Package ‘col2hex2col’

February 18, 2026

Type Package

Title Convert Between Color Names and Hex Codes

Version 0.5.3

Description Provides simple functions to convert between color names and hexadecimal color codes using an extensive database of over 32,000 colors. Includes all 657 R built-in colors plus the comprehensive color-names database. The package supports bidirectional conversion with backward compatibility, prioritizing R colors when available.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), gt, farver, cli

Config/testthat/edition 3

Depends R (>= 3.5.0)

URL `https://github.com/AnttiRask/col2hex2col`

BugReports `https://github.com/AnttiRask/col2hex2col/issues`

NeedsCompilation no

Author Antti Rask [aut, cre, cph],
Yann Cohen [ctb] (GitHub: iamYannC)

Maintainer Antti Rask <`anttilennartrask@gmail.com`>

Repository CRAN

Date/Publication 2026-02-18 00:40:08 UTC

Contents

color_to_hex 2
create_color_table 3
get_color_data 4
hex_to_color 6

Index **8**

`color_to_hex` *Convert Color Names to Hex Codes*

Description

Converts color names to their hexadecimal color code representations. This function accepts color names from an extensive database of over 32,000 color names, including all 657 built-in R colors from `colors` plus the comprehensive color-names database from `https://github.com/meodai/color-names`.

Usage

```
color_to_hex(color)
```

Arguments

`color`	A character vector of color names (e.g., "red", "blue", "sunset orange"). Color names are case-insensitive and whitespace is trimmed. Spaces within color names are preserved (e.g., "sky blue" is different from "skyblue").

Details

The function performs input validation and will raise an error if:

- The input is not a character vector
- Any NA values are present
- Any invalid color names are provided

This function is vectorized and efficiently handles both single colors and vectors of multiple colors. The extended database includes over 32,000 unique color names from various sources, making it suitable for a wide range of color specification needs.

Color name matching is case-insensitive: "Red", "red", and "RED" all match the same color.

Value

A character vector of hexadecimal color codes in the format "#RRGGBB", where each pair of characters represents the red, green, and blue components in hexadecimal notation (00-FF). The returned vector has the same length as the input.

See Also

`hex_to_color` for the reverse conversion, `colors` for R's built-in color names

Examples

```
# Convert a single color
color_to_hex("red")

# Convert multiple colors
color_to_hex(c("red", "blue", "green"))

# Works with all R color names
color_to_hex(c("skyblue", "coral", "chartreuse"))

# Also works with extended color names
color_to_hex(c("sunset orange", "arctic ocean", "forest green"))

# Case insensitive
color_to_hex(c("Red", "BLUE", "Green"))

# Use in a data visualization context
colors <- c("steelblue", "firebrick", "forestgreen")
hex_codes <- color_to_hex(colors)
# hex_codes can now be used with plotting functions
```

`create_color_table` *Create Color Swatch Table*

Description

Creates a visual table displaying color names, hex codes, and color swatches. Requires the `gt` package to be installed.

Usage

```
create_color_table(df)
```

Arguments

`df`	A data frame with at minimum a column named `hex` containing hexadecimal color codes. Any additional columns will be included in the table. The first column (if not `hex`) will be automatically labeled as "Color Name" in the output table, regardless of its original name.

Details

This function creates an enhanced table using the `gt` package. The table includes:

- Color names (if the `name` column exists in the input)
- Hexadecimal color codes
- Visual color swatches (cells filled with the actual colors)

The gt package must be installed to use this function. If it's not installed, the function will provide instructions on how to install it.

You can pass the output of get_color_data directly to this function, or create a custom data frame with your own color selection.

Value

A gt table object displaying the colors with visual swatches. The table will have columns for color names (if provided), hex codes, and a color swatch column where each cell is filled with the corresponding color.

See Also

get_color_data for obtaining the complete color database, color_to_hex for converting color names to hex codes

Examples

```
# Create a table with a few colors
colors_df <- data.frame(
  name = c("red", "blue", "forestgreen"),
  hex = c("#FF0000", "#0000FF", "#228B22")
)

# Only run if gt is available
if (requireNamespace("gt", quietly = TRUE)) {
  create_color_table(colors_df)

  # Use with get_color_data() - show first 10 colors
  all_colors <- get_color_data()
  create_color_table(head(all_colors, 10))

  # Create a table with only specific colors
  blue_colors <- all_colors[grepl("blue", all_colors$name), ]
  create_color_table(head(blue_colors, 20))

  # Minimal example with just hex codes
  hex_only <- data.frame(hex = c("#FF0000", "#00FF00", "#0000FF"))
  create_color_table(hex_only)
}
```

---

get_color_data *Get Color Database*

Description

Returns the complete color database as a data frame containing all 32,000+ color names and their corresponding hexadecimal codes.

