

Package ‘colt’

October 10, 2017

Type Package

Title Command-Line Color Themes

Version 0.1.1

Maintainer Stefan Fleck <stefan.b.fleck@gmail.com>

Description A collection of command-line color styles based on the ‘crayon’ package. ‘Colt’ styles are defined in themes that can easily be switched, to ensure command line output looks nice on dark as well as light consoles.

License MIT + file LICENSE

Imports crayon (>= 1.3.4)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, rstudioapi

Depends R (>= 2.10)

URL <https://github.com/s-fleck/colt>

BugReports <https://github.com/s-fleck/colt/issues>

NeedsCompilation no

Author Stefan Fleck [aut, cre]

Repository CRAN

Date/Publication 2017-10-10 04:20:30 UTC

R topics documented:

clt_theme_dark	2
clt_theme_light	2
colt_styles	2
colt_theme	3
print.colt_theme	5

Index	6
--------------	----------

clt_theme_dark *Colt Theme for Dark Consoles*

Description

Colt Theme for Dark Consoles

See Also

colt_theme

clt_theme_light *Colt Theme for Light Consoles*

Description

Colt Theme for Light Consoles

See Also

colt_theme

colt_styles *Colt style elements*

Description

Colt style elements

Usage

```
clt_default(x, fun = options("colt.theme")[[1]]$clt_default)
```

```
clt_h1(x, fun = options("colt.theme")[[1]]$clt_h1)
```

```
clt_h2(x, fun = options("colt.theme")[[1]]$clt_h2)
```

```
clt_chr(x, fun = options("colt.theme")[[1]]$clt_chr)
```

```
clt_chr_accent(x, fun = options("colt.theme")[[1]]$clt_chr_accent)
```

```
clt_chr_subtle(x, fun = options("colt.theme")[[1]]$clt_chr_subtle)
```

```
clt_true(x, fun = options("colt.theme")[[1]]$clt_true)
```

```
clt_false(x, fun = options("colt.theme")[[1]]$clt_false)
clt_maybe(x, fun = options("colt.theme")[[1]]$clt_maybe)
clt_num(x, fun = options("colt.theme")[[1]]$clt_num)
clt_num_accent(x, fun = options("colt.theme")[[1]]$clt_num_accent)
clt_num_subtle(x, fun = options("colt.theme")[[1]]$clt_num_subtle)
clt_bg(x, fun = options("colt.theme")[[1]]$clt_bg)
clt_bg_subtle(x, fun = options("colt.theme")[[1]]$clt_bg_subtle)
clt_bg_accent(x, fun = options("colt.theme")[[1]]$clt_bg_accent)
clt_na(x, fun = options("colt.theme")[[1]]$clt_na)
clt_nan(x, fun = options("colt.theme")[[1]]$clt_nan)
clt_inf(x, fun = options("colt.theme")[[1]]$clt_inf)
clt_null(x, fun = options("colt.theme")[[1]]$clt_null)
clt_info(x, fun = options("colt.theme")[[1]]$clt_info)
clt_message(x, fun = options("colt.theme")[[1]]$clt_message)
clt_warning(x, fun = options("colt.theme")[[1]]$clt_warning)
clt_error(x, fun = options("colt.theme")[[1]]$clt_error)
clt_emph(x, fun = options("colt.theme")[[1]]$clt_emph)
clt_emph2(x, fun = options("colt.theme")[[1]]$clt_emph2)
```

Arguments

x	a character vector
fun	a styling function

Description

Colt comes with two default color themes: `clt_theme_light` and `clt_theme_dark`. A `colt_theme` is just a list of functions, so it is easy to create your own theme. Use `set_colt_theme()` to switch between themes.

