

# Package ‘unikn’

August 15, 2022

**Type** Package

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

**Version** 0.5.0

**Date** 2022-08-15

**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 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 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 institutions and purposes.

**Depends** R (>= 3.4.0)

**Imports** utils, ggplot2

**Suggests** knitr, rmarkdown, spelling

**Collate** 'color\_def\_1.R' 'color\_def\_2.R' 'color\_util.R' 'color\_fun.R'  
'plot\_util.R' 'plot\_box.R' 'plot\_box\_calls.R' 'plot\_text.R'  
'plot\_text\_calls.R' 'plot\_themes.R' 'plot\_kn.R' 'start\_unikn.R'

**Encoding** UTF-8

**LazyData** false

**License** CC BY-SA 4.0

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

**BugReports** <https://github.com/hneth/unikn/issues>

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**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** 2022-08-15 09:40:02 UTC

## R topics documented:

ac	3
Bordeaux	4
Grau	5
grepal	5
heading	7
Karpfenblau	9
mark	9
newpal	11
pal_bordeaux	13
pal_grau	14
pal_karpfenblau	15
pal_peach	16
pal_petrol	17
pal_pinky	18
pal_seeblau	19
pal_seegruen	20
pal_signal	21
pal_unikn	22
pal_unikn_dark	23
pal_unikn_light	24
pal_unikn_pair	25
pal_unikn_ppt	26
pal_unikn_pref	27
pal_unikn_web	28
Peach	29
Petrol	29
Pinky	30
post	31
Seeblau	33
seecol	34
Seegruen	36
shades_of	37
Signal	38
simcol	39
slide	40
theme_grau	41
theme_unikn	43
uline	44
unikn.guide	46
url_unikn	46
usecol	47
xbox	48

---

ac	<i>Adjust the transparency of a color or color palette.</i>
----	---

---

### Description

ac adjusts the transparency of a color or color palette `col` to an opacity level `alpha`.

### Usage

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

### Arguments

<code>col</code>	A (required) color or color palette (as a vector).
<code>alpha</code>	A factor modifying the opacity <code>alpha</code> (as <code>alpha.f</code> in <a href="#">adjustcolor</a> ) to a value in $[0, 1]$ . Default: <code>alpha = .50</code> (i.e., medium opacity).
<code>use_names</code>	A logical value indicating whether color names should be adjusted to include the values of <code>alpha</code> . Default: <code>use_names = TRUE</code> .

### Details

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

### 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: [grepal\(\)](#), [newpal\(\)](#), [seecol\(\)](#), [shades\\_of\(\)](#), [simcol\(\)](#), [usecol\(\)](#)

### Examples

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

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

# multiple alphas:
blacks <- ac("black", alpha = 5:0/5)
seecol(blacks, title = "One col several alpha values")

bgc <- ac(c("black", "gold"), alpha = 1:6/6)
```

```

seecol(bgc, title = "More alpha values than cols")

# Using a color palette:
seecol(ac(pal_unikn_pref, 2/3), title = "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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

## See Also

`pal_bordeaux` for the corresponding color palette; `pal_unikn` for the unkn default color palette with all 5 colors of `pal_seeblau`; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` to show color palettes; `usecol` to use color palettes.

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

## Examples

```

Bordeaux # HEX character "#8E2043" (as value)
all.equal(Bordeaux, pal_bordeaux[[4]]) # TRUE (same HEX values)

seecol(Bordeaux) # view color and details

```

---

Grau	<i>uni.kn color Grau.</i>
------	---------------------------

---

### 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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` to show color palettes; `usecol` to use color palettes.

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

### Examples

```
Grau # HEX character "#9AA0A7" (as value)
all.equal(Grau, pal_grau[[3]]) # TRUE (same HEX values)

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

---

grepal	<i>Get a vector of colors whose names match a regular expression.</i>
--------	---

---

### Description

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

### Usage

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

**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: <code>x = colors()</code> .
ignore_case	Should the case of pattern be ignored (passed to <code>ignore.case</code> of the <code>grep</code> function)? Default: <code>ignore_case = TRUE</code> .
plot	Boolean: Plot the output (using <a href="#">seecol</a> )? Default: <code>plot = TRUE</code> .

**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**

[seepal](#) for plotting 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\(\)](#), [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", title = "See 'white' colors()")

olives <- grepal("olive", plot = FALSE)
oranges <- grepal("orange", plot = FALSE)
```

```
seecol(list(olives, oranges),
       pal_names = c("olives", "oranges"),
       title = "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

```
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 = .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).

