Package: emo (via r-universe)

June 20, 2024

Title Easily Insert 'Emoji'

Version 0.0.0.9000

Description Makes it easy to insert 'emoji' based on either their name
or a descriptive keyword.

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

LazyData true

Imports stringr, glue (>= 1.3.0), crayon, utils, magrittr, assertthat,
lubridate, rlang, purrr

RoxygenNote 6.0.1.9000

URL https://github.com/hadley/emo

BugReports https://github.com/hadley/emo/issues

Suggests testthat, dplyr

Repository https://thinkr-open.r-universe.dev

RemoteUrl https://github.com/hadley/emo

RemoteRef HEAD

RemoteSha 3f03b11491ce3d6fc5601e210927eff73bf8e350

Contents

`clock` *emoji version of time*

Description

emoji version of time

Usage

```
clock(time)
```

Arguments

`time`	a ‘POSIXct‘ object

Value

an emoji clock that is the closest to the given time

Examples

```
## Not run:
clock( Sys.time() )

## End(Not run)
```

day_in_synodic_cycle *Calculate the number of days in the synodic month*

Description

This uses the approximation described in the [lunar phase wikipedia page](https://en.wikipedia.org/wiki/Lunar_phase), i.e. the number of days since ‘1900/01/01‘ modulo the length of a synodic month (‘29.530588853‘ days)

Usage

```
day_in_synodic_cycle(date)
```

Arguments

date	a date

Examples

```
## Not run:
day_in_synodic_cycle( today() )

## End(Not run)
```

fisher_lst *Letter to emoji list*

Description

Letter to emoji list

Usage

```
fisher_lst
```

Format

List

flag *Flag emoji*

Description

Flag emoji

Usage

```
flag(pattern)
```

Arguments

`pattern`	pattern suitable for [stringr::str_detect] matched against the name of the flag

Value

If the pattern matches a single flag, the emoji (with classes "flag" and "emoji") is returned

Examples

```
## Not run:

flag( "^Fra" )
flag( "New Zealand" )

# name of all the flags
if( require(dplyr) ){
 emo::jis %>%
   filter( group == "Flags" ) %>%
   pull(name)
}


## End(Not run)
```

ji *Find a single emoji*

Description

Find a single emoji

Usage

```
ji(keyword)
```

Arguments

`keyword`	Either name or keyword. If more than one emoji has the specified keyword, will pick one at random.

Examples

```
emo::ji("banana")
emo::ji("monkey")
```

`jis` *full list of emojis*

Description

full list of emojis

Usage

```
jis
```

Format

tibble with columns - id: identifier - emoji: character representation of the emoji - name: name - group: group, e.g. "Smileys & People" - subgroup: sub group, e.g. "face-positive" - keywords: vector of keywords - runes: vector of unicode runes, i.e. hexadecimal representations prefixed with "U+" - nrunes: number of runes the emoji uses - apple ... windows: logical indicating if the given vendor supports the emoji

Source

[Unicode® Emoji Charts v5.0](http://unicode.org/emoji/charts/index.html)

`ji_completion` *emoji completion*

Description

emoji completion

Usage

```
ji_completion(token)
```

Arguments

`token`	start of an emoji alias

Examples

```
ji_completion( "key" )
```

ji_count *Count the number of emojis in a string*

Description

Vectorised over ‘string‘

Usage

```
ji_count(string)
```

Arguments

string	Input vector

Value

An integer vector

ji_detect *Detect the presence or absence of emojis in a string*

Description

Vectorised over ‘string‘

Usage

```
ji_detect(string)
```

Arguments

string	Input vector. Either a character vector, or something coercible to one

Value

A logical vector

See Also

[stringr::str_detect()]

ji_extract *Extract emojis from a string*

Description

vectorised over ‘string‘

Usage

```
ji_extract(string)

ji_extract_all(string, simplify = FALSE)
```

Arguments

`string`	Input vector.
`simplify`	see [stringr::str_extract_all()]

Value

A character vector

See Also

[stringr::str_extract()] and [stringr::str_extract_all()]

ji_find *List all emoji with a given keyword*

Description

Note that this is unlikely to print correctly on your R console, but it will work in (e.g.) the RStudio viewer.

Usage

```
ji_find(keyword)
```

Arguments

`keyword`	Emoji keyword

Examples

```
emo::ji_find("happy")
```

ji_fisher *Convert text to emoji (by letter)*

Description

Convert text to emoji (by letter)

Usage

```
ji_fisher(x)
```

Arguments

`x`	Character string. Text you'd like to be emojified.

