Package ‘unikn’

August 10, 2023

Type Package

Title Graphical Elements of the University of Konstanz's Corporate Design

Version 0.9.0

Date 2023-08-09

Maintainer Hansjoerg Neth <h.neth@uni.kn>

Description Define and use graphical elements of corporate design manuals in R. The 'unikn' package provides color functions (by defining dedicated colors and color palettes, and commands for finding, changing, viewing, and using them) and styled text elements (e.g., for marking, underlining, or plotting colored titles). The pre-defined range of colors and text decoration functions is based on the corporate design of the University of Konstanz <https://www.uni-konstanz.de/>, but can be adapted and extended for other purposes or institutions.

Depends R (>= 3.4.0)

Imports utils, cli, ggplot2

Suggests knitr, rmarkdown, spelling


Encoding UTF-8

License CC BY-SA 4.0

URL https://CRAN.R-project.org/package=unikn

BugReports https://github.com/hneth/unikn/issues

VignetteBuilder knitr

RoxygenNote 7.2.3

Language en-US

NeedsCompilation no

Author Hansjoerg Neth [aut, cre] (<https://orcid.org/0000-0001-5427-3141>),
Nico Gradwohl [aut] (<https://orcid.org/0000-0002-8703-905X>)}
Repository CRAN
Date/Publication 2023-08-10 15:00:02 UTC

R topics documented:

ac ................................................................. 3
Bordeaux ..................................................... 4
demopal ........................................................ 5
get_col_names ............................................... 6
Grau .............................................................. 7
grepal ........................................................... 7
heading ........................................................ 9
Karpfenblau .................................................. 11
mark .............................................................. 11
newpal .......................................................... 13
pal_bordeaux ................................................ 15
pal_grau ....................................................... 16
pal_karpfenblau ............................................. 17
pal_peach ..................................................... 18
pal_petrol ..................................................... 19
pal_pinky ..................................................... 20
pal_seeblau .................................................. 21
pal_seegruen ............................................... 22
pal_signal .................................................... 23
pal_unikn ..................................................... 24
pal_unikn_dark ............................................. 25
pal_unikn_light ............................................ 26
pal_unikn_pair .............................................. 27
pal_unikn_ppt .............................................. 28
pal_unikn_pref ............................................. 29
pal_unikn_web .............................................. 30
Peach ........................................................... 31
Petrol .......................................................... 31
Pinky ........................................................... 32
post .............................................................. 33
Seeblau ........................................................ 35
seecol .......................................................... 36
Seegruen ...................................................... 38
shades_of .................................................... 39
Signal .......................................................... 40
simcol ........................................................ 41
slide ............................................................ 43
theme_grau .................................................. 44
theme_unikn ................................................ 45
uiline ........................................................... 46
unikn_guide .................................................. 48
url_unikn ..................................................... 48
**Adjust color transparency**

**Description**

`ac` adjusts the transparency of a color or color palette `col` to an opacity level `alpha` (in $[0, 1]$).

**Usage**

```r
ac(col, alpha = 0.5, use_names = TRUE)
```

**Arguments**

- `col`  
  A (required) color or color palette (as a vector).

- `alpha`  
  A factor modifying the opacity `alpha` (as `alpha.f` in `adjustcolor`) to a value in $[0, 1]$. Default: `alpha = .50` (i.e., medium opacity).

- `use_names`  
  A logical value indicating whether color names should be adjusted to include the values of `alpha`. Default: `use_names = TRUE`.

**Details**

`ac` is mostly a wrapper for `adjustcolor` of the `grDevices` package, but allows for more flexible combinations of (multiple) `col` and `alpha` values.

The name `ac` is an abbreviation of "adjust color", but is also a mnemonic aid for providing "air conditioning".

**Value**

A color vector of the same length as `col`, transformed by `adjustcolor`.

**See Also**

- `seecol` for plotting/seeing color palettes; `usecol` for using color palettes; `simcol` for finding similar colors; `newpal` for defining new color palettes; `grepal` for finding named colors.

Other color functions: `demopal()`, `grepal()`, `newpal()`, `seecol()`, `shades_of()`, `simcol()`, `usecol()`
Examples

ac("black")  # using alpha = .5 by default

# multiple colors:
cols <- ac(c("black", "gold", "deeppskyblue"), alpha = .50)
seecol(cols, main = "Transparent colors")

# multiple alphas:
blacks <- ac(c("black", "gold"), alpha = 5:6/5)
seecol(blacks, main = "One col several alpha values")

bgc <- ac(c("black", "gold"), alpha = 1:6/6)
seecol(bgc, main = "More alpha values than cols")

# Using a color palette:
seecol(ac(pal_unikn_pref, 2/3), main = "Adding color transparency by ac()")

# Color names:
seecol(ac(col = pal_unikn_pref, alpha = c(1/5, 4/5), use_names = TRUE))
seecol(ac(col = pal_unikn_pref, alpha = c(1/5, 4/5), use_names = FALSE))

--------
Bordeaux  uni.kn color Bordeaux
--------

Description

Bordeaux provides the preferred color of pal_bordeaux (as an atomic HEX character value) and is defined as pal_bordeaux[[4]].

Usage

Bordeaux

Format

An object of class character of length 1.

Details


See Also

pal_bordeaux for the corresponding color palette; pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_unikn_pref for a uni.kn color palette with all preferred colors; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other preferred colors: Grau, Karpfenblau, Peach, Petrol, Pinky, Seeblau, Seegruen, Signal
Examples

```r
Bordeaux  # HEX character "#8E2043" (as value)
all.equal(Bordeaux, pal_bordeaux[[4]])  # TRUE (same HEX values)
seecol(Bordeaux)  # view color and details
```

Description

demopal provides an example plot of some type to illustrate a color palette pal.

Usage

demopal(pal = pal_unikn, type = NA, pal_name = NULL, ...)

Arguments

- `pal` A color palette (to be illustrated). Default: `pal = pal_unikn`.
- `type` The type of plot to be used (as character string or integer value). Permissible types are "bar", "curve", "mosaic", "polygon", or "scatter" (or an integer value from 1 to 5, respectively).
- `pal_name` A name for the input color palette `pal` (shown on bottom-right margin). Default: `pal_name = NULL` (deparising to input name).
- `...` Auxiliary arguments passed to type-specific plots (see details).

Details

The `demopal` wrapper function passes a range of arguments to more specific functions. Common arguments include:

- `col_par` Default color for `par(col)`;
- `alpha` Default value for color transparency (in 0:1);
- `n` A scaling parameter (for random data generation);
- `main` plot title (on top);
- `sub` plot subtitle (on right margin);
- `seed` A random seed value (for reproducible randomness).

The fit between a color palette `pal` and plot type depends on the uses of colors in a plot. For instance, overlaps of transparent color areas can be evaluated with plot type = "curve" or plot type = "scatter" (and $0 < \alpha < 1$).

Some functions additionally accept type-specific arguments (e.g., `beside`, `horiz`, and `as_prop` for plot type = "bar", and `cex` for plot type = "scatter").

The type-specific functions usually generate some random data (scaled by a parameter `n`) that is being plotted. This data is returned (as an invisible R object) to enable a plot’s reconstruction.
Value

The random data that was plotted (as an invisible R object).

See Also

seepal for plotting color palettes; usecol for using color palettes; shades_of to defining shades of a given color; ac for adjusting color transparency; pal_unikn for the default uni.kn color palette.

Other color functions: ac(), grepal(), newpal(), seecol(), shades_of(), simcol(), usecol()

Examples

demopal(pal = pal_petrol, type = 1)

my_pal <- c(rev(pal_pinky), pal_seeblau)
# Selecting plot type:
demopal(my_pal, type = 2) # by numeric index
demopal(my_pal, type = "polygon") # by name

# Passing type-specific arguments:
demopal(type = "scatter", col_par = "black", n = 200, cex = c(2, 4, 6), seed = 101)

get_col_names

Description

get_col_names gets color names from color palettes.

Usage

get_col_names(col, custom_pals = all_pals)

Arguments

col A vector of colors.
custom_pals A vector of color palettes.

Value

A vector of color names.
Grau

uni.kn color Grau

Description

Grau provides the preferred color of `pal_grau` (as an atomic HEX character value) and is defined as `pal_grau[[3]]`.

Usage

Grau

Format

An object of class character of length 1.

Details


See Also

`pal_grau` for the corresponding color palette; `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Karpfenblau, Peach, Petrol, Pinky, Seeblau, Seegruen, Signal

Examples

```r
Grau # HEX character "#9AA0A7" (as value)
all.equal(Grau, pal_grau[[3]]) # TRUE (same HEX values)
seecol(Grau) # view color and details
```

grepal

Get a vector of colors whose names match a regular expression

Description

grepal returns a vector of colors whose names match a regular expression (regex).

Usage

grepal(pattern, x = colors(), ignore_case = TRUE, plot = TRUE)
**grepal**

**Arguments**

- **pattern**: A regular expression (specified as a string/character object).
- **x**: A vector of R color names or a data frame of named colors (i.e., whose names can be searched). Default: `x = colors()`.
- **ignore_case**: Should the case of pattern be ignored (passed to `ignore.case` of the `grep` function)? Default: `ignore_case = TRUE`.
- **plot**: Boolean: Plot the output (using `seecol`)? Default: `plot = TRUE`.

**Details**

By default, the base R vector of named colors (i.e., `colors()`) is searched for names matching a pattern (which can be a simple string or regular expression).

