

# Package ‘styler’

December 11, 2017

**Title** Non-Invasive Pretty Printing of R Code

**Version** 1.0.0

**Description** Pretty-prints R code without changing the user's formatting intent.

**Imports** backports, cli, enc, magrittr, purrr, rematch2, rlang,  
rprojroot, tibble, withr

**Suggests** data.tree, dplyr, here, knitr, mockr, rmarkdown, rstudioapi,  
testthat

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Date** 2017-12-05

**BugReports** <https://github.com/r-lib/styler/issues>

**URL** <https://github.com/r-lib/styler>, <https://r-lib.github.io/styler/>

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**Collate** 'adds.R' 'communicate.R' 'compat-tidyr.R' 'dplyr.R'  
'expr-is.R' 'indent.R' 'initialize.R' 'nest.R'  
'nested\_to\_tree.R' 'parse.R' 'reindent.R' 'token-define.R'  
'relevel.R' 'rules-line\_break.R' 'rules-other.R'  
'rules-replacement.R' 'rules-spacing.R' 'serialize.R'  
'serialized\_tests.R' 'set-assert-args.R' 'style\_guides.R'  
'styler.R' 'token-create.R' 'transform-code.R'  
'transform-files.R' 'ui.R' 'unindent.R' 'utils.R' 'vertical.R'  
'visit.R' 'zzz.R'

**NeedsCompilation** no

**Author** Kirill Müller [aut],  
Lorenz Walthert [cre, aut]

**Maintainer** Lorenz Walthert <[lorenz.walthert@icloud.com](mailto:lorenz.walthert@icloud.com)>

**Repository** CRAN

**Date/Publication** 2017-12-11 12:52:26 UTC

## R topics documented:

styler-package . . . . .	2
construct_vertical . . . . .	3
create_style_guide . . . . .	3
default_style_guide_attributes . . . . .	4
math_token_spacing . . . . .	5
reindentation . . . . .	6
styler_addins . . . . .	7
style_dir . . . . .	7
style_file . . . . .	8
style_guides . . . . .	9
style_pkg . . . . .	10
style_text . . . . .	11
tidyverse_style . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

styler-package	<i>Non-invasive pretty printing of R code</i>
----------------	---

---

### Description

styler allows you to format .R files, packages or entire R source trees according to a style guide. The following functions can be used for styling:

- `style_text()` to style a character vector.
- `style_file()` to style a single .R file.
- `style_dir()` to style all .R files in a directory.
- `style_pkg()` to style the source files of an R package.
- `styler_addins` (RStudio Addins) to style either selected code or the active file.

### Author(s)

**Maintainer:** Lorenz Walthert <lorenz.walthert@icloud.com>

Authors:

- Kirill Müller <kr1mlr+r@mailbox.org>

### See Also

Useful links:

- <https://github.com/r-lib/styler>
- <https://r-lib.github.io/styler/>
- Report bugs at <https://github.com/r-lib/styler/issues>

**Examples**

```

style_text("call( 1)")
style_text("1 + 1", strict = FALSE)
style_text("a%>b", scope = "spaces")
style_text("a%>b; a", scope = "line_breaks")
style_text("a%>b; a", scope = "tokens")

```

---

construct\_vertical     *Construct an object of class vertical*

---

**Description**

Sole purpose of the class vertical is to have a print method that aligns the output vertically.

**Usage**

```
construct_vertical(x)
```

**Arguments**

x                    A character vector or an object of class "vertical".

---

create\_style\_guide     *Create a style guide*

---

**Description**

This is a helper function to create a style guide, which is technically speaking a named list of groups of transformer functions where each transformer function corresponds to one styling rule. The output of this function can be used as an argument for style in top level functions like [style\\_text\(\)](#) and friends.

**Usage**

```

create_style_guide(initialize = default_style_guide_attributes,
  line_break = NULL, space = NULL, token = NULL, indention = NULL,
  use_raw_indention = FALSE, reindention = tidyverse_reindention())

```

**Arguments**

initialize	The bare name of a function that initializes various variables on each level of nesting.
line_break	A list of transformer functions that manipulate line_break information.
space	A list of transformer functions that manipulate spacing information.
token	A list of transformer functions that manipulate token text.
indentation	A list of transformer functions that manipulate indentation.
use_raw_indentation	Boolean indicating whether or not the raw indentation should be used.
reindentation	A list of parameters for regex re-indentation, most conveniently constructed using <a href="#">specify_reindentation()</a> .

