
Pearson
Broadview
www.broadview.com.cn
一路编程
Learning to Program
[美] Steven Foote 著　佟达 译
中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

一路编程

Learning to Program

[美] Steven Foote 著　佟达 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

这是一本编程入门书籍，然而，如果以书中所讲内容作为入门标准，估计十有八九的在职程序员都不能算已入门。现代软件开发，已经不仅仅是写出正确的代码这么简单，环境、依赖、构建、版本、测试及文档，每一项都对软件是否成功交付起到至关重要的作用，这些都是每一个程序员在开发软件过程中必备的技能。本书对于上述的每一种技能都做了简洁而精练的介绍，以满足最基本的日常软件开发。换句话说，本书实际上是为现代软件开发的入门设下了最基本的门槛。相信每一位本书的读者，不论是即将进入软件行业，还是已经在软件行业工作多年，都会获得收获。

强烈推荐刚刚或将要成为程序员的人及其朋友们阅读本书。

版权贸易合同登记号　图字：01-2015-6645

图书在版编目（CIP）数据

一路编程 /（美）史蒂夫·富特（Steven Foote）著；佟达译. —北京：电子工业出版社，2017.1
书名原文：Learning to Program
ISBN 978-7-121-30478-1

Ⅰ. ①一… Ⅱ. ①史… ②佟… Ⅲ. ①程序设计 Ⅳ. ①TP311

中国版本图书馆 CIP 数据核字(2016)第 287278 号

策划编辑：符隆美
责任编辑：徐津平
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：18.5　字数：350 千字
版　　次：2017 年 1 月第 1 版
印　　次：2017 年 1 月第 1 次印刷
定　　价：65.00 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）88254888，88258888。

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

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

推荐序

编程魔法的麻瓜入门手册

在看一个涉及旧房改造的电视综艺节目时，我突然意识到一件事：同样是需要大量知识和实践作为基础的专业人士，程序员似乎从来没有得到像建筑师那样来自普罗大众的认可与尊重。究其原因，我想可能部分原因是程序员的工作太缺乏可见性，以至于那些不懂技术的麻瓜们对程序员的印象要么是盲目崇拜，要么是盲目轻视，总之没有那种平等而深入的交流方式。尤其在英语极不普及的中国，天书一般的程序代码更是让麻瓜们望而却步，遑论去理解和交流。像本书译者、我的同事佟达所遭遇的“家人不知道我一天到晚在干些啥”的困扰，对于广大中国程序员而言恐怕是家常便饭。

本着帮助麻瓜进入魔法世界的崇高理想，Steven Foote 写了这么一本精彩的编程入门教材。相比常见的高校编程教材，这本《一路编程》最大的好处是它的“不求甚解”——作者堂而皇之地宣称：你不知道一行程序代码背后那些魔法是怎么发生的吗？无所谓！“你不需要知道那些指令是什么，或者理解它们的工作原理，只要它们能用就行。”这么一来，读这本书入门的麻瓜们可就比它们大部分在学校里上编程课的同侪们更有希望：当那些抱着大部头教材的学生们正在被指针地址等高深的魔法细节吓破胆时，这本书的读者倒是——像已经在干着程序员这份工作的人一样——放心大胆地忽视绝大部分细节，写出一两个能运行的程序，并从中获得继续学习必不

可少的成就感。

其实——出乎麻瓜们的意料——绝大部分的软件开发根本不是什么火箭科学。就像我经常跟朋友打趣说的，我们写这些程序连四则运算都用不全，主要除法不怎么用。相比科学，软件开发其实更像一种工匠手艺（craft）。使一个优秀的程序员区别于平庸程序员的，正如 Neal Ford 在《卓有成效的程序员》中所说，是好的习惯和趁手的工具。Foote 很准确地抓住了这一点。在快速介绍 Javascript 编程之后，他迅速而不失全面地介绍了体面的前端程序员应该用到的全套工具箱：命令行、构建工具、数据库、正则表达式、测试工具、文档……招聘时我们常会对着候选人提交的一个光秃秃的源代码文件挠头，只有经历过这种痛苦的人才会知道，候选人提交的代码作业有 Grunt 构建脚本甚至有单元测试是一件多么令人愉悦的事。

