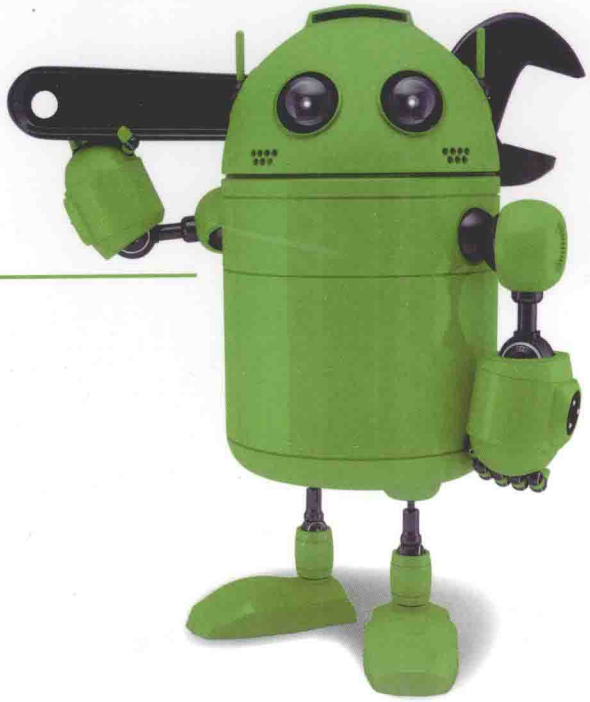
Broadview®
www.broadview.com.cn

Android开发工具系列详解

Android 群英传

神兵利器



徐宜生◎编著

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

Android 辞典传

神兵利器

徐宜生◎编著

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

内 容 简 介

本书以通俗易懂的语言介绍了 Android 开发的工具使用。全书共分为 7 章。第 1 章主要讲解如何搭建一个优雅、令人愉悦的开发环境。第 2 章主要讲解协同开发最重要的工具 Git。第 3 章主要讲解 Android Studio 的一些不为人知的使用技巧。第 4 章主要讲解 Android 最新的编译工具 Gradle 的使用技巧。第 5 章主要讲解 SDK 和开发者选项中提供的工具的使用方式。第 6 章主要讲解 Android 提供的一些性能优化的工具及其使用技巧。第 7 章主要讲解个人开发者和团队开发者在学习、工作中经常使用的一些工具。

本书适用于各个层次的 Android 开发者，不论是初出茅庐的开发者还是资深的开发者。工具的使用永远是一门讲不完的学问，笔者希望抛砖引玉，让开发者能够驾驭好各种工具，为己所用。

图书在版编目（CIP）数据

Android 群英传：神兵利器 / 徐宜生编著. —北京：电子工业出版社，2016.9
ISBN 978-7-121-29602-4

Ⅰ. ①A… Ⅱ. ①徐… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2016)第 179688 号

策划编辑：官　杨
责任编辑：徐津平
印　　刷：北京嘉恒彩色印刷有限责任公司
装　　订：北京嘉恒彩色印刷有限责任公司
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：787×1092　1/16　印张：25.75　字数：589 千字
版　　次：2016 年 9 月第 1 版
印　　次：2016 年 9 月第 2 次印刷
定　　价：79.00 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

在看到这本书之前，我作为一名程序员已经在各种工具的海洋里摸爬滚打了多年，而各种新事务的层出不穷，不断提高的开发效率，越来越完善的开发环境，也是沉浸在开发里的一大乐趣。在工作中也常常听到同事们发现了新工具时的欢呼，发现工具另一种用法时的喜悦，以及在深入学习和思考后，自行修改工具以实现更高目标的专业。甚至有的时候，找到一款好用的工具，比写出一段高效无误的代码或是解决一个实际难题更让人兴奋。

诚然，对于开发者来说，现在已是工具之争，好的开发工具变得越来越重要。产品设计的需求总让人觉得他们欲求不满，总要不停地解决各种问题。也正是这样的客观现实，使我们不得不借助各种工具来应对挑战，也有了越来越多的人加入到开发工具的行列。要简单易用，且能使开发变得高效和稳定着实不易。