If `x` (i.e., the object to be searched) is provided, it must be a vector of color names or a data frame of named color objects (i.e., a color palette).

If `plot = TRUE`, `grepal` also visualizes the detected colors (by passing its result to `seecol`, as a side-effect).

This function facilitates searching colors by name and yields (a vector of) colors of similar color hue (provided that the color’s hue is expressed in a color’s name). Its name `grepal` is an abbreviation of `grep` and "pal".

**See Also**

- `seecol` for viewing and comparing color palettes; `usecol` for using color palettes; `simcol` for finding similar colors; `newpal` for defining new color palettes; `shades_of` to defining shades of a given color; `ac` for adjusting color transparency; `pal_unikn` for the default uni.kn color palette.

Other color functions: `ac()`, `demopal()`, `newpal()`, `seecol()`, `shades_of()`, `simcol()`, `usecol()`

**Examples**

grepal("tan")

# With regular expressions:
some_grey  <- grepal("gr(a|e)y", plot = FALSE)
start_grey  <- grepal("^gr(a|e)y", plot = FALSE)
only_grey   <- grepal("^gr(a|e)y$", plot = FALSE)

length(some_grey)
length(only_grey)

# With other color objects (df as x):
grepal("blau", x = pal_unikn)
grepal("SEE", x = pal_unikn_pref, ignore_case = FALSE)

# Applications:
seecol(grepal("white"), col_bg = "lightblue2", main = "See 'white' colors()")

olives  <- grepal("olive", plot = FALSE)
oranges <- grepal("orange", plot = FALSE)
```r
seecol(list(olives, oranges),
  pal_names = c("olives", "oranges"),
  main = "Comparing olives and oranges")
```

---

**heading**

**Plot a heading (as marked text elements)**

**Description**

`heading` plots 1 or more text strings (provided as a character vector `labels`) as a heading to an (existing or new) plot and places a colored box behind each label to mark it (i.e., highlighting the heading).

**Usage**

```r
heading(
  labels,
  x = 0,
  y = 0.8,
  x_layout = NA,
  y_layout = "flush",
  col = "black",
  col_bg = "default",
  cex = 2,
  font = 2,
  new_plot = "slide"
)
```

**Arguments**

- `labels`: A character vector specifying the text labels to be plotted.
- `x`: A numeric vector of x-coordinates at which the text labels in `labels` should be written. If the lengths of `x` and `y` differ, the shorter one is recycled. Default: `x = 0`.
- `y`: A numeric vector of y-coordinates at which the text labels in `labels` should be written. If the lengths of `x` and `y` differ, the shorter one is recycled. Default: `y = 0.8`.
- `x_layout`: An optional numeric vector or character string to control the horizontal positions of `labels`. Numeric values are interpreted as increments to values of `x` and recycled (to enable stepwise or alternating patterns). 3 character string options are: "center" (i.e., center wrt. first label or plot center), "left" (i.e., left wrt. first label or plot center), "right" (i.e., right wrt. first label or plot center). Default: `x_layout = NA` (i.e., using values of `x`).
heading

y_layout A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of y[1] and recycled (to enable stepwise or alternating patterns). 2 character string options are: "even" (i.e., even distribution of labels across available y-space) and "flush" (i.e., no space between adjacent labels, i.e., y_layout = 0). Default: y_layout = "flush".

col The color(s) of the text label(s). Default: col_lbl = "black".

col_bg The color(s) to highlight or fill the rectangle(s) with. Default: col_bg = "default" (to automatically select different shades of pal_seeblau).

cex Numeric character expansion factor(s), multiplied by par("cex") to yield the character size(s). Default: cex = 2.

font The font type(s) to be used. Default: font = 2 (i.e., bold).

ew_plot Boolean: Should a new plot be generated? Set to "blank" or "slide" to create a new plot, and to "none" to add to an existing plot. Default: new_plot = "slide" (i.e., create a new slide).

Details

Text formatting parameters (like col, col_bg, cex, font) are recycled to match length(labels). heading uses the base graphics system graphics:::

See Also

slide and xbox to create simple plots (without text).

Examples

heading(labels = c("This is a headline", "containing two lines."))

# Note the warning:
heading(labels = c("Headlines", "with 3 or more lines",
                "should not be arranged", "in such a step-wise fashion."))

# Avoiding the warning:
heading(labels = c("Headlines with", "3 or more lines should",
                "not be arranged", "in a step-wise fashion."))

# Using non-default colors:
heading(labels = c("Ene," "mene, miste.", "es rappelt", "in der Kiste."),
        cex = 1.6, col = "white", col_bg = usecol(c(Pinky, Seegruen, Bordeaux, Karpfenblau)))

# Using x_layout and y_layout:
heading(labels = c("Ene," "mene, miste.", "es rappelt", "in der Kiste."),
        cex = 1.6, col = "white", col_bg = usecol(pal_pinky[2:5]),
        x = NA, y = .6, x_layout = "right", y_layout = "flush")

# @family text functions
Karpfenblau

uni.kn color Karpfenblau

Description

Karpfenblau provides the preferred color of `pal_karpfenblau` (as an atomic HEX character value) and is defined as `pal_karpfenblau[[4]]`.

Usage

Karpfenblau

Format

An object of class character of length 1.

Details


See Also

- `pal_karpfenblau` for the corresponding color palette; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `pal_unikn` for the default uni.kn color palette; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Peach, Petrol, Pinky, Seeblau, Seegruen, Signal

Examples

```r
Karpfenblau  # HEX character "#3E5496" (as value)
all.equal(Karpfenblau, pal_karpfenblau[[4]])  # TRUE (same HEX values)

seecol(Karpfenblau)  # view color and details
```

mark

Plot marked (or highlighted) text elements

Description

mark plots 1 or more text strings (provided as a character vector `labels`) to an (existing or new) plot and places a colored box behind each label to mark it (i.e., highlight or make it stand out from the background).
Usage

```r
mark(
  labels,
  x = 0,
  y = 0.55,
  x_layout = NA,
  y_layout = "even",
  col = "black",
  col_bg = Seeblau,
  cex = 2,
  font = 2,
  new_plot = "none"
)
```

Arguments

- `labels` A character vector specifying the text labels to be plotted.
- `x` A numeric vector of x-coordinates at which the text labels in `labels` should be written. If the lengths of `x` and `y` differ, the shorter one is recycled. Default: `x = 0`.
- `y` A numeric vector of y-coordinates at which the text labels in `labels` should be written. If the lengths of `x` and `y` differ, the shorter one is recycled. Default: `y = 0.55`.
- `x_layout` An optional numeric vector or character string to control the horizontal positions of `labels`. Numeric values are interpreted as increments to values of `x` and recycled (to enable stepwise or alternating patterns). 3 character string options are: "center" (i.e., center wrt. first label or plot center), "left" (i.e., left wrt. first label or plot center), "right" (i.e., right wrt. first label or plot center). Default: `x_layout = NA` (i.e., using values of `x`).
- `y_layout` A numeric value or character string to control the vertical positions of `labels`. Numeric values are interpreted as increments to values of `y[1]` and recycled (to enable stepwise or alternating patterns). 2 character string options are: "even" (i.e., even distribution of labels across available y-space) and "flush" (i.e., no space between adjacent labels, i.e., `y_layout = 0`). Default: `y_layout = "even"`.
- `col` The color(s) of the text label(s). Default: `col_lbl = "black"`.
- `col_bg` The color(s) to highlight or fill the rectangle(s) with. Default: `col_bg = Seeblau`.
- `cex` Numeric character expansion factor(s), multiplied by `par("cex")` to yield the character size(s). Default: `cex = 2`.
- `font` The font type(s) to be used. Default: `font = 2` (i.e., bold).
- `new_plot` Should a new plot be generated? Set to "blank" or "slide" to create a new plot. Default: `new_plot = "none"` (i.e., add to an existing plot).

Details

The positions of the text elements in `labels` can be specified by providing their coordinates (as `x` and `y` arguments) or by providing an initial position and an `y_layout` (see below).
Text formatting parameters (like col, col_bg, cex, font) are recycled to match length(labels). mark uses the base graphics system graphics::.

See Also

slide and xbox to create simple plots (without text).

Other text functions: post(), uline(), url_unikn()

Examples

# Basics:
mark(labels = "This is a test.", new_plot = "blank")  # create a new blank plot
mark(labels = "More testing here...", y = .45, col_bg = pal_pinky[[2]])  # add to plot

# Example:
# (a) Mark text on an existing plot:
plot(x = 0, y = 0, type = "n", xlab = ", ylab = "
mark(x = 0, y = .8, labels = "Mark (on an existing plot)"")  # uses existing plot

# (b) Mark text on a new plot:
mark(x = 0, y = .8, labels = "Mark (and create a new plot)",
    new_plot = "slide")  # starts a new plot

# (c) More text and decorations:
mark(x = 0, y = c(.60, .50),
    labels = c("Highlighting text is simple", "and effective"),
    cex = 1.5, col_bg = c(pal_seeblau[[2]], pal_seeblau[[1]]))

mark(labels = c("It is also flexible", "but to be handled with care"),
    x = .4, y = .3, y_layout = "flush", cex = 1.2,
    col = c("white", "black"), col_bg = c(pal_seeblau[[5]], "gold"))

# Using x_layout and y_layout:
mark(labels = c("Ene,", "mene, miste,", "es rappelt", "in der Kiste."),
    cex = 1.4, font = 2, col = "white", col_bg = Petrol,
    x = NA, y = .85, x_layout = "center", y_layout = "even", new_plot = "slide")

mark(labels = c("One, and", "two, and", "three and four is", "plenty and perhaps enough..."),
    cex = 1.4, font = 2, col = "white", col_bg = Bordeaux,
    x = .5, y = .6, x_layout = c(-.25, +.25), y_layout = 0, new_plot = "slide")

newpal

Define a new color palette

newpal allows defining new color palettes (as data frames or vectors).
Usage

newpal(col, names = NULL, as_df = FALSE)

Arguments

col A required vector of colors (specified as R color names, HEX codes, or RGB values).

names An optional character vector of color names. Default: names = NULL, using default color names. Setting names = NA removes all color names.

as_df Should the new color palette be returned as a data frame (rather than as a vector)? Default: as_df = FALSE.