处在当今这个数字化浪潮方兴未艾的时代，能让麻瓜顺利学会编程（或者至少对编程有点概念），其意义超出家庭和谐的层面。中国无比巨大的 IT 需求，与有限的 IT 人才供应之间的矛盾，近年来不仅没有缓解，倒有愈演愈烈之势。另外，众多出身、学历平平的青年人有志于投身 IT 这个大有前途（至少有“钱途”）的行业，却苦于迈不过入门的门槛。如果有更多的学校、企业、培训机构照着 Foote 这个思路，用深入浅出、符合直觉、又贴合行业最佳实践的方式来给这些青年人提供指导，不仅能创造若干就业机会，说大一点更是为“互联网＋”宏观战略提供人才支撑，善莫大焉。

最后，祝各位读者阅读愉快，祝佟达及广大程序员都得到家人更多的理解与支持，祝更多麻瓜走进编程这一神秘的魔法世界。

ThoughtWorks 总监咨询师　熊节

2016 年 11 月

译者序

这是最好的时代，也是最坏的时代

在中国，IT 从业者有数百万人之多，但这其中，称得上会编程的，不会超过十分之一。

我所说的会编程，绝不仅仅是会写代码，而是包括环境搭建、版本管理、构建管理、单元测试、文档编写、团队合作，以及任务拆分在内的综合技能。很多人——包括我自己——一开始都无法理解，一个程序员除了写代码，为什么还需要懂这么多东西。

当年在学校，有幸参加一个学生团体，利用课余时间做点小项目。第一次几个人一起写代码，还有点小激动。然而激动并没有延续很久，过了两天，当大家准备把各自写的代码合到一起时，发现这是个根本无法完成的任务，每个人都有自己的想法，从代码组织方式，到代码风格，甚至连 IDE 都不一样。那是第一次感受到，真正的软件开发，和写 C 语言程序设计的作业完全不一样。

2009 年年底，我到微软亚洲研究院的创新工程中心实习，当时的部门负责人是邹欣，他是《移山之道》和《构建之法》的作者。进组的第一天，我就拿到几张纸，上面写着一些基本的编码规范。分配给我的电脑上环境已经准备好，从 TFS 上签出

项目代码，在项目文件中的编译选项已经配置好，直接运行编译，之后执行脚本就可以将服务跑起来。从新人进组到可以开始工作，不到一天时间，真是让我眼界大开。当然，这才是开始，后面从代码提交，到工作项分配，再到上线部署，每一件事都在刷新着我对软件开发的理解。不过，作为一个小小的实习生，那时的我只想安安静静地写代码，没有仔细思考这些专业工作背后的意义。

我毕业后的第一份工作就职于一家研究所，所在的部门人员能力都很强，但是因为信息相对闭塞，对于现代软件开发方法并不是很了解，所以开发团队的很多做法都比较原始，导致需要花费大量的时间和精力在管理代码版本、修复由于更新代码导致一些已有功能不能用的 bug 等上。我尝试将在微软亚洲研究院学到的那些知识引入到团队中，觉得只要搭建起 TFS，就水到渠成了。可惜现实狠狠给我上了一课，搭建 TFS 其实是所有事情中最简单的，设定编码规范、规范代码提交流程、统一编程环境、编写自动化脚本等，每一件事都非常困难。这时我才意识到，如果一个团队中大部分人都不了解这些现代软件开发的知识，靠一两个人去推动，几乎不可能。

后来，我来到 ThoughtWorks，发现这里每个人都能够熟练使用 Git 管理代码，使用 Gulp、Maven、SBT 等管理构建构成，还会写大量的自动化测试来保证质量。从代码修改到测试环境上线，只需要 5 分钟时间，整个过程不需要人参与，程序员们只要看着屏幕上的流水线走到最后亮起绿灯，就可以安心地做下一个任务。后来有人给这种工作方式起了个很直观的名字，DevOps，中文叫作“开发自运维”。 在这里，我了解到了为什么需要敏捷开发，为什么要做持续集成、持续交付，为什么要组建全功能团队。以前对于软件开发的很多疑惑，都慢慢解开了。我常常想，要是我在还没毕业的时候，就知道这些事情，会少走多少弯路啊！

最近几年，参与了几次技术咨询项目，接触到更多软件开发者。很多业界有名的公司，实际上软件开发人员的技能非常不足。完全不理解软件工程任何概念的程序员大有人在，不会使用命令行工具、不知道如何处理代码冲突、从不做单元测试、基本上全靠网上搜索一些代码片段来完成任务，这样的人真心不能算会编程。当然也不乏一些在日常工作中有思考，能够理解软件开发的痛点，但是苦于不知道如何改进的程序员。因为从来没有人告诉他们怎么做才是正确的编程方式。

