

Package ‘r2symbols’

August 15, 2022

Type Package

Title Symbols for 'Markdown' and 'Shiny' Application

Version 1.1

Author Obinna Obianom

Maintainer Obinna Obianom <idonshayo@gmail.com>

Description Direct insertion of symbols (e.g. currencies, letters, arrows, mathematical symbols and so on) into 'Rmarkdown' documents and 'Shiny' applications by incorporating 'HTML' hex codes.

License MIT + file LICENSE

URL <https://r2symbols.obianom.com>

BugReports <https://github.com/oobianom/r2symbols>

Depends R (>= 3.4)

Imports utils, htmltools, jsonlite

Suggests rmarkdown, knitr, qpdf

Encoding UTF-8

VignetteBuilder knitr

Language en-US

LazyData false

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-15 05:40:07 UTC

R topics documented:

as.html	2
is.empty	2
sym	3
sym.setting	4

symbol	5
symbol.setting	6
symbol.units	6
symload	7
trans.r2symbols	8

Index	9
--------------	----------

as.html	<i>Transform to html</i>
---------	--------------------------

Description

Transform string into HTML ready for Rmarkdown and Shiny

Usage

```
as.html(h)
```

Arguments

h	string
---	--------

Value

HTML form of the string for easy insertion in Shiny or Rmarkdown documents

Examples

```
string <- "<b>smile</b>" #html b tag included to make the string bold
string_trans <- as.html(string)
cat(string) #when inserted in Rmarkdown will not show up as bolded
cat(string_trans) #when inserted into Rmarkdown will show up as bolded after knitting
```

is.empty	<i>Is string empty</i>
----------	------------------------

Description

Return TRUE or FALSE after checking if a string is empty

Usage

```
is.empty(string)
```

Arguments

string the string to be checked

Value

Returns TRUE or FALSE

Examples

```
string <- "r2resume"
string2 <- NULL
is.empty(string)
is.empty(string2)
```

sym

Get a desired symbol

Description

Get a symbol by name or number for use in Rmarkdown or Shiny apps

Usage

```
sym(
  ...,
  font.size = NULL,
  font.weight = NULL,
  font.color = NULL,
  if.error = FALSE,
  search.units.each = FALSE
)
```

Arguments

... the symbols to display

font.size font size as integer

font.weight font weight as integer

font.color font color e.g. red, blue, cyan

if.error error response

search.units.each whether to search single letter columns

Value

symbol hex for HTML displays

Examples

```
sym("trademark")
sym("alpha")
sym("arrow-right")
sym(144)
```

sym.setting

Symbol setting

Description

Font styling for symbols

Usage

```
sym.setting(font.size = NULL, font.weight = NULL, font.color = NULL)
```

Arguments

font.size	font size as integer
font.weight	font weight eg. normal, bold or lighter
font.color	font color e.g. red, blue, cyan

Value

No values returned

Examples

```
symbol.setting(40, "bold", "green") # ascribe settings for all symbols
symbol("alpha") # display the alpha symbol
symbol.setting() # clear settings for symbols
symbol("alpha") # display the alpha symbol
```

symbol	<i>Get a desired symbol</i>
--------	-----------------------------

Description

Get a symbol by name or number for use in Rmarkdown or Shiny apps, could be a mathematical symbol or other kinds of symbols

Usage

```
symbol(  
  ...,  
  font.size = NULL,  
  font.weight = NULL,  
  font.color = NULL,  
  if.error = FALSE,  
  search.units.each = FALSE  
)
```

Arguments

...	the symbols to display
font.size	font size as integer
font.weight	font weight as integer
font.color	font color e.g. red, blue, cyan
if.error	error response
search.units.each	whether to search single letter columns

Value

symbol hex for HTML integration

Complete cheatcheat for symbols

Common symbols include alpha, beta, copyright, registered, theta, sigma. See the complete list at <https://github.com/oobianom/r2symbols>.

Examples

```
symbol("trademark")  
symbol("alpha")  
symbol("arrow-right")  
symbol(144)
```

symbol.setting *Symbol setting*

Description

Font styling for symbols

Usage

```
symbol.setting(font.size = NULL, font.weight = NULL, font.color = NULL)
```

Arguments

font.size	font size as integer
font.weight	font weight eg. normal, bold or lighter
font.color	font color e.g. red, blue, cyan

Value

No values returned

Examples

```
symbol.setting(40, "bold", "green") # ascribe settings for all symbols  
symbol("alpha") # display the apha symbol  
symbol.setting() # clear settings for symbols  
symbol("alpha") # display the alpha symbol
```

symbol.units *Transform custom units*

Description

Enter a unit and it will automatically be transformed to the appropriate form

Usage

```
symbol.units(units, error = FALSE)
```

Arguments

units	the desired unit e.g ug/mL,pg/mL, ab, yL
error	whether to report any errors

Value

Transformed units with substitution of symbols with appropriate symbol hex

Examples

```
symbol.units("ug/mL") # u will be transformed to mu
symbol.units("pg/mL") # no letter will be transformed
```

symload

Symbol repository loader

Description

Loads all the available symbols

Usage

```
symload(
  font.size = NULL,
  font.weight = NULL,
  font.color = NULL,
  if.error = TRUE
)
```

Arguments

font.size	font size as integer
font.weight	font weight as integer
font.color	font color e.g. red, blue, cyan
if.error	error response

Value

No returns, but symbols are loaded

Examples

```
symload(40, "normal", "purple") # load symbols and add symbol settings
symload() # load symbols without custom settings
```

trans.r2symbols	<i>Transform HTML into symbols</i>
-----------------	------------------------------------

Description

Transform texts in HTML to symbols

Usage

```
trans.r2symbols(  
  font.size = NULL,  
  font.weight = c("normal", "bold", "lighter"),  
  font.color = NULL  
)
```

Arguments

font.size	font size as integer
font.weight	font weight eg. normal, bold or lighter
font.color	font color e.g. red, blue, cyan

Value

A rendered HTML where simple text is transformed to symbols using HTML codes

Note

This function simply inserts a script within the output HTML, which transforms the content using the relevant HTML codes.

Examples

```
trans.r2symbols()
```


Index

`as.html`, 2

`is.empty`, 2

`sym`, 3

`sym.setting`, 4

`symbol`, 5

`symbol.setting`, 6

`symbol.units`, 6

`symload`, 7

`trans.r2symbols`, 8