

件是从 Java 编译来的，还是从 Kotlin 编译来的，只要是符合规格的 class 文件，它都能识别。也正是这个原因，JetBrains 才能以一个第三方公司的身份设计出一门用来开发 Android 应用程序的编程语言。

现在你已经明白了 Kotlin 的工作原理，但是 Kotlin 究竟凭借什么魅力能够迅速得到广大开发者的支持，并且仅在 1.0 版本发布一年后就成为 Android 官方支持的开发语言呢？

这就有很多原因了，比如说 Kotlin 的语法更加简洁，对于同样的功能，使用 Kotlin 开发的代码量可能会比使用 Java 开发的减少 50% 甚至更多。另外，Kotlin 的语法更加高级，相比于 Java 比较老旧的语法，Kotlin 增加了很多现代高级语法的语法特性，使得开发效率大大提升。还有，Kotlin 在语言安全性方面下了很多工夫，几乎杜绝了空指针这个全球崩溃率最高的异常，至于是如何做到的，我们在稍后就会学到。

然而 Kotlin 在拥有众多出色的特性之外，还有一个最为重要的特性，那就是它和 Java 是 100% 兼容的。Kotlin 可以直接调用使用 Java 编写的代码，也可以无缝使用 Java 第三方的开源库。这使得 Kotlin 在加入了诸多新特性的同时，还继承了 Java 的全部财富。

那么既然 Kotlin 和 Java 之间有这样千丝万缕的关系，学习 Kotlin 之前是不是必须先会 Java 呢？我的回答是：如果你掌握了 Java 再来学习 Kotlin，你将会学得更好。如果你没学过 Java，但是学过其他编程语言，那么直接学习 Kotlin 也是可以的，只是可能在某些代码的理解上，相比有 Java 基础的人会比较吃力一些。而如果你之前没有任何编程基础，那么本书可能不太适合你阅读，建议你还是先从最基础的编程入门书看起。

另外，本书不会讲解任何 Java 基础方面的知识，所以如果你准备先去学习 Java 的话，请参考其他相关书。

好了，对 Kotlin 的介绍就先讲这么多吧，在正式开始学习 Kotlin 之前，我们先来学习一下如何将一段 Kotlin 代码运行起来。

## 2.2 如何运行 Kotlin 代码

本章的目标是快速入门 Kotlin 编程，因此我只会讲解 Kotlin 方面的知识，整个章节都不会涉及 Android 开发。既然暂时和 Android 无关了，那么我们首先要解决的一个问题就是怎样独立运行一段 Kotlin 代码。

方法大概有以下 3 种，下面逐个进行介绍。

第一种方法是使用 IntelliJ IDEA。这是 JetBrains 的旗舰 IDE 开发工具，对 Kotlin 支持得非常好。在 IntelliJ IDEA 里直接创建一个 Kotlin 项目，就可以独立运行 Kotlin 代码了。但是这种方法的缺点是你还要再下载安装一个 IDE 工具，有点麻烦，因此这里我们就不使用这种方法了。

第二种方法是在线运行 Kotlin 代码。为了方便开发者快速体验 Kotlin 编程，JetBrains 专门提