<code>y_layout</code>	A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of <code>y[1]</code> 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., <code>y_layout = 0</code> ). Default: <code>y_layout = "flush"</code> .
<code>col</code>	The color(s) of the text label(s). Default: <code>col_lbl = "black"</code> .
<code>col_bg</code>	The color(s) to highlight or fill the rectangle(s) with. Default: <code>col_bg = "default"</code> (to automatically select different shades of <a href="#">pal_seeblau</a> ).
<code>cex</code>	Numeric character expansion factor(s), multiplied by <code>par("cex")</code> to yield the character size(s). Default: <code>cex = 2</code> .
<code>font</code>	The font type(s) to be used. Default: <code>font = 2</code> (i.e., bold).
<code>new_plot</code>	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: <code>new_plot = "slide"</code> (i.e., create a new <a href="#">slide</a> ).

### 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, Seegrue, 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	<i>uni.kn color Karpfenblau.</i>
-------------	----------------------------------

---

### 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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` to show color palettes; `usecol` to use color palettes.

Other preferred colors: [Bordeaux](#), [Grau](#), [Peach](#), [Petrol](#), [Pinky](#), [Seebrau](#), [Seegrueen](#), [Signal](#)

### Examples

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

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

---

mark	<i>Plot marked (or highlighted) text elements.</i>
------	--

---

### 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**

```
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 = .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", xlim = c(0, 1), ylim = c(0, 1), 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 new color palettes.*

---

### Description

`newpal` allows defining new color palettes (as data frames).

**Usage**

```
newpal(col, names = NA, as_df = FALSE)
```

**Arguments**

col	A required vector of colors (specified by their R color names, HEX codes, or RGB values).
names	An optional character vector of names. Default: names = NA, yielding numeric 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**

[seepal](#) for plotting color palettes; [usecol](#) for using color palettes; [simcol](#) 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\(\)](#), [grepal\(\)](#), [seecol\(\)](#), [shades\\_of\(\)](#), [simcol\(\)](#), [usecol\(\)](#)

**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"))

seecol(pal_flag_de, title = "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)
seecol(pal_google, title = "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")
)
seecol(pal_flag_de_2, title = "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, title = "Colors of the Max Planck Society")

# (3) From RGB values: -----
# 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), # vermilion
               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,
       title = "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,
       title = "Comparing custom color palettes")

```

---

pal\_bordeaux

*uni.kn bordeaux color palette.*

---

## Description

pal\_bordeaux provides an additional uni.kn color palette as a data frame containing 5 colors (shades of [Bordeaux](#)).

## Usage

```
pal_bordeaux
```

## Format

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

**Details**

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

**See Also**

`pal_unikn` for the unikn default color palette with all 5 colors of `pal_seeblau`; `pal_peach` and `pal_pinky` for alternative redish uni.kn color palettes; `pal_unikn_pref` for a uni.kn color palette with all preferred colors; `seecol` to show and use color palettes.

Other color palettes: `pal_grau`, `pal_karpfenblau`, `pal_peach`, `pal_petrol`, `pal_pinky`, `pal_seeblau`, `pal_seegrueen`, `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	<i>uni.kn grau color palette.</i>
----------	-----------------------------------

---

**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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

**See Also**

[pal\\_unikn](#) for the unkn default color palette with all 5 colors of [pal\\_seeblau](#); [pal\\_unikn\\_pref](#) for a uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use 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 karpfenblau color palette.*

---

**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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

**See Also**

[pal\\_unikn](#) for the unkn 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](#) to show and use 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**

```
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	<i>uni.kn peach color palette.</i>
-----------	------------------------------------

---

**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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrueen](#), [pal\\_signal](#), [pal\\_unikn\\_dark](#), [pal\\_unikn\\_light](#), [pal\\_unikn\\_pair](#), [pal\\_unikn\\_ppt](#), [pal\\_unikn\\_pref](#), [pal\\_unikn\\_web](#), [pal\\_unikn](#)

**Examples**

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

# Plotting palette:
```



```
seecol(pal_peach)
```

---

pal_petrol	<i>uni.kn petrol color palette.</i>
------------	-------------------------------------

---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details, and [https://en.wikipedia.org/wiki/New\\_riddle\\_of\\_induction](https://en.wikipedia.org/wiki/New_riddle_of_induction) for the portmanteau "grue".

