Package: colourspace (via r-universe)

May 26, 2026

Title Convert from One Colour Space to Another, Print a Ready-to-Paste Modern 'CSS' Syntax

Version 0.1.1

Description Provides a comprehensive 'API' for colour conversion between popular colour spaces ('RGB', 'HSL', 'OKLab', 'OKLch', 'hex', and named colours) along with clean, modern 'CSS' Color Level 4 syntax output. Integrates seamlessly into 'Shiny' and 'Quarto' workflows. Includes nearest colour name lookup powered by a curated database of over 30,000 colour names. 'OKLab'/'OKLCh' colour spaces are described in Ottosson (2020) <https://bottosson.github.io/posts/oklab/>. 'CSS' Color Level 4 syntax follows the W3C specification <https://www.w3.org/TR/css-color-4/>.

License MIT + file LICENSE

URL https://github.com/iamyannc/colourspace

BugReports https://github.com/iamyannc/colourspace/issues

Language en-GB

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

Imports farver, RANN

Suggests testthat (>= 3.0.0)

Repository https://iamyannc.r-universe.dev

Date/Publication 2026-03-27 03:06:51 UTC

RemoteUrl https://github.com/iamyannc/colourspace

RemoteRef HEAD

RemoteSha 6996fd82ecfd4ce55e6974ea2062c91af599eaf2

Contents

`color_map` *Precomputed colour map for nearest-name lookup*

Description

A data frame containing each colour name, hex code, and coordinates in multiple colour spaces for fast nearest-neighbour search.

Usage

```
color_map
```

Format

A data frame with the following columns:

hex Lowercase hex code starting with '#'.

name Colour name (character).

source Origin of the colour name: `"r"` or `"extended"` (see `color_names`).

lab_l, lab_a, lab_b CIELAB components.

oklch_l, oklch_c, oklch_h OKLCH components.

rgb_r, rgb_g, rgb_b sRGB components (0-255).

hsl_h, hsl_s, hsl_l HSL components.

Author(s)

Data curated by David Aerne (`https://github.com/meodai`).

Source

Derived from `color_names` using farver decoders.

`color_names` *CSS color names to HEX mapping*

Description

A dataset containing 31k+ color names compiled by David Aerne's meodai/color-names project, merged with R's built-in colour names.

Usage

```
color_names
```

Format

A data frame with three columns:

hex Lowercase hex triplet starting with '#'.

name Color name as provided by the source (character).

source Origin of the colour name: `"r"` for R's built-in colours (`grDevices::colors()`) or `"extended"` for the meodai/color-names community database.

Author(s)

Data curated by David Aerne (https://github.com/meodai).

Source

https://github.com/meodai/color-names

`convert_colourspace` *Convert between colour spaces*

Description

Convert between colour spaces

Usage

```
convert_colourspace(value, from, to, fallback = c("all", "r", "none"))
```

Arguments

value	Colour input. For from = "hex" or from = "name", a character vector. For numeric spaces (rgb, hsl, oklch), a numeric vector of length 3, matrix/data frame with three columns, or a list of such vectors.
from	Source colour space. One of "hex", "rgb", "hsl", "oklab", "oklch", or "name".
to	Target colour space. One of "hex", "rgb", "hsl", "oklab", "oklch", or "name" (reverse lookup).
fallback	Behaviour when mapping to = "name" and no exact hex/name match is found. One of:
	"all" (default) Return the closest named colour from the full 31 000+ colour database.
	"r" Always return the nearest R built-in colour (from grDevices::colors()). Useful when the result will be used in base-R or ggplot2 plotting functions.
	"none" Return NA for colours without an exact name match.

Details

All conversions and nearest-colour calculations are powered by the **farver** package. Hex inputs may include an alpha channel (#rgba/#rrggbbaa), but alpha is currently ignored (stripped before decoding).

Value

For scalar inputs, a named numeric vector (or hex string or colour name). For vectorised inputs, a matrix with one row per input colour or a character vector for to = "name".