Usage

```
get_color_data()
```

Details

The returned data frame is sorted alphabetically by color name. Each color name maps to exactly one hex code, though multiple color names may share the same hex code.

This function is useful for:

- Exploring available color names
- Creating custom color palettes
- Searching for colors by name pattern
- Analyzing color distributions
- Building color visualization tools

Value

A data frame with the following columns:

name Character vector of color names (lowercase)

hex Character vector of hexadecimal color codes (uppercase, format: #RRGGBB)

lab_l, lab_a, lab_b (Optional) LAB color space coordinates, added when the `farver` package is available

The data frame contains over 32,000 rows representing all available colors, including both R's 657 built-in colors and the extended color-names database.

See Also

`color_to_hex` for converting specific color names to hex codes, `hex_to_color` for the reverse conversion, `create_color_table` for creating visual color tables

Examples

```
# Get the complete color database
colors_df <- get_color_data()
head(colors_df)

# See dimensions
dim(colors_df)

# Find all colors containing "blue"
blue_colors <- colors_df[grepl("blue", colors_df$name), ]
head(blue_colors)

# Get a random sample of colors
set.seed(123)
sample_colors <- colors_df[sample(nrow(colors_df), 10), ]
sample_colors
```

`hex_to_color` *Convert Hex Codes to Color Names*

Description

Converts hexadecimal color codes to their corresponding color names. This function searches through an extensive database of over 32,000 color names, prioritizing R's built-in color names when available.

Usage

```
hex_to_color(hex, fallback_nearest_color = TRUE, fallback_distance = c("lab"))
```

Arguments

`hex`	A character vector of hexadecimal color codes in the format "#RRGGBB" or "#RRGGBBAA" (e.g., "#FF0000", "#0000FF", "#FF0000FF"). The hash symbol (#) is required, and the hex code is case-insensitive. Each component (RR, GG, BB) must be a two-digit hexadecimal value (00-FF). If an 8-digit code with alpha channel (AA) is provided, the alpha channel is ignored.
`fallback_nearest_color`	Logical. If `TRUE` (default), when a hex code has no exact match in the database, the function will find the nearest named color using LAB color distance. Requires the `farver` package. If `FALSE`, unmatched hex codes return `NA`.
`fallback_distance`	Character. The color distance metric to use for fallback matching. Currently only `"lab"` is supported.

Details

The function performs input validation and will raise an error if:

- The input is not a character vector
- Any NA values are present
- Any hex codes are not in the correct "#RRGGBB" or "#RRGGBBAA" format

This function is case-insensitive for the hex values (e.g., "#FF0000" and "#ff0000" are treated identically). When 8-digit hex codes with alpha channel are provided (e.g., "#FF0000FF"), the alpha channel is automatically stripped and only the RGB portion is used for color name lookup.

Name Selection Strategy: When multiple color names map to the same hex code:

1. R's built-in color names are prioritized (from `colors`)
2. If no R color exists, the shortest name from the extended database is returned

This ensures backward compatibility with R's color system while providing coverage for the 32,161 unique hex codes in the extended database.

The extended database includes colors from `https://github.com/meodai/color-names`, significantly increasing the likelihood of finding a named match for any given hex code.

Value

A character vector of color names (in lowercase). If a hex code does not have a corresponding named color in the database and `fallback_nearest_color` = `FALSE`, `NA` is returned for that element. If `fallback_nearest_color` = `TRUE`, the nearest named color is returned instead (with a warning). The returned vector has the same length as the input.

See Also

`color_to_hex` for the reverse conversion, `colors` for R's built-in color names

Examples

```
# Convert a single hex code
hex_to_color("#FF0000")

# Convert multiple hex codes
hex_to_color(c("#FF0000", "#0000FF", "#00FF00"))

# Works with 8-digit hex codes (alpha channel ignored)
hex_to_color(c("#FF0000FF", "#0000FFFF"))

# Case insensitive
hex_to_color("#ff0000")  # Same as "#FF0000"

# Works with extended color database
hex_to_color("#FF6347")  # Returns a descriptive color name

# Fallback to nearest color when no exact match (default behavior)
hex_to_color("#859900")  # Returns nearest named color with a warning

# Disable fallback to get NA for unmatched colors
hex_to_color("#859900", fallback_nearest_color = FALSE)

# Round-trip conversion
original <- c("red", "blue", "green")
hex_codes <- color_to_hex(original)
hex_to_color(hex_codes)  # Returns original color names
```

Index

color_to_hex, 2, *4*, *5*, *7*
colors, *2*, *6*, *7*
create_color_table, 3, *5*

get_color_data, *4*, 4

hex_to_color, *2*, *5*, 6