

# Package: stringfix (via r-universe)

August 25, 2024

**Type** Package

**Title** Strings manipulations in a piping way of coding

**Version** 0.1.0

**Author** Guillaume Pressiat

**Maintainer** Guillaume Pressiat <guillaume.pressiat@chu-brest.fr>

**Description** Some infix operators and other utility functions to make syntax more and more left to right. In a manner of coding, I just want to say `%>%` nothing.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** stringr, tibble

**URL** <https://guillaumepressiat.github.io/stringfix/>

**BugReports** <https://github.com/GuillaumePressiat/stringfix/issues>

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

**RemoteUrl** <https://github.com/GuillaumePressiat/stringfix>

**RemoteRef** HEAD

**RemoteSha** e4c5bf69f572503a4c53a9084edf872b6d53f83a

## Contents

run_transcoder	2
tolower_names	2
toupper_names	3
<code>%c%</code>	3
<code>%count%</code>	4
<code>%e%</code>	5
<code>%ee%</code>	5
<code>%g%</code>	6

<code>%gic%</code>	7
<code>% %</code>	7
<code>%lpad%</code>	8
<code>%out%</code>	9
<code>%pad%</code>	9
<code>%+%</code>	10
<code>%re%</code>	11
<code>%reall%</code>	11
<code>%rm%</code>	12
<code>%rmall%</code>	13
<code>%rpad%</code>	13
<code>%s%</code>	14

## Index 15

---

<code>run_transcoder</code>	<i>Run transcoder shiny app</i>
-----------------------------	---------------------------------

---

### Description

Run transcoder shiny app

### Usage

```
run_transcoder(
  launch.browser = getOption("shiny.launch.browser", interactive())
)
```

---

<code>tolower_names</code>	<i>Make names of a tibble tolower</i>
----------------------------	---------------------------------------

---

### Description

Make names of a tibble tolower

### Usage

```
tolower_names(d)
```

### Arguments

`d` tibble

### Value

A tibble

**Examples**

```
library(magrittr)
iris %>% tolower_names %>% head
```

---

toupper_names	<i>Make names of a tibble toupper</i>
---------------	---------------------------------------

---

**Description**

Make names of a tibble toupper

**Usage**

```
toupper_names(d)
```

**Arguments**

d                    tibble

**Value**

A tibble

**Examples**

```
library(magrittr)
iris %>% toupper_names %>% head
```

---

%c%	<i>Collapse a character vector to a character string</i>
-----	--

---

**Description**

Collapse a character vector to a character string

**Usage**

```
x %c% y
```

**Arguments**

x                    A character vector  
y                    Character string, collapse parameter of paste0

**Details**

This function is a pipe version of paste0 with collapse parameter to specify

**Value**

A character string

**Examples**

```
c('N', 'F', 'K', 'A') %c% ' '
'All I wanna do' %e% ' ' %c% ' '
```

---

%count%

*count pattern in strings the way of pipe operator*

---

**Description**

count pattern in strings the way of pipe operator

**Usage**

```
x %count% y
```

**Arguments**

x	Pattern to count in string
y	string where to count pattern

**Details**

This functions is a pipe version of stringr::count

**Value**

A vector of integer

**Examples**

```
fruit <- c("apple", "banana", "pear", "pineapple")
"a" %count% fruit
c("a", "b", "p", "p") %count% fruit
```

---

%e% *Explode a character string*

---

**Description**

Explode a character string

**Usage**

x %e% p

**Arguments**

x            A character string  
p            A pattern

**Details**

This function is a pipe version of `stringr::str_split` for a character string, with `simplify = TRUE`

**Value**

a character vector

**Examples**

```
'The little horse under the weather' %e% ' '  
'The little horse under the weather' %e% ''
```

---

%ee% *Explode a character vector*

---

**Description**

Explode a character vector

**Usage**

x %ee% p

**Arguments**

x            A character vector  
p            A pattern

**Details**

This function is a pipe version of `stringr::str_split` for a character vector, with `simplify = TRUE`

**Value**

a tibble

**Examples**

```
c('The little horse under the weather',
  "He who gave courage",
  "It was a little white horse",
  'All behind, all behind',
  "It was time of the little white horse",
  "All behind him before") %>% " |' |, "
```

---

`%g%` *grepl in the way of pipe operator (case sensitive)*

---

**Description**

`grepl` in the way of pipe operator (case sensitive)

**Usage**

```
x %g% y
```

**Arguments**

`x` string where to search pattern with `grepl`  
`y` pattern submitted to `grepl`

**Details**

This functions is a pipe version of `grepl`

**Value**

A boolean

**Examples**

```
'pig' %g% 'The pig is in the cornfield'
'Pig' %g% 'The pig is in the cornfield'
```

---

%gic% *grepl in the way of pipe operator (ignore case)*

---

**Description**

grepl in the way of pipe operator (ignore case)

**Usage**

x %gic% y

**Arguments**

x string where to search pattern with grepl  
y pattern submitted to grepl

**Details**

This functions is a pipe version of grepl with ignore.case = TRUE

**Value**

A boolean

**Examples**

```
'pig' %gic% 'The pig is in the cornfield'  
'PIG' %gic% 'The PiG is in the cornfield'
```

---

% % *paste ', ' in the way of pipe operator*

---

**Description**

paste ', ' in the way of pipe operator

**Usage**

a % % b  
a %, % b

**Arguments**

x A string (or coerced to)  
y A string (or coerced to)

