

Which Option to Choose?

We'll cover the following ^

- Options to run Kotlin code

Options to run Kotlin code

Kotlin doesn't dictate which option you choose to run the code; it's entirely up to you, depending on your needs and liking. Here are a few things to consider when choosing your option:

- To intermix Kotlin code with Java or other languages on the JVM, compile the code using `kotlinc-jvm`. Then, simply throw the jar file into the classpath or modulepath along with a reference to the `kotlin-stdlib.jar` file, and use it like you'd use any jar created from Java source code using `javac`.
- If you have multiple Kotlin files and intend to run the program as a Kotlin program, then use the `kotlin` tool to run the code.
- To implement system-level or back-end tasks using Kotlin, create a single Kotlin file and run it as script using the `-script` option. Alternatively, use the shebang to directly run the file from the command line or a cron task.
- To enjoy static typing and compile-time verification that comes with Kotlin for creating web applications, use the option to compile Kotlin to JavaScript.
- To run Kotlin code on different native platforms, like iOS and WebAssembly, use Kotlin/Native to compile the code to the desired targets.
- During active development, run the code from within the IDE for ease of development and the convenience offered by the IDEs.
- To experiment with small snippets of code, run `kotlinc-jvm` as REPL.

The next lesson concludes the discussion for this chapter.

