

Package: certestyle (via r-universe)

September 6, 2024

Title A Certe R Package for Applying Certe Organisational Style

Version 1.6.9

Description A Certe R Package for applying the organisational colours and style of Certe, plus some additional formatting functions. This package is part of the 'certedata' universe.

URL <https://certe-medical-epidemiology.github.io/certestyle>,
<https://github.com/certe-medical-epidemiology/certestyle>

Depends R (>= 4.1.0)

Imports cleaner (>= 1.5.2), crayon (>= 1.4.0), lubridate (>= 1.7.0), purrr (>= 0.3.4), readr (>= 1.4.0), rstudioapi (>= 0.8), viridisLite (>= 0.4.0), yaml (>= 2.2.0)

Suggests certemail, certetoolbox, certepplot2, certeprojects, hms (>= 1.0.0), knitr (>= 1.0.0), pillar (>= 1.5.0), rmarkdown, styler (>= 1.3.2), testthat (>= 2.0.0), tibble (>= 3.0.0)

License GPL-2

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/testthat/edition 2

VignetteBuilder knitr

Repository <https://certe-medical-epidemiology.r-universe.dev>

RemoteUrl <https://github.com/certe-medical-epidemiology/certestyle>

RemoteRef HEAD

RemoteSha 5c105cdb640f7abef7da22c9267c272f3014c7a6

Contents

certe.colours	2
certe_style_transformer	3
colourpicker	4

current_markdown_colour	7
dec_mark	8
font_colours	9
format2	10
is.double2	12
plain_html_table	13
rmarkdown	14
rstudio_install_certe_themes	14
toproper	15

Index 16

certe.colours	<i>Certe Colour Vector</i>
---------------	----------------------------

Description

This is a character vector with all Certe colours. It is used by `colourpicker()`.

Usage

```
certe.colours
```

Format

An object of class character of length 42.

Details

The colours are:

```
#> certeblauw certegroen certeroze certegeel certelila certebrauin
#> "#4A647D" "#93984C" "#B4527F" "#E4D559" "#CEB9D6" "#998961"
#> certeblauw2 certegroen2 certeroze2 certegeel2 certelila2 certebrauin2
#> "#69849C" "#AEB278" "#C581A0" "#E4DD9C" "#DED3E2" "#B9B09C"
#> certeblauw3 certegroen3 certeroze3 certegeel3 certelila3 certebrauin3
#> "#97AABB" "#C9CCA5" "#D5ACBF" "#ECE6B1" "#E6DDE9" "#CAC3B2"
#> certeblauw4 certegroen4 certeroze4 certegeel4 certelila4 certebrauin4
#> "#C5D0DB" "#DFE1C8" "#E7CCD8" "#F5F1CF" "#F0EAF2" "#DFDBD0"
#> certeblauw5 certegroen5 certeroze5 certegeel5 certelila5 certebrauin5
#> "#E2E7EC" "#EEEFE4" "#F2E6EB" "#F9F7E8" "#F7F4F8" "#EEECE8"
#> certeblauw6 certegroen6 certeroze6 certegeel6 certelila6 certebrauin6
#> "#F6F7F8" "#F9F9F6" "#FAF7F8" "#FCFBF8" "#FCFBFC" "#F9F8F7"
#> certeblauw0 certegroen0 certeroze0 certegeel0 certelila0 certebrauin0
#> "#3A4D5D" "#5A5D33" "#7F3C5B" "#D4C230" "#BEA5C7" "#675D45"
```

In RGB:

```

#>      certeblauw      certegroen      certeroze      certegeel
#> rgb(74, 100, 125) rgb(147, 152, 76) rgb(180, 82, 127) rgb(228, 213, 89)
#>      certelila      certebruin      certeblauw2      certegroen2
#> rgb(206, 185, 214) rgb(153, 137, 97) rgb(105, 132, 156) rgb(174, 178, 120)
#>      certeroze2      certegeel2      certelila2      certebruin2
#> rgb(197, 129, 160) rgb(228, 221, 156) rgb(222, 211, 226) rgb(185, 176, 156)
#>      certeblauw3      certegroen3      certeroze3      certegeel3
#> rgb(151, 170, 187) rgb(201, 204, 165) rgb(213, 172, 191) rgb(236, 230, 177)
#>      certelila3      certebruin3      certeblauw4      certegroen4
#> rgb(230, 221, 233) rgb(202, 195, 178) rgb(197, 208, 219) rgb(223, 225, 200)
#>      certeroze4      certegeel4      certelila4      certebruin4
#> rgb(231, 204, 216) rgb(245, 241, 207) rgb(240, 234, 242) rgb(223, 219, 208)
#>      certeblauw5      certegroen5      certeroze5      certegeel5
#> rgb(226, 231, 236) rgb(238, 239, 228) rgb(242, 230, 235) rgb(249, 247, 232)
#>      certelila5      certebruin5      certeblauw6      certegroen6
#> rgb(247, 244, 248) rgb(238, 236, 232) rgb(246, 247, 248) rgb(249, 249, 246)
#>      certeroze6      certegeel6      certelila6      certebruin6
#> rgb(250, 247, 248) rgb(252, 251, 248) rgb(252, 251, 252) rgb(249, 248, 247)
#>      certeblauw0      certegroen0      certeroze0      certegeel0
#>   rgb(58, 77, 93)   rgb(90, 93, 51)   rgb(127, 60, 91)   rgb(212, 194, 48)
#>      certelila0      certebruin0
#> rgb(190, 165, 199)   rgb(103, 93, 69)

