

Table of Contents

Advanced topics 1

Advanced topics

As this paper is primarily an introduction to the Kotlin language itself, it does not include certain advanced use-cases.

The following is a list of incredibly useful language features and links to their references.

- The standard library
 - Kotlin has an immense standard library, which is best learned through experience. Often, searching for an implementation of a complex utility function will reveal that it already exists in the `stdlib`.
- Reflection
 - Kotlin has its own advanced reflection system — however, its complexities and gotchas are too great to discuss in an introductory paper.
- Collections
 - Kotlin has an extensive amount of collection types and a large support library providing operations like `map`, `filter`, and `fold` as extension functions.
- **Coroutines** (`suspend` functions)
 - Enables safe, structured, asynchronous programming without much need for synchronization
- **Kotlin/Native**
 - Compiles Kotlin to native binaries — this is *very* experimental.
- **Multiplatform projects** (Gradle only)
 - Builds projects for multiple platforms (JVM, JS, Native) using a single common project