**Examples**

```

set_line_break_before_curly_opening <- function(pd_flat) {
  op <- pd_flat$token %in% "'{'"
  pd_flat$lag_newlines[op] <- 1L
  pd_flat
}
set_line_break_before_curly_opening_style <- function() {
  create_style_guide(line_break = set_line_break_before_curly_opening)
}
style_text("a <- function(x) { x }", style = set_line_break_before_curly_opening_style)

```

---

default\_style\_guide\_attributes

*Initialize default style guide attributes*

---

**Description**

This function initialises and removes various variables from the parse table.

**Usage**

```
default_style_guide_attributes(pd_flat)
```

**Arguments**

pd_flat	A parse table.
---------	----------------

**Examples**

```

string_to_format <- "call( 3)"
pd <- styler:::compute_parse_data_nested(string_to_format)
styler:::pre_visit(pd, c(default_style_guide_attributes))

```

---

math_token_spacing	<i>Specify spacing around math tokens</i>
--------------------	---

---

### Description

Helper function to create the input for the argument `math_token_spacing` in `tidyverse_style()`.

### Usage

```
specify_math_token_spacing(zero = NULL, one = c("'+'", "'-'", "'*'", "'/'",  
  "'^'"))
```

```
tidyverse_math_token_spacing()
```

### Arguments

<code>zero</code>	Character vector of tokens that should be surrounded with zero spaces.
<code>one</code>	Character vector with tokens that should be surrounded by at least one space (depending on <code>strict = TRUE</code> in the styling functions <code>style_text()</code> and friends). See 'Examples'.

### Functions

- `specify_math_token_spacing`: Allows to fully specify the math token spacing.
- `tidyverse_math_token_spacing`: Simple forwarder to `specify_math_token_spacing` with spacing around math tokens according to the tidyverse style guide.

### Examples

```
style_text(  
  "1+1 -3",  
  math_token_spacing = specify_math_token_spacing(zero = "'+'"),  
  strict = FALSE  
)  
style_text(  
  "1+1 -3",  
  math_token_spacing = specify_math_token_spacing(zero = "'+'"),  
  strict = TRUE  
)  
style_text(  
  "1+1 -3",  
  math_token_spacing = tidyverse_math_token_spacing(),  
  strict = TRUE  
)  
style_text(  
  "1+1 -3",  
  math_token_spacing = tidyverse_math_token_spacing(),  
  strict = TRUE  
)
```

---

reindentation	<i>Specify what is re-indented how</i>
---------------	--

---

## Description

This function returns a list that can be used as an input for the argument `reindentation` of the function `tidyverse_style()`. It features sensible defaults, so the user can specify deviations from them conveniently without the need of setting all arguments explicitly.

## Usage

```
specify_reindentation(regex_pattern = NULL, indentation = 0,  
  comments_only = TRUE)  
  
tidyverse_reindentation()
```

## Arguments

<code>regex_pattern</code>	Character vector with regular expression patterns that are to be re-indented with spaces, NULL if no reindentation needed.
<code>indentation</code>	The indentation tokens should have if they match <code>regex_pattern</code> .
<code>comments_only</code>	Whether the <code>regex_reindentation_pattern</code> should only be matched against comments or against all tokens. Mainly added for performance.

## Functions

- `specify_reindentation`: Allows to specify which tokens are reindented and how.
- `tidyverse_reindentation`: Simple forwarder to `specify_reindentation` with reindentation according to the tidyverse style guide.

## Examples

```
style_text("a <- xyz", reindentation = specify_reindentation(  
  regex_pattern = "xyz", indentation = 4, comments_only = FALSE)  
)  
style_text("a <- xyz", reindentation = tidyverse_reindentation())
```

---

styler_addins	<i>Stylers for RStudio Addins</i>
---------------	-----------------------------------

---

### Description

Helper functions for styling via RStudio Addins.

### Usage

```
style_active_file()
```

```
style_selection()
```

### Functions

- `style_active_file`: Styles the active file with `tidyverse_style()` and `strict = TRUE`.
- `style_selection`: Styles the highlighted selection in a `.R` or `.Rmd` file.

### Auto-Save Option

By default, both of the RStudio Addins will apply styling to the (selected) file contents without saving changes. Automatic saving can be enabled by setting the environment variable `save_after_styling` to `TRUE`.

Consider setting this in your `.Rprofile` file if you want to persist this setting across multiple sessions. Untitled files will always need to be saved manually after styling.

### See Also

[Sys.setenv\(\)](#)

Other stylers: [style\\_dir](#), [style\\_file](#), [style\\_pkg](#), [style\\_text](#)

---

style_dir	<i>Prettify arbitrary R code</i>
-----------	----------------------------------

---

### Description

Performs various substitutions in all `.R` files in a directory. Carefully examine the results after running this function!

