

## Android Developers

# Kotlin on Android FAQ

### Why did Android make Kotlin a first-class supported language?

Kotlin is an Android-compatible language that is concise, expressive, and designed to be type- and null-safe. It works with the Java language seamlessly, so it makes it easy for developers who love the Java language to keep using it but also incrementally add Kotlin code and leverage Kotlin libraries. Also, many Android developers have already found that Kotlin makes development faster and more fun, so we want to better support these Kotlin users. You can read more about Kotlin and Android here (<https://developer.android.com/kotlin/>).

### I'm already using the Kotlin language, what changes for me?

In the short term, we think the biggest change you'll notice is that Android Studio 3.0 and higher includes fully tested Kotlin support. We think this will result in a much easier, more stable development experience for you.

### How do I use Kotlin with Android Studio?

Kotlin is fully supported in Android Studio 3.0 and higher. Previously, using Kotlin required that you add the Kotlin plugin, but all new releases of Android Studio ship with these tools built in. So you can create new projects with Kotlin files, convert Java language code to Kotlin, debug Kotlin code, and more, without any extra steps. See [Get Started with Kotlin](#).

### How do I debug Kotlin in Android Studio?

Debugging Kotlin works just like debugging Java code. You don't need to do anything differently.

### What kind of other IDE support is provided for Kotlin (like lint, autocomplete, refactorings, etc.)?

As of Android Studio 3.0, the IDE has full tooling support for Kotlin. There are, however, some known issues and limitations (<https://developer.android.com/studio/preview/kotlin-issues.html>) that we're working on.

### What's the future of Kotlin?

JetBrains' thoughtful work on Kotlin's design is one of the reasons we're embracing the language. Google is partnering with JetBrains to ensure a wonderful overall developer story—from language, to framework, to tools. And, we are excited to be working together to move the Kotlin language into a non profit foundation.

## Is Kotlin open source?

The preferred license for Kotlin is the Apache Software License, Version 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>) ("Apache 2.0"), and the majority of the Kotlin software is licensed under Apache 2.0. While the project will strive to adhere to the preferred license, there may be exceptions that will be handled on a case-by-case basis. For example, certain third-party dependencies used by Kotlin may be licensed under different open source licenses which are still compatible with the Apache 2 license.

## How do I choose between the Java and Kotlin languages?

You don't have to pick! You can use both together as you see fit. If you need help discovering whether Kotlin is a good fit for you, you can try it on Android (<https://developer.android.com/kotlin/get-started.html>) or learn more about the language with these Kotlin resources (<https://developer.android.com/kotlin/resources.html>)

## Can I call Android or other Java language library APIs from Kotlin?

Yes. Kotlin provides Java language interoperability. This is a design that allows Kotlin code to transparently call Java language methods, coupled with annotations that make it easy to expose Kotlin-only functionality to Java code. Kotlin files that don't use any Kotlin-specific semantics can be directly referenced from Java code without any annotations at all. Combined, this allows you to granularly mix Java code with Kotlin code. To learn more, see Kotlin's interop documentation (<https://kotlinlang.org/docs/reference/java-interop.html>).

## Can I use both Java files and Kotlin files in the same project?

Yes. You can adopt as much or as little Kotlin as you like and mix it with Java code using Kotlin's interoperability with Java (<https://kotlinlang.org/docs/reference/java-interop.html>).

## Can I use Kotlin with C++?

Yes, JNI is fully supported with Kotlin. Simply mark the JNI method with the external modifier (<https://kotlinlang.org/docs/reference/java-interop.html#using-jni-with-kotlin>).

## How do I add Kotlin to my new projects?

When you create a new project in Android Studio, just check the **Include Kotlin support** checkbox. For more information, see Get Started with Kotlin (<https://developer.android.com/kotlin/get-started.html>).

## How do I add Kotlin to my existing projects?

Select your module in the **Project** window, and then select **File > New**, select any Android template, and then choose **Kotlin** as the **Source language**. For more information, see [Get Started with Kotlin \(https://developer.android.com/kotlin/get-started.html\)](https://developer.android.com/kotlin/get-started.html).

### How do I convert Java language code to Kotlin?

Open a Java file and select **Code > Convert Java File to Kotlin File**. Or, create a new Kotlin file (**File > New > Kotlin File/Class**), and then paste your Java code into that file—when prompted, click **Yes** to convert the code to Kotlin. For more information, see [Get Started with Kotlin \(https://developer.android.com/kotlin/get-started.html\)](https://developer.android.com/kotlin/get-started.html).

### Will there be parallel docs, samples, codelabs, and templates in Kotlin?

We're working to make our documentation and educational materials as useful as possible to both Java and Kotlin language users. In the meantime, developers can rely on Kotlin's excellent interoperability with the Java language and the ability to automatically translate Java language code to Kotlin in Android Studio.

### Do Kotlin coroutines work on Android? How about async/await?

Kotlin coroutines should currently work, but they are currently an experimental design. As such, Kotlin makes no guarantees about future status, and thus, neither does Android.

### What's Kotlin's impact on APK size / method count?

The Kotlin runtime adds about 7,000 methods and ~1MB to your debug APK. The net impact might be less if you use Kotlin to replace another library in your project, such as Guava or RxJava. This size also reduces when you optimize the APK for release with Proguard, just like other app code and libraries.

### Does using Kotlin have any performance impact?

Kotlin doesn't have a direct performance impact, but just as with the Java language, you should be thoughtful about how you use it. For example, repeated copying between new collection instances can impact GC performance, and calling a method that accepts non-null types adds a method call for the null check (though you can disable runtime null checks in the compiler with `-Xno-param-assertions`).

### Which versions of Android does Kotlin support?

All of them! Kotlin is compatible with JDK 6, so apps with Kotlin safely run on older Android versions.

### Where can I learn more about using Kotlin?

Check out Resources to Learn Kotlin (<https://developer.android.com/kotlin/resources.html>).

---