```

certe_style_transformer

Syntax Format According to Certe Coding Guidelines

Description

Use this styler for formatting code by setting this as an option:

```
options(styler.addins_style_transformer = "certe_style_transformer()")
```

Then use the keyboard shortcut of the styler package to apply the formatting.

Usage

```
certe_style_transformer(...)
```

Arguments

... arguments passed on to `styler::tidyverse_style()`

Details

The Certe styler keeps the first argument of a function on the same line, puts all following arguments on a new line, and does not add another new line after the last argument. This makes these lines:

```
example_isolates %>% count(hospital_id, gender) %>% plot2(x.title = "Hospital", y.title = "Count", titl

example_isolates %>%
  count(hospital_id, gender) %>%
  plot2(x.title = "Hospital", y.title = "Count", title = "Count isolates per hospital/gender")
```

Change into:

```
example_isolates %>%
  count(hospital_id,
        gender) %>%
  plot2(x.title = "Hospital",
        y.title = "Count",
        title = "Count isolates per hospital/gender")
```

colourpicker

Colours from R, Certe, Viridis and More

Description

Colours from R, Certe, viridis and more. The output prints in the console with the actual colours.

Usage

```
colourpicker(x, length = 1, opacity = 0, ...)

## S3 method for class 'colourpicker'
as.character(x, ...)

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

add_white(x, white)
```

Arguments

x colour or colour palette name. Certe colours will be used from the [certe.colours](#) object. Input can be:

- "certe"
- "certe0" to "certe6" (higher numbers give lighter colours)
- "certeblauw", "certegroen", "certeroze", "certegeel", "certelila", or "certebruin" (or any of these followed by a 0 to 6)
- "certe_sir" or "certe_sir2" for certeroze/certegeel/certegroen (will **always** return length 5, with names "S", "SI", "I", "IR", "R")
- One of the colourblind-safe viridisLite palettes:
 - "viridis"

- "magma"
- "inferno"
- "plasma"
- "cividis"
- "rocket"
- "mako"
- "turbo"
- One of the built-in palettes in R (currently R 4.4.1):
 - "R3"
 - "R4"
 - "ggplot2"
 - "Okabe-Ito"
 - "Accent"
 - "Dark 2"
 - "Paired"
 - "Pastel 1"
 - "Pastel 2"
 - "Set 1"
 - "Set 2"
 - "Set 3"
 - "Tableau 10"
 - "Classic Tableau"
 - "Polychrome 36"
 - "Alphabet"
 - "topo"
 - "heatmap"
 - "rainbow"
 - "terrain"
 - "greyscale"
 - "grayscale"
- One of the 657 built-in `colours()` in R, such as "azure3", "chocolate4", "darkolivegreen", "seagreen", "steelblue1"

length	size of the vector to be returned
opacity	amount of opacity (0 = solid, 1 = transparent)
...	not used at the moment
white	number between [0, 1] to add white to x

Details

Certe colours will be chosen as divergent as possible if the required output length is not too high. For example:

- `x = "certe"` tries to only return the "certe" colours ("certeblauw", "certegroen", ...), the "certe3" colours ("certeblauw3", "certegroen3", ...) and the "certe5" colours ("certeblauw5", "certegroen5", ...)
- `x = "certe2"` tries to only return the regular "certe2" colours ("certeblauw2", "certegroen2", ...), the "certe4" colours ("certeblauw4", "certegroen4", ...) and the "certe6" colours ("certeblauw6", "certegroen6", ...)
- `x = "certe3"` tries to only return the "certe3" colours ("certeblauw3", "certegroen3", ...) and the "certe5" colours ("certeblauw5", "certegroen5", ...)