## See Also

[pal\\_unikn](#) for the unikn default color palette with all 5 colors of [pal\\_seeblau](#); [pal\\_seegrue](#) for an alternative green/grue uni.kn color palette; [pal\\_unikn\\_pref](#) for a uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrue](#), [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                      *uni.kn pinky color palette.*

---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use 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)
```

---

pal_seeblau	<i>uni.kn seeblau color palette.</i>
-------------	--------------------------------------

---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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; [seecol](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seegrueen](#), [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)
```

---

pal_seegruen	<i>uni.kn seegruen color palette.</i>
--------------	---------------------------------------

---

### 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use 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

```
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)
```

---

pal_signal	<i>uni.kn signal (Ampel) color palette.</i>
------------	---

---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

## 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](#) to show and use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrueen](#), [pal\\_unikn\\_dark](#), [pal\\_unikn\\_light](#), [pal\\_unikn\\_pair](#), [pal\\_unikn\\_ppt](#), [pal\\_unikn\\_pref](#), [pal\\_unikn\\_web](#), [pal\\_unikn](#)

## Examples

```
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)
```

---

pal\_unikn                      *unikn default color palette (11 colors).*

---

### 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 palette containing 11 color values.

### 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 palette with 11 values:

`pal_unikn[[6]]` is white or `"#FFFFFF"`.

This is useful when creating color gradients.

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

### See Also

[pal\\_unikn](#) for the default uni.kn color palette; [pal\\_seeblau](#) for the uni.kn seeblau color palette; [seecol](#) to show color palettes; [usecol](#) to use 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"

# Plotting palette:
```

```
seecol(pal_unikn)

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

---

pal_unikn_dark	<i>uni.kn dark colors in a color palette.</i>
----------------	---

---

## Description

pal\_unikn\_dark provides an additional uni.kn color palette that collects 2 dark colors of 4 color palettes as a data frame containing 8 colors (in 4 pairs).

## Usage

```
pal_unikn_dark
```

## Format

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

## Details

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use 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\\_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

# Plotting palette:
seecol(pal_unikn_dark)
```

---

pal_unikn_light	<i>uni.kn light colors in a color palette.</i>
-----------------	--

---

## Description

pal\_unikn\_light provides an additional uni.kn color palette that collects 2 light colors of 4 color palettes as a data frame containing 8 colors (in 4 pairs).

## Usage

```
pal_unikn_light
```

## Format

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

## Details

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrueen](#), [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 8

# 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"

# Plotting palette:
seecol(pal_unikn_light)
```



---

pal_unikn_pair	<i>uni.kn pairwise colors in a color palette.</i>
----------------	---

---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrueen](#), [pal\\_signal](#), [pal\\_unikn\\_dark](#), [pal\\_unikn\\_light](#), [pal\\_unikn\\_ppt](#), [pal\\_unikn\\_pref](#), [pal\\_unikn\\_web](#), [pal\\_unikn](#)

## Examples

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

# Plotting palette:
seecol(pal_unikn_pair)
```

---

pal\_unikn\_ppt            *uni.kn secondary color palette (ppt version).*

---

### 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 [pal\\_unikn](#) for the primary/default (web/sRGB) scale and <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

### 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](#) to show color palettes; [usecol](#) to use color palettes.

Other color palettes: [pal\\_bordeaux](#), [pal\\_grau](#), [pal\\_karpfenblau](#), [pal\\_peach](#), [pal\\_petrol](#), [pal\\_pinky](#), [pal\\_seeblau](#), [pal\\_seegrueen](#), [pal\\_signal](#), [pal\\_unikn\\_dark](#), [pal\\_unikn\\_light](#), [pal\\_unikn\\_pair](#), [pal\\_unikn\\_pref](#), [pal\\_unikn\\_web](#), [pal\\_unikn](#)

### Examples

