

# Introduction to Kotlin

## We'll cover the following ^

- Kotlin?
- Features of Kotlin

## Kotlin? #

Ah, Kotlin—that's an island off St. Petersburg, Russia, but this book is about its namesake programming language. Programmers who use Kotlin don't just like the language—they say they love it. What are the reasons for such affection? That's the question we'll quickly start with. Then we'll be on our way to install the Kotlin Software Developer Kit (SDK), write some code, and compile and execute it so we can see it working.

Imagine taking the best of many different languages—C++, C#, Erlang, Groovy, Java, JavaScript, Python, Ruby, Scala, Smalltalk—throwing them into a blender and turning it on; the resulting cocktail is Kotlin. The strength of Kotlin is in its diversity.

[Andrey Breslav](#) and the team of developers behind the language at [JetBrains](#) set out to create a fluent, expressive, pragmatic, and easy to use language that is less verbose than many mainstream languages. As programmers pick up Kotlin, they quickly recognize good parts of their familiar languages and, at the same time, are intrigued by other awesome capabilities they've not been exposed to before. The familiar ideas in Kotlin makes programmers feel at home as they learn and adopt the language, yet the ideas that are new to them make them more productive compared to the languages they're used to. That's part of the reason why programmers are passionate about Kotlin.

The biggest uptick in interest for Kotlin came right after Google's announcement that Kotlin is an official language for [Android development](#). An endorsement from Google is certainly significant, but there are more reasons to be excited about Kotlin

Kotlin is one of the few languages that can be used for server-side, mobile/Android, and front-end development. Code, written appropriately, can compile down to Java bytecode or may be transpiled (compiled from the source code of one language to the source code of another language) to JavaScript. Kotlin/Native supports targeting platforms, including iOS, macOS, Linux, Windows, and WebAssembly, to compile your source code to native binaries. That makes Kotlin one of the few languages you can use for full-stack development.

## Features of Kotlin #

As you journey through Kotlin, you may recognize a number of these features and trace their roots:

- Though syntactically different, Kotlin is semantically similar to Java, making it easy for Java programmers to adapt.
- Without inheriting from a class, you may add your own domain-specific convenience methods to classes. These methods, called extension functions, may be used just like methods that are part of the original class, with full integrated development environment (IDE) support. That's like C#-style extension methods in Kotlin, although Kotlin has much richer capabilities.
- Delegation is often a better design tool than inheritance to reuse code. Kotlin, inspired by languages like Groovy and Ruby, is versatile in the ways you can delegate method calls from one object to another, without compromising type safety.
- You can use the concise and elegant argument-matching syntax in Kotlin, which is similar to Erlang and Scala syntax, instead of the more verbose series of nested if-else statements.
- Extending existing functions and methods is easy in Kotlin (though it requires recompilation due to binary incompatibility), thanks to its default parameters' capabilities, similar to JavaScript, Scala, Ruby, and Python.
- Named arguments, like in Groovy, Scala, and Ruby, make the code highly expressive, easier to read, and less error prone.
- Where it makes sense, you may overload operators on your own classes or on

third-party classes, much like in languages like C++ and Groovy.

- The elegance, fluency, and concise nature of Kotlin comes together to support creating internal DSLs, similar to languages like Groovy and Ruby, but with full support for static type checking.
- You can write C-style procedures, Scala-style scripts, Java-like OO code, and Smalltalk/Erlang-like functional-style code in Kotlin.
- Kotlin is leading innovation in the area of asynchronous programming with coroutines and continuations.

---

These are just a few of the significant features that are prominent in Kotlin. The next lesson describes more extensive functionalities of Kotlin.