

Package ‘backpipe’

October 12, 2022

Type Package

Title Backward Pipe (Right-to-Left) Operator

Version 0.2.3

Date 2018-06-25

Description Provides a backward-pipe operator for 'magrittr' (`%<%`) or 'pipeR' (`%<<%`) that allows for a performing operations from right-to-left. This allows writing more legible code where right-to-left ordering is natural. This is common with hierarchies and nested structures such as trees, directories or markup languages (e.g. HTML and XML). The package also includes a R-Studio add-in that can be bound to a keyboard shortcut.

Depends R (\geq 3.1.0)

Suggests magrittr (\geq 1.5), pipeR (\geq 0.6.0.6), R6 (\geq 2.1.1), testthat (\geq 1.0.0), shiny (\geq 0.12.1), knitr (\geq 1.11), rstudioapi (\geq 0.7), rmarkdown

License GPL-2 | file LICENSE

URL <https://github.com/decisionpatterns/backpipe>

BugReports <https://github.com/decisionpatterns/backpipe/issues>

RoxygenNote 6.0.1.9000

Encoding UTF-8

VignetteBuilder knitr

Repository CRAN

NeedsCompilation no

Author Christopher Brown [aut, cre],
Decision Patterns [cph]

Maintainer Christopher Brown <chris.brown@decisionpatterns.com>

Date/Publication 2018-06-26 04:38:34 UTC

R topics documented:

backpipe	2
insert_backpipe	3
%<%	3

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

backpipe	<i>backpipe</i>
----------	-----------------

Description

Creates backpiping operators

Usage

```
backpipe(pipe, backpipe = gsub(">", "<", pipe))
```

Arguments

pipe	character; string representing the existing pipe operator
backpipe	character; string representing the desired backpipe operator

Details

Only pipe is necessary. Arbitrary mixing of forward and reverse are not allowed.

References

<https://github.com/smbache/magrittr/issues/26>
<http://stackoverflow.com/questions/31305342/is-right-to-left-operator-associativity-in-r-possible>

See Also

[%>%](#)
[%>>%](#)

Examples

```
backpipe('%>%')
backpipe('%>>%', '%<<%')
```

insert_backpipe	<i>Insert backpipe</i>
-----------------	------------------------

Description

Call this function to insert `<code>%<t;-%</code>` at the cursor position.

Usage

```
insert_backpipe()
```

Details

Use **Tools > Modify Keyboard Shortcuts** to bind a `->.</code>; ->.</code>`

<code>%<%</code></code>	<i>backpipes: %<%</code>, %<<%</code></i>
----------------------------------	--

Description

Provides back-piping operators `%<%</code> (magrittr) or %<<%</code> (pipeR) allowing a reverses (right-to-left) order of arguments`

Usage

```
lhs %<% rhs
```

```
lhs %<<% rhs
```

Arguments

lhs argument on the lhs of the backpipe

rhs argument on the rhs of the backpipe

Details

- `%<%</code> works with magrittr. %<<%</code> works with piper. They are both based on backpipe.`
- Other magrittr operators and complex expressions are not supported yet.
- It is not possible to mix forward and backward piping in the same expression because of likely ambiguous results.

References

<https://github.com/smbache/magrittr/issues/26>

<http://stackoverflow.com/questions/31305342/is-right-to-left-operator-associativity-in-r-possible>

See Also

[%>%](#)

[%>>%](#)

Examples

```
## Not run:
require(magrittr)
letters %>% paste0( 1:26 ) # forward pipe
paste0( 1:26 ) %<% letters # backward pipe

mean %<% range( na.rm = TRUE ) %<% c(1:5, NA)

## End(Not run)

## Not run:
require(pipeR)
letters %>>% paste0( 1:26 ) # forward pipe
paste0( 1:26 ) %<<% letters # backward pipe

mean %<<% range( na.rm = TRUE ) %<<% c(1:5, NA)

## End(Not run)

## Not run:
require(shiny)
div( class="outer-outer" ) %<%
  div( class="outer" ) %<%
    div( class="inner" ) %<%
      h1( "content", role="heading" )

## End(Not run)
```

Index

`%<<% (%<%)`, 3

`%<%`, 3

`%>%`, 2, 4

`%>>%`, 2, 4

`backpipe`, 2, 3

`insert_backpipe`, 3