Package ‘sinx’

May 17, 2020

Version 0.0.13

Date 2020-05-16

Title Sino Xmen Said

Maintainer Peng Zhao <pzhao@pzhao.net>

Imports utils, cowsay, jsonlite, xaringan, pagedown, bookdownplus, rosr, crayon, multicolor, rmsfact, clipr

License GPL-3

Depends R (>= 3.1.0)

Suggests knitr, rmarkdown

Description Displays a pseudorandom message from a database of quotations. It works as an advanced version of the package 'fortunes', while 'sinx' supports multi-byte languages such as Chinese. The databases of 'sinx' can be given in markdown format, which is easier and more friendly than spread sheets for users.

URL https://github.com/pzhaonet/sinx

BugReports https://github.com/pzhaonet/sinx/issues

RoxygenNote 7.1.0

NeedsCompilation no

LazyData true

VignetteBuilder knitr

Author Peng Zhao [aut, cre]

Repository CRAN

Date/Publication 2020-05-17 07:40:02 UTC

R topics documented:

cscx *Create a Skeleton in a Clipboard for a new sinX*

Description

Create a Skeleton in a Clipboard for a new sinX

Usage

```
cscx()
```

Value

a skeleton text for contributing to the database

Examples

```
cscx()
```

md2df *create sinx data spread sheet*

Description

create sinx data spread sheet

Usage

```
md2df(mdfile)
```

Arguments

`mdfile`	filename of the original .md file

print.sinx *Print R sinx sayings.*

Description

Print R sinx sayings.

Usage

```
## S3 method for class 'sinx'
print(x, ...)
```

Arguments

x	an object of class "sinx", usually a single row from ‘sinxs.data‘.
...	potential further arguments passed to ‘grep‘.

Value

print.

read.sinxs *Read sayings from spread sheets.*

Description

Read sayings from spread sheets.

Usage

```
read.sinxs(file = NULL, sep = ",", lib = "sinxs")
```

Arguments

file	a character string giving a sinx sayings database in csv format (in UTF-8 encoding). By default all csv files in the data directory of the sinx package are used.
sep	seperator of the columns. See ’?read.table()’.
lib	library name of the sayings. - ’sinxs’: (default) from cosx.org - ’tangshi’ - ’songshi’ - ’yangsheng’ - ’chinese’ - ’english’ - ’jinyong’

Value

a data frame of sayings, each row contains: - quote: the quote, main part of the sayings, - author: the author of the quote, - context: the context in which it was quoted (if available, otherwise NA), - source : where it was quoted (if available, otherwise NA), - date: when it was quoted (if available, otherwise NA).

Examples

```
libs <- read.sinxs()

libs <- read.sinxs(lib = "jinyong")

libs <- read.sinxs(lib = c("tangshi", "songshi", "chinese", "yangsheng", "english",
    "jinyong"))

path_f <- system.file("fortunes/fortunes.csv", package = "fortunes")
path_s <- system.file("sinxs/sinxs.csv", package = "sinx")
libs <- sinx::read.sinxs(c(path_f, path_s), sep = c(";", ","))
```

---

secx *Create a book/slides from a sinx library*

Description

Create a book/slides from a sinx library

Usage

```
secx(
  lib = "tangshi",
  file = NULL,
  title = "A Sinx Book",
  author = "dapeng",
  section = "author",
  subsection = "context",
  style = c("pagedown", "bookdown", "xaringan"),
  bookdown_template = "demo_zh",
  to = "sinx_book",
  if_render = TRUE
)
```

Arguments

`lib`	library name of the sayings. See ‘?read.sinxs‘.
`file`	file name of source spread sheet. See ’?read.sinxs()’.
`title`	title of the book
`author`	author’s name
`section`	name of a sinx libary column, in which the characters are the sections in the book
`subsection`	name of a sinx libary column, in which the characters are the subsections in the book
`style`	’pagedown’ or ’bookdown’ or ’xaringan’

`bookdown_template`
template name. only valid when the style is ’bookdown’. See ’?bookdownplus’.

`to` the book directory.

`if_render` if render the book

Value

a book

Examples

```
## Not run:
secx()

secx(lib = "sinxs", title = "cosx", subsection = "date")

## End(Not run)
```

sinx *Sino Xmen’s sayings the R community.*

Description

Sino Xmen’s sayings the R community.

Usage

```
sinx(
  which = NULL,
  sinxs.data = NULL,
  fixed = TRUE,
  showMatches = FALSE,
  author = character(),
  ...
)
```

Arguments

`which` an integer specifying the row number of ‘sinxs.data‘. Alternatively ‘which“ can be a character and ‘grep“ is used to try to find a suitable row.