### Usage

```
style_dir(path = ".", ..., style = tidyverse_style,  
          transformers = style(...), filetype = "R", recursive = TRUE,  
          exclude_files = NULL)
```

**Arguments**

path	Path to a directory with files to transform.
...	Arguments passed on to the style function.
style	A function that creates a style guide to use, by default <code>tidyverse_style()</code> (without the parentheses). Not used further except to construct the argument transformers. See <code>style_guides()</code> for details.
transformers	A set of transformer functions. This argument is most conveniently constructed via the style argument and .... See 'Examples'.
filetype	Vector of file extensions indicating which filetypes should be styled. Case is ignored, and the . is optional, e.g. <code>c(".R", ".Rmd")</code> or <code>c("r", "rmd")</code> .
recursive	A logical value indicating whether or not files in subdirectories of path should be styled as well.
exclude_files	Character vector with paths to files that should be excluded from styling.

**Value**

Invisibly returns a data frame that indicates for each file considered for styling whether or not it was actually changed.

**Warning**

This function overwrites files (if styling results in a change of the code to be formatted). It is strongly suggested to only style files that are under version control or to create a backup copy.

**See Also**

Other stylers: [style\\_file](#), [style\\_pkg](#), [style\\_text](#), [styler\\_addins](#)

**Examples**

```
## Not run:
style_dir(file_type = "r")

## End(Not run)
```

---

style\_file

*Style .R and/or .Rmd files*

---

**Description**

Performs various substitutions in the files specified. Carefully examine the results after running this function!

**Usage**

```
style_file(path, ..., style = tidyverse_style, transformers = style(...))
```



## Arguments

path	A character vector with paths to files to style.
...	Arguments passed on to the style function.
style	A function that creates a style guide to use, by default <code>tidyverse_style()</code> (without the parentheses). Not used further except to construct the argument transformers. See <code>style_guides()</code> for details.
transformers	A set of transformer functions. This argument is most conveniently constructed via the style argument and ... See 'Examples'.

## Value

Invisibly returns a data frame that indicates for each file considered for styling whether or not it was actually changed.

## Warning

This function overwrites files (if styling results in a change of the code to be formatted). It is strongly suggested to only style files that are under version control or to create a backup copy.

## See Also

Other stylers: [style\\_dir](#), [style\\_pkg](#), [style\\_text](#), [styler\\_addins](#)

## Examples

```
# the following is identical but the former is more convenient:
file <- tempfile("styler", fileext = ".R")
enc::write_lines_enc("1++1", file)
style_file(file, style = tidyverse_style, strict = TRUE)
style_file(file, transformers = tidyverse_style(strict = TRUE))
enc::read_lines_enc(file)
unlink(file)
```

---

style\_guides

*Style guides*

---

## Description

Format code according to a style guide. Style guides are the input to the argument style in `style_file()` and friends. The available style guides are:

- the tidyverse style guide (see `tidyverse_style()`).

---

style\_pkg

*Prettify R source code*

---

## Description

Performs various substitutions in all .R files in a package (code and tests). Carefully examine the results after running this function!

## Usage

```
style_pkg(pkg = ".", ..., style = tidyverse_style,  
          transformers = style(...), filetype = "R",  
          exclude_files = "R/RcppExports.R")
```

## Arguments

pkg	Path to a (subdirectory of an) R package.
...	Arguments passed on to the style function.
style	A function that creates a style guide to use, by default <code>tidyverse_style()</code> (without the parentheses). Not used further except to construct the argument transformers. See <code>style_guides()</code> for details.
transformers	A set of transformer functions. This argument is most conveniently constructed via the style argument and <code>...</code> . See 'Examples'.
filetype	Vector of file extensions indicating which filetypes should be styled. Case is ignored, and the . is optional, e.g. <code>c(".R", ".Rmd")</code> or <code>c("r", "rmd")</code> .
exclude_files	Character vector with paths to files that should be excluded from styling.

## Warning

This function overwrites files (if styling results in a change of the code to be formatted). It is strongly suggested to only style files that are under version control or to create a backup copy.

## Value

Invisibly returns a data frame that indicates for each file considered for styling whether or not it was actually changed.

## See Also

Other stylers: [style\\_dir](#), [style\\_file](#), [style\\_text](#), [styler\\_addins](#)

**Examples**

```
## Not run:

style_pkg(style = tidyverse_style, strict = TRUE)
style_pkg(
  scope = "line_breaks",
  math_token_spacing = specify_math_token_spacing(zero = "'+'")
)

## End(Not run)
```

---

style_text	<i>Style a string</i>
------------	-----------------------

---

**Description**

Styles a character vector. Each element of the character vector corresponds to one line of code.

