

# Package ‘listcomp’

May 8, 2026

**Title** List Comprehensions

**Version** 0.4.1

**Description** An implementation of list comprehensions as purely syntactic sugar with a minor runtime overhead. It constructs nested for-loops and executes the byte-compiled loops to collect the results.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** testthat

**Imports** rlang, compiler

**URL** <https://github.com/dirkschumacher/listcomp>

**BugReports** <https://github.com/dirkschumacher/listcomp/issues>

**NeedsCompilation** no

**Author** Dirk Schumacher [aut, cre, cph]

**Maintainer** Dirk Schumacher <mail@dirk-schumacher.net>

**Repository** CRAN

**Date/Publication** 2022-01-31 16:10:02 UTC

## Contents

gen\_list . . . . . 2

**Index** 3

---

 gen\_list

*List comprehensions*


---

### Description

Create lists of elements using an expressive syntax. Internally nested for-loops are created and compiled that generate the list.

### Usage

```
gen_list(element_expr, ..., .compile = TRUE, .env = parent.frame())
```

### Arguments

element_expr	an expression that will be collected
...	either a logical expression that returns a length 1 result. A named list of equal length sequences that are iterated over in parallel or a named parameter with an iterable sequence.
.compile	compile the resulting for loop to bytecode before eval
.env	the parent environment in which all the elements are being evaluated.

### Details

For parallel iterations all elements in the list need to be of equal length. This is not checked at runtime at the moment.

### Value

A list of all generated values. The element-type is determined by the parameter element\_expr.

### Examples

```
gen_list(c(x, y), x = 1:10, y = 1:10, x + y == 10, x < y)
z <- 10
gen_list(c(x, y), x = 1:10, y = 1:10, x + y == !!z, x < y)

# it is also possible to iterate in parallel by passing a list of
# sequences
gen_list(c(x, y), list(x = 1:10, y = 1:10), (x + y) %in% c(4, 6))
```

# Index

`gen_list`, 2