`sinxs.data` data frame containing a saying in each row. By default the data from the ’sinx’ package are used.

`fixed` logical passed to ‘grep‘ if ‘which“ is a character, indicating if it should work (if ‘TRUE‘, as by default) with a simple character string or (if ‘FALSE‘) with regular expressions.

`showMatches` if ‘which‘ is character, a logical indicating if ‘sinx()‘ should print all the row numbers of ‘sinxs.data‘ which match the ‘grep‘ search.

`author` a character string to match (via ‘grep‘) to the "authors" column of ‘sinxs.data‘.

`...` potential further arguments passed to ‘grep‘.

Value

an object of class "sinx" which is a row from a data frame of sayings (like those read in from read.sinxs).

Examples

```
sinx()

for (i in 1:4) print(sinx(i))

path_f <- system.file("fortunes/fortunes.csv", package = "fortunes")
path_s <- system.file("sinxs/sinxs.csv", package = "sinx")
ftns <- sinx::read.sinxs(c(path_f, path_s), sep = c(";", ","))
sinx::sinx(sinxs.data = ftns)

jinyong <- read.sinxs(lib = "jinyong")
sinx(sinxs.data = jinyong)

libs <- read.sinxs(lib = c("tangshi", "songshi", "chinese", "yangsheng", "english",
    "jinyong"))
sinx(sinxs.data = libs)
```

tanx *TAlked by SiNo Xmen's Pets*

Description

TAlked by SiNo Xmen's Pets

Usage

```
tanx(
  which = NULL,
  sinxs.data = NULL,
  what = "Hello world!",
  by = "random",
  type = "message",
  what_color = "steelblue3",
  by_color = NULL,
  length = 18,
  ...
)
```

Arguments

`which` An integer specifying the row number of sinxs.data. Alternatively which can be a character and grep is used to try to find a suitable row.

`sinxs.data` data frame containing a saying in each row. By default the data from the 'sinx' package are used.

`what` (character) What do you want to say? See details.

`by` (character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use `match.arg` internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".

`type` (character) One of message (default), warning, or string (returns string). If multiple colors are supplied to `what_color` or `by_color`, type cannot be warning. (This is a limitation of the multicolor packcage :/.)

`what_color` (character or crayon function) One or more `crayon`-suported text color(s) or `crayon style function` to color `what`. You might try `colors()` or `?rgb` for ideas. Use "rainbow" for c("red", "orange", "yellow", "green", "blue", "purple").

`by_color` (character or crayon function) One or more `crayon`-suported text color(s) or `crayon style function` to color `who`. Use "rainbow" for c("red", "orange", "yellow", "green", "blue", "purple").

`length` (integer) Length of longcat. Ignored if other animals used.

`...` Further args passed on to `sinx`

Details

You can put in any phrase you like, OR you can type in one of a few special phrases that do particular things. They are:

- catfact A random cat fact from https://catfact.ninja
- fortune A random quote from an R coder, from fortunes library
- time Print the current time
- rms Prints a random 'fact' about Richard Stallman from the `rmsfact` package. Best paired with `by = "rms"`.

Note that if you choose `by='hypnotoad'` the quote is forced to be, as you could imagine, 'All Glory to the HYPNO TOAD!'. For reference see http://knowyourmeme.com/memes/hypnotoad

Signbunny: It's not for sure known who invented signbunny, but this article http://www.vox.com/2014/9/18/6331753/sign-bunny-meme-explained thinks they found the first use in this tweet: https://twitter.com/wei_bluebear/status/32910164578077

Trilobite: from http://www.retrojunkie.com/asciiart/animals/dinos.htm (site down though)

Note to Windows users: there are some animals (shortcat, longcat, fish, signbunny, stretchycat, anxiouscat, longtailcat, grumpycat, mushroom) that are not available because they use non-ASCII characters that don't display properly in R on Windows.

Examples

```
tanx()

for (i in 1:4) tanx(i)

path_f <- system.file("fortunes/fortunes.csv", package = "fortunes")
path_s <- system.file("sinxs/sinxs.csv", package = "sinx")
ftns <- sinx::read.sinxs(c(path_f, path_s), sep = c(";", ","))
sinx::tanx(sinxs.data = ftns)

jinyong <- read.sinxs(lib = "jinyong")
tanx(sinxs.data = jinyong)

libs <- read.sinxs(lib = c("tangshi", "songshi", "chinese", "yangsheng", "english",
    "jinyong"))
tanx(42, sinxs.data = libs)
```

Index