现在市面上对于每一个流行的技术都有大量的书籍文档做介绍，然而，唯独缺

少一类书，告诉读者如何才能做一名合格的程序员。事实上，我自己以前也一直认为，要想成为一名合格的程序员，需要读很多不同方面的书。直到我看到了本书。

当时接下这本书的翻译，初衷是想要将这本书送给我当时的女朋友，现在的妻子，因为她刚好也是会计，和本书作者在转行做程序员之前的职业一样。我的本意是通过这本书，让我的妻子也可以对编程感兴趣，能够理解我每天对着电脑到底是在干什么。然而当我翻译到第 4 章，介绍 JavaScript 构建工具那部分时，我发现这本书并不是我一开始想的那样，它不只是一本介绍如何写 JavaScript 代码的入门书。在只有两百多页的书中，作者对所有软件开发相关的技能都做了介绍。对于每个编程必备技能，作者仅仅介绍其在日常开发过程中最常用到的一些知识，用 20%的篇幅，把 80%的场景都覆盖到了。不仅如此，因为作者自己从对编程一无所知开始学习，所以在介绍一些相对难理解的概念时，能够设身处地地从初学者角度着想，用直白的语言，将一些概念解释出来。尽管可能从专业人士角度看，不算非常严谨，但对于日常开发工作来说，刚好够用。

对于程序员来说，这是最好的时代，物联网几乎改变了所有行业，甚至已经有人在提“程序员拯救世界”这种说法，作为程序员，我们应感到自豪；然而，这也是最坏的时代，软件开发对程序员的要求越来越高，20 年前也许会用 HTML 设计网页已经非常厉害，但今天你需要会很多不同的技能才能成为一名合格的程序员。如果你希望能够在软件开发这条路上一直走下去，本书可以帮你迈出坚实的一步。

感谢本书的编辑符隆美，在翻译本书期间给我不少鼓励，让我能够坚持译完本书，没有她的帮助，本书不可能完成。

佟达

2016 年 11 月

前言

我为什么写这本书

像大多数伟大的（会计师）故事一样，我的故事开始于一个 Excel 电子表格。那是 2008 年，我正在犹他州普若佛的杨百翰大学学习会计，同时在法学院图书馆做接待员。一天，我的老板问我知不知道如何在 Excel 里把一列名字随机打乱顺序。“当然，”我骗了他，然后去做了每一个说了大话的人都会做的事：谷歌一下。谷歌给我展示了至少三种能够在 Excel 中打乱列表顺序的不同方法。2 分钟后，我就把打乱顺序的列表交给了老板。就在那时，她认定我很擅长计算机，应该去图书馆系统部工作。我不太确定是不是用谷歌检索东西的天赋就等于擅长计算机，但我很感激她这么认为。

我的编程之旅就是从法学院图书馆系统部开始的。第一天，我的老板给了我一本有着 25 年历史的书——《Perl 编程》，并向我展示了我的桌子，坐落在一个没有窗户的房间，上面杂乱地堆放着老旧的计算机、键盘和显示器。临走之前，他告诉我，他正忙着其他事情，但是在他回来之前，这个 Perl 教材会让我一直有事可做。我翻开书，开始阅读，那是我从没有过的感受。

那本书写于 Windows 尚未出现之时，所以它假定读者在用 UNIX 操作系统（我从没听说过 UNIX）。我忽略了这条，继续使用我的 Windows XP 计算机，它刚刚从图书馆计算机实验室退休（在此之前，我从没听说一台计算机对于一个图书馆计算机实验室来说太老了的这种事）。那本书说，给某个地址寄一封贴好邮票的回邮信封，

就会收到一张存着 Perl 的软盘。我决定不邮寄而使用我的谷歌技能，找到下载和安装 Perl 的说明。我以前下载过软件，但我并不知道程序语言也需要下载安装，我有点紧张，怕把我的计算机搞坏，但我发现没法跳过这一步，只能硬着头皮继续。

那本书第 1 章展示了一些示例代码，用来打印 `Hello, World!`，看起来就像这样：

```
print ("Hello, world!");
```

书中让我把这段示例代码保存到一个名为 `hello_world.pl` 的文件中。我之前唯一用过能往计算机里输入文本的软件就是 Word，于是我打开 Word，开始打字。我大概花了 45 秒意识到 Word 不是写代码的地方，然后花了 45 分钟才找到正确的地方。穷尽我所有谷歌技能，都想不出来该搜索什么才能找到写代码的正确方式。我现在进入了一个全新的、不熟悉的世界，我熟悉的那些检索词都没用。最终，我找到了答案——记事本（正如你将在第 1 章“‘Hello, World!’写下你的第一个程序”中会学到的，记事本并不是正确答案，但它确实可以工作），然后我继续艰难前进，只是有一点点沮丧。