而面对琳琅满目的工具，很多开发者就迷失了方向，到底什么才是适合自己的，适合自己的项目的，甚至与自己的开发理念吻合的。本书作者另辟蹊径，从工具本身着手，针对 Android 开发的每一处细节，对每一个常用工具都给予了详细的剖析讲解，对于 Android 程序员来说，着实是省了很大的力气，也更容易在工具的帮助下，快速实现开发的需求。当然，这一切都是在有趣的前提下，作者的行文风格直白清爽，读起来非常轻松，让读者在潜移默化之中就认识，熟悉，并掌握了这些工具。

同样的，作为一名程序员，我深知这本书的难写，相较于使用，总结和知识的传承更显难得和珍贵。不论是哪个层次的程序员，在这本书的引领下都会遇见一个新天地，这确实是程序员们的一大福音。在 Android 开发之外，理解工具的使用也同样重要，我想作者要传达的也是这个意思。

最后也感谢本书作者徐宜生先生的邀请，让我为这本书写序，也让我有幸提前读到了本书，实在是人生一大快事。

何晓杰

沪江高级架构师、知名开发者、技术投资人

前　言

在笔者的第一本书《Android 群英传》上市之后，得到了很多读者的好评，也收到了很多读者对于该书的意见和建议。在此，笔者对广大读者朋友表示最衷心的感谢，感谢你们一直以来的支持。

写书一直都是一件苦差事，能支撑我走下去的，就是读者们的支持。只要笔者的书对读者有一点点帮助，不论是解决了一个项目中的 Bug，还是成功回答了面试官的问题，对笔者来说，都是莫大的鼓励。也正是这些鼓励，让笔者坚持到了今天，坚持到了第二本书的出版。

➘ 第二本书

由于书籍的篇幅和内容限制，笔者有很多内容都无法在《Android 群英传》中尽善尽美地表述出来，因此笔者在写完《Android 群英传》之后，萌生了创作后续作品的想法。最终，笔者将第二本书命名为——《Android 群英传：神兵利器》。第一本书《Android 群英传》，以 Android 开发中的重、难点知识点为基础，对如何学习、理解并掌握这些知识点进行讲解。而第二本书，笔者不再继续讲解 Android 中的知识点，而是向大家介绍如何使用工具进行高效的 Android 开发，很明显两本书的重点各不相同，内容相辅相成。

➘ 工具之道

古人有云，工欲善其事，必先利其器。好的工具，可以事半而功倍。人类的发展历程，也是一个工具革新的历程。人类不断创造工具，改善生活，从而推动着社会的进步。对于程序员来说，工具更是有着举足轻重的意义。在软件开发界，有一句非常有名的话——Stop Trying to Reinvent the Wheel，即不要重复造轮子。这也是本书的宗旨——让读者善于使用工具以提高开发的效率。

笔者一直认为工具是程序员最好的伙伴。普通程序员使用工具，高级程序员驾驭工具，神级程序员创造工具。这也是一个开发者，从普通程序员到优秀程序员的进阶之路。普通程序员也许只是懂得在合适的场合使用合适的工具。而优秀程序员，则是那些能够驾驭这些工具的开发者，他们是设计师，通过工具创造美妙的程序。开发者需要了解、驾驭你的工具，知道何时、何地该怎样使用工具，以便快速、准确地解决问题。

笔者相信，这个世界上没有什么事情是不能通过工具来解决的。如果有，那么就创造

一个工具去解决。

关于本书

本书共分为 7 章，分别是：

第 1 章主要讲解如何搭建一个优雅、令人愉悦的开发环境。开发者绝不是“码农”，而是要去享受创造的乐趣的，所以一个高效的开发环境就显得尤为重要了。正所谓——开发环境搭得好，程序设计乐逍遥。