Examples

```
convert_colourspace("#ff0000", from = "hex", to = "rgb")
convert_colourspace(c(255, 255, 0), from = "rgb", to = "hex")
convert_colourspace(c("#ff0000", "#00ff00"), from = "hex", to = "oklch")
```

---

from_css *Parse CSS color strings*

Description

Parse CSS color function strings (e.g., oklch(...), rgb(...), hsl(...)) or hex colors and convert them to a target color space. Automatically detects the input format from the CSS syntax.

Usage

```
from_css(css, to = "hex", fallback = c("all", "r", "none"))
```

Arguments

`css`	Character vector of CSS color strings. Supported formats: • `oklch(L C H)`, `oklch(L C H / A)` • `oklab(L A B)`, `oklab(L A B / A)` • `rgb(R G B)`, `rgb(R G B / A)`, `rgb(R, G, B)`, `rgb(R, G, B, A)` • `rgba(R, G, B, A)` • `hsl(H S L)`, `hsl(H S L / A)`, `hsl(H, S, L)`, `hsl(H, S, L, A)` • `hsla(H, S, L, A)` • Hex colors: `#rgb`, `#rrggbb`, `#rrggbbaa`
`to`	Target colour space. One of `"hex"` (default), `"rgb"`, `"hsl"`, `"oklab"`, `"oklch"`, or `"name"`.
`fallback`	Passed to `convert_colourspace()`. One of `"all"` (default), `"r"`, or `"none"`.

Details

Both modern (space-separated) and legacy (comma-separated) CSS notations are supported:

- Modern: `rgb(255 0 0)`, `rgb(255 0 0 / 0.5)`
- Legacy: `rgb(255, 0, 0)`, `rgb(255, 0, 0, 0.5)`
- Legacy with explicit alpha: `rgba(255, 0, 0, 0.5)`

Alpha channels are currently parsed but **ignored** during conversion.

Value

For scalar inputs, a named numeric vector (or hex string or colour name). For vectorised inputs, a matrix with one row per input colour or a character vector for `to = "name"` or `to = "hex"`.

Examples

```
# Parse OKLCH CSS string to hex
from_css("oklch(62.792% 0.258 29.221 / 1)")

# Parse RGB CSS string (modern & legacy)
from_css("rgb(255 0 0 / 1)", to = "oklch")
from_css("rgb(255, 0, 0)", to = "hex")

# Parse HSL CSS string
from_css("hsl(210 50% 40% / 1)", to = "rgb")

# Also works with hex colors
from_css("#ff0000", to = "oklch")

# Vectorized
from_css(c("oklch(62.792% 0.258 29.221 / 1)", "rgb(0 255 0 / 1)"))
```

hex_to_hsl *Convert HEX to HSL*

Description

Convert HEX to HSL

Usage

```
hex_to_hsl(hex)
```

Arguments

hex	Character vector of hex colour strings.

Value

Numeric vector (length 3) or matrix with columns h, s, l.

Examples

```
hex_to_hsl("#336699")
```

hex_to_name *Convert HEX to colour name*

Description

Reverse lookup using the bundled name database. When an exact match is not found, behaviour depends on fallback:

Usage

```
hex_to_name(hex, fallback = c("all", "r", "none"))
```

Arguments

hex	Character vector of hex colour strings.
fallback	One of "all" (default), "r", or "none".

Details

"all" (default) Return the closest named colour from the full 31 000+ colour database.

"r" Always return the nearest R built-in colour (from grDevices::colors()). Useful when the result will be used in base-R or ggplot2 plotting functions.

"none" Return NA for colours without an exact name match.

Value

Character vector of colour names (or NA).

Examples

```
hex_to_name("#c93f38")
hex_to_name("#111114", fallback = "all")
hex_to_name("#ff5733", fallback = "r")
```

---

hex_to_oklab *Convert HEX to OKLAB*

Description

Convert HEX to OKLAB

Usage

```
hex_to_oklab(hex)
```

Arguments

hex	Character vector of hex colour strings.

Value

Numeric vector (length 3) or matrix with columns l, a, b.

Examples

```
hex_to_oklab("#ff0000")
```

---

hex_to_oklch *Convert HEX to OKLCH*

Description

Convert HEX to OKLCH

Usage

```
hex_to_oklch(hex)
```

Arguments

hex	Character vector of hex colour strings.