See Also

see_col for viewing and comparing color palettes; use_col for using color palettes; sim_col for finding similar colors; grepal for finding named colors; shades_of to defining shades of a given color; ac for adjusting color transparency; pal_unikn for the default uni.kn color palette.

Other color functions: ac(), demopal(), grepal(), see_col(), shades_of(), sim_col(), use_col()

Examples

newpal(col = c("black", "white"), names = c("dark", "bright"))

# Example: 3 ways of defining a new color palette:

# (1) From R color names: -----
pal_flag_de <- newpal(col = c("black", "firebrick3", "gold"),
  names = c("Schwarz", "Rot", "Gold"))
see_col(pal_flag_de, main = "Colors in the flag of Germany")

# (2) From HEX values: -----

# (a) Google logo colors:
# Source: https://www.schemecolor.com/google-logo-colors.php
color_google <- c("#4285f4", "#34a853", "#fbbc05", "#ea4335")
names_google <- c("blueberry", "sea green", "selective yellow", "cinnabar")
pal_google <- newpal(color_google, names_google)
see_col(pal_google, main = "Colors of the Google logo", col_brd = "white", lwd_brd = 10)

# (b) German flag (revised):
# Based on a different source at
# <https://www.schemecolor.com/germany-flag-colors.php>:
pal_flag_de_2 <- newpal(col = c("#000000", "#dd0000", "#ffce00"),
  names = c("black", "red", "gold")

see_col(pal_flag_de_2, main = "Colors of the German flag (www.schemecolor.com)")

# (c) MPG colors:
pal_mpg <- newpal(col = c("#007367", "white", "#D0D3D4"),
                  names = c("mpg green", "white", "mpg grey"))
seecol(pal_mpg, main = "Colors of the Max Planck Society")

# (3) From RGB values: -----

# A barrier-free color palette
# Source: Okabe & Ito (2002): Color Universal Design (CUD):
#   Fig. 16 of <https://jfly.uni-koeln.de/color/>:

# (a) Vector of colors (as RGB values):
o_i_colors <- c(rgb(0, 0, 0, maxColorValue = 255), # black
                 rgb(230, 159, 0, maxColorValue = 255), # orange
                 rgb(86, 180, 233, maxColorValue = 255), # skyblue
                 rgb(0, 158, 115, maxColorValue = 255), # green
                 rgb(240, 228, 66, maxColorValue = 255), # yellow
                 rgb(0, 114, 178, maxColorValue = 255), # blue
                 rgb(213, 94, 0, maxColorValue = 255), # vermillion
                 rgb(204, 121, 167, maxColorValue = 255) # purple
                 )

# (b) Vector of color names:
o_i_names <- c("black", "orange", "skyblue", "green", "yellow", "blue", "vermillion", "purple")

# (c) Use newpal() to combine colors and names:
pal_okabe_ito <- newpal(col = o_i_colors,
                        names = o_i_names)
seecol(pal_okabe_ito,
       main = "Color-blind friendly color scale (Okabe & Ito, 2002)")

# (+) Compare custom color palettes: -----

my_pals <- list(pal_flag_de, pal_flag_de_2, pal_google, pal_mpg, pal_okabe_ito)
seecol(my_pals, col_brd = "white", lwd_brd = 5,
       main = "Comparing custom color palettes")

---

### Description

`pal_bordeaux` provides an additional uni.kn color palette as a data frame containing 5 colors (shades of **Bordeaux**).

### Usage

`pal_bordeaux`
pal_grau

Format
An object of class data.frame with 1 rows and 5 columns.

Details

See Also
col, usecol for viewing and comparing color palettes; usecol for using color palettes.
Other color palettes: pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pink, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples
pal_bordeaux
dim(pal_bordeaux) # 1 5
pal_bordeaux[4] # preferred (named) color “bordeaux4”
pal_bordeaux[[4]] # preferred color “bordeaux4” OR “#8E2043”

# Plotting palette:
seecol(pal_bordeaux)

pal_grau

Description
pal_grau provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Grau or grey).

Usage
pal_grau

Format
An object of class data.frame with 1 rows and 5 columns.

Details
See Also

pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_unikn_pref for a unikn color palette with all preferred colors; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

    pal_grau
    dim(pal_grau)  # 1 5
    pal_grau[3]    # preferred (named) color "grau3"
    pal_grau[[3]] # preferred color "grau3" OR "#9AA0A7"

# Plotting palette:
seecol(pal_grau)

pal_karpfenblau
uni.kn color palette karpfenblau

Description

pal_karpfenblau provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Karpfenblau or blue carp).

Usage

pal_karpfenblau

Format

An object of class data.frame with 1 rows and 5 columns.

Details


See Also

pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_seeblau for the default seeblau uni.kn color palette; pal_unikn_pref for a uni.kn color palette with all preferred colors; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn
Examples

```r
pal_karpfenblau
  dim(pal_karpfenblau)  # 1 5
  pal_karpfenblau[4]   # preferred (named) color "karpfenblau4"
  pal_karpfenblau[[4]] # preferred color "karpfenblau4" OR "#3E5496"

# Plotting palette:
  seecol(pal_karpfenblau)
```

---

**pal_peach**

**uni.kn color palette peach**

Description

`pal_peach` provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Peach).

Usage

`pal_peach`

Format

An object of class `data.frame` with 1 rows and 5 columns.

Details


See Also

`pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_pinky` and `pal_bordeaux` for alternative redish uni.kn color palettes; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other color palettes: `pal_bordeaux, pal_grau, pal_karpfenblau, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn`

Examples

```r
pal_peach
  dim(pal_peach)  # 1 5
  pal_peach[4]    # preferred (named) color "peach4"
  pal_peach[[4]]  # preferred color "peach4" OR "#FEA090"
```
pal_petrol

# Plotting palette:
seecol(pal_peach)

---

| pal_petrol | uni.kn color palette petrol |

Description

pal_petrol provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Petrol or grue).

Usage

pal_petrol

Format

An object of class data.frame with 1 rows and 5 columns.

Details


See Also

pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_seegruen for an alternative green/grue uni.kn color palette; pal_unikn_pref for a uni.kn color palette with all preferred colors; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

pal_petrol
dim(pal_petrol) # 1 5
pal_petrol[4] # preferred (named) color "petrol4"
pal_petrol[[4]] # preferred color "petrol4" OR "#077187"

# Plotting palette:
seecol(pal_petrol)
pal_pinky

----

pal_pinky  uni.kn color palette pinky

Description

pal_pinky provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Pinky or pink).

Usage

pal_pinky

Format

An object of class data.frame with 1 rows and 5 columns.

Details


See Also

pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_peach and pal_bordeaux for alternative redish uni.kn color palettes; pal_unikn_pref for a uni.kn color palette with all preferred colors; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

pal_pinky
  dim(pal_pinky)  # 1 5
  pal_pinky[4]  # preferred (named) color "pinky4"
  pal_pinky[[4]]  # preferred color "pinky4" OR "#E0607E"

  # Plotting palette:
  seecol(pal_pinky)
Description

pal_seeblau provides an additional uni.kn color palette as a data frame containing 5 colors (shades of Seeblau).

Usage

pal_seeblau

Format

An object of class data.frame with 1 rows and 5 columns.

Details


See Also

pal_unikn for the unikn default color palette with all 5 colors of pal_seeblau; pal_karpfenblau for an alternative blue uni.kn color palette; pal_unikn_pref for a uni.kn color palette with all preferred colors; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

pal_seeblau
dim(pal_seeblau) # 1 5

# Preferred color:
pal_seeblau[3] # preferred (named) color "seeblau3" (as df)
pal_seeblau[[3]] # preferred color value "#59C7EB"

# Access by position:
pal_seeblau[3] # named color "seeblau3" (as df)
pal_seeblau[[3]] # color value "#59C7EB"

# Access by name:
pal_unikn["seeblau3"] # color "seeblau3" (as df)
pal_unikn[["seeblau3"]]) # color value "#59C7EB"

# Plotting palette:
seecol(pal_seeblau)
Description

`pal_seegruen` provides an additional uni.kn color palette as a data frame containing 5 colors (shades of `Seegruen`).

Usage

`pal_seegruen`

Format

An object of class `data.frame` with 1 rows and 5 columns.