When using a single Certe colour with length > 1, a palette will be generated based on `*`, `*2`, `*3`, `*4`, and `*5`. For example, `colourpicker("certeblauw", 5)` will return the colours "certeblauw", "certeblauw2", "certeblauw3", "certeblauw4", and "certeblauw5".

A palette from `R` will be expanded where needed, so even `colourpicker("R4", length = 20)` will work, despite "R4" only supporting a maximum of eight colours.

Value

character vector in HTML format (i.e., "#AABBCC") with new class `colourpicker`

Examples

```
colourpicker("certegroen")
colourpicker("certe", 5)
colourpicker(c("certeblauw", "red", "tan1", "#ffa", "FFAA00"))

par(mar = c(0.5, 2.5, 1.5, 0)) # set plot margins for below plots

# Certe colours
barplot(12:1,
       col = colourpicker("certe", 12),
       main = "'certe': uses 'certe' + 'certe3'")
barplot(12:1,
       col = colourpicker("certe2", 12),
       main = "'certe2': uses 'certe2' + 'certe4'")
barplot(12:1,
       col = colourpicker("certe3", 12),
       main = "'certe3': uses 'certe3' + 'certe5'")
barplot(5:1,
       col = colourpicker("certeblauw", 5),
       main = "'certeblauw'")
barplot(12:1,
       col = colourpicker("certeblauw", 12),
       main = "'certeblauw': auto-extended range")

# all colourblind-safe colour palettes from the famous viridisLite package
barplot(1:7,
       col = colourpicker("viridis", 7))
barplot(1:7,
       col = colourpicker("magma", 7))

barplot(8:1,
```

```
      col = colourpicker("R4", 8),
      main = "Some palettes have only 8 colours...")
barplot(20:1,
      col = colourpicker("R4", 20),
      main = "Not anymore!")

colours <- colourpicker("R4", 6)
colours
add_white(colours, 0.25)
add_white(colours, 0.5)
add_white(colours, 0.75)
```

current_markdown_colour

Get Current Markdown Colour

Description

This function determines the Certe theme colour currently used in a markdown document (Quarto or R Markdown), based on the YAML header.

Usage

```
current_markdown_colour(default = "certeblauw")
```

Arguments

default default colour

Details

It returns a Certe colour if one is set in the YAML header, and checks in this order:

1. colour-main
2. colour-titlepage-titlebanner
3. colour-titlepage-full
4. colour-verticalbars
5. colour-heading1

If none is set, it returns the default setting, which is "certeblauw".

This function is the default to set the theme for `tbl_flextable()`.

It can be also be used for `plot2()`:

```
# will turn e.g. 'certeblauw' or 'certeroze' based on PDF format settings
library(certepplot2)
admitted_patients |>
  plot2(colour = current_markdown_colour())
```

`dec_mark`*Use Decimal Comma?*

Description

These functions determine which characters the decimal mark and big mark should be that are used in the Certe R package functions. They base the determination on the R [locale user settings](#).

Usage

```
dec_mark()
```

```
big_mark()
```

```
dec_mark_english()
```

Details

For `dec_mark()`: this returns a comma (",") on Dutch systems, and a full stop (".") otherwise. If the [option](#) "dec_mark" is set, that value will be used if it is either a comma or a full stop.

For `big_mark()`: this returns a full stop if `dec_mark()` returns a comma, and a space otherwise. If the [option](#) "big_mark" is set, that value will be used if it is either a comma (",") or a full stop (".") or a space (" ") or an empty character ("").

The function `dec_mark_english()` is short for `options(dec_mark = ".", big_mark = " ")` and useful for using the Certe R packages in English-based academic research. This function is session-specific, meaning that it must be set in every new R session (which is intended).

Examples

```
# according the current user settings / OS language:
dec_mark()
big_mark()

options(dec_mark = ",")
dec_mark()
big_mark()

options(dec_mark = ".")
dec_mark()
big_mark()

options(big_mark = ",")
dec_mark()
big_mark()

# clean up
options(dec_mark = NULL, big_mark = NULL)
```

font_colours	<i>Console Font Colours</i>
--------------	-----------------------------

Description

Add colours and font formatting to console text.

Usage

```
font_black(..., collapse = " ")
font_blue(..., collapse = " ")
font_green(..., collapse = " ")
font_magenta(..., collapse = " ")
font_red(..., collapse = " ")
font_red_bg(..., collapse = " ")
font_white(..., collapse = " ")
font_yellow(..., collapse = " ")
font_subtle(..., collapse = " ")
font_silver(..., collapse = " ")
font_grey(..., collapse = " ")
font_bold(..., collapse = " ")
font_italic(..., collapse = " ")
font_underline(..., collapse = " ")
font_stripstyle(...)
```

Arguments

...	character (vector)
collapse	character to separate the text elements. Use collapse = NULL to <i>not</i> collapse the input.