Usage

```
colt_theme(clt_default = crayon::mypaste, clt_h1 = crayon::mypaste,
  clt_h2 = crayon::mypaste, clt_chr = crayon::mypaste,
  clt_chr_accent = crayon::mypaste, clt_chr_subtle = crayon::mypaste,
  clt_true = crayon::mypaste, clt_false = crayon::mypaste,
  clt_maybe = crayon::mypaste, clt_num = crayon::mypaste,
  clt_num_accent = crayon::mypaste, clt_num_subtle = crayon::mypaste,
  clt_bg = crayon::mypaste, clt_bg_subtle = crayon::mypaste,
  clt_bg_accent = crayon::mypaste, clt_na = crayon::mypaste,
  clt_nan = crayon::mypaste, clt_inf = crayon::mypaste,
  clt_null = crayon::mypaste, clt_info = crayon::mypaste,
  clt_message = crayon::mypaste, clt_warning = crayon::mypaste,
  clt_error = crayon::mypaste, clt_emph = crayon::mypaste,
  clt_emph2 = crayon::mypaste)
```

```
set_theme(x)
```

Arguments

<code>clt_default</code>	Default text style
<code>clt_h1</code>	Header Level 1
<code>clt_h2</code>	Header Level 2
<code>clt_chr</code>	Text
<code>clt_chr_accent</code>	Text (accented)
<code>clt_chr_subtle</code>	Text (subtle)
<code>clt_true</code>	TRUE
<code>clt_false</code>	FALSE
<code>clt_maybe</code>	Undecided
<code>clt_num</code>	Numeric values
<code>clt_num_accent</code>	Numeric values (accented)
<code>clt_num_subtle</code>	Numeric values (subtle)
<code>clt_bg</code>	Text background
<code>clt_bg_subtle</code>	Text background (subtle)
<code>clt_bg_accent</code>	Text background (accented)
<code>clt_na</code>	NA values
<code>clt_nan</code>	NAN values
<code>clt_inf</code>	INF values

clt_null	NULL values
clt_info	INFO level messages
clt_message	MESSAGE level messages
clt_warning	Warnings
clt_error	Errors
clt_emph	Emphasis
clt_emph2	stronger Emphasis
x	A colt_theme Object. If x is NULL, colors will be switched off

Value

colt_theme() creates a new colt_theme Object.
 set_theme() returns the input theme Objects invisibly.

Examples

```
mytheme <- colt_theme(
  clt_h1 = crayon::combine_styles(crayon::red, crayon::bold)
)

mytheme # preview theme

cat(clt_h1("Important"))

set_theme(mytheme)
cat(clt_h1("Important"))

set_theme(clt_theme_light)
cat(clt_h1("Important"))
```

print.colt_theme *Printing Colt Themes*

Description

Printing Colt Themes

Usage

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

Arguments

x	a colt_theme
...	ignored

Index

*Topic **data**

- `clt_theme_dark`, 2
- `clt_theme_light`, 2

- `clt_bg (colt_styles)`, 2
- `clt_bg_accent (colt_styles)`, 2
- `clt_bg_subtle (colt_styles)`, 2
- `clt_chr (colt_styles)`, 2
- `clt_chr_accent (colt_styles)`, 2
- `clt_chr_subtle (colt_styles)`, 2
- `clt_default (colt_styles)`, 2
- `clt_emph (colt_styles)`, 2
- `clt_emph2 (colt_styles)`, 2
- `clt_error (colt_styles)`, 2
- `clt_false (colt_styles)`, 2
- `clt_h1 (colt_styles)`, 2
- `clt_h2 (colt_styles)`, 2
- `clt_inf (colt_styles)`, 2
- `clt_info (colt_styles)`, 2
- `clt_maybe (colt_styles)`, 2
- `clt_message (colt_styles)`, 2
- `clt_na (colt_styles)`, 2
- `clt_nan (colt_styles)`, 2
- `clt_null (colt_styles)`, 2
- `clt_num (colt_styles)`, 2
- `clt_num_accent (colt_styles)`, 2
- `clt_num_subtle (colt_styles)`, 2
- `clt_theme_dark`, 2
- `clt_theme_light`, 2
- `clt_true (colt_styles)`, 2
- `clt_warning (colt_styles)`, 2
- `colt_styles`, 2
- `colt_theme`, 3

- `print.colt_theme`, 5

- `set_theme (colt_theme)`, 3