Details


See Also

`pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_petrol` for an alternative green uni.kn color palette; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other color palettes: `pal_bordeaux`, `pal_grau`, `pal_karpfenblau`, `pal_peach`, `pal_petrol`, `pal_pinky`, `pal_seeblau`, `pal_signal`, `pal_unikn_dark`, `pal_unikn_light`, `pal_unikn_pair`, `pal_unikn_ppt`, `pal_unikn_pref`, `pal_unikn_web`, `pal_unikn`

Examples

```r
pal_seegruen
dim(pal_seegruen)  # 1 5
pal_seegruen[4]    # preferred (named) color "seegruen4"
pal_seegruen[[4]]  # preferred color "seegruen4" OR "#0A9086"
```

# Plotting palette:
`seecol(pal_seegruen)`
Description

`pal_signal` provides an additional uni.kn color palette as a data frame containing 3 colors (Ampel or traffic signal colors).

Usage

`pal_signal`

Format

An object of class `data.frame` with 1 rows and 3 columns.

Details

The colors are arranged as in a traffic light ("Ampel"):

1. top: red or "bad"
2. mid: yellow or "alert"
3. bot: green or "good"


See Also

`pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other color palettes: `pal_bordeaux`, `pal_grau`, `pal_karpfenblau`, `pal_peach`, `pal_petrol`, `pal_pinky`, `pal_seeblau`, `pal_seegruen`, `pal_unikn_dark`, `pal_unikn_light`, `pal_unikn_pair`, `pal_unikn_ppt`, `pal_unikn_pref`, `pal_unikn_web`, `pal_unikn`

Examples

```r
pal_signal
dim(pal_signal) # 1 3
pal_signal[2]  # (named) color "signal2"
pal_signal[[2]] # color "signal2" OR "#EFDC60"

# Plotting palette:
seecol(pal_signal)
```
Description

pal_unikn combines the 5 shades of blue colors from color palette pal_seeblau with the 6 non-blue colors of pal_unikn_web to a divergent palette of 11 colors.

Usage

pal_unikn

Format

An object of class data.frame with 1 rows and 11 columns.

Details

Adding seeblau5 (i.e., pal_seeblau[1]) to the default color palette pal_unikn also puts white at the central (middle) position of a color palette with 11 values:
pal_unikn[[6]] is white or "#FFFFFF".
A divergent palette is useful for creating color gradients.

See Also

pal_unikn for the default uni.kn color palette; pal_seeblau for the uni.kn seeblau color palette; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web

Examples

pal_unikn
dim(pal_unikn) # 1 11

# Access by position:
pal_unikn[1] # new color "seeblau5" (as df)
pal_unikn[[1]] # new color value "#008ECE"

# Access by name:
pal_unikn["seeblau5"] # new color "seeblau5" (as df)
pal_unikn["seeblau5"] # new color value "#008ECE"

# Viewing/using color palette:
pal_unikn_dark

seecol(pal_unikn)
demopal(pal_unikn, type = "curve", main = "Default colors of Konstanz University")

# Note:
pal_unikn[6] # "white" or "#FFFFFF" as central of 11 colors

pal_unikn_dark

uni.kn color palette of dark colors (10 colors)

Description

pal_unikn_dark provides an additional uni.kn color palette that collects 2 dark colors of 5 color palettes as a data frame containing 10 colors (in 5 pairs).

Usage

pal_unikn_dark

Format

An object of class data.frame with 1 rows and 10 columns.

Details


See Also

pal_unikn_light for a lighter uni.kn color palette; pal_unikn_pair for a pairwise uni.kn color palette; pal_unikn for the default uni.kn color palette; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblaublau, pal_seegruen, pal_signal, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

pal_unikn_dark
dim(pal_unikn_dark) # 1 8
pal_unikn_dark[1] # color "karpfenblau5" by position
pal_unikn_dark[[1]] # color value by position: #324376
pal_unikn_dark["karpfenblau5"] # color value by name

# Viewing/using color palette:
seecol(pal_unikn_dark)
demopal(pal_unikn_dark, type = "points", main = "Dark colors of Konstanz University")
pal_unikn_light

uni.kn color palette of light colors (10 colors)

Description

pal_unikn_light provides an additional uni.kn color palette that collects 2 light colors of 5 color palettes as a data frame containing 10 colors (in 5 pairs).

Usage

pal_unikn_light

Format

An object of class data.frame with 1 rows and 10 columns.

Details


See Also

pal_unikn_dark for a darker uni.kn color palette; pal_unikn_pair for a pairwise uni.kn color palette; pal_unikn for the default uni.kn color palette; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_pair, pal_unikn_ppt, pal_unikn_pref, pal_unikn_web, pal_unikn

Examples

pal_unikn_light
dim(pal_unikn_light) # 1 10

# Access by position:
pal_unikn_light[1] # color "seeblau3" (as df)
pal_unikn_light[[1]] # color value "#59C7EB"

# Access by name:
pal_unikn_light["seeblau3"] # color "seeblau3" (as df)
pal_unikn_light[["seeblau3"]] # color value "#59C7EB"

# Viewing/using color palette:
seecol(pal_unikn_light)
demopal(pal_unikn_light, type = "bar", main = "Light colors of Konstanz University")
pal_unikn_pair

| pal_unikn_pair | uni.kn color palette of pairwise colors (16 colors) |

Description

pal_unikn_pair provides an additional uni.kn color palette that collects 16 paired colors of 8 color palettes as a data frame containing 16 colors (in 8 pairs).

Usage

pal_unikn_pair

Format

An object of class data.frame with 1 rows and 16 columns.

Details


See Also

pal_unikn_light for a lighter uni.kn color palette; pal_unikn_dark for a darker uni.kn color palette; pal_unikn for the default uni.kn color palette; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegruen, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_ppt, pal_unikn pref, pal_unikn_web, pal_unikn

Examples

```r
pal_unikn_pair
dim(pal_unikn_pair) # 1 16
tp_unikn_pair[1] # color "karpfenblau4" by position
tp_unikn_pair[[1]] # color value by position: #3E5496
pal_unikn_pair["karpfenblau4"] # color value by name

# Viewing/using color palette:
seecol(pal_unikn_pair)
demopal(pal_unikn_pair, type = "polygon", main = "A pair-wise color palette")
```
### Description

`pal_unikn_ppt` provides an alternative uni.kn color palette as a data frame containing 10 colors.

### Usage

`pal_unikn_ppt`

### Format

An object of class `data.frame` with 1 rows and 10 columns.

### Details

This is a secondary (ppt) variant with more muted colors.


### See Also

- `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.
- Other color palettes: `pal_bordeaux`, `pal_grau`, `pal_karpfenblau`, `pal_peach`, `pal_petrol`, `pal_pinky`, `pal_seeblau`, `pal_seegruen`, `pal_signal`, `pal_unikn_dark`, `pal_unikn_light`, `pal_unikn_pair`, `pal_unikn_pref`, `pal_unikn_web`, `pal_unikn`

### Examples

```r
pal_unikn_ppt
dim(pal_unikn_ppt)  # 1 10

# Access by position:
pal_unikn_ppt[2]  # 2nd named color "seeblau3" (as df)
pal_unikn_ppt[[2]] # 2nd color value "#59B6DC"

# Access by name:
pal_unikn_ppt["seeblau3"] # color "seeblau3" (as df)
pal_unikn_ppt[["seeblau3"]][] # color value "#59B6DC"

# Plotting palette:
seecol(pal_unikn_ppt)
```
Description

pal_unikn_pref provides an additional uni.kn color palette that collects the preferred color of each palette as a data frame containing 9 (or 8 + 1) colors.

Usage

pal_unikn_pref

Format

An object of class data.frame with 1 rows and 9 columns.

Details

The colors are arranged in a sequence that provides high contrasts between adjacent colors.

Note that the (alert) color Signal is not a preferred color according to the official color definition.


See Also

pal_unikn for the default uni.kn color palette; seecol for viewing and comparing color palettes; usecol for using color palettes.

Other color palettes: pal_bordeaux, pal_grau, pal_karpfenblau, pal_peach, pal_petrol, pal_pinky, pal_seeblau, pal_seegrue, pal_signal, pal_unikn_dark, pal_unikn_light, pal_unikn_pair, pal_unikn_ppt, pal_unikn_web, pal_unikn

Examples

pal_unikn_pref
dim(pal_unikn_pref) # 1 9

# Access by position:
pal_unikn_pref[1] # color Seeblau (as df)
pal_unikn_pref[[1]] # color value "#59C7EB"

# Access by name:
pal_unikn_pref["Seeblau"] # color "seeblau3" (as df)
pal_unikn_pref[["Seeblau"]]] # color value "#59C7EB"

# Viewing/using color palette:
seecol(pal_unikn_pref)
demopal(pal_unikn_pref, type = "mosaic", main = "Preferred colors of Konstanz University")
Description

dal_unikn_web provides the default uni.kn color palette as a data frame containing 10 colors.

Usage

dal_unikn_web

Format

An object of class data.frame with 1 rows and 10 columns.

Details

This is the primary (web/sRGB) scale.
Note that dal_unikn provides a divergent color palette (of 11 colors).

See Also

dal_unikn for the unikn default color palette with all 5 colors of dal_seeblau; dal_unikn_ppt
for an alternative (ppt) version; dal_unikn_pref for a unikn color palette with all preferred colors;
dalcol for viewing and comparing color palettes; dalcol for using color palettes.