**Usage**

```
style_text(text, ..., style = tidyverse_style, transformers = style(...))
```

**Arguments**

text	A character vector with text to style.
...	Arguments passed on to the style function.
style	A function that creates a style guide to use, by default <a href="#">tidyverse_style()</a> (without the parentheses). Not used further except to construct the argument transformers. See <a href="#">style_guides()</a> for details.
transformers	A set of transformer functions. This argument is most conveniently constructed via the style argument and .... See 'Examples'.

**See Also**

Other stylers: [style\\_dir](#), [style\\_file](#), [style\\_pkg](#), [styler\\_addins](#)

**Examples**

```
style_text("call( 1)")
style_text("1 + 1", strict = FALSE)
style_text("a%>%b", scope = "spaces")
style_text("a%>%b; a", scope = "line_breaks")
style_text("a%>%b; a", scope = "tokens")
# the following is identical but the former is more convenient:
style_text("a<-3++1", style = tidyverse_style, strict = TRUE)
style_text("a<-3++1", transformers = tidyverse_style(strict = TRUE))
```

---

tidyverse_style	<i>The tidyverse style</i>
-----------------	----------------------------

---

## Description

Style code according to the tidyverse style guide.

## Usage

```
tidyverse_style(scope = "tokens", strict = TRUE, indent_by = 2,
  start_comments_with_one_space = FALSE,
  reindentation = tidyverse_reindentation(),
  math_token_spacing = tidyverse_math_token_spacing())
```

## Arguments

scope	The extent of manipulation. Can range from "none" (least invasive) to "token" (most invasive). See 'Details'. This argument is a vector of length one.
strict	A logical value indicating whether a set of strict or not so strict transformer functions should be returned. Compare the functions returned with or without <code>strict = TRUE</code> . For example, <code>strict = TRUE</code> means force <i>one</i> space e.g. after <code>","</code> and <i>one</i> line break e.g. after a closing curly brace. <code>strict = FALSE</code> means to set spaces and line breaks to one if there is none and leave the code untouched otherwise. See 'Examples'.
indent_by	How many spaces of indentation should be inserted after operators such as <code>'(</code> .
start_comments_with_one_space	Whether or not comments should start with only one space (see <code>start_comments_with_space()</code> ).
reindentation	A list of parameters for regex re-indentation, most conveniently constructed using <a href="#">specify_reindentation()</a> .
math_token_spacing	A list of parameters that define spacing around math token, conveniently constructed using <a href="#">specify_math_token_spacing()</a> .

## Details

The following options for scope are available.

- "none": Performs no transformation at all.
- "spaces": Manipulates spacing between token on the same line.
- "indentation": In addition to "spaces", this option also manipulates the indentation level.
- "line\_breaks": In addition to "indentation", this option also manipulates line breaks.
- "tokens": In addition to "line\_breaks", this option also manipulates tokens.

As it becomes clear from this description, more invasive operations can only be performed if all less invasive operations are performed too.

**Examples**

```
style_text("call( 1)", style = tidyverse_style, scope = "spaces")
style_text("call( 1)", transformers = tidyverse_style(strict = TRUE))
style_text(c("ab <- 3", "a <-3"), strict = FALSE) # keeps alignment of "<-"
style_text(c("ab <- 3", "a <-3"), strict = TRUE) # drops alignment of "<-"
```

# Index

`construct_vertical`, 3  
`create_style_guide`, 3

`default_style_guide_attributes`, 4

`math_token_spacing`, 5

`reindentation`, 6

`specify_math_token_spacing`  
  (`math_token_spacing`), 5  
`specify_math_token_spacing()`, 12  
`specify_reindentation` (`reindentation`), 6  
`specify_reindentation()`, 4, 12  
`style_active_file` (`styler_addins`), 7  
`style_dir`, 7, 7, 9–11  
`style_dir()`, 2  
`style_file`, 7, 8, 8, 10, 11  
`style_file()`, 2, 9  
`style_guides`, 9  
`style_guides()`, 8–11  
`style_pkg`, 7–9, 10, 11  
`style_pkg()`, 2  
`style_selection` (`styler_addins`), 7  
`style_text`, 7–10, 11  
`style_text()`, 2, 3, 5  
`styler` (`styler-package`), 2  
`styler-package`, 2  
`styler_addins`, 2, 7, 8–11  
`Sys.setenv()`, 7

`tidyverse_math_token_spacing`  
  (`math_token_spacing`), 5  
`tidyverse_reindentation` (`reindentation`), 6  
`tidyverse_style`, 12  
`tidyverse_style()`, 5–11