```
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)
```

---

pal_unikn_pref	<i>uni.kn preferred colors in a color palette.</i>
----------------	--

---

### 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

### See Also

`pal_unikn` for the default uni.kn color palette; `seecol` to show color palettes; `usecol` to use color palettes.

Other color palettes: `pal_bordeaux`, `pal_grau`, `pal_karpfenblau`, `pal_peach`, `pal_petrol`, `pal_pinky`, `pal_seeblau`, `pal_seegrueen`, `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"

# Plotting palette:
seecol(pal_unikn_pref)
```

---

pal_unikn_web	<i>uni.kn default color palette.</i>
---------------	--------------------------------------

---

## Description

pal\_unikn\_web provides the default uni.kn color palette as a data frame containing 10 colors.

## Usage

```
pal_unikn_web
```

## Format

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

## Details

This is the primary (web/sRGB) scale.

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

## See Also

[pal\\_unikn](#) for the unikn default color palette with all 5 colors of [pal\\_seeblau](#); [pal\\_unikn\\_ppt](#) for an alternative (ppt) version; [pal\\_unikn\\_pref](#) for a uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use 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](#)

## Examples

```
pal_unikn_web
dim(pal_unikn_web) # 1 10

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

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

# Plotting palette:
seecol(pal_unikn_web)
```

---

Peach	<i>uni.kn color Peach.</i>
-------	----------------------------

---

### 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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` to show and use color palettes.

Other preferred colors: `Bordeaux`, `Grau`, `Karpfenblau`, `Petrol`, `Pinky`, `Seeblau`, `Seegruen`, `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	<i>uni.kn color Petrol.</i>
--------	-----------------------------

---

### Description

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

### Usage

Petrol

**Format**

An object of class character of length 1.

**Details**

See <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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 uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use color palettes.

Other preferred colors: [Bordeaux](#), [Grau](#), [Karpfenblau](#), [Peach](#), [Pinky](#), [Seeblau](#), [Seegrueen](#), [Signal](#)

**Examples**

```
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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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 uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use color palettes.

Other preferred colors: [Bordeaux](#), [Grau](#), [Karpfenblau](#), [Peach](#), [Petrol](#), [Seeblau](#), [Seegrueen](#), [Signal](#)

**Examples**