Other color palettes: dal_bordeaux, dal_grau, dal_karpfenblau, dal_peach, dal_petrol, dal_pinsky,
dal_seebblau, dal_seegruen, dal_signal, dal_unikn_dark, dal_unikn_light, dal_unikn_pair,
dal_unikn_ppt, dal_unikn_pref, dal_unikn

Examples

dal_unikn_web

dim(dal_unikn_web)  # 1 10

# Access by position:
dal_unikn_web[2]    # 2nd named color "seeblau3" (as df)
dal_unikn_web[[2]]  # 2nd color value "#59C7EB"

# Access by name:
dal_unikn_web["seeblau3"]  # color "seeblau3" (as df)
dal_unikn_web["seeblau3"]  # color value "#59C7EB"

# Plotting palette:
dalcol(dal_unikn_web)
Peach

uni.kn color Peach

Description

Peach provides the preferred color of `pal_peach` (as an atomic HEX character value) and is defined as `pal_peach[[4]]`.

Usage

Peach

Format

An object of class `character` of length 1.

Details


See Also

`pal_peach` for the corresponding color palette; `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Karpfenblau, Petrol, Pinky, Seeblau, See gruen, Signal

Examples

Peach  # HEX character "#FEA090" (as value)
all.equal(Peach, pal_peach[[4]])  # TRUE (same HEX values)
seecol(Peach)  # view color and details

Petrol

uni.kn color Petrol

Description

Petrol provides the preferred color of `pal_petrol` (as an atomic HEX character value) and is defined as `pal_petrol[[4]]`.

Usage

Petrol
Pinky

Format

An object of class character of length 1.

Details


See Also

- `pal_petrol` for the corresponding color palette;
- `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`;
- `pal_unikn_pref` for a unikn color palette with all preferred colors;
- `seecol` for viewing and comparing color palettes;
- `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Karpfenblau, Peach, Pinky, Seeblau, Seegruen, Signal

Examples

```r
Petrol # HEX character "#077187" (as value)
all.equal(Petrol, pal_petrol[[4]]) # TRUE (same HEX values)

seecol(Petrol) # view color and details
```

---

Pinky

*uni.kn color Pinky*

Description

Pinky provides the preferred color of `pal_pinky` (as an atomic HEX character value) and is defined as `pal_pinky[[4]]`.

Usage

Pinky

Format

An object of class character of length 1.

Details

See Also

*pal_pinky* for the corresponding color palette; *pal_unikn* for the unikn default color palette with all 5 colors of *pal_seeblau*; *pal_unikn_pref* for a unikn color palette with all preferred colors; *seecol* for viewing and comparing color palettes; *usecol* for using color palettes.

Other preferred colors: *Bordeaux*, *Grau*, *Karpfenblau*, *Peach*, *Petrol*, *Seeblau*, *Seegruen*, *Signal*

Examples

```r
Pinky # HEX character "#E0607E" (as value)
all.equal(Pinky, pal_pinky[4]) # TRUE (same HEX values)
seecol(Pinky) # view color and details
```

---

**post**

*Post text (in an xbox)*

Description

`post` plots 1 or more text strings (provided as a character vector *labels*) to an (existing or new) *xbox*.

Usage

```r
post(
  labels,
  x = 0.03,
  y = 0.55,
  x_layout = NA,
  y_layout = "even",
  col = "white",
  col_bg = Seeblau,
  cex = 1,
  font = 1,
  new_plot = "none"
)
```

Arguments

- **labels**
  A character vector specifying the text labels to be plotted.
- **x**
  A numeric vector of x-coordinates at which the text labels in *labels* should be written. If the lengths of *x* and *y* differ, the shorter one is recycled. Default: *x* = 0.03.
- **y**
  A numeric vector of y-coordinates at which the text labels in *labels* should be written. If the lengths of *x* and *y* differ, the shorter one is recycled. Default: *y* = 0.55.
x_layout An optional numeric vector or character string to control the horizontal positions of labels. Numeric values are interpreted as increments to values of x and recycled (to enable stepwise or alternating patterns). 3 character string options are: "center" (i.e., center wrt. first label or plot center), "left" (i.e., left wrt. first label or plot center), "right" (i.e., right wrt. first label or plot center). Default: x_layout = NA (i.e., using values of x).

y_layout A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of y[1] and recycled (to enable stepwise or alternating patterns). 2 character string options are: "even" (i.e., even distribution of labels across available y-space) and "flush" (i.e., no space between adjacent labels, i.e., y_layout = 0). Default: y_layout = "even".

col The color(s) of the text label(s). Default: col_lbl = "white".

col_bg The background color(s) of the xbox. Default: col_bg = Seeblau.

cex Numeric character expansion factor(s), multiplied by par("cex") to yield the character size(s). Default: cex = 1.0.

font The font type(s) to be used. Default: font = 1 (i.e., plain text).

new_plot Should a new plot be generated? Set to "xbox" to plot to a basic xbox (with square dimensions, i.e., dim = c(1, 1)). Default: new_plot = "none" (i.e., assumes a pre-existing xbox).

Details

The positions of the text elements in labels can be specified by providing their coordinates (as x and y arguments) or by providing an initial position and an y_layout (see below). Text formatting parameters (like col, col_bg, cex, font) are recycled to match length(labels).

post uses the base graphics system graphics::.

See Also

xbox to create a new xbox (without text).

Other text functions: mark(), uline(), url_unikn()

Examples

post(labels = "Post this line with default settings.", new_plot = "xbox")

# Create a new xbox:
post(labels = "This is a test.", new_plot = "xbox",
     cex = 1.2, font = 2, col_bg = pal_seeblau[5])

# Add text to an existing xbox:
post(labels = c("More text follows here,",
             "yet another line here,",
             "and even more here.")
     y = .4, y_layout = .04,
     new_plot = "none")
# Using x_layout and y_layout:
post(labels = c("Ene," ,"mene, miste.", "es rappelt", "in der Kiste."),
cex = 1.4, font = 2, col = "white", col_bg = Pinky,
  x = .1, y = .5, x_layout = "left", y_layout = .05, new_plot = "xbox")

post(labels = c("Hello world!", "Does this work?", "That\'s good!", "Please carry on..."),
cex = 1.4, font = 2, col = "white", col_bg = Karpfenblau,
  x = .01, y = .6, x_layout = .10, y_layout = .05, new_plot = "xbox")

---

Seeblau   uni.kn color Seeblau

**Description**

Seeblau provides the preferred color of `pal_seeblau` (as an atomic HEX character value) and is defined as `pal_seeblau[[3]]`.

**Usage**

Seeblau

**Format**

An object of class character of length 1.

**Details**


**See Also**

`pal_seeblau` for the corresponding color palette; `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Karpfenblau, Peach, Petrol, Pinky, Seegruen, Signal

**Examples**

```
Seeblau # HEX character "#59C7EB" (as value)
all.equal(Seeblau, pal_seeblau[[3]]) # TRUE (same HEX values)
seecol(Seeblau) # view color and details
```
seecol provides an interface to plotting (or "seeing") the colors of a palette or comparing multiple color palettes.

Usage

```r
seecol(
  pal = "unikn_all",
  n = "all",
  alpha = NA,
  hex = NULL,
  rgb = NULL,
  col_bg = NULL,
  col_brd = NULL,
  lwd_brd = NULL,
  grid = TRUE,
  main = NA,
  sub = NULL,
  title = NULL,
  mar_note = NA,
  pal_names = NA,
  ...
)
```

Arguments

- **pal**: A single color palette (as a vector of colors), multiple color palettes (as a list), or a recognized keyword (as a character string). Default: `pal = "unikn_all"` (i.e., plot all color palettes provided by the `unikn` package).

  Recognized keywords are:
  1. "all": All color palettes of the `unikn` package.
  2. "all_unikn" or "unikn_all": All uni.kn color palettes (of the University of Konstanz).
  3. "unikn_basic": All basic uni.kn palettes.
  4. "grad_all": All uni.kn palettes with color gradients.
  5. "pair_all": All uni.kn palettes with pairwise colors.
  6. "pref_all": All preferred uni.kn colors and their gradients.
  7. "add": Additional/contributed color palettes (migrated to the `unicol` package).

  seecol does also recognize keywords (e.g., "all_unikn") or keywords without "unikn" (e.g., "basic").
Numbers to show or use. If \( n \) is lower or higher than the length of the current color palette \( \text{pal} \), the color palette is reduced or extrapolated (using \texttt{grDevices::colorRampPalette}). Default: \( n = "\text{all}" \) (i.e., show all colors in palette).

- **alpha**: A factor modifying the opacity \( \alpha \) (as \( \alpha.f \) in \texttt{adjustcolor}) to a value in \([0, 1]\). Default: \( \alpha = \text{NA} \) (i.e., no modification of opacity).

- **hex**: Should HEX color values be shown? Default: \( \text{hex} = \text{NULL} \) (i.e., show HEX color values when there is sufficient space to print them).

- **rgb**: Should RGB color values be shown? Default: \( \text{rgb} = \text{NULL} \) (i.e., show RGB color values when there is sufficient space to print them).

- **col\_bg**: Color of plot background. Default: \( \text{col\_bg} = \text{NULL} \).

- **col\_brd**: Color of shape borders (if shown). Default: \( \text{col\_brd} = \text{NULL} \).

- **lwd\_brd**: Line width of shape borders (if shown). Default: \( \text{lwd\_brd} = \text{NULL} \).

- **grid**: Show grid in the color plot? Default: \( \text{grid} = \text{TRUE} \).

- **main**: Main plot title (as a character string). Default: \( \text{main} = \text{NA} \) creates a default title.

- **sub**: Optional subtitle (as a character string). Default: \( \text{sub} = \text{NULL} \) (i.e., no subtitle).