第 2 章讲解协同开发最重要的工具——Git。它可以说是目前团队开发的基础，也是版本控制的核心工具。正所谓——项目要想跑得好，版本控制不可少。

第 3 章主要讲解 Android Studio 的一些不为人知的使用技巧，发掘出 Android Studio 作为一个强大工具的巨大力量。正所谓——Android Studio 大揭秘，省出时间玩游戏。

第 4 章主要讲解 Android 最新的编译工具 Gradle 的使用技巧。虽然 Gradle 的学习曲线比较陡峭，但如果说 Android Studio 是一把宝剑，那么掌握好 Gradle，就好比一块磨刀石，可以把宝剑打磨得愈发锋利。正所谓——与 Gradle 的爱恨情仇，让你一次爱个够。

第 5 章主要讲解 SDK 和开发者选项中提供的工具的使用方式。这些工具也是开发者最容易忽视的工具。正所谓——珍视身边的朋友，从开发者工具做起。

第 6 章主要讲解 Android 提供的一些性能优化的工具及其使用技巧。利用好这些工具，是进行性能优化的必备前提。正所谓——探究性能秘史，了解尘封往事。

第 7 章主要讲解个人开发者和团队开发者在学习、工作中经常使用的一些工具。正所谓——个人团队轮流转，工具真情长相伴。

本书读者对象

本书适用于各个层次的 Android 开发者，不论是初出茅庐的开发者还是资深的开发者。工具的使用永远是一门讲不完的学问，笔者希望抛砖引玉，让开发者能够驾驭好各种工具，为己所用。

致谢

感谢朋友、群友在我写书的这段时间内对我的帮助，也感谢电子工业出版社的官杨女士和出版社的编辑们对我文章的核对和建议，没有你们的帮助也就没有这本书的诞生。此

外，还要特别感谢我的妻子朱佳，感谢你一直以来对我的包容和支持，没有你也就没有这两本书的诞生，我会爱你一辈子。

资源与勘误

由于个人能力的局限，虽已竭尽全力，但对于书中的一些问题的分析难免会有纰漏，实例中的解决方法可能也不是尽善尽美，请读者海涵。希望读者朋友能将发现的问题及时向我反馈，我将感激不尽。本书的勘误与读者的反馈内容都将在我的个人博客上不断更新。

目　录

第 1 章

程序员小窝——搭建高效的开发环境

程序员的电脑、书桌就是程序员的小窝。在这块小天地里面，程序员要完成开发、学习的任务，那么一个高效、优雅的开发环境就显得尤为重要。古人云：居不可一日无竹。一个良好的环境是提高工作效率的保证。本章将向大家讲解如何做一名程序员中的雅士，在优雅的开发环境中完成自己的工作。

1.1 搭建高效的开发环境之操作系统

与大多数开发者一样，笔者最早接触的也是 Windows 系列操作系统，当然身边也有一些使用 MacBook 和 Ubuntu 的人。对于个人用户来说，MacBook 的优势或许只是在于优美的外观颜值，而对于开发者来说，笔者认为 MacBook 的优势在于它集 Windows 的易用性与 Linux 的高可开发性于一体，因此特别适合开发者使用。

在国外很多的极客大会上，开发者、工程师最钟爱的就是 Mac 系统。相对于 Windows，MacBook 使用的是 Unix 系统，它是 Linux 系统的始祖，与 Linux 一样具有一切对开发者友好的优点。首先要提的必须是系统的终端命令行工具（Terminal）。

每个操作系统基本都有自己的终端命令行工具（Terminal）。在 Windows 中，可以通

过快捷键“Win+R”，输入 CMD 调出命令行工具，默认的命令行工具如图 1.1 所示。

图 1.1　CMD 窗口

但是相信用过 CMD 窗口的朋友一定对 Windows 提供的这个终端工具有很多“吐槽”。其中一部分原因是由于 Linux 在程序开发界的流行导致的。大部分的开发者在终端中都熟悉 Linux 的操作指令，而很少使用 Windows 下 CMD 的操作指令，这就导致开发者在 CMD 终端中的各种操作不便。而 MacBook 则不存在这些问题，它本身就是 Linux 的鼻祖，因此它几乎支持所有的 Linux 指令。在 MacBook 的终端中操作，与在 Linux 的终端中操作几乎没有什么区别。一个最普通的命令行终端如图 1.2 所示。