```
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**

```
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 = .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 = .55.

<code>x_layout</code>	An optional numeric vector or character string to control the horizontal positions of labels. Numeric values are interpreted as increments to values of <code>x</code> 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: <code>x_layout = NA</code> (i.e., using values of <code>x</code> ).
<code>y_layout</code>	A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of <code>y[1]</code> and recycled (to enable stepwise or alternating patterns). 2 character string options are: "even" (i.e., even distribution of labels across available <code>y</code> -space) and "flush" (i.e., no space between adjacent labels, i.e., <code>y_layout = 0</code> ). Default: <code>y_layout = "even"</code> .
<code>col</code>	The color(s) of the text label(s). Default: <code>col_lbl = "white"</code> .
<code>col_bg</code>	The background color(s) of the <code>xbox</code> . Default: <code>col_bg = Seeblau</code> .
<code>cex</code>	Numeric character expansion factor(s), multiplied by <code>par("cex")</code> to yield the character size(s). Default: <code>cex = 1.0</code> .
<code>font</code>	The font type(s) to be used. Default: <code>font = 1</code> (i.e., plain text).
<code>new_plot</code>	Should a new plot be generated? Set to "xbox" to plot to a basic <code>xbox</code> (with square dimensions, i.e., <code>dim = c(1, 1)</code> ). Default: <code>new_plot = "none"</code> (i.e., assumes a pre-existing <code>xbox</code> ).

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for 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` to show color palettes; `usecol` to use color palettes.

Other preferred colors: `Bordeaux`, `Grau`, `Karpfenblau`, `Peach`, `Petrol`, `Pinky`, `Seegrueen`, `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 *Plot color palettes (to see their colors).*

---

### Description

seecol provides an interface to plotting (or "seeing") the colors of a palette or comparing multiple color palettes.

### Usage

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

### Arguments

pal	<p>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 <b>unikn</b> package).</p> <p>Recognized keywords are:</p> <ol style="list-style-type: none"> <li>1. "unikn_all": All color palettes defined in <b>unikn</b></li> <li>2. "unikn_basic": All basic palettes.</li> <li>3. "pair_all": All palettes with pairwise colors.</li> <li>4. "pref_all": All preferred colors and their gradients.</li> <li>5. "grad_all":</li> </ol> <p>seecol does also recognize keywords (e.g., "all_unikn") or keywords without "unikn" (e.g., "basic").</p>
n	<p>Number of colors to show or use. If n is lower or higher than the length of the current color palette pal, the color palette is reduced or extrapolated (using <code>grDevices::colorRampPalette</code>). Default: n = "all" (i.e., show all colors in palette).</p>
alpha	<p>A factor modifying the opacity alpha (as alpha.f in <code>adjustcolor</code>) to a value in <math>[\emptyset, 1]</math>. Default: alpha = NA (i.e., no modification of opacity).</p>

hex	Should HEX color values be shown? Default: hex = NULL (i.e., show HEX color values when there is sufficient space to print them).
rgb	Should RGB color values be shown? Default: rgb = NULL (i.e., show RGB color values when there is sufficient space to print them).
col_bg	Color of plot background. Default: col_bg = NULL.
col_brd	Color of shape borders (if shown). Default: col_brd = NULL.
lwd_brd	Line width of shape borders (if shown). Default: lwd_brd = NULL.
grid	Show grid in the color plot? Default: grid = TRUE.
title	Plot title (as a character string). Default: title = NA creates a default title.
mar_note	Optional margin note (on bottom right). Default: mar_note = NA (i.e., no margin note).
pal_names	Names of color palettes or colors (as a character vector). Default: pal_names = NA (for default names).
...	Other graphical parameters (passed to plot).

## Details

seecol has two main modes, based on the contents of its pal argument:

1. if pal is set to a *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 pal = "unikn\_all" or a list of *multiple* color palettes:  
Plot visual vectors of all current color palettes for comparing them.

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

The title and pal\_names arguments add control over plotted text labels. However, the length of a character vector provided to pal\_names must correspond to the number of (custom) color palettes or colors.

## See Also

[usecol](#) for using color palettes; [simcol](#) for finding similar colors; [newpal](#) for defining new color palettes; [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\(\)](#), [grepal\(\)](#), [newpal\(\)](#), [shades\\_of\(\)](#), [simcol\(\)](#), [usecol\(\)](#)

## 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,
       title = "Reduced version of pal_unikn (n = 5)") # reducing pal_unikn
seecol(pal_seeblau, n = 8,
       title = "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,
       title = "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, title = "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"),
       title = "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

```

---

Seegruen

*uni.kn color Seegruen.*


---

## 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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

**See Also**

[pal\\_seegruen](#) for the corresponding color palette; [pal\\_unikn](#) for the unkn default color palette with all 5 colors of [pal\\_seeblau](#); [pal\\_unikn\\_pref](#) for a uni.kn color palette with all preferred colors; [seecol](#) to show color palettes; [usecol](#) to use 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 an initial color col\_1.

**Usage**

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

**Arguments**

n	Number of desired colors. Default: n = 5.
col_1	Initial color. Default: col_1 = "black".
col_n	Final (n-th) color. Default: col_n = "white".
alpha	A factor modifying the opacity alpha (as alpha.f in <a href="#">adjustcolor</a> ) to a value in [0, 1]. Default: alpha = NA (i.e., no modification of opacity).

**Details**

By default, the colors range from the initial color col\_1 = "black" to col\_n = "white", but specifying different initial and final colors yields other color ranges.

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

**See Also**

[seepal](#) for plotting color palettes; [usecol](#) for using color palettes; [simcol](#) for finding similar colors; [newpal](#) for defining new color palettes; [grepal](#) for finding named colors; [ac](#) for adjusting color transparency.

