ATOMIC KOTLIN

Bruce Eckel

Svetlana Isakova

Atomic Kotlin

Bruce Eckel and Svetlana Isakova

This book is for sale at http://leanpub.com/AtomicKotlin

This version was published on 2020-12-21

* * * * *

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

* * * * *

ISBN for EPUB version: 978-0-9818725-4-4

ISBN for MOBI version: 978-0-9818725-4-4

Table of Contents

Copyright

Atomic Kotlin

By Bruce Eckel, President, MindView, LLC, and Svetlana Isakova, JetBrains sro.

eBook ISBN 978-0-9818725-4-4

Print Book ISBN 978-0-9818725-5-1

The eBook ISBN covers the Stepik and Leanpub eBook distributions, both available through *AtomicKotlin.com*.

Please purchase this book through www.AtomicKotlin.com, to support its continued maintenance and updates.

Created in Crested Butte, Colorado, USA, and Munich, Germany.

Text printed in the United States

Ebook: Version 1.0, December 2020

First printing January 2021

Cover design by Daniel Will-Harris, www.Will-Harris.com

Visit us at www.AtomicKotlin.com.

Source Code

All the source code for this book is available as copyrighted freeware, distributed via Github. To ensure you have the most current version, this is the official code distribution site. You may use this code in classroom and other educational situations as long as you cite this book as the source.

The primary goal of this copyright is to ensure that the source of the code is properly cited, and to prevent you from republishing the code without permission. (As long as this book is cited, using examples from the book in most media is generally not a problem.)

In each source-code file you find a reference to the following copyright notice:

```
// Copyright.txt
This computer source code is Copyright ©2021 MindView LLC.
All Rights Reserved.

Permission to use, copy, modify, and distribute this
computer source code (Source Code) and its documentation
without fee and without a written agreement for the
purposes set forth below is hereby granted, provided that
```

```
please submit a correction at:
https://github.com/BruceEckel/AtomicKotlinExamples/issues
```

Section I: Programming Basics

There was something amazingly enticing about programming—**Vint Cerf**

This section is for readers who are learning to program. If you're an experienced programmer, skip forward to Summary 1 and Summary 2.

Introduction

> This book is for dedicated novices and experienced programmers.

You're a novice if you don't have prior programming knowledge, but "dedicated" because we give you just enough to figure it out on your own. When you're finished, you'll have a solid foundation in programming and in Kotlin.

If you're an experienced programmer, skip forward to Summary 1 and Summary 2, then proceed from there.

The "Atomic" part of the book title refers to atoms as the smallest indivisible units. In this book, we try to introduce only one concept per chapter, so the chapters cannot be further subdivided—thus we call them *atoms*.

Concepts

All programming languages consist of features. You apply these features to produce results. Kotlin is powerful—not only does it have a rich set of features, but you can usually express those features in numerous ways.

If everything is dumped on you too quickly, you might come away thinking Kotlin is "too complicated."

This book attempts to prevent overwhelm. We teach you the language carefully and deliberately, using the following principles:

1. **Baby steps and small wins**. We cast off the tyranny of the chapter. Instead, we present each small step as an *atomic concept* or simply *atom*, which looks like a tiny chapter. We try to present only one new concept per atom. A typical atom contains one or more small, runnable pieces of code and the output it produces.
2. **No forward references**. As much as possible, we avoid saying, "These features are explained in a later atom."
3. **No references to other programming languages**. We do so only when necessary. An analogy to a feature in a language you don't understand isn't