图 1.2　Mac 终端窗口

既然 Linux 中的终端与 MacBook 的终端几乎一模一样，那么我们为什么要花那么高的价钱去买 MacBook，而不选择免费的 Linux 呢？如果你的经济不是太富裕（单身的程序员除外），那么 Linux 也是一个非常好的选择。但是 Linux 相对于 Windows 和 MacBook 来说，

又有一点太过于极客了。使用 Linux 几乎可以使用“折腾”一词，要使用好 Linux，上手难度还是有一点的。由此可见，MacBook 几乎就成了开发者最好的选择，处于 Windows 和 Linux 中间，鱼和熊掌可以兼得。

Mac 系统的优势不仅仅在于终端的易用性，搭上了 Linux 的顺风车，大量的开源软件和开发工具可以非常容易地用来开发 Mac 版。同时，Mac 系统还不用担心 Windows 下的各种电脑病毒和木马，也不用清理磁盘碎片，甚至不用安装各种驱动程序，对于这一点，相信做 Android 开发的同学深有体会。在 Windows 上，不同的 Android 手机需要安装不同的驱动软件，否则系统无法连接到 Android 设备。而在 Mac 系统下，由于 Mac 与 Android 内核都是 Unix\Linux 架构，不需要任何驱动程序就可以直接使用。

另外，不得不说 Mac 系统的设计哲学，将一切操作都简化到了极致。很多细小的设计点，不得不让人佩服乔布斯的眼界与思考能力。比如 Mac 最早不惜成本引入 SSD 硬盘，将系统的性能提升到一个新的境界；再比如 Mac 的多窗口环境，可以最大化地利用桌面，同时还能方便地在不同工作区中进行切换；再比如 Mac 系统的触控板、触发角等快捷工具，将各种操作集于一身极大地降低了操作成本。当然，Mac 系统也并不是完美的。由于 Windows 系统最早的窗口可视化设计，让它占领了 PC 的大部分江山，所以很多游戏基本上只支持 Windows 系统（不过笔者觉得这也许也是一个好处，可以帮助开发者远离游戏的诱惑）。同时，由于 Mac 系统是基于以安全著名的 BSD Unix 系统改进而来，所以 Mac 系统的安全性是非常高的，这也导致很多软件在 Windows 上能实现的功能在 Mac 上是无法实现的。这一点对比 Windows 版的 QQ 和 Mac 版的 QQ 就可以发现。这一点也是有利有弊的双刃剑，一方面在 Mac 系统下，有些软件无法展现在 Windows 中的强大功能；但另一方面，基于 Unix 系统的架构却可以让 Mac 使用非常多的高质量开发工具。

由于在 Mac 系统中，有些按键与常用的 Windows 按键有所不同，所以有时候在看一些配置的时候，初学者可能找不到对应的按键，因此这里对常用的按键进行一下讲解。

Command ⌘　Shift ⇧　Option ⌥　Control ^　Caps Lock ⇪

初学者应该多使用快捷键，如下所示的网址正是 Apple 官网上提供的 Mac 快捷键一览表。

https://support.apple.com/zh-cn/HT201236

在这个网址上，读者可以找到几乎所有的 Mac 快捷键，用好这些快捷键，是让 Mac 为你高效工作的基础。这里笔者列举一些常用的快捷键。

- 窗口操作

切换同一应用的多个窗口——Command + ~，这个快捷键非常有用，例如打开了多个

Android Studio 窗口，就可以通过这个快捷键进行切换。

关闭当前窗口——Command + W，这个快捷键可以关掉当前所在的这个应用的窗口，如果这个应用有多个窗口，那么这个应用不会被关闭，除非使用 Command + Q 快捷键。

