Wrapping Up

Kotlin extends Java collections and at the same time brings along improved compile-time safety via read-only views. Use the read-only views when working with functional style, writing concurrent code, or when programming asynchronously.

Kotlin's `Pair` and `Triple` are useful to create finite small-sized collections. For a larger fixed-sized collection, reach for the `Array` class. However, for flexibility to grow the collection, choose between `List` or `Set`. When working with collections, you have to choose methods, like `listOf()` to get a read-only view of the JDK collection, or methods like `mutableListOf()`, to get a read-write view. To work with associative maps, use the Kotlin view interfaces to interact with Java's Map implementations.

We've been exploring the imperative way to use iterators in this chapter. A functional style, using internal iterators, is a very powerful way to iterate, but there are performance consequences to it. We'll explore that later in the book, along with when to use iterators vs. sequences for lazy evaluation.

In the next chapter, you'll learn how Kotlin takes compile-time type safety to a whole new level.