

# Package: paint (via r-universe)

July 5, 2024

**Title** paint data.frames summaries in colour

**Version** 0.1.7

**Description** Provides an alternative to the print() methods provided by the authors of `data.frame` objects: `tibble`, `sf`, `data.table`, `data.frame`.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**URL** <https://github.com/MilesMcBain/paint>

**BugReports** <https://github.com/MilesMcBain/paint/issues>

**Depends** R (>= 3.6.0)

**Imports** crayon, keypress, pryr, purrr, RColorBrewer, stringi, utils, vctrs, viridisLite

**Suggests** spData, tibble, tsibble, data.table, sf, testthat (>= 3.0.0), dplyr, tidyr, palmerpenguins, rlang

**Config/testthat/edition** 3

**Repository** <https://milesmbain.r-universe.dev>

**RemoteUrl** <https://github.com/milesmbain/paint>

**RemoteRef** v0.1.7

**RemoteSha** a7e97ddfba3a2a37d6836124e376ff27a6e0bac8

## Contents

ipaint	2
mask_print	2
paint	3
rainbow_6	4
rainbow_text	4
unpaint	5

**Index****6**


---

ipaint	<i>Interactively scroll through rows of a painted dataframe</i>
--------	---

---

**Description**

This function calls `paint()` repeatedly on a dataframe, updating the rows shown in response to left or right keypresses.

**Usage**

```
ipaint(df = .Last.value)
```

**Arguments**

`df` the dataframe to scroll through, defaults to `.Last.value`

**Details**

It will only work in terminals supported by keypress - Not many!

---

mask_print	<i>mask print methods for paintable dataframes</i>
------------	--

---

**Description**

This function replaces the original paint methods for supported data.frame sub classes with calls to `paint()` in the current session. The utility of this is that `paint()` will be called whenever `print()` would, for example at the end of an expression when coding interactively.

**Usage**

```
mask_print()
unmask_print()
```

**Details**

This also makes `paint` usable with other tools that output to console e.g. `{breakerofchains}`.

If you really like `paint()` you could put `paint::mask_print()` in your `.Rprofile`.

By default a colourful reminder message is emitted whenever `print` methods are replaced, which may not happen until the package that contains them is actually loaded, e.g. after `library(tibble)`. This can be disabled with `options(paint_remind_mask_print = FALSE)`

**Functions**

- `unmask_print`: the reverse operation to `mask_print`: return all print methods to defaults.

**See Also**

`unpaint()` A method for calling the default print functions once for a single object or `.Last.value`.

---

paint	<i>Paint a rectangle in the data.frame family</i>
-------	---

---

**Description**

`paint` is an alternative print method of objects that are of class `data.frame`. It has been described as a "pretty `pillar::glimpse()`", since it also favours printing columns running horizontally.

**Usage**

```
paint(df, ...)
```

**Arguments**

<code>df</code>	the dataframe to paint
<code>...</code>	further arguments to class specific paint methods

**Details**

`paint` tries to be low noise by using colour instead of markup characters. It also highlights important values like NA, NULL, NaN, and Inf.

Each supported sub-class of `data.frame` has a custom meta data header, designed to show useful information for wrangling.

The appearance of the output can be customised with options:

- `paint_palette` the palette to paint row colours with, defaults to `rainbow_6()`, 6 standard ANSI colours that are supported in most terminals.
- `paint_n_rows` the number of rows to print for each column. Defaults to the number of colours in the `paint_palette`, but can be set higher for repeating sequences.
- `paint_max_width` the maximum width of the output. 60 is the default. `{paint}` does not resize based on terminal width by design.
- `paint_align_row_head = c("left", "center", "right")`. How to align the column title and type. Defaults to "left".
- `paint_dark_mode` darken the `paint_palette` using `crayon::blurred` - not supported in all terminals.

**Value**

`df` invisibly. Text outputted printed to console.

**See Also**

[unpaint\(\)](#) for calling the default print method on a dataframe

**Examples**

```
paint(mtcars)
```

---

rainbow\_6

*paintpals*

---

**Description**

paintpals

**Usage**

```
rainbow_6()
```

```
viridis_6()
```

```
brewer_set3_12()
```

```
brewer_pastel1_8()
```

```
brewer_pastel2_7()
```

```
brewer_dark2_7()
```

```
brewer_accent_7()
```

---

rainbow\_text

*Colour some text with a paint palette*

---

**Description**

Exported just for fun.

**Usage**

```
rainbow_text(msg)
```

**Arguments**

`msg` a length one character vector (not length checked) to be coloured using the palette set in the `paint_palette` option.

**Value**

a character vector with ANSI colour codes embedded. use `cat()` to see the coloured message.

---

unpaint

*Call the default print methods instead of paint once*

---

**Description**

If you're using `paint()` as your default `print()` for dataframes, you may occasionally want to see an object printed using its default method. This function does that.

**Usage**

```
unpaint(df = .Last.value)
```

**Arguments**

`df` the object to print.

**Details**

Either supply the object as the `df` arg, or if it has just been painted, you can call `unpaint()` with no arguments to print the `.Last.value`.

**See Also**

[mask\\_print\(\)](#) to use [paint\(\)](#) by default for dataframes in the current session over `print()`.

# Index

brewer\_accent\_7 (rainbow\_6), 4  
brewer\_dark2\_7 (rainbow\_6), 4  
brewer\_pastel1\_8 (rainbow\_6), 4  
brewer\_pastel2\_7 (rainbow\_6), 4  
brewer\_set3\_12 (rainbow\_6), 4

ipaint, 2

mask\_print, 2  
mask\_print(), 5

paint, 3  
paint(), 2, 5  
pillar::glimpse(), 3

rainbow\_6, 4  
rainbow\_text, 4

unmask\_print (mask\_print), 2  
unpaint, 5  
unpaint(), 3, 4

viridis\_6 (rainbow\_6), 4