- **title**: Deprecated plot title. Use \( \text{main} \) instead.

- **mar\_note**: Optional margin note (on bottom right). Default: \( \text{mar\_note} = \text{NA} \) (i.e., no margin note).

- **pal\_names**: Names of color palettes or colors (as a character vector). Default: \( \text{pal\_names} = \text{NA} \) (for default names).

- **...**: Other graphical parameters (passed to \texttt{plot}).

### Details

\texttt{seeicol} has two main modes, based on the contents of its \texttt{pal} argument:

1. if \( \text{pal} \) is set to a \textit{specific} color palette (or a vector of multiple colors or color palettes):
   
   Plot the current color palette and optional details on its colors.

2. if \( \text{pal} = \text{"unikn\_all"} \) or a list of \textit{multiple} color palettes:
   
   Plot visual vectors of all current color palettes for comparing them.

Specifying \texttt{distinct = TRUE} removes visual duplicate colors (based on HEX values, ignoring transparency), but only when showing an individual color palette \( \text{pal} \).

Various title options (i.e., \texttt{main}, \texttt{sub}, and \texttt{mar\_note}) and a \texttt{pal\_names} argument add control over plotted text labels. However, the length of a character vector provided to \texttt{pal\_names} must correspond to the number of (custom) color palettes or colors.

### See Also

\texttt{useicol} for using color palettes; \texttt{simcol} for finding similar colors; \texttt{newpal} for defining new color palettes; \texttt{grepal} for finding named colors; \texttt{shades\_of} to defining shades of a given color; \texttt{ac} for adjusting color transparency; \texttt{pal\_unikn} for the default \texttt{uni\_kn} color palette.

Other color functions: \texttt{ac()}, \texttt{demopal()}, \texttt{grepal()}, \texttt{newpal()}, \texttt{shades\_of()}, \texttt{simcol()}, \texttt{useicol()}

Examples

# See multiple color palettes:
seecol()  # default: seecol(pal = "all")

# See details of one color palette:
seecol(pal_unikn)  # see a specific color palette

# Combining colors or color palettes:
seecol(c(rev(pal_seeblau), pal_seegruen))  # combine color palettes
seecol(c(rev(pal_seeblau), "white", pal_pinky))  # combine color palettes and color names
seecol(c("black", "firebrick", "gold"))  # combine color names

# Using n to reduce or extend color palettes:
seecol(n = 3)  # viewing reduced ranges of all palettes
seecol(n = 12)  # viewing extended ranges of all palettes
seecol(pal_unikn, n = 5,
       main = "Reduced version of pal_unikn (n = 5)")  # reducing pal_unikn
seecol(pal_seeblau, n = 8,
       main = "Extended version of pal_seeblau (n = 8)")  # extending pal_seeblau

# Combining and extending color palettes:
seecol(c(rev(pal_seeblau), "white", pal_bordeaux), n = 17,
       main = "Diverging custom color palette (with 17 colors)")

# Defining custom color palettes:
pal_mpg <- c("#007367", "white", "#D0D3D4")  # mixing hex values and color names
names(pal_mpg) <- c("mpg green", "mpg white", "mpg grey")  # color names

pal_bdg <- usecol(c(Bordeaux, "gold"), n = 10)  # using usecol

# Viewing extended color palette:
seecol(pal_mpg, n = 9, main = "Custom color palette of the Max Planck Society")

# Comparing (and labeling) custom color palettes:
seecol(list(pal_mpg, pal_bdg, pal_unikn), n = 7,
       pal_names = c("Max Planck", "Bordeaux-Gold", "Uni Konstanz"),
       main = "Comparing and labeling custom color palettes")

## Viewing color palettes from other packages:
## library(RColorBrewer)
## seecol(brewer.pal(name = "RdBu", n = 11))  # viewing "RdBu" palette from RColorBrewer

## Extending color palettes:
## seecol(brewer.pal(name = "RdBu", n = 11), n = 15)  # extending palette to 15 colors
Description

Seegruen provides the preferred color of `pal_seegruen` (as an atomic HEX character value) and is defined as `pal_seegruen[[4]]`.

Usage

Seegruen

Format

An object of class character of length 1.

Details


See Also

`pal_seegruen` for the corresponding color palette; `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Karpfenblau, Peach, Petrol, Pinky, Seeblau, Signal

Examples

Seegruen # HEX character "#0A9086" (as value)
all.equal(Seegruen, pal_seegruen[[4]]) # TRUE (same HEX values)

seecol(Seegruen) # view color and details

---

**shades_of**

*Get n shades of a color*

Description

`shades_of` returns a vector of `n` colors that are shades of a color gradient ranging from an initial color `col_1` to a final color `col_n`.

Usage

`shades_of(n = 5, col_1 = "black", col_n = "white", alpha = NA)`
Arguments

\begin{itemize}
  \item \textbf{n} \hspace{1cm} Number of desired colors. Default: \texttt{n = 5.}
  \item \textbf{col\_1} \hspace{1cm} Initial color. Default: \texttt{col\_1 = "black".}
  \item \textbf{col\_n} \hspace{1cm} Final (n-th) color. Default: \texttt{col\_n = "white".}
  \item \textbf{alpha} \hspace{1cm} A factor modifying the opacity alpha (as \texttt{alpha.f} in \texttt{adjustcolor}) to a value in [0, 1]. Default: \texttt{alpha = NA} (i.e., no modification of opacity).
\end{itemize}

Details

By default, the color gradient returned contains \texttt{n = 5} colors that range from the initial color \texttt{col\_1 = "black"} to the final color \texttt{col\_n = "white"}. Specifying different values for \texttt{n} and the initial or final colors yields different color ranges.

\texttt{shades\_of} is mostly a wrapper for a special \texttt{usecol} command. However, \texttt{usecol} allows defining more complex color gradients (e.g., by specifying more than two colors).

See Also

\texttt{seecol} for viewing and comparing color palettes; \texttt{usecol} for using color palettes; \texttt{simcol} for finding similar colors; \texttt{newpal} for defining new color palettes; \texttt{grepal} for finding named colors; \texttt{ac} for adjusting color transparency.

Other color functions: \texttt{ac()}, \texttt{demopal()}, \texttt{grepal()}, \texttt{newpal()}, \texttt{seecol()}, \texttt{simcol()}, \texttt{usecol()}

Examples

\begin{verbatim}
grey50 <- shades_of(50, col_1 = 'grey1')
seecol(grey50, main = "50 shades of grey1")

blue_black <- shades_of(5, Seebblau, col_n = "black")
seecol(blue_black, main = "5 shades from Seebblau to black")

wine_white <- shades_of(6, Bordeaux, alpha = 1/2)
seecol(wine_white, col_br = "black", lwd_br = .5,
       main = "Shades of semi-transparent Bordeaux")
\end{verbatim}

Description

Signal provides the alert color of \texttt{pal\_signal} (as an atomic HEX character value) and is defined as \texttt{pal\_signal[2]}.

Usage

Signal
*simcol*

**Format**

An object of class character of length 1.

**Details**

The official specification of `pal_signal` does not identify a preferred color. We provide Signal as a dedicated color as it is suited for creating color gradients (see `usecol`).


**See Also**

`pal_signal` for the corresponding color palette; `pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` for viewing and comparing color palettes; `usecol` for using color palettes.

Other preferred colors: Bordeaux, Grau, Karpfenblau, Peach, Petrol, Pinky, Seeblau, Seegruen

**Examples**

```r
Signal  # HEX character "#EFDC60" (as value)
all.equal(Signal, pal_signal[[2]])  # TRUE (same HEX values)
seecol(Signal)  # view color and details
```

---

**simcol**

*Find similar colors*

**Description**

`simcol` finds and shows colors from a palette of color candidates `col_candidates` that are similar to some target color `col_target`.