Details

Most colours are adapted based on the RStudio theme (dark/light).

Examples

```

cat(font_black("TEXT TEST"), "\n")
cat(font_red("TEXT TEST"), "\n")
cat(font_green("TEXT TEST"), "\n")
cat(font_blue("TEXT TEST"), "\n")
cat(font_bold("TEXT TEST"), "\n")
cat(font_italic("TEXT TEST"), "\n")
cat(font_underline("TEXT TEST"), "\n")
cat(font_red_bg(font_white("TEXT TEST")), "\n")

```

format2

*Formatting with readable format settings and Dutch defaults***Description**

Formatting with readable format settings and Dutch defaults

Usage

```

format2(x, ...)

## Default S3 method:
format2(x, ...)

## S3 method for class 'numeric'
format2(
  x,
  round = ifelse(percent, 1, 2),
  force_decimals = ifelse(percent, TRUE, FALSE),
  decimal.mark = dec_mark(),
  big.mark = big_mark(),
  min_length = 0,
  percent = FALSE,
  ...
)

## S3 method for class 'percentage'
format2(
  x,
  round = 1,
  force_decimals = TRUE,
  decimal.mark = dec_mark(),
  big.mark = big_mark(),
  ...
)

## S3 method for class 'Date'

```

```

format2(x, format = "d mmmm yyyy", locale = "nl", ...)

## S3 method for class 'POSIXt'
format2(x, format = "yyyy-mm-dd HH:MM:SS", locale = "nl", ...)

## S3 method for class 'hms'
format2(x, format = "HH:MM:SS", ...)

## S3 method for class 'difftime'
format2(
  x,
  round = 2,
  force_decimals = FALSE,
  decimal.mark = dec_mark(),
  big.mark = big_mark(),
  ...
)

## S3 method for class 'object_size'
format2(x, round = 1, decimal.mark = dec_mark(), ...)

format2_scientific(x, round = 2, decimal.mark = dec_mark(), ...)

```

Arguments

x	vector of values
...	arguments given to methods such as format()
round	number of decimals to round to
force_decimals	force printing decimals, even with trailing zeroes
decimal.mark, big.mark	decimal and thousands limiters
min_length	minimal length of output, overwrites force_decimals
percent	logical to transform numeric to percentage (character)
format	format to use, can be set with human-readable text such as "d mmmm yyyy" or POSIX such as "%e %B %Y"
locale	language to set for dates

Details

The [format2_scientific\(\)](#) function returns an [expression](#) and can be used in [ggplot2](#) plots.

Value

[format2\(\)](#) always returns a [character](#).

Examples

```

format2("2021-01-01")
format2("2021-01-01", "yyyy-qq")

format2(Sys.time(), "d mmmm yyyy HH:MM")

# content-aware of decimal use
format2(1024)
format2(c(1024, 0.123))

format2(2.1)
format2(2.1, force_decimals = TRUE) # since default is 2 decimals

p <- cleaner::as.percentage(0.123)
format2(p)
format2_scientific(c(12345, 12345678))

format2_scientific(c(12345, 12345678), round = 1)

# use format2_scientific for scientific labels in plots:
if (require("certepplot2")) {

  # y axis without scientific notation
  plot2(mtcars,
        y = hp * 1000)

  # y axis with scientific notation
  plot2(mtcars,
        y = hp * 1000,
        y.labels = format2_scientific)

}

```

is.double2

is.double() for Comma-Decimal Input

Description

This works like [is.double\(\)](#) and [as.double\(\)](#), but is vectorised and can also check (and transform) comma-decimal input such as "0,1".

Usage

```
is.double2(x, dec = c(".", ","))
```

```
as.double2(x, dec = c(".", ","))
```

Arguments

x vector of values
dec characters to be treated as comma

Examples

```
is.double(0.1)
is.double("0.1")
is.double("0,1")

is.double2(0.1)
is.double2("0.1")
is.double2("0,1")

is.double(c(0.1, "0.1", "0,1"))
is.double2(c(0.1, "0.1", "0,1"))

as.double(c(0.1, "0.1", "0,1"))
as.double2(c(0.1, "0.1", "0,1"))
```

plain_html_table *Format Data Set as HTML*

Description

Formats a [data.frame](#) into HTML code, so it can be used in e.g. [mail\(\)](#) or [teams\(\)](#).