**Details**

This functions is a pipe version of paste

**Value**

The concatenation of x and y strings

**Examples**

```
'Hello' % % 'world'
'Your pastas taste like' % % '%>%'
'Hello' %,% 'world...'
'Your pastas taste like ' %+% '%>%...' %,% 'or %>>%...'
```

---

%lpad%

*Left pad strings the way of pipe operator*

---

**Description**

Left pad strings the way of pipe operator

**Usage**

```
string %lpad% pad.width
```

**Arguments**

string	Character vector or NULL
pad.width	string or .digit where to indicate the pad and widht

**Details**

This functions is a pipe version of stringr::pad

**Value**

A character vector or NULL

**Examples**

```
5 %lpad% '0.5'
5 %lpad% .5
5 %lpad% '.5'
5 %lpad% '2.5'
'é' %lpad% 'é.5'
```



---

%out% *negation of in infix operator*

---

### Description

negation of in infix operator

### Usage

a %out% b

### Arguments

a                    Vector or NULL, values to be matched  
b                    Vector or NULL, values to be matched

### Details

This function comes from <https://github.com/pierucci/rmngb/tree/master/rmngb>

### Value

a boolean

### Examples

```
'coca' %out% c('cola')  
'coca' %out% c('coca', 'cola')
```

---

%pad% *Both side pad strings the way of pipe operator*

---

### Description

Both side pad strings the way of pipe operator

### Usage

string %pad% pad.width

### Arguments

string                Character vector or NULL  
pad.width            string or .digit where to indicate the pad and width

**Details**

This functions is a pipe version of `stringr::pad`

**Value**

A character vector or NULL

**Examples**

```
5 %pad% '.3'
5 %rpad% '0.3'
```

---

%+%

*paste0 in the way of pipe operator*

---

**Description**

`paste0` in the way of pipe operator

**Usage**

```
a %+% b
```

**Arguments**

x	A string (or coerced to)
y	A string (or coerced to)

**Details**

This functions is a pipe version of `paste0`

**Value**

The concatenation of x and y strings

**Examples**

```
'Hello ' %+% 'world'
'Your pastas taste like ' %+% '%>%'
'coco' %+% 'bolo'
```

---

%re% *Replace matched patterns in a string*

---

### Description

Replace matched patterns in a string

### Usage

x %re% y.z

### Arguments

x                    A character vector  
y.z                  A vector of two string specifying pattern for match and replace

### Details

This function is a pipe version of stringr::str\_replace

### Value

A character vector

### Examples

```
'totoro12_13' %re% c('[0-9]+', 'to')
```

---

%reall% *Replace matched patterns in a string*

---

### Description

Replace matched patterns in a string

### Usage

x %reall% y.z

### Arguments

x                    A character vector  
y.z                  A vector of two string specifying pattern for match and replace

**Details**

This function is a pipe version of `stringr::str_replace_all`

**Value**

A character vector

**Examples**

```
'totoro12_13' %real1% c('[0-9]+', 'to')
```

---

%rm%

*Remove matched patterns in a string*

---

**Description**

Remove matched patterns in a string

**Usage**

```
x %rm% y
```

**Arguments**

x	A character vector
y	Character string or pattern to remove

**Details**

This function is a pipe version of `stringr::str_remove`

**Value**

A character vector

**Examples**

```
'totoro' %rm% 'to'
```

---

`%rmall%` *Remove matched patterns in a string*

---

**Description**

Remove matched patterns in a string

**Usage**

`x %rmall% y`

**Arguments**

`x`                    A character vector  
`y`                    Character string or pattern to remove

**Details**

This function is a pipe version of `stringr::str_remove_all`

**Value**

A character vector

**Examples**

`'totoro' %rmall% 'to'`

---

`%rpad%` *Right pad strings the way of pipe operator*

---

**Description**

Right pad strings the way of pipe operator

**Usage**

`string %rpad% pad.width`

**Arguments**

`string`                Character vector or NULL  
`pad.width`            string or `.digit` where to indicate the pad and width

**Details**

This functions is a pipe version of stringr::pad

**Value**

A character vector or NULL

**Examples**

```
5 %rpad% '.2'
'é' %rpad% 'è.5'
```

---

%s%

*substring in the way of pipe operator*

---

**Description**

substring in the way of pipe operator

**Usage**

```
x %s% yz
```

**Arguments**

x	Vector or NULL, values to be substringed
y.z	A character or 0.digit to call substr

**Details**

This functions is a pipe version of substr

**Value**

A string or NULL

**Examples**

```
'NFKA008' %s% '1.4'
'NFKA008' %s% .4
'where is' % % ('the pig is in the cornfield' %s% '1.7') +% '?'
'NFKA008' %s% 5.7
```

# Index

`%,% (% %), 7`

`% %, 7`

`%+%, 10`

`%c%, 3`

`%count%, 4`

`%e%, 5`

`%ee%, 5`

`%g%, 6`

`%gic%, 7`

`%lpad%, 8`

`%out%, 9`

`%pad%, 9`

`%re%, 11`

`%reall%, 11`

`%rm%, 12`

`%rmall%, 13`

`%rpad%, 13`

`%s%, 14`

`run_transcoder, 2`

`tolower_names, 2`

`toupper_names, 3`