# Sensible Warnings

## Getting ahead of errors

Even if a piece of code is valid syntactically, some potential problems may be lurking. Getting an early warning, during compilation time, can help us to proactively fix such possible issues. The Kotlin compiler looks out for quite a few potential issues in code.

For example, if a parameter that's received in a function or a method isn't used, then the compiler will give a warning. In the following script, the parameter passed to `compute()` isn't used.

```
fun compute(n: Int) = 0

println(compute(4))
```

unused.kts

When you run this script, in addition to displaying the result, Kotlin will also report any warnings for unused parameters:

```
0
unused.kts:1:13: warning: parameter 'n' is never used
fun compute(n: Int) = 0
            ^
```

It's a good software development practice to treat warnings as errors—an agile practice emphasized in Practices of an Agile Developer. Kotlin makes that easy with the -Werror option. To use this option, place it on the command line when you compile the code or run it as a script, like so:

```
kotlinc-jvm -Werror -script unused.kts
```

This option will fail the build or execution. Unlike the previous run without that option, there will be no output when the script is run; instead an error is reported:

```
error: warnings found and -Werror specified
unused.kts:1:13: warning: parameter 'n' is never used
fun compute(n: Int) = 0
            ^
```

## Warnings in the `main()` function

The Kotlin compiler is sensible when giving warnings. For example, it's not uncommon for programs to ignore command-line arguments. Forcing us to use parameters given to `main()` is considered draconian, so Kotlin doesn't complain about unused parameters for `main()`, as we see in the next example. But if you have an unused parameter in `main()` within a script (a `.kts` file instead of a `.kt` file), then Kotlin will give you a warning—it decides based on the context.

```
fun compute(n: Int) = 0

fun main(args: Array<String>) = println(compute(4))
```

UnusedInMain.kt

When you compile the code using `kotlinc-jvm` and then run it using either java or kotlin, you'll get the following output. The warning is from `kotlinc` and the output is from the execution of the generated jar file:

```
0

UnusedInMain.kt:1:13: warning: parameter 'n' is never used
fun compute(n: Int) = 0
            ^
```

Starting from `Kotlin 1.3`, you may leave out the parameters to `main()` if you don't need them.

We saw how Kotlin tries to make a preemptive strike against potential errors. Along those lines, the language wants you to be decisive about immutability.

Let’s explore that choice in the next lesson.