Other color functions: [ac\(\)](#), [grepal\(\)](#), [newpal\(\)](#), [seecol\(\)](#), [simcol\(\)](#), [usecol\(\)](#)

**Examples**

```
grey50 <- shades_of(50, col_1 = 'grey1')
seecol(grey50, title = "50 shades of grey1")

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

wine_white <- shades_of(6, Bordeaux, alpha = 1/2)
seecol(wine_white, col_brd = "black", lwd_brd = .5,
       title = "Shades of semi-transparent Bordeaux")
```

---

Signal

*uni.kn color Signal or alert.*

---

**Description**

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

**Usage**

Signal

**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 <https://www.uni-konstanz.de/en/university/news-and-media/create-online-and-print-media/corporate-design/> for details.

**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](#) to show color palettes; [usecol](#) to use color palettes.

Other preferred colors: [Bordeaux](#), [Grau](#), [Karpfenblau](#), [Peach](#), [Petrol](#), [Pinky](#), [Seeblau](#), [Seegrueen](#)

**Examples**

```
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**

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

**Arguments**

<code>col_target</code>	A (required) target color.
<code>col_candidates</code>	Palette of color candidates. Default: <code>col_candidates = colors()</code> .
<code>tol</code>	Numeric tolerance value(s) (must be either 1 or 3 numeric values, in RGB range from 0 to 255). Default: <code>tol = c(25, 50, 75)</code> .
<code>distinct</code>	Boolean: Return only visually distinct colors? Default: <code>distinct = TRUE</code> (i.e., remove duplicate colors).
<code>plot</code>	Boolean: Plot the output (using <a href="#">seecol</a> )? Default: <code>plot = TRUE</code> .

**Details**

simcol returns a vector of the (named) colors or color values in `col_candidates` (set to `'colors()'` of **grDevices** per default) that are similar to the specified target color `col_target`.

If `plot = TRUE`, simcol also visualizes the detected colors (by passing its result to [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 `tol` (with  $0 \leq \text{tol} \leq 255$ ). Higher `tol` values correspond to more permissive similarity judgments.

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

### Value

A named vector of colors or color values.

### See Also

[seecol](#) for plotting/seeing color palettes; [usecol](#) for using color palettes; [newpal](#) for defining new color palettes; [grepal](#) for finding named colors; [shades\\_of](#) to defining shades of a given color; [ac](#) for adjusting color transparency.

Other color functions: [ac\(\)](#), [grepal\(\)](#), [newpal\(\)](#), [seecol\(\)](#), [shades\\_of\(\)](#), [usecol\(\)](#)

### Examples

```
# Basic uses:
simcol(col_target = "red")
simcol("tan", tol = 15)
simcol(Seeblau, tol = c(20, 30, 40))
simcol("blue", col_candidates = pal_unikn_pref, tol = 120)

# Fine-tuning the range of color matching:
simcol(Seeblau, tol = 30) # = simcol(Seeblau, tol = c(30, 30, 30))
simcol(Seeblau, tol = c(20, 20, 80))

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

---

slide

*Plot a slide (or frame).*

---

### Description

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

### Usage

```
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(.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**

```
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**

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

**Details**

`theme_grau` is no-nonsense, but fills panel backgrounds in "grau" (specifically, `pal_seegrau[[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

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, Seegrueen))) +
  labs(tag = "B",
       title = "Iris sepals",
       caption = "Data from datasets::iris") +
  coord_fixed(ratio = 3/2) +
  theme_grau()
```

---

theme_unikn	<i>Basic unikn theme for ggplot2.</i>
-------------	---------------------------------------

---

## Description

theme\_unikn provides a basic **unikn** theme to use in **ggplot2** commands.

## Usage

```
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

col_title	Color of title (text) elements (optional, numeric). Default: col_title = pal_seeblau[[4]]. Consider using col_title = "black" when data uses Seeblau colors.
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, Seegrueen))) +
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

