

# Package ‘trekcolors’

June 1, 2021

**Title** Star Trek Color Palettes

**Version** 0.1.3

**Description** Provides a dataset of predefined color palettes based on the Star Trek science fiction series, associated color palette functions, and additional functions for generating customized palettes that are on theme. The package also offers functions for applying the palettes to plots made using the 'ggplot2' package.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/leonawicz/trekcolors>

**BugReports** <https://github.com/leonawicz/trekcolors/issues>

**Depends** R (>= 2.10)

**Imports** ggplot2

**Suggests** testthat, knitr, rmarkdown, covr

**RoxygenNote** 7.1.1

**Language** en-US

**NeedsCompilation** no

**Author** Matthew Leonawicz [aut, cre] (<<https://orcid.org/0000-0001-9452-2771>>)

**Maintainer** Matthew Leonawicz <[mflleonawicz@gmail.com](mailto:mflleonawicz@gmail.com)>

**Repository** CRAN

**Date/Publication** 2021-06-01 16:40:02 UTC

## R topics documented:

lcars_colors	2
lcars_pals	3
scale_lcars	4
scale_trek	5
trekpals	6
trek_pal	7

---

lcars_colors	<i>Hex colors from LCARS color names</i>
--------------	--

---

### Description

Obtain hex colors from standard LCARS color names.

### Usage

```
lcars_colors(...)
```

```
lcars_2357(...)
```

```
lcars_2369(...)
```

```
lcars_2375(...)
```

```
lcars_2379(...)
```

### Arguments

... character, LCARS color names.

### Details

These functions return the hex colors for LCARS color names. If no argument is provided, the functions return their respective LCARS color set. These functions correspond to LCARS color palettes that contain named colors. Other predefined LCARS color palettes are available but do not have names for each color. You can see all LCARS palettes with [lcars\\_pals](#).

### See Also

[lcars\\_pals](#)

### Examples

```
lcars_colors()  
lcars_2357()  
lcars_colors("rust", "danub")
```

---

`lcars_pals`*Palettes and palette generating functions based on LCARS colors*

---

**Description**

Predefined and custom palettes based on LCARS colors.

**Usage**

```
lcars_pals()

lcars_pal(palette = "2357", reverse = FALSE)

lcars_colors_pal(palette, reverse = FALSE, ...)
```

**Arguments**

<code>palette</code>	character, name of a single predefined LCARS palette; or a vector of LCARS or other colors. See details.
<code>reverse</code>	logical, reverse color order.
<code>...</code>	additional arguments to pass to <code>colorRampPalette</code> .

**Details**

`lcars_pal` returns a predefined, qualitative LCARS color palette. `lcars_color_pal` returns a palette generator based on specific colors. `lcars_pals` is a function that takes no arguments and returns a list of all predefined LCARS palettes.

Predefined palettes options for `palette` are "2357", "2369", "2375", "2379", "alt", "first\_contact", "nemesis", "nx01", "23c", "29c", "red\_alert" and "cardassian".

Custom palettes can also be constructed by passing a vector of colors to `palette` in `lcars_color_pal`. This is useful for sequential and divergent palettes. This is essentially a wrapper around `colorRampPalette` that understands LCARS color names. It accepts any color, allowing you to still use a color like "white" or "#FFFFFF" as the midpoint in a divergent palette for example. A special case is when you provide only a single color, e.g., `lcars_color_pal("husk")`; this is equivalent to `lcars_color_pal(c("white", "husk"))`.

**Value**

a palette generating function that takes an argument, `n`, the number of colors.

**Examples**

```
# All predefined LCARS palettes
lcars_pals()
# predefined palette
lcars_pal("2357")
# custom palettes
```

```
lcars_colors_pal("rust")(8) # sequential
lcars_colors_pal(c("pale-canary", "rust"))(8)
lcars_colors_pal(c("pale-canary", "rust"))(4)
lcars_colors_pal(c("mariner", "white", "rust"))(9) # divergent
```

---

scale\_lcars

*Color and fill scale functions for LCARS colors*

---

## Description

Scale functions used with ggplot2.

