

Package ‘partialised’

July 23, 2025

Type Package

Title Partialised Functions

Version 0.1.1

Description Provides a 'partialised' class that extends the partialising function of 'purrr' by making it easier to change the arguments. This is similar to the function-like object in 'Julia' (<https://docs.julialang.org/en/v1/manual/methods/#Function-like-objects>).

License MIT + file LICENSE

Encoding UTF-8

Imports pillar, purrr, rlang, vctrs

RoxygenNote 7.2.1

URL <https://github.com/UchidaMizuki/partialised>

BugReports <https://github.com/UchidaMizuki/partialised/issues>

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Mizuki Uchida [aut, cre]

Maintainer Mizuki Uchida <uchidamizuki@vivaldi.net>

Repository CRAN

Date/Publication 2022-11-04 11:20:02 UTC

Contents

arguments	2
extract	2
new_partialised	3

Index	5
--------------	----------

arguments	<i>Argument lists for partialised functions</i>
-----------	---

Description

Argument lists for partialised functions

Usage

```
arguments(x)
```

```
arguments(x) <- value
```

Arguments

x	Partialised function.
value	A list of arguments.

Value

arguments() returns a list of arguments.

extract	<i>Extract or replace arguments for partialised functions</i>
---------	---

Description

Extract or replace arguments for partialised functions

Usage

```
## S3 method for class 'partialised'
x[i, ...]
```

```
## S3 replacement method for class 'partialised'
x[i] <- value
```

```
## S3 method for class 'partialised'
x[[i, ...]]
```

```
## S3 replacement method for class 'partialised'
x[[i]] <- value
```

```
## S3 method for class 'partialised'
x$i
```

```
## S3 replacement method for class 'partialised'  
x$i <- value
```

Arguments

x	Partialised function.
i	Indices specifying arguments to extract or replace.
...	Additional arguments.
value	An object, the new value of the argument.

Value

[, [[and \$ return arguments.

new_partialised	<i>Create partialised functions</i>
-----------------	-------------------------------------

Description

Create partialised functions

Usage

```
new_partialised(f, args = list(), ..., class = character())
```

Arguments

f	A function.
args	A list of default arguments.
...	Additional arguments for attributes.
class	Name of subclass.

Value

A partialised function.

See Also

[purrr::partial\(\)](#)

Examples

```
dist <- function(x, y) {
  sqrt(x ^ 2 + y ^ 2)
}
pdist <- new_partialised(dist,
                          list(x = 3))
pdist(y = 4)
```

Index

`[.partialised(extract)`, [2](#)
`[<-.partialised(extract)`, [2](#)
`[[.partialised(extract)`, [2](#)
`[[<-.partialised(extract)`, [2](#)
`$.partialised(extract)`, [2](#)
`$<-.partialised(extract)`, [2](#)

`arguments`, [2](#)
`arguments<- (arguments)`, [2](#)

`extract`, [2](#)

`new_partialised`, [3](#)

`purrr::partial()`, [3](#)