**Usage**

```r
simcol(
  col_target,
  col_candidates = colors(),
  tol = c(25, 50, 75),
  distinct = TRUE,
  plot = TRUE
)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>col_target</td>
<td>A (required) target color.</td>
</tr>
<tr>
<td>col_candidates</td>
<td>A palette of color candidates to be considered. Default: \texttt{col_candidates = colors().}</td>
</tr>
<tr>
<td>tol</td>
<td>Numeric tolerance value(s) (either 1 or 3 numeric values, in the RGB range</td>
</tr>
<tr>
<td></td>
<td>from 0 to 255). Values are considered in the order of the RGB value rank in</td>
</tr>
<tr>
<td></td>
<td>\texttt{col_target}. Default: \texttt{tol = c(25, 50, 75)}.</td>
</tr>
<tr>
<td>distinct</td>
<td>Boolean: Return only visually distinct colors? Default: \texttt{distinct = TRUE} (i.e., remove visual duplicates).</td>
</tr>
<tr>
<td>plot</td>
<td>Boolean: Plot the output (using \texttt{seecol})? Default: \texttt{plot = TRUE}.</td>
</tr>
</tbody>
</table>

Details

\texttt{simcol} returns a vector of the (named) colors or color values in \texttt{col_candidates} (set to \texttt{colors()} of \texttt{grDevices} per default) that are similar to the specified target color \texttt{col_target}.

If \texttt{plot = TRUE}, \texttt{simcol} also visualizes the detected colors (by passing its result to \texttt{seecol}, as a side-effect).

Color similarity is defined in terms of the distance between colors' RGB values, which must be within the numeric tolerance threshold(s) specified by \texttt{tol} (with \(0 \leq \texttt{tol} \leq 255\)). Higher \texttt{tol} values correspond to more permissive similarity judgments.

If \texttt{tol} is a scalar, the values of all three RGB dimensions of \texttt{col_candidates} must be within the corresponding values of \texttt{col_target} to be judged as 'similar'. If \texttt{tol} contains three values, the three RGB dimension are compared in order of the dimensions’ rank in \texttt{col_target} (i.e., the primary dimension must be within \texttt{tol[1]}, etc.). Thus, providing three \texttt{tol} values allows for more fine-grained similarity matching.

Value

A named vector of colors or color values.

See Also

\texttt{seecol} for plotting/seeing color palettes; \texttt{usecol} for using color palettes; \texttt{newpal} for defining new color palettes; \texttt{grepal} for finding named colors; \texttt{shades_of} to defining shades of a given color; \texttt{ac} for adjusting color transparency.

Other color functions: \texttt{ac()}, \texttt{demopal()}, \texttt{grepal()}, \texttt{newpal()}, \texttt{seecol()}, \texttt{shades_of()}, \texttt{usecol()}.

Examples

# Basic uses:
\texttt{simcol(col_target = "red")}
\texttt{simcol("tan", tol = 15)}
\texttt{simcol("Seeblau", tol = c(20, 30, 40))}
\texttt{simcol("blue", col_candidates = pal_unikn_pred, tol = 120)}

# Fine-tuning the range of color matching:
\texttt{simcol("Seeblau", tol = 30)} # = \texttt{simcol("Seeblau", tol = c(30, 30, 30))}
**slide**

simcol(Seeblau, tol = c(20, 20, 80))

# Increasing tolerance widens range:
simcol("grey", c("white", "grey", "black"), tol = 255, distinct = FALSE, plot = FALSE)

---

**Description**

`s`lide plots an empty slide (or frame) as a colored rectangle.

**Usage**

```r
slide(col = NA, dim = c(4/3, 1), border = grey(0.33, 1), lwd = 1.5)
```

**Arguments**

- `col` The color to fill the slide with (i.e., its background color). Default: `col = NA` (i.e., system default for transparency).
- `dim` The x- and y-dimensions of the slide. Default: `dim = c(4/3, 1)` (i.e., unit height, 4/3 wider than high).
- `border` The color of the slide’s border. Setting `border = NA` hides border. Default: `border = grey(0.33, 1)`.
- `lwd` The line width of the slide’s border. Setting `lwd = 0` or `lwd = NA` removes border. Default: `lwd = 1.5`.

**See Also**

- `heading`, `line`, or `mark` to add text to a slide; `xbox` to plot a box.

Other plot functions: `theme_grau()`, `theme_unikn()`, `xbox()`

**Examples**

```r
slide() # default slide (or frame)
slide(lwd = NA) # borderless slide

# Dimensions:
slide(dim = c(18, 9)) # larger and 2:1 dimensions
slide(dim = c(1/3, 1)) # smaller and 1:3 dimensions

# Formatting:
slide(col = pal_seeblau[[1]], border = pal_seeblau[[5]], lwd = 2)
```
theme_grau

Alternative theme (for ggplot2)

Description

theme_grau provides an alternative unikn theme to use in ggplot2 commands.

Usage

theme_grau(
  col_title = grey(0, 1),
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)

Arguments

- **col_title**: Color of title (text) elements (optional, numeric). Default: col_title = grey(0, 1) (i.e., "black"). Consider using col_title = unikn::pal_seeblau[[4]].
- **base_size**: Base font size (optional, numeric). Default: base_size = 11.
- **base_family**: Base font family (optional, character). Default: base_family = "". Options include "mono", "sans" (default), and "serif".
- **base_line_size**: Base line size (optional, numeric). Default: base_line_size = base_size/22.
- **base_rect_size**: Base rectangle size (optional, numeric). Default: base_rect_size = base_size/22.

Details

theme_grau is no-nonsense, but fills panel backgrounds in "grau" (specifically, pal_seeggrau[1]). This theme works well for dark colors and bright color accents, but is of limited use with transparent colors.

See Also

- theme_unikn for default theme.

Other plot functions: slide(), theme_unikn(), xbox()

Examples

# Plotting iris dataset (using ggplot2, theme_grau, and unikn colors):

library('ggplot2') # theme_unikn requires ggplot2
theme_unikn

```r
ggplot(datasets::iris) +
  geom_jitter(aes(x = Sepal.Length, y = Sepal.Width, color = Species), size = 3, alpha = 2/3) +
  facet_wrap(~Species) +
  scale_color_manual(values = usecol(pal = c(Pinky, Seeblau, Seegruen))) +
  labs(tag = "B",
       title = "Iris sepals",
       caption = "Data from datasets::iris") +
  coord_fixed(ratio = 3/2) +
  theme_grau()
```

---

**theme_unikn**

Basic unikn theme (for ggplot2)

Description

theme_unikn provides a basic unikn theme to use in ggplot2 commands.

Usage

```r
theme_unikn(
  col_title = pal_seeblau[[4]],
  base_size = 11,
  base_family = ",",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Arguments

- **base_size**: Base font size (optional, numeric). Default: `base_size = 11`.
- **base_family**: Base font family (optional, character). Default: `base_family = ""`. Options include "mono", "sans" (default), and "serif".
- **base_line_size**: Base line size (optional, numeric). Default: `base_line_size = base_size/22`.
- **base_rect_size**: Base rectangle size (optional, numeric). Default: `base_rect_size = base_size/22`.

Details

The theme is lightweight and no-nonsense, but somewhat opinionated (e.g., in using mostly grey scales to allow emphasizing data points with color accents).
See Also

theme_grau for an alternative theme.

Other plot functions: slide(), theme_grau(), xbox()

Examples

# Plotting iris dataset (using ggplot2, theme_unikn, and unikn colors):
library('ggplot2') # theme_unikn requires ggplot2

ggplot(datasets::iris) +
geom_jitter(aes(x = Petal.Length, y = Petal.Width, color = Species), size = 3, alpha = 2/3) +
scale_color_manual(values = usecol(pal = c(Pinky, Seeblau, Seegruen))) +
labs(tag = "A", title = "Iris petals", caption = "Data from datasets::iris") +
theme_unikn()

uline

Plot underlined text elements

Description

uline plots 1 or more text strings (provided as a character vector labels) to an (existing or new) plot and places a colored line underneath each label (to underline it).

Usage

uline(
labels,
x = 0,
y = 0.55,
x_layout = NA,
y_layout = "even",
col = "black",
col_bg = Seeblau,
cex = 1.5,
font = 1,
new_plot = "none"
)
Arguments

labels A character vector specifying the text labels to be plotted.

x A numeric vector of x-coordinates at which the text labels in labels should be written. If the lengths of x and y differ, the shorter one is recycled. Default: x = 0.

y A numeric vector of y-coordinates at which the text labels in labels should be written. If the lengths of x and y differ, the shorter one is recycled. Default: y = .55.

x_layout An optional numeric vector or character string to control the horizontal positions of labels. Numeric values are interpreted as increments to values of x and recycled (to enable stepwise or alternating patterns). 3 character string options are: "center" (i.e., center wrt. first label or plot center), "left" (i.e., left wrt. first label or plot center), "right" (i.e., right wrt. first label or plot center). Default: x_layout = NA (i.e., using values of x).

y_layout A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of y[1] and recycled (to enable stepwise or alternating patterns). 2 character string options are: "even" (i.e., even distribution of labels across available y-space) and "flush" (i.e., no space between adjacent labels, i.e., y_layout = 0). Default: y_layout = "even".

col The color(s) of the text label(s). Default: col_lbl = "black".

col_bg The color(s) of the line (under the text labels of labels). Default: col_bg = Seeblau.

cex Numeric character expansion factor(s), multiplied by par("cex") to yield the character size(s). Default: cex = 1.5.

font The font type(s) to be used. Default: font = 1 (i.e., plain text).

new_plot Boolean: Should a new plot be generated? Set to "blank" or "slide" to create a new plot. Default: new_plot = "none" (i.e., add to an existing plot).

Details

The positions of the text elements in labels can be specified by providing their coordinates (as x and y arguments) or by providing an initial position and an y_layout (see below).

Text formatting parameters (like col, col_bg, cex, font) are recycled to match length(labels).

uline uses the base graphics system graphics::.

See Also

slide and xbox to create simple plots (without text).

Other text functions: mark(), post(), url_unikn()
Examples

```r
uline(labels = "This is a test.", new_plot = "blank") # create a new blank plot
uline(labels = "More testing here...", y = .33, col_bg = pal_pinky[[2]]) # add to plot

# 2 basic cases:
# (a) Underline text on an existing plot:
plot(x = 0, y = 0, type = "n", xlim = c(0, 1), ylim = c(0, 1), xlab = "", ylab = "")
uline(x = 0, y = .8, labels = "Underline text (on an existing plot)") # add to plot

# (b) Underline text on a new plot:
uline(x = .02, y = .80, labels = "Underline text (on a new plot)",
     new_plot = "slide") # create a new plot

# Example:
lbl_line <- c("This is neat, true, and terribly important."")
uline(labels = lbl_line, new_plot = "blank") # create a new plot
uline(labels = "(which is why we underline it).", y = .40, cex = 1.2) # add to plot

# Using x_layout and y_layout:
uline(labels = c("Ene,", "mene, miste,", "es rappelt", "in der Kiste."),
        cex = 1.4, font = 2, col = Grau, col_bg = Pinky,
        x = 1.2, y = .85, x_layout = "right", y_layout = "even", new_plot = "slide")
```

---

**unikn.guide**

*Open the unikn package guides*

**Description**

Open the unikn package guides

**Usage**

```
unikn.guide()
```

---

**url_unikn**

*url_unikn formats an URL the uni.kn way*

**Description**

`url_unikn` removes various patterns (e.g., "http", "https", "://", "www.")) from the front of a given URL and returns the remaining character string with a figure dash prefix.