## Usage

```
scale_color_lcars(palette = "2357", discrete = TRUE, reverse = FALSE, ...)
```

```
scale_color_lcars1(
  color = "atomic-tangerine",
  discrete = TRUE,
  reverse = FALSE,
  dark = FALSE,
  ...
)
```

```
scale_color_lcars2(
  low = "atomic-tangerine",
  high = "near-blue",
  discrete = TRUE,
  reverse = FALSE,
  dark = FALSE,
  divergent = FALSE,
  ...
)
```

```
scale_fill_lcars(palette = "2357", discrete = TRUE, reverse = FALSE, ...)
```

```
scale_fill_lcars1(
  color = "atomic-tangerine",
  discrete = TRUE,
  reverse = FALSE,
  dark = FALSE,
  ...
)
```

```
scale_fill_lcars2(
  low = "atomic-tangerine",
  high = "near-blue",
```

```

    discrete = TRUE,
    reverse = FALSE,
    dark = FALSE,
    divergent = FALSE,
    ...
  )

```

### Arguments

palette	character, name of palette in lcars_pals.
discrete	logical, discrete or continuous palette.
reverse	logical, reverse color order.
...	additional arguments passed to <code>ggplot2::discrete_scale</code> or <code>ggplot2::scale*_gradientn</code> , for discrete or continuous palettes, respectively.
color	character, LCARS color name for sequential palette.
dark	logical, use black instead of white for the base color in sequential palette or midpoint in divergent palette.
low	character, LCARS color name.
high	character, LCARS color name.
divergent	logical, use a divergent palette instead of two-color sequential palette.

### Examples

```

library(ggplot2)
p <- ggplot(diamonds, aes(carat, stat(count), fill = cut)) +
  geom_density(position = "fill")
p + scale_fill_lcars("2357")
p + scale_fill_lcars1("atomic-tangerine", dark = TRUE)
p + scale_fill_lcars2("pale-canary", "danub")

```

---

scale\_trek

*Color and fill scale functions for Star Trek palettes*

---

### Description

Scale functions used with `ggplot2`.

### Usage

```
scale_color_trek(palette = "starfleet", discrete = TRUE, reverse = FALSE, ...)
```

```
scale_fill_trek(palette = "starfleet", discrete = TRUE, reverse = FALSE, ...)
```

**Arguments**

palette	character, name of Star Trek palette. See <code>trek_pal()</code> for list of palette names.
discrete	logical, discrete or continuous palette.
reverse	logical, reverse color order.
...	additional arguments passed to <code>ggplot2::discrete_scale</code> or <code>ggplot2::scale_*_gradientn</code> , for discrete or continuous palettes, respectively.

**Details**

Most palettes should be used as qualitative palettes. See `trekpals` to see how many colors are in each predefined palette. Use `view_trek_pals()` to plot all palettes to see which may work best for your purposes.

**Examples**

```
library(ggplot2)
d <- diamonds[diamonds$cut >= "Very Good", ]
ggplot(d, aes(carat, stat(count), fill = cut)) +
  geom_density(position = "fill") +
  scale_fill_trek("starfleet")
```

---

trekpals                      *Star Trek color palettes.*

---

**Description**

A named list of 35 Star Trek color palettes.

**Usage**

```
trekpals
```

**Format**

A named list.

---

trek_pal	<i>Star Trek color palettes</i>
----------	---------------------------------

---

### Description

Return a predefined Star Trek color palette from the trekpals dataset.

### Usage

```
trek_pal(palette, reverse = FALSE)

view_trek_pals(palette)
```

### Arguments

palette	character, name of predefined palette. If missing, return all available palette names.
reverse	logical, reverse color order.

### Details

Many of the palettes are qualitative, and not necessarily evenly spaced in terms of hue, saturation or brightness. This is because many palettes come from logos, symbols, insignia and other simple representations. However, several palettes have specifically been constructed as sequential or divergent if it made sense to do so based on the dominant colors present. Additional special functions exist for the subset of LCARS palettes.

### Value

character vector of hex colors or palette names

### See Also

[lcars\\_pals](#), [scale\\_lcars](#)

### Examples

```
trek_pal("lcars_2357")
# leave palette blank to list available names:
trek_pal()

# to view all palettes
view_trek_pals()
```

# Index

## \* datasets

trekpals, [6](#)

lcars\_2357 (lcars\_colors), [2](#)

lcars\_2369 (lcars\_colors), [2](#)

lcars\_2375 (lcars\_colors), [2](#)

lcars\_2379 (lcars\_colors), [2](#)

lcars\_colors, [2](#)

lcars\_colors\_pal (lcars\_pals), [3](#)

lcars\_pal (lcars\_pals), [3](#)

lcars\_pals, [2](#), [3](#), [7](#)

scale\_color\_lcars (scale\_lcars), [4](#)

scale\_color\_lcars1 (scale\_lcars), [4](#)

scale\_color\_lcars2 (scale\_lcars), [4](#)

scale\_color\_trek (scale\_trek), [5](#)

scale\_fill\_lcars (scale\_lcars), [4](#)

scale\_fill\_lcars1 (scale\_lcars), [4](#)

scale\_fill\_lcars2 (scale\_lcars), [4](#)

scale\_fill\_trek (scale\_trek), [5](#)

scale\_lcars, [4](#), [7](#)

scale\_trek, [5](#)

trek\_pal, [7](#)

trekpals, [6](#)

view\_trek\_pals (trek\_pal), [7](#)