina，她们支持我们完成了过程看似并不简单的第一本书。

关于作者

Matt Henderson 从 2009 年开始在苹果平台上进行软件开发，现在是 MartianCraft 的一名 Cocoa 工程师。他在不同的用户群体和会议上做过几次演讲，包括 360iDev、Cocoaheads Denver 和 Boulder iOS Meetup。当他发现用图形计算器来解方程比数学学习更简单时，他意识到自己应该在软件发展方面大有潜力。他认为最佳调试技术就像走在有阳光或雪的户外。

Dave Wood 从 2008 年开始在 iOS 上进行开发，2009 年开始进行 OS X 开发，在 9 岁时，他就在 TI/99/4A 设备上编写代码，并立马爱上了编程。他曾开发过不同类型的工程，包括股票交易系统的接口、新闻媒体、银行系统，以及报纸网站，当然也包括移动应用程序，涉及游戏、社交网络、财务软件、工作效率和开发应用程序。如果可能，他喜欢激流皮划艇和深海潜水。目前他自己开了一个开发工作室 Cerebral Gardens，并且是 MartianCraft 的一名 Cocoa 自由职业工程师。

目录

第 1 章 Swift 介绍	1
1.1 基本语法	4
1.1.1 变量和常量	4
1.1.2 字符串插值	7
1.1.3 流程控制	8
1.2 基本数据类型	12
1.2.1 整型	12
1.2.2 Double 和 Float	14
1.2.3 布尔型	15
1.2.4 字符串	16
1.2.5 数组	17
1.2.6 字典	21
习题	22