<code>labels</code>	A character vector specifying the text labels to be plotted.
<code>x</code>	A numeric vector of x-coordinates at which the text labels in <code>labels</code> should be written. If the lengths of <code>x</code> and <code>y</code> differ, the shorter one is recycled. Default: <code>x = 0</code> .
<code>y</code>	A numeric vector of y-coordinates at which the text labels in <code>labels</code> should be written. If the lengths of <code>x</code> and <code>y</code> differ, the shorter one is recycled. Default: <code>y = .55</code> .
<code>x_layout</code>	An optional numeric vector or character string to control the horizontal positions of labels. Numeric values are interpreted as increments to values of <code>x</code> and recycled (to enable stepwise or alternating patterns). 3 character string options are: <code>"center"</code> (i.e., center wrt. first label or plot center), <code>"left"</code> (i.e., left wrt. first label or plot center), <code>"right"</code> (i.e., right wrt. first label or plot center). Default: <code>x_layout = NA</code> (i.e., using values of <code>x</code> ).

<code>y_layout</code>	A numeric value or character string to control the vertical positions of labels. Numeric values are interpreted as increments to values of <code>y[1]</code> 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., <code>y_layout = 0</code> ). Default: <code>y_layout = "even"</code> .
<code>col</code>	The color(s) of the text label(s). Default: <code>col_lbl = "black"</code> .
<code>col_bg</code>	The color(s) of the line (under the text labels of labels). Default: <code>col_bg = Seeblau</code> .
<code>cex</code>	Numeric character expansion factor(s), multiplied by <code>par("cex")</code> to yield the character size(s). Default: <code>cex = 1.5</code> .
<code>font</code>	The font type(s) to be used. Default: <code>font = 1</code> (i.e., plain text).
<code>new_plot</code>	Boolean: Should a new plot be generated? Set to "blank" or "slide" to create a new plot. Default: <code>new_plot = "none"</code> (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

```
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	<i>Opens the unikn package guides</i>
-------------	---------------------------------------

---

### Description

Opens the unikn package guides

### Usage

```
unikn.guide()
```

---

url_unikn	<i>url_unikn formats an URL the uni.kn way.</i>
-----------	---

---

### 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 = <a href="#">pal_unikn</a> .
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 <b>unikn</b> , 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 <a href="#">colorRampPalette</a> .
alpha	A factor modifying the opacity alpha (as alpha.f in <a href="#">adjustcolor</a> ) 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 <a href="#">colorRampPalette</a> 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 plotting/seeing color palettes; [simcol](#) for finding similar colors; [newpal](#) for defining new color palettes; [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\(\)](#), [grepal\(\)](#), [newpal\(\)](#), [seecol\(\)](#), [shades\\_of\(\)](#), [simcol\(\)](#)

**Examples**

```
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")
```

---

xbox

*Plot a box (with x).*


---

**Description**

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

**Usage**

```
xbox(col = Seeblau, dim = c(1, 1))
```

**Arguments**

**col** The color to fill the box with (i.e., its background color). Default: `col = unlist(seeblau)`.

**dim** The x- and y-dimensions of the box. Default: `dim = c(1, 1)` (i.e., a unit square).



**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**

