K

PROGRAMMING CHECKLIST

Kotlin Syllabus Checklist

1 Chapters

Philip Osztromok

Generated with Claude

Contents

Chapter 1 Kotlin Checklist 2

Kotlin Syllabus Checklist

What you can learn here, and where

✅ Your Kotlin Checklist

This page lists what the Kotlin courses on this site teach, as things you can *do*. Tick an item when you can do it, use the "Where to learn it" column to find the course and chapter, and use the "Prove it" column for a short task to try yourself. Print the PDF to keep a copy that you can tick with a pen.

> **HOW TO USE THIS PAGE**
>
> The sections follow the order of the courses: **Kotlin Fundamentals** (beginner), **Kotlin Intermediate**, and a short section on reading Kotlin in an **Android** app. The "Prove it" tasks are short prompts for you to try, not graded exercises. Ticks on the website are not saved when you leave the page, so the printable PDF is the way to keep your progress.

1. Kotlin Fundamentals (beginner)

Eight chapters: the basics of the language.

	I CAN…	WHERE TO LEARN IT	PROVE IT
☐	install a Kotlin setup, run a program, and use `val`, `var` and string templates	**Kotlin Fundamentals**, Chapter 1 (Getting Started)	Declare `val name = "Ada"` and `var age = 30`, then print `Ada is 30` using a string template.
☐	write functions with default and named arguments, and pass lambdas	**Kotlin Fundamentals**, Chapter 2 (Functions & Lambdas)	Write `greet(name, greeting = "Hello")` and call it with the greeting named. Then pass the lambda `{ it * 2 }` to a function that applies it to 5.
☐	use nullable types safely with `?.`, `?:` and smart casts	**Kotlin Fundamentals**, Chapter 3 (Null Safety)	Write a function that takes a `String?` and returns its length, or 0 when it is null.
☐	write classes and data classes	**Kotlin Fundamentals**, Chapter 4 (Classes & Data Classes)	Declare `data class Point(val x: Int, val y: Int)`, print one, and make a changed copy with `copy(y = 5)`.

	I CAN…	WHERE TO LEARN IT	PROVE IT
☐	use inheritance, interfaces and sealed classes	**Kotlin Fundamentals**, Chapter 5 (Inheritance & Interfaces)	Declare a sealed class `Shape` with a `Circle` and a `Square`, and a `when` that returns a description of each.
☐	work with lists, sets and maps using `map`, `filter` and `reduce`	**Kotlin Fundamentals**, Chapter 6 (Collections & Functional Operators)	From `listOf(1, 2, 3, 4, 5, 6)` get the squares of the even numbers.
☐	use `when`, ranges, loops and destructuring	**Kotlin Fundamentals**, Chapter 7 (Control Flow & Destructuring)	Write a `when` that turns a score into a grade, and destructure a `Pair` into two variables.
☐	write extension functions and `object` declarations	**Kotlin Fundamentals**, Chapter 8 (Extension Functions & Object Declarations)	Write `fun String.shout()` that returns the text in capitals with `!` added, and an `object Counter` with an `increment()` function.

2. Kotlin Intermediate

Eight chapters: coroutines, generics, delegation, DSLs and Multiplatform.

	I CAN…	WHERE TO LEARN IT	PROVE IT
☐	launch coroutines and use suspend functions	**Kotlin Intermediate**, Chapter 1 (Coroutines Basics)	Inside `runBlocking`, launch two coroutines that each print after a `delay`, and show that they overlap.
☐	use `Flow`, cancellation and structured concurrency	**Kotlin Intermediate**, Chapter 2 (Coroutines Advanced)	Write a `Flow` that emits 1 to 3 with a delay, collect it, and cancel the collecting job after the first value.
☐	write generic classes and functions, and explain `in` and `out`	**Kotlin Intermediate**, Chapter 3 (Generics & Variance)	Write `fun <T> List<T>.secondOrNull(): T?` and call it on a list of `Int` and a list of `String`.
☐	use property delegation (`by lazy`, observable, custom)	**Kotlin Intermediate**, Chapter 4 (Delegation)	Use `by lazy` so a message prints only the first time the property is read.
☐	write a small type-safe DSL with lambdas with receivers	**Kotlin Intermediate**, Chapter 5 (DSL Building)	Write a function that takes a `StringBuilder.() -> Unit` lambda and use it to build two lines of text.

	I CAN…	WHERE TO LEARN IT	PROVE IT
☐	use reflection and write a custom annotation	**Kotlin Intermediate**, Chapter 6 (Reflection & Annotations)	Declare an annotation, put it on a class, and print the class's simple name with `MyClass::class.simpleName`. (Listing properties needs the `kotlin-reflect` library.)
☐	use value classes, operator overloading and type aliases	**Kotlin Intermediate**, Chapter 7 (Advanced Patterns)	Give `Money(val cents: Int)` an `operator fun plus`, and add `typealias UserName = String`.
☐	explain how Kotlin Multiplatform shares code with `expect` / `actual`	**Kotlin Intermediate**, Chapter 8 (Kotlin Multiplatform Basics)	Write an `expect fun platformName(): String` and the matching `actual` for one target. (Needs a Multiplatform project.)

3. Kotlin in an Android app

One item: the Android course uses Kotlin and Compose but is about building the app, so it is listed here as practice rather than as a language lesson.

	I CAN…	WHERE TO LEARN IT	PROVE IT
☐	read the Kotlin and Compose code of an Android app	**Personal Catalogue: Android (Kotlin & Compose)** (the language is used, not taught)	Create an Empty Activity project, open `MainActivity.kt`, and name the function that calls `setContent` and one function marked `@Composable`.

4. Not yet on this site

Some things are not taught yet, so this checklist does not claim them. This list is based on the chapter topic lines, so some of these may appear inside a chapter.

- Exceptions and error handling (`try`, `Result`), and the standard library's string and file functions.
- Scope functions (`let`, `apply`, `run`) and inline functions.
- Testing in Kotlin, Gradle in practice, and working with Java code.
- Serialization, and Compose as a topic in its own right (it appears only inside the Android course).
- There is no Kotlin Advanced course yet.

ABOUT ACCURACY

The checklist items come from the chapters' banner and topic lines, and nothing was checked against the official Kotlin documentation. **The "Prove it" tasks have not been run**: there was no Kotlin compiler available when this page was written, so they were written by hand and may need small fixes. If a task does not work for you, check the chapter and the current Kotlin documentation.

KEEP GOING

Revisit a chapter whenever an item on this list is still unticked. The chapters have worked examples you can copy and change.