Accumulate in Scala

Readme (readme) Test Suite (../accumulate)

Accumulate

Implement the `accumulate` operation, which, given a collection and an operation to perform on each element of the collection, returns a new collection containing the result of applying that operation to each element of the input collection.

For example, given the collection of numbers:

- 1, 2, 3, 4, 5

And the operation:

- square a number

Your code should be able to produce the collection of squares:

- 1, 4, 9, 16, 25

Check out the test suite to see the expected function signature.

Restrictions

Keep your hands off that collect/map/fmap/whatchamacallit functionality provided by your standard library! Solve this one yourself using other basic tools instead.

Elixir specific: it's perfectly fine to use `Enum.reduce` or `Enumerable.reduce`.

Lisp specific: it's perfectly fine to use `MAPCAR` or the equivalent, as this is idiomatic Lisp, not a library function.

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

Conversation with James Edward Gray II view source (https://twitter.com/jeg2)

exercism.io
(/)

About (/about) - Donate (/donate)

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

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

SPONSORS

(https://bugsnag.com/blog/bugsnag-loves-open-source)

(http://www.rackspace.com/)

(http://www.shopify.com/)