Value

Character string of emoji.

Examples

```
ji_fisher("Carrie Fisher is a delight")
```

ji_glue *emoji glue*

Description

emoji glue

Usage

```
ji_glue(..., .envir = parent.frame())
```

Arguments

`...`	strings to format, where ':x:' is replaced by an emoji for "x", using [ji()] and ':y*:' is replaced by all emojis that match "y", using [ji_find()].
`.envir`	see [glue::glue()]

See Also

[glue::glue()] for how the strings are concatenated

Examples

```
## Not run:
  ji_glue("one :heart:")
  ji_glue("many :heart*:")

## End(Not run)
```

`ji_keyword` *emoji keywords*

Description

emoji keywords

Usage

```
ji_keyword
```

Format

An object of class `list` of length 5586.

`ji_locate` *Lodate the positio of emojis in a string*

Description

Vectorised over ‘string‘

Usage

```
ji_locate(string)

ji_locate_all(string)
```

Arguments

`string`	Input vector

Value

For ‘ji_locate‘ an integer matrix, for ‘ji_locate_all‘ a list of integer matrices

`ji_match` *Extract emojis from a string*

Description

Vectorized over ‘string‘

Usage

```
ji_match(string)

ji_match_all(string)
```

Arguments

`string`	Input vector

Value

see [stringr::str_match()]

See Also

[stringr::str_match]

`ji_name` *emoji names*

Description

emoji names

Usage

```
ji_name
```

Format

An object of class `character` of length 4239.

`ji_p` *Summarise your p-values with emoji*

Description

Summarise your p-values with emoji

Usage

```
ji_p(x)
```

Arguments

`x` A vector of p-values.

Examples

```
emo::ji_p(1)
emo::ji_p(0.1)
emo::ji_p(0.05)
emo::ji_p(0.01)
emo::ji_p(1e-6)

emo::ji_p(rbeta(50, 2, 5))
```

`ji_replace` *Replace emojis in a string*

Description

Vectorised over ‘string‘ and ‘replacement‘

Usage

```
ji_replace(string, replacement)

ji_replace_all(string, replacement)
```

Arguments

`string` Input vector

`replacement` A character vector of replacements. Should either be of length 1 or the same length as ‘string‘. See [stringr::str_replace()] for details

Value

A character vector

`ji_rx` *A regular expression to catch all emojis*

Description

A regular expression to catch all emojis

Usage

```
ji_rx
```

Format

character vector

`ji_subset` *Keep strings containing an emoji, or find positions*

Description

Keep strings containing an emoji, or find positions

Usage

```
ji_subset(string)

ji_which(string)
```

Arguments

`string`	input vector

Value

A character vector

See Also

[stringr::str_subset()]

keycap *Keycap emoji sequence*

Description

Keycap emoji sequence

Usage

```
keycap(x)
```

Arguments

x	character to emoji keycap

Value

a keycap version of ‘x‘

Examples

```
## Not run:
  keycap(3)
  keycap(10)
  keycap('#')

## End(Not run)
```

medal *medals*

Description

medals

Usage

```
medal(position)
```

Arguments

position	1, 2, 3, or ‘first‘, ‘second‘, ‘third‘, or ‘gold‘, ‘silver‘, ‘bronze‘

Examples

```
## Not run:
medal(gold)
medal(third)
medal(2)

## End(Not run)
```

`moon` *moon phase*

Description

moon phase

Usage

```
moon(date, day = day_in_synodic_cycle(date))
```

Arguments

`date`	a date
`day`	number of days since new moon If not supplied, ‘day‘ is calculated using the approximation of [day_in_synodic_cycle], i.e the number of days since a known new moon modulo ‘29.530588853‘ days

Value

a moon emoji

Examples

```
## Not run:
moon( today() )

## End(Not run)
```

`square` *geometric emoji*

Description

geometric emoji

Usage

```
square(size = c("small", "medium", "medium-small", "large"),
  color = c("white", "black"))

diamond(size = c("large", "small"), color = c("orange", "blue"))
```

Arguments

size	size
color	color

Details

For ‘square‘: - ‘size‘ should be one of ‘c("small", "medium", "medium-small", "large")‘ - ‘color‘ should be "white" or "black"

For ‘diamond‘ - ‘size‘ should be "large" or "small" - ‘color‘ should be "orange" or "blue"

Examples

```
## Not run:
square( "small", "black" )
square( "large", "white" )

diamond( "small", "orange")

## End(Not run)
```

Index