最终，我把书中的代码敲进记事本，并保存文件。什么都没发生。但我“很擅长计算机”，而且我知道如果有什么东西不工作，你应该尝试重启。所以我关掉了这个文件，尝试通过双击重新打开它。一个黑底白字的小窗口在屏幕上一闪而过，持续不到 1 秒就消失了，而且记事本也没有打开。那时，我觉得我把计算机给搞坏了，而且我不太相信我的老板不会因为一台计算机坏了而发火，尽管它已经很旧了。然后我的脑中闪过一个想法，就是我的程序事实上已经执行完了。我穿过房间，跑到打印机旁，想看看我是不是已经成功打印了“`Hello world!`”。打印机空闲着，托盘里也没有任何新打印出来的纸张。我试着重启打印机，以防万一。不走运（在很久之后我才知道 print 意味着打印到屏幕，而不是打印到一张纸上），那天下午剩下的时间就在重复着这种令人抓狂的事情。

那天结束时，我感觉我并不擅长计算机；事实上，计算机在用它们的方式对付我，而且它们好像乐在其中。我很想放弃，但我需要工作，而且我不愿言败，我不想让计算机获胜。接下来的几个月里，我只取得缓慢而零星的一些进展。我不断在原地打转，不得不一遍又一遍学习一些相同的东西。唯一可能有能力指导我、回答我问题的人，不是特别忙，就是太有经验以至于帮不上什么忙。就像我在读的那本

有关 Perl 的书（它假设我已经知道怎么用 C 编程，鬼才知道那是什么东西），这些可能的指导老师都认为我懂得比我实际知道的多得多。我不想泄露我什么都不知道，怕丢掉工作（这可能不是个明智的举动）。然而，即使一开始感到沮丧，我也能看出程序是多么强大，而且甚至还挺享受的。最终，知识碎片开始汇集到一起，我慢慢开始理解。

我的编程入门之路经历了许多的坎坷，而且我意识到，每个希望自学编程的人，都可能有类似的经历。有几次，我想要放弃，觉得编程就是给那些计算机科学系的书呆子学生准备的。编程在刚开始学习时确实有些吓人，而有经验的程序员知道那么多东西，以至于让人不敢对他们提问题。但是不管有多难，不管多少次你可能想要用头撞桌子或是把计算机扔到地上，编程最终会带给你惊人的乐趣和回报。当我意识到编程有多么棒时，我放弃了会计学硕士学位和一个在顶级会计师事务所工作的机会，转而从事编程。我从没后悔。我写下这本书，是因为当年刚开始学编程时，就想有这么一本书。

为什么你应该读这本书

计算机就在我们身边，遍布我们生活的方方面面，然而大多数人还并不真正理解它们的工作方式，或者如何让计算机为我们工作。我们都被限制在计算机已经知道怎么做的那些事情上，但是计算机一直以来都意图成为可编程的机器。你可以为你桌上的计算机编程，让它做任何你想要做的事。随着世界越来越依赖于计算机，编程技能对每个人来说都将是必需的，不仅仅对专业软件工程师或开发者来说是如此。这本书会帮助你学习编程，并且乐在其中。

在接下来的几页，你会建立编程基础，为你实现全部编程目标做好准备。不论你想要成为一个专业的软件开发者，还是想要学习如何更高效地和程序员沟通，或者只是对于程序如何工作感到好奇，这本书都非常适合作为帮你达成所愿的第一步。学习编程仍将困难，但却有可能，并且有希望变得有趣，而不是让人沮丧。

你的项目

最好的学习编程的方式就是真正去编程。贯穿本书，你将编写一个完整的 Chrome 浏览器扩展程序。Chrome 浏览器扩展程序是能够增强（或者扩展）Chrome 浏览器功

能的程序。我们即将搭建的这个扩展程序会向用户询问名字和手机号，然后把用户 Facebook 新鲜事里的所有照片，都根据用户的位置（通过电话号码确定）改成小猫小狗。这个扩展程序（我们称之为猫咪书）可能没什么特别的用处，但搭建它的过程会很有趣，而且期间会帮你学到很多重要的编程概念。

目录