新建窗口——Command + N，这个快捷键在很多应用中都是适用的，例如终端、浏览器等，通过这个快捷键可以快速创建该应用的新窗口。

- 截图

自由截图——Command + Shift + 4，这个快捷键可以像 QQ 截图那样截取任意大小的窗口，截图会保存在 Desktop 上

截取当前窗口——Command + Shift + 4 + 空格键，如果要截取当前窗口，那么只需要在自由截取的基础上，按一下空格键即可截取。

- 编辑

行首行尾——Command + Left\Right，通过这个快捷键，可以快速移动光标到行首或者行尾。

按单词移动——Option + Left\Right，通过这个快捷键，可以按单词进行光标的移动。

页首页尾——Command + Up\Down，通过这个快捷键，可以在一页的页首和页尾中快速切换。

删除行——Command + Delete，通过这个快捷键，可以快速删除一行。

虽然本章中提到的很多软件都是 Mac 中的，但是在 Linux 平台和 Windows 平台上，几乎都可以找到类似的替代软件。特别是 Linux 平台，Mac 上能够使用的软件，在 Linux 上基本都能找到，毕竟它们同根同源。而最近微软也在 Windows 10 中增加了对 Linux Bash 的支持，这意味着 Windows 平台对于 Linux 的支持也指日可待了（目前 Windows 平台上也可以使用 cygwin 来模拟 Linux 系统环境）。

1.2 搭建开发环境之高效配置

由于 Mac 系统的种种便利，笔者认为最合适的开发系统依次是 Mac > Linux > Windows。下面笔者将与大家分享开发环境的配置方法，帮助读者搭建一个高效的开发环境。

基本环境配置

这一小节，笔者将分享一些开发环境的基础配置。

Fn 键

Fn 键在 Mac 系统中默认是需要在按住 fn 功能键后才能使用的。但是 Fn 键在各种 IDE 中的功能是非常重要的，很多快捷键都包含有 Fn 键，因此笔者认为最好将 Fn 键改为标准的功能键，而不是需要按住 fn 功能键后才能使用的辅助快捷键。

通过在“系统偏好设置-键盘”中选中“将 F1、F2 等键用作标准功能键”，如图 1.3 所示，即可将 Fn 键恢复为标准快捷键，这样在各种 IDE 中就可以直接使用，而不需要配合 Fn 键了。

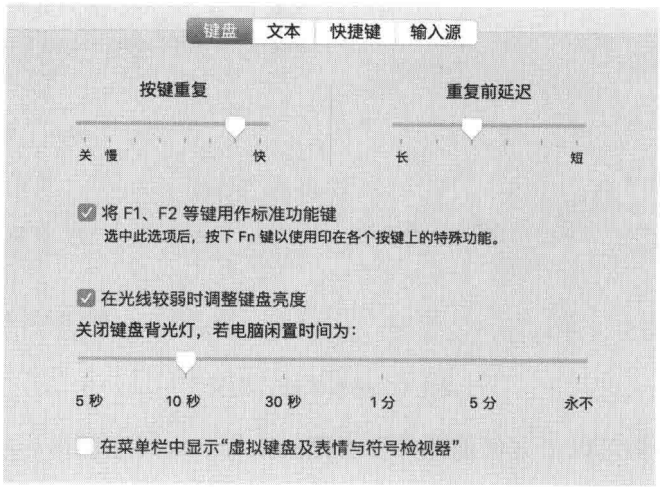


图 1.3　修改 Fn 功能键

这样修改的一个原因就是在很多 IDE、编辑器中，Fn 键都是一些快捷键，这样设置可以比较方便地使用这些快捷键。

虽然本文主要是以 Mac 为例进行讲解，但其实现在很多笔记本电脑默认都是这样的设置。例如 IBM、华硕的一些笔记本电脑，Fn 键都是与 fn 功能键一起使用的（例如 F8 实际上需要按住 fn 功能键和 F8 键才能使用）。这点在调试代码的时候会比较麻烦，建议开发者使用单独的 Fn 键。

Trackpad 触控板

MacBook 一个非常好的设计就是将触控板变得非常强大。大部分未使用过 Mac 的读