

Package ‘arpr’

May 7, 2026

Type Package

Title Advanced R Pipes

Version 0.1.2

Description Provides convenience functions for programming with 'magrittr' pipes. Conditional pipes, a string prefixer and a function to pipe the given object into a specific argument given by character name are currently supported. It is named after the dadaist Hans Arp, a friend of Rene Magritte.

License GPL (>= 3)

URL <https://github.com/statnmap/arpr>

BugReports <https://github.com/statnmap/arpr/issues>

Imports magrittr, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Jirka Lewandowski [aut],
Sébastien Rochette [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-1565-9313>>)

Maintainer Sébastien Rochette <sebastien@thinkr.fr>

Repository CRAN

Date/Publication 2021-08-02 15:50:05 UTC

Contents

browse_r	2
const	2
iff	3
pipe_into	4

prefix	5
prefix_path	5
remove_names	6

Index	7
--------------	----------

browse_r	<i>browser() in a magrittr pipe</i>
----------	-------------------------------------

Description

browser() in a magrittr pipe

Usage

```
browse_r(x, ...)
```

Arguments

x	input
...	passed on to browser()

Value

Used for side effect. Open a browser inside the pipe workflow.

const	<i>Create a constant function</i>
-------	-----------------------------------

Description

Create a constant function

Usage

```
const(val = NULL)
```

Arguments

val	return value of constant function (defaults to NULL)
-----	--

Value

A function always returning val accepting arbitrary arguments (dots)

`iff`*Apply a function depending on test output*

Description

`iff` returns output of the function if and only if test is TRUE. `iffn` returns output of the function if and only if test is FALSE. They return the original value otherwise. `iffelse` returns output of the first function if test is TRUE, output of the second function otherwise.

Usage

```
iff(obj, test, fun, ...)
```

```
iffn(obj, test, fun, ...)
```

```
iffelse(obj, test, true_fun, false_fun, ...)
```

Arguments

<code>obj</code>	object to apply test and fun to
<code>test</code>	logical or function to apply to test
<code>fun</code>	function to apply
<code>...</code>	passed on to test
<code>true_fun</code>	function to apply when test is true
<code>false_fun</code>	function to apply when test is false

Value

Output of function `fun` applied to the original value or the original value, depending on the test.

Examples

```
x <- 1
x %>%
  iff(is.na, const(0))
x <- NA
x %>%
  iff(is.na, const(0))

x <- 1
x %>%
  iff(x <= 0, function(x) { x - 2 })
x <- -1
x %>%
  iff(x <= 0, function(x) { x - 2 })

x <- NA
```

```
x %>%  
  iffn(is.na, exp)  
x <- 10  
x %>%  
  iffn(is.na, exp)
```

pipe_into

Pipe into specific formal argument

Description

This rotates the order of the arguments such that the one named in `param_name` comes first and then calls the function.

Usage

```
pipe_into(x, param_name, fun, ...)
```

Arguments

<code>x</code>	value to be piped into fun
<code>param_name</code>	name of the argument that x should be assigned to
<code>fun</code>	function
<code>...</code>	further arguments for fun

Value

Output of fun.

Examples

```
require(magrittr)  
5L %>%  
  pipe_into("digits", format, 2.731234567)
```

prefix	<i>Prefix a string of text</i>
--------	--------------------------------

Description

Convenience function to use with magrittr wraps `paste0()`, hence vectorised as `paste0()`

Usage

```
prefix(text, ...)
```

Arguments

text	goes to the end, rest
...	goes to the front.

Value

Character. Character chain with the prefix added.

Examples

```
require(magrittr)
"xyz" %>%
  prefix("abc")
```

prefix_path	<i>Prefix a path</i>
-------------	----------------------

Description

file.path with arguments reversed

Usage

```
prefix_path(path, prefix, ...)
```

Arguments

path	path to be prefixed
prefix	path to be appended before
...	passed on to file.path

Value

```
file.path(prefix, path, ...)
```

remove_names	<i>Remove names of an object</i>
--------------	----------------------------------

Description

Remove names of an object

Usage

```
remove_names(x)
```

Arguments

x object to unname

Value

x without names.

Index

`browse_r`, 2

`const`, 2

`iff`, 3

`iffelse(iff)`, 3

`iffn(iff)`, 3

`paste0()`, 5

`pipe_into`, 4

`prefix`, 5

`prefix_path`, 5

`remove_names`, 6