Usage

```
plain_html_table(x, max_col = Inf)
```

Arguments

x a [data.frame](#)
max_col maximum number of columns to return

Examples

```
plain_html_table(mtcars[1:2, 1:2])
```

 rmarkdown

Markdown Template Properties

Description

These functions can be used in R Markdown documents.

Usage

```
rmarkdown_author(user_id = NULL)

rmarkdown_department()

rmarkdown_date(date = Sys.Date())

rmarkdown_table(...)

rmarkdown_template(type = "latex")

rmarkdown_logo(logo_type = "certe")
```

Arguments

<code>user_id</code>	user ID at Certe, defaults to currently logged in user
<code>date</code>	Date to print in "d mmmm yyyy"
<code>...</code>	data set (and options) to pass on to tbl_flextable()
<code>type</code>	defaults to "latex", must be "latex" or "word" (case-insensitive)
<code>logo_type</code>	type of logo, must be one of the files in /inst/rmarkdown/latextemplate of the certestyle package. For the LaTeX template, the front logo must be 16x7 cm, and the footer logo must be 16x0.7 cm.

 rstudio_install_certe_themes

Certe Themes for RStudio

Description

Install and apply RStudio syntax highlighting in Certe theme colours.

Usage

```
rstudio_install_certe_themes(apply_theme = "Certe Light")  
  
rstudio_set_certe_light()  
  
rstudio_set_certe_dark()  
  
rstudio_set_certe_light_nonbold()  
  
rstudio_set_certe_dark_nonbold()
```

Arguments

`apply_theme` The theme to apply after install

Details

This package comes with four RStudio editor themes that can be installed with `rstudio_install_certe_themes()`: "Certe Light", "Certe Light Non-Bold", "Certe Dark" and "Certe Dark Non-Bold".

Quickly apply the light theme with `rstudio_set_certe_light()` and the dark theme with `rstudio_set_certe_dark()` (they will be installed if needed).

toproper

Proper Case for Text

Description

Proper Case for Text

Usage

```
toproper(text, every_word = FALSE, ...)
```

Arguments

`text` text to transform
`every_word` set captial to every word
`...` unused, allows for backwards compatibility and future extension

Index

- * **datasets**
 - certe.colours, 2
- add_white (colourpicker), 4
- as.character.colourpicker (colourpicker), 4
- as.double(), 12
- as.double2 (is.double2), 12

- big_mark (dec_mark), 8
- big_mark(), 8

- certe.colours, 2, 4
- certe_style_transformer, 3
- character, 6, 11
- colourpicker, 4
- colourpicker(), 2
- colours(), 5
- current_markdown_colour, 7

- data.frame, 13
- dec_mark, 8
- dec_mark(), 8
- dec_mark_english (dec_mark), 8
- dec_mark_english(), 8

- expression, 11

- font_black (font_colours), 9
- font_blue (font_colours), 9
- font_bold (font_colours), 9
- font_colours, 9
- font_green (font_colours), 9
- font_grey (font_colours), 9
- font_italic (font_colours), 9
- font_magenta (font_colours), 9
- font_red (font_colours), 9
- font_red_bg (font_colours), 9
- font_silver (font_colours), 9
- font_stripstyle (font_colours), 9
- font_subtle (font_colours), 9

- font_underline (font_colours), 9
- font_white (font_colours), 9
- font_yellow (font_colours), 9
- format(), 11
- format2, 10
- format2(), 11
- format2_scientific (format2), 10
- format2_scientific(), 11

- is.double(), 12
- is.double2, 12

- locale user settings, 8

- mail(), 13

- option, 8

- plain_html_table, 13
- plot2(), 7
- print.colourpicker (colourpicker), 4

- rmarkdown, 14
- rmarkdown_author (rmarkdown), 14
- rmarkdown_date (rmarkdown), 14
- rmarkdown_department (rmarkdown), 14
- rmarkdown_logo (rmarkdown), 14
- rmarkdown_table (rmarkdown), 14
- rmarkdown_template (rmarkdown), 14
- rstudio_install_certe_themes, 14
- rstudio_install_certe_themes(), 15
- rstudio_set_certe_dark (rstudio_install_certe_themes), 14
- rstudio_set_certe_dark(), 15
- rstudio_set_certe_dark_nonbold (rstudio_install_certe_themes), 14
- rstudio_set_certe_light (rstudio_install_certe_themes), 14

`rstudio_set_certe_light()`, [15](#)
`rstudio_set_certe_light_nonbold`
 (`rstudio_install_certe_themes`),
 [14](#)

`tbl_flextable()`, [7](#), [14](#)
`teams()`, [13](#)
`toproper`, [15](#)