

# Kotlin Multiplatform

# Pros:

- Shared logic
  - Common API for platform-specific functionalities easier than Flutter/React
    - I.e. Drawing in Android vs Windows, can be streamlined by one common API
  - Allows for scaling of existing applications across multiple platforms.
    - I.e. both can use the same codebase for business logic.
- Kotlin more popular than Flutter's Dart (2022)
- More compile-time safety than JS (React Native)
  - Errors can be found during compilation, not as a part of running the application.
- Easy to use (for Android devs)

# Cons

- Started 2017, was in Beta until recently (November 1st, 2023).
  - <https://blog.jetbrains.com/kotlin/2023/11/kotlin-multiplatform-stable/>
- Community is in its infancy, as is the structure methodology - as it is reported as "experimental" (August, 2023)
- Steep learning curve as a result; few official examples/tutorials, even fewer dedicated forums.
- Various issues when Kotlin is translated to Objective-C, WIP.

# How it works

- Multiple compilers for shared module:
  - Android => Kotlin/JVM
  - iOS => Kotlin/Native code.
- The shared module can be added to projects, allowing Developers of UI to focus on one instead of 2 developments, so to speak.
- Converts Kotlin to Intermediate Representation, which is then compiled to native code.

# Architecture

- Common code shared between different platforms.
  - "shared/src/commonMain"
- Targets (platforms) are designated in Gradle, and the Common code is then compiled to those platforms.
- Additional sources (platform-specific) for code that has to be run in the target platform explicitly.
  - "shared/src/[platform]Main"
- During compilation, Kotlin selects sources labeled with [platform] for the target platform, i.e. "commonMain" and "jsMain".
- Intermediate source set, compile to some but not all platforms. I.e. "all Apple".
  - "shared/src/[group]Main".

# Releases

- First release: 2017
- Alpha release: April 15th 2020
- Stable releases: November 1st 2023
- Latest update: May 7th 2024

# IDEs/Tools

## IDEs:

- Recommended: JetBrains Fleet (WIP)
- Stable: Android Studio (limited support for Kotlin Multiplatform)
- Minimal support: Others, IntelliJ, etc.
- If for iOS: Xcode, obviously.

## Resources:

- KMP Library Wizard (<https://terrakok.github.io/kmp-web-wizard/>)
- KMP Mobile plugin for Android Studio.

# Community / Documentation

- Community is small, but growing
- Documentation (examples, a few tutorials) available on official site:
  - <https://www.jetbrains.com/help/kotlin-multiplatform-dev/get-started.html>