```
xbox() # default box
```

```
# Options:
```

```
xbox(col = Bordeaux)
```

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

# Index

## \* color functions

- ac, 3
- grepal, 5
- newpal, 11
- seecol, 34
- shades\_of, 37
- simcol, 39
- usecol, 47

## \* color palettes

- pal\_bordeaux, 13
- pal\_grau, 14
- pal\_karpfenblau, 15
- pal\_peach, 16
- pal\_petrol, 17
- pal\_pinky, 18
- pal\_seeblau, 19
- pal\_seegruen, 20
- pal\_signal, 21
- pal\_unikn, 22
- pal\_unikn\_dark, 23
- pal\_unikn\_light, 24
- pal\_unikn\_pair, 25
- pal\_unikn\_ppt, 26
- pal\_unikn\_pref, 27
- pal\_unikn\_web, 28

## \* datasets

- Bordeaux, 4
- Grau, 5
- Karpfenblau, 9
- pal\_bordeaux, 13
- pal\_grau, 14
- pal\_karpfenblau, 15
- pal\_peach, 16
- pal\_petrol, 17
- pal\_pinky, 18
- pal\_seeblau, 19
- pal\_seegruen, 20
- pal\_signal, 21
- pal\_unikn, 22

- pal\_unikn\_dark, 23
- pal\_unikn\_light, 24
- pal\_unikn\_pair, 25
- pal\_unikn\_ppt, 26
- pal\_unikn\_pref, 27
- pal\_unikn\_web, 28
- Peach, 29
- Petrol, 29
- Pinky, 30
- Seeblau, 33
- Seegruen, 36
- Signal, 38

## \* plot functions

- slide, 40
- theme\_grau, 41
- theme\_unikn, 43
- xbox, 48

## \* preferred colors

- Bordeaux, 4
- Grau, 5
- Karpfenblau, 9
- Peach, 29
- Petrol, 29
- Pinky, 30
- Seeblau, 33
- Seegruen, 36
- Signal, 38

## \* text functions

- mark, 9
- post, 31
- uline, 44
- url\_unikn, 46

ac, 3, 6, 12, 35, 38, 40, 48  
adjustcolor, 3, 34, 37, 47

Bordeaux, 4, 5, 9, 13, 29–31, 33, 37, 38

colorRampPalette, 47

defcol (newpal), 11

- defpal (newpal), 11
- Grau, 4, 5, 9, 14, 29–31, 33, 37, 38
- grepal, 3, 5, 12, 35, 38, 40, 48
- heading, 7, 41
- Karpfenblau, 4, 5, 9, 15, 29–31, 33, 37, 38
- line, 41
- mark, 9, 32, 41, 45, 46
- newpal, 3, 6, 11, 35, 38, 40, 48
- pal\_bordeaux, 4, 13, 15–28
- pal\_grau, 5, 14, 14, 15–28
- pal\_karpfenblau, 9, 14, 15, 15, 16–28
- pal\_peach, 14, 15, 16, 17–29
- pal\_petrol, 14–16, 17, 18–30
- pal\_pinky, 14–17, 18, 19–28, 30, 31
- pal\_seeblau, 4, 5, 8, 14–19, 19, 20–31, 33, 37, 38
- pal\_seegruen, 14–19, 20, 21–28, 36, 37
- pal\_signal, 14–20, 21, 22–28, 38
- pal\_unikn, 4–6, 9, 12, 14–22, 22, 23–31, 33, 35, 37, 38, 47, 48
- pal\_unikn\_dark, 14–22, 23, 24–28
- pal\_unikn\_light, 14–23, 24, 25–28
- pal\_unikn\_pair, 14–24, 25, 26–28
- pal\_unikn\_ppt, 14–25, 26, 27, 28
- pal\_unikn\_pref, 4, 5, 9, 14–26, 27, 28–31, 33, 37, 38
- pal\_unikn\_web, 14–27, 28
- Peach, 4, 5, 9, 16, 29, 30, 31, 33, 37, 38
- Petrol, 4, 5, 9, 17, 29, 29, 31, 33, 37, 38
- Pinky, 4, 5, 9, 18, 29, 30, 30, 33, 37, 38
- post, 11, 31, 45, 46, 49
- Seeblau, 4, 5, 9, 19, 29–31, 33, 37, 38
- seecol, 3–6, 9, 12, 14–31, 33, 34, 37–40, 48
- Seegruen, 4, 5, 9, 20, 29–31, 33, 36, 38
- seepal, 6, 12, 38
- seepal (seecol), 34
- shades\_of, 3, 6, 12, 35, 37, 40, 48
- Signal, 4, 5, 9, 27, 29–31, 33, 37, 38
- simcol, 3, 6, 12, 35, 38, 39, 48
- slide, 8, 11, 40, 42, 43, 45, 49
- theme\_grau, 41, 41, 43, 49
- theme\_unikn, 41, 42, 43, 49
- uline, 11, 32, 44, 46
- unikn.guide, 46
- url\_unikn, 11, 32, 45, 46
- usecol, 3–6, 9, 12, 15–20, 22–28, 30, 31, 33, 35, 37, 38, 40, 47
- xbox, 8, 11, 31, 32, 41–43, 45, 46, 48