

Android Developers

Kotlin and Android



We have been watching Kotlin adoption on Android steadily rise over the years, with increasing excitement among developers. Kotlin is expressive, concise, extensible, powerful, and a joy to read and write. It has wonderful safety features in terms of nullability and immutability, which aligns with our investments to make Android apps healthy and performant by default. Best of all, it's interoperable with our existing Android languages and runtime. So we're thrilled to make Kotlin an official language on Android.

If you're interested in using Kotlin, it's easy to get started because it works side by side with Java and C++ on Android. So you can keep your existing code, continue to use the various Android libraries, and incrementally add Kotlin code to your project. Unlike almost any other language, Kotlin is a drop-in replacement you can use bi-directionally—you can call into the Java language from Kotlin, and you can call into Kotlin from the Java language.

Of course, IDE support is also crucial, and we have it. Android Studio is built upon IntelliJ IDEA, an IDE built by JetBrains—the same company that created the Kotlin language. The JetBrains team has been working for years to make sure Kotlin works great with IntelliJ IDEA. So we're inheriting all their hard work. Starting with Android Studio 3.0, tooling support for Kotlin is bundled directly into Android Studio.

We believe Kotlin is an excellent fit for Android not only because it gives developers what they want, but also because it matches the spirit of Android. Just like Android, Kotlin has always been an open source project, primarily under Apache 2. It involves an entire community and does not belong to just one company. So we're working with JetBrains to move Kotlin into a non-profit foundation. Our choice of Kotlin reaffirms our commitment to an open developer ecosystem as we evolve and grow the Android platform, and we are excited to see the language evolve.

And while we're adding Kotlin as an official Android language, we're also expanding our investment in our existing languages. For example, in Android O we added support for more Java 8 libraries

(<https://developer.android.com/studio/preview/features/java8-support.html>). In Android Studio 3.0, Java 8 language features are

now directly supported with the javac compiler (<https://android-developers.googleblog.com/2017/04/java-8-language-features-support-update.html>). And on C++, we're making ongoing investments such as expanding performance profiling tools (<https://android.googlesource.com/platform/system/extras/+/master/simpleperf/README.md>) and APK debugging tools (<https://developer.android.com/studio/preview/features/apk-debugger.html>) to fully cover the native experience; and significantly improving libc support (<https://github.com/android-ndk/ndk/wiki/Changelog-r13#ndk>) and updating the NDK to ensure you can access modern headers (<https://android-developers.googleblog.com/2017/03/introducing-android-native-development.html>) even when you target older APIs.

In conclusion, we think Kotlin is a great addition to the Android language family and a delightful language to try. If you're not already a Kotlin user, read [Get Started with Kotlin on Android](https://developer.android.com/kotlin/get-started.html) (<https://developer.android.com/kotlin/get-started.html>). If you have more questions, check out the [Kotlin on Android FAQ](https://developer.android.com/kotlin/faq.html) (<https://developer.android.com/kotlin/faq.html>).

More information

YOUTUBE

Kotlin: Android Support Announced at Google I/O

Visit these sites for more information: Kotlin website - <https://goo.gl/W6bJJJ>, Kotlin and Android documentation - <https://goo.gl/NWt6O7> Android announced full support for the Kotlin language at Google I/O along with ; ...

YOUTUBE

Getting Started with Kotlin

Learn how to start a new project with the Kotlin language enabled or how to convert your existing classes using the Convert Java file to Kotlin refactoring.

DEVELOP**Resources to Learn Kotlin**

If you're new to Kotlin, we recommend you check out the following resources to get started with the language.