Value

Numeric vector (length 3) or matrix with columns `l`, `c`, `h`.

Examples

```
hex_to_oklch("#ff0000")
```

`hex_to_rgb` *Convert HEX to RGB*

Description

Convert HEX to RGB

Usage

```
hex_to_rgb(hex)
```

Arguments

`hex`	Character vector of hex colour strings.

Value

Numeric vector (length 3) or matrix with columns `r`, `g`, `b`.

Examples

```
hex_to_rgb("#336699")
```

`hsl_to_hex` *Convert HSL to HEX*

Description

Convert HSL to HEX

Usage

```
hsl_to_hex(hsl)
```

Arguments

`hsl`	Numeric vector/matrix of HSL values (h: 0-360, s/l: 0-100).

Value

Character vector of hex colours.

Examples

```
hsl_to_hex(c(210, 50, 40))
```

---

name_to_hex *Convert colour name to HEX*

Description

Looks up CSS-style colour names from the bundled meodai list and returns hex values.

Usage

```
name_to_hex(name)
```

Arguments

name	Character vector of colour names (case-insensitive).

Value

Character vector of hex colours.

Examples

```
name_to_hex("100 Mph")
```

---

oklab_to_hex *Convert OKLAB to HEX*

Description

Convert OKLAB to HEX

Usage

```
oklab_to_hex(oklab)
```

Arguments

oklab	Numeric vector/matrix of OKLAB values (l in 0-1).

Value

Character vector of hex colours.

Examples

```
oklab_to_hex(c(0.628, 0.225, 0.126))
```

`oklch_to_hex` *Convert OKLCH to HEX*

Description

Convert OKLCH to HEX

Usage

```
oklch_to_hex(oklch)
```

Arguments

`oklch`	Numeric vector/matrix of OKLCH values (`l` in 0-1, `c` >= 0, `h` in degrees).

Value

Character vector of hex colours.

Examples

```
oklch_to_hex(c(0.628, 0.258, 29.221))
```

`rgb_to_hex` *Convert RGB to HEX*

Description

Convert RGB to HEX

Usage

```
rgb_to_hex(rgb)
```

Arguments

`rgb`	Numeric vector/matrix of RGB values (0-255).

Value

Character vector of hex colours.

Examples

```
rgb_to_hex(c(51, 102, 153))
```

---

to_css *Format colours as modern CSS color functions*

Description

Convert colours between supported spaces and return a character vector in modern CSS Color 4 functional notation (space-separated components with an optional alpha channel introduced by `/`).

Usage

```
to_css(
  value,
  from = NULL,
  to = c("oklch", "oklab", "rgb", "hsl", "hex"),
  alpha = 1,
  fallback = c("all", "r", "none")
)
```

Arguments

`value`	Colour input. For `from = "hex"` or `from = "name"`, a character vector. For numeric spaces (`rgb`, `hsl`, `oklch`, `oklab`), a numeric vector of length 3, matrix/data frame with three columns, or a list of such vectors.
`from`	Source colour space. One of `"hex"`, `"rgb"`, `"hsl"`, `"oklab"`, `"oklch"`, or `"name"`. If `NULL` (default), `to_css()` will infer `"hex"` vs `"name"` for character inputs.
`to`	Target CSS function. One of `"oklch"` (default), `"oklab"`, `"rgb"`, `"hsl"`, or `"hex"`.
`alpha`	Alpha channel as numbers in `[0, 1]`. Recycled to match the number of colours.
`fallback`	Passed to `convert_colourspace()`. One of `"all"` (default), `"r"`, or `"none"`.

Value

A character vector of CSS colors.

See Also

OKLCH in CSS: why we moved from RGB and HSL for a detailed explanation of why OKLCH is the recommended colour space for modern CSS.

Examples

```
to_css("red")
to_css("#ff5a3c", from = "hex", to = "oklch", alpha = 0.8)
to_css(c("#ff0000", "#00ff00"), to = "rgb", alpha = c(1, 0.5))
```

Index