Wrapping Up

Using the imperative style to iterate is a lot more fluent in Kotlin than in many mainstream languages. With a higher level of abstraction, through specialized classes for range of values, Kotlin makes it easier to do forward iteration, reverse iteration, skip values in the range, and so on. The same approach may be used for iterating over a collection of values as well. And, to process different values, of different types, the argument-matching syntax—the `when` construct—removes a lot of noisy conditional code.

We've covered a lot of ground so far in this part about fundamental everyday programming tasks.

In the next lesson we'll take a look at the collections that are available when programming with Kotlin.