Prime Factors in Scala

Readme (readme) | Test Suite (../prime-factors)

Prime Factors

Compute the prime factors of a given natural number.

A prime number is only evenly divisible by itself and 1.

Note that 1 is not a prime number.

Example

What are the prime factors of 60?

- Our first divisor is 2. 2 goes into 60, leaving 30.
- 2 goes into 30, leaving 15.
  - 2 doesn't go cleanly into 15. So let's move on to our next divisor, 3.
- 3 goes cleanly into 15, leaving 5.
  - 3 does not go cleanly into 5. The next possible factor is 4.
  - 4 does not go cleanly into 5. The next possible factor is 5.
- 5 does go cleanly into 5.
- We're left only with 1, so now, we're done.

Our successful divisors in that computation represent the list of prime factors of 60: 2, 2, 3, and 5.

You can check this yourself:

- 2 * 2 * 3 * 5
- = 4 * 15
- = 60
- Success!

The Scala exercises assume an SBT project scheme. The exercise solution source should be placed within the exercise directory/src/main/scala. The exercise unit tests can be found within the exercise directory/src/test/scala.

To run the tests simply run the command `sbt test` in the exercise directory.

For more detailed info about the Scala track see the help page (http://help.exercism.io/getting-started-with-scala.html).

Beta

Source

The Prime Factors Kata by Uncle Bob view source (http://butunclebob.com/ArticleS.UncleBob.ThePrimeFactorsKata)

exercism.io (/)

About (/about) - Donate (/donate)

GitHub (https://github.com/exercism/exercism.io) Twitter (https://twitter.com/exercism_io)

Newsletter (https://tinyletter.com/exercism)