

# Package: decorators (via r-universe)

September 13, 2024

**Type** Package

**Title** Extend the Behaviour of a Function without Explicitly Modifying it

**Version** 0.3.0

**Date** 2022-06-23

**Maintainer** Harel Lustiger <tidylab@gmail.com>

**Description** A decorator is a function that receives a function, extends its behaviour, and returned the altered function. Any caller that uses the decorated function uses the same interface as it were the original, undecorated function. Decorators serve two primary uses: (1) Enhancing the response of a function as it sends data to a second component; (2) Supporting multiple optional behaviours. An example of the first use is a timer decorator that runs a function, outputs its execution time on the console, and returns the original function's result. An example of the second use is input type validation decorator that during running time tests whether the caller has passed input arguments of a particular class. Decorators can reduce execution time, say by memoization, or reduce bugs by adding defensive programming routines.

**License** MIT + file LICENSE

**URL** <https://tidylab.github.io/decorators/>,  
<https://github.com/tidylab/decorators>

**BugReports** <https://github.com/tidylab/decorators/issues>

**Depends** R (>= 3.5)

**Imports** purrr, methods

**Suggests** testthat

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-GB

2

**Roxygen** list(markdown = TRUE, r6 = TRUE)

**RoxygenNote** 7.2.0

**Repository** <https://tidylab.r-universe.dev>

**RemoteUrl** <https://github.com/tidylab/decorators>

**RemoteRef** HEAD

**RemoteSha** c74e1ad6488db1ef7f8ac78da9f7324a11a4cb99