**Usage**

```
url_unikn(url = "https://www.uni-konstanz.de/")
```
Arguments

url  The url to be written (as copied from a web browser).

See Also

xbox to create a new xbox (without text).

Other text functions: mark(), post(), uline()

Examples

url_unikn("https://www.uni-konstanz.de/")

---

usecol  Use a color or color palette

Description

usecol allows using a color or color palette pal (e.g., for plotting).

Usage

usecol(
  pal = pal_unikn,
  n = "all",
  alpha = NA,
  distinct = FALSE,
  use_names = FALSE,
  use_col_ramp = FALSE
)

Arguments

pal  A color palette (as a vector of colors or color palettes). Default: pal = pal_unikn.

n  An integer value specifying the desired number of colors from the palette. Default: n = "all" (i.e., use all colors of a color palette). For the palettes defined by unikn, n is set to a pre-defined selection of colors if the desired number of colors is smaller than the available number. For all other palettes and values of n larger than length(pal), n compresses or extends the palette using colorRampPalette.

alpha  A factor modifying the opacity alpha (as alpha.f in adjustcolor) to a value in [0, 1]. Default: alpha = NA (i.e., no modification of opacity).

distinct  Boolean: Return only visually distinct colors? Default: distinct = FALSE (i.e., include duplicate colors).

use_names  A logical value indicating whether colors should be returned as a named vector. Default: use_names = FALSE, for compatibility with ggplot.
use_col_ramp: A logical value specifying whether the default of using pre-selected colors should be overridden and `colorRampPalette` should be used to process `n`. Default: `use_col_ramp = FALSE`.

**Details**

`usecol` also allows modifying and combining color palettes in various ways.

**Value**

A (named) vector of colors (of type character).

**See Also**

`seecol` for viewing and comparing color palettes; `simcol` for finding similar colors; `newpal` for defining new color palettes; `grepal` for finding named colors; `shades_of` for defining shades of a given color; `ac` for adjusting color transparency; `pal_unikn` for the default uni.kn color palette.

Other color functions: `ac()`, `demopal()`, `grepal()`, `newpal()`, `seecol()`, `shades_of()`, `simcol()`

**Examples**

```r
usecol(pal = pal_unikn, n = "all") # default color palette
usecol(pal = pal_unikn, n = 4)    # selecting n dedicated colors
usecol(pal = pal_unikn, n = 20)   # extending color palette

# Mixing a new color palette:
pal_1 <- usecol(pal = c(rev(pal_seeblau), "white", pal_pinky))
seecol(pal_1)

# Mixing and extending a color palette:
pal_2 <- usecol(pal = c(rev(pal_seegruen), "white", pal_bordeaux), n = 20)
seecol(pal_2)

# Defining and using a custom color palette:
pal_princeton_1 <- c("#E77500", "white", "black")
names(pal_princeton_1) <- c("orange_w", "white", "black")
pal_3 <- usecol(pal_princeton_1, n = 7)
seecol(pal_3)

# Removing visual duplicates:
usecol(c("black", "#000000", "gray", "grey", "red", "red1"), distinct = TRUE)
seecol(usecol(c(pal_unikn, pal_seeblau), distinct = TRUE), title = "Using distinct colors")
```
**Description**

xbox plots a box with a cross (x) in its top-right corner.

**Usage**

```r
xbox(col = Seeblau, dim = c(1, 1), use_x = TRUE)
```

**Arguments**

- `col` The color to fill the box with (i.e., its background color). Default: `col = Seeblau`.
- `dim` The x- and y-dimensions of the box (as numeric). Default: `dim = c(1, 1)` (i.e., a unit square).
- `use_x` Plot a cross in upper right corner (as logical)? Default: `use_x = TRUE`.

**Details**

The cross (x) appears rectangular when viewing the plot at the correct aspect ratio (as defined by `dim`).

**See Also**

- `post` to add text to an xbox; `slide` to plot a new slide (or frame).
- Other plot functions: `slide()`, `theme_grau()`, `theme_unikn()`

**Examples**

```r
xbox() # default box

# Options:
xbox(col = Bordeaux)
xbox(dim = c(2, 1)) # 2:1 dimension (wider than high)
```
Index

* **color functions**
  - ac, 3
  - demopal, 5
  - grepal, 7
  - newpal, 13
  - seeocol, 36
  - shades_of, 39
  - simcol, 41
  - usecol, 49

* **color palettes**
  - pal_bordeaux, 15
  - pal_grau, 16
  - pal_karpfenblau, 17
  - pal_peach, 18
  - pal_petrol, 19
  - pal_pinky, 20
  - pal_seeblau, 21
  - pal_seegruen, 22
  - pal_signal, 23
  - pal_unikn, 24
  - pal_unikn_dark, 25
  - pal_unikn_light, 26
  - pal_unikn_pair, 27
  - pal_unikn_ppt, 28
  - pal_unikn_pref, 29
  - pal_unikn_web, 30

* **datasets**
  - Bordeaux, 4
  - Grau, 7
  - Karpfenblau, 11
  - pal_bordeaux, 15
  - pal_grau, 16
  - pal_karpfenblau, 17
  - pal_peach, 18
  - pal_petrol, 19
  - pal_pinky, 20
  - pal_seeblau, 21
  - pal_seegruen, 22
  - pal_signal, 23

* **plot functions**
  - slide, 43
  - theme_grau, 44
  - theme_unikn, 45
  - xbox, 51

* **preferred colors**
  - Bordeaux, 4
  - Grau, 7
  - Karpfenblau, 11
  - Peach, 31
  - Petrol, 31
  - Pinky, 32
  - Seeblau, 35
  - Seegruen, 38
  - Signal, 40

* **text functions**
  - mark, 11
  - post, 33
  - uline, 46
  - url_unikn, 48

* **utility functions**
  - get_col_names, 6

ac, 3, 6, 8, 14, 37, 40, 42, 50
adjustcolor, 3, 37, 40, 49

Bordeaux, 4, 7, 11, 15, 31–33, 35, 39, 41
INDEX

colorRampPalette, 49, 50

defcol (newpal), 13
defpal (newpal), 13
democol (demopal), 5
demofun (demopal), 5
demopal, 3, 5, 8, 14, 37, 40, 42, 50
demoplot (demopal), 5
get_col_names, 6
Grau, 4, 7, 11, 16, 31–33, 35, 39, 41
grepal, 3, 6, 7, 14, 37, 40, 42, 50
heading, 9, 43
Karpfenblau, 4, 7, 11, 17, 31–33, 35, 39, 41
line, 43
mark, 11, 34, 43, 47, 49
newpal, 3, 6, 8, 13, 37, 40, 42, 50

pal_bordeaux, 4, 15, 17–30
pal_grau, 7, 16, 16, 17–30
pal_karpfenblau, 11, 16, 17, 17, 18–30
pal_peach, 16, 17, 18, 19–31
pal_petrol, 16–18, 19, 20–32
pal_pinky, 16–19, 20, 21–30, 32, 33
pal_seebau, 4, 7, 10, 16–21, 21, 22–33, 35, 39, 41
pal_seegruen, 16–21, 22, 23–30, 39
pal_signal, 16–22, 23, 24–30, 40, 41
pal_unikn, 4, 6–8, 11, 14, 16–24, 24, 25–33, 35, 37, 39, 41, 49, 50
pal_unikn_dark, 16–24, 25, 26–30
pal_unikn_light, 16–25, 26, 27–30
pal_unikn_pair, 16–26, 27, 28–30
pal_unikn_ppt, 16–27, 28, 29, 30
pal_unikn_pref, 4, 7, 11, 16–28, 29, 30–33, 35, 39, 41
pal_unikn_web, 16–29, 30
Peach, 4, 7, 11, 18, 31, 32, 33, 35, 39, 41
Petrol, 4, 7, 11, 19, 31, 33, 35, 39, 41
Pinky, 4, 7, 11, 20, 31, 32, 35, 39, 41
post, 13, 33, 47, 49, 51

Seeblau, 4, 7, 11, 21, 31–33, 35, 39, 41
seeocol, 3, 4, 6–8, 11, 14, 16–33, 35, 36, 39–42, 50
Seeegruen, 4, 7, 11, 22, 31–33, 35, 38, 41
seepal, 6
seepal (seeocol), 36
shades_of, 3, 6, 8, 14, 37, 39, 42, 50
Signal, 4, 7, 11, 29, 31–33, 35, 39, 40
simcol, 3, 6, 8, 14, 37, 40, 41, 50
slide, 10, 13, 43, 44, 46, 47, 51

theme_grau, 43, 44, 46, 51
theme_unikn, 43, 44, 45, 51

uline, 13, 34, 46, 49
unikn_guide, 48
url_unikn, 13, 34, 47, 48
usecol, 3, 4, 6–8, 11, 14, 16–33, 35, 37, 39–42, 49

xbox, 10, 13, 33, 34, 43, 44, 46, 47, 49, 51