

# Package ‘RTLknitr’

May 7, 2026

**Type** Package

**Title** Right to Left Dynamic Documents Using 'knitr'

**Version** 1.1.1

**Date** 2025-11-21

**Maintainer** Foad Esmaeili <foadesmaeili5@gmail.com>

**Description** Provide seamless support for right-to-left (RTL) languages, such as Persian and Arabic, in R Markdown documents and 'LaTeX' output. It includes functions and hooks that enable easy integration of RTL language content, allowing users to create documents that adhere to RTL writing conventions. For in-depth insights into dynamic documents and the 'knitr' package, consider referring to Xie, Y (2014) <ISBN: 978-1-482-20353-0>.

**License** GPL (>= 2)

**URL** <https://github.com/FoadEsmaeili5/RTLknitr>

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** R (>= 4.1)

**Imports** knitr, magrittr, gt, bookdown

**NeedsCompilation** no

**Author** Foad Esmaeili [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0002-9638-0807>>)

**Repository** CRAN

**Date/Publication** 2025-11-21 16:30:02 UTC

## Contents

Centering . . . . .	2
hooks_rtl . . . . .	2
markdown_rtl . . . . .	3
markdown_rtl_header . . . . .	4
remove_escape_latex . . . . .	5
tab_header_lbl . . . . .	5

**Index**[7](#)


---

Centering	<i>Centering Caption of LaTeX Table</i>
-----------	---

---

**Description**

This function adds the "\centering" command to the caption of a LaTeX table, allowing the caption to be centered in the final output.

**Usage**

```
Centering(x)
```

**Arguments**

x                    LaTeX-formatted table with a caption.

**Value**

LaTeX-formatted table with a centered caption.

**Examples**

```
latex_table <- "\\begin{table}\\caption{This is a caption.}\\end{table}"
centered_table <- Centering(latex_table)
print(centered_table)
```

---

hooks_rtl	<i>Customize knitr Hooks for Right-to-Left (RTL) Text Formatting</i>
-----------	--

---

**Description**

This function defines customized knitr hooks to handle RTL (right-to-left) text formatting for languages like Persian and Arabic for Sweave. It allows you to modify the appearance of output, messages, warnings, and errors when producing RTL documents.

**Usage**

```
hooks_rtl(fontcolor = "black", bgcolor = "gray")
```

**Arguments**

fontcolor            Font color for inline output. Default is "black".  
 bgcolor              Background color for inline output. Default is "gray".

## Value

The function returns a list of customized `knit_hooks` that handle RTL text formatting. It modifies the output, messages, warnings, errors, inline expressions, and source code to include appropriate commands for RTL languages.

When `'echo=FALSE'` you need to use `'rtl=TRUE'` in order to get the output appropriately.

## References

Xie, Y. (2014). *Dynamic Documents with R and knitr*. Chapman & Hall/CRC. ISBN: 978-1482203530.

## See Also

Use [markdown\\_rtl](#) to enable RTL formatting for Markdown documents.

## Examples

```
# Define custom RTL hooks
rtl_hooks <- hooks_rtl()

# Apply custom RTL hooks to `knit_hooks`
knitr::knit_hooks$set(rtl_hooks)

# After setting the hooks, knit your document as usual to produce RTL formatting.
```

---

markdown\_rtl

*Enable Right-to-Left (RTL) Formatting for Markdown Documents*

---

## Description

This function modifies the behavior of `knit` output hooks to handle RTL (right-to-left) text formatting for Markdown documents, particularly for languages like Persian and Arabic.

## Usage

```
markdown_rtl(...)
```

## Arguments

... Additional arguments to be passed to the underlying `knit_hooks$set` function.

## Details

The `markdown_rtl` function wraps the existing `knit_hooks` with custom commands that switch the text direction between Latin and Persian. It enhances the appearance of the output, messages, warnings, and errors in RTL languages.

**Value**

The function has no explicit return value. It modifies the behavior of `knit_hooks` for RTL formatting.

**References**

Xie, Y. (2014). *Dynamic Documents with R and knitr*. Chapman & Hall/CRC. ISBN: 978-1482203530.

**See Also**

Use [hooks\\_rtl](#) for handling RTL formatting within code chunks.

**Examples**

```
# Enable RTL formatting for the current R Markdown document
markdown_rtl()

# After enabling RTL formatting, proceed to knit the R Markdown document as usual.
# The output and messages will be formatted for RTL languages.
```

---

markdown_rtl_header	<i>Generate a YAML header for an R Markdown document with RTL support</i>
---------------------	---

---

**Description**

This function generates a YAML header for an R Markdown document that will produce a PDF output with support for Persian typesetting using the `xepersian` package and the `Arial` font.

**Usage**

```
markdown_rtl_header()
```

**Value**

A character string containing the YAML header.

**Examples**

```
markdown_rtl_header()
```

---

remove\_escape\_latex     *Remove Escaped LaTeX Formulas from Text*

---

### Description

This function removes escaped LaTeX formulas and formatting commands from LaTeX-formatted text.

### Usage

```
remove_escape_latex(x)
```

### Arguments

x                    A character vector or string containing LaTeX-formatted text.

### Value

A character vector with escaped LaTeX formulas and formatting commands removed.

### Examples

```
# Remove escaped LaTeX formulas from a string
cleaned_text <- remove_escape_latex("\\\\textbf\\{Hello\\}, $\\alpha$ formula")
print(cleaned_text)
```

---

tab\_header\_lbl             *Create a Labeled Header for gt Tables*

---

### Description

This function enhances the gt package's tab\_header function by allowing the inclusion of a label for referencing the table in LaTeX documents.

### Usage

```
tab_header_lbl(data, title, subtitle = NULL, label = NULL)
```

### Arguments

data                 A gt table object.  
title                Title of the table.  
subtitle             Subtitle of the table.  
label                Label for referencing the table in LaTeX.

**Details**

Be sure to use this function at the end of `gt` format functions. This function returns a LaTeX-formatted table header.

**Value**

LaTeX-formatted table header.

**Examples**

```
require(gt)
head(iris) |> gt() |> tab_header_lbl(title = "iris",label = "tab:iris") |> cat()
```

# Index

Centering, [2](#)

hooks\_rtl, [2](#), [4](#)

knit, [3](#)

markdown\_rtl, [3](#), [3](#)

markdown\_rtl\_header, [4](#)

remove\_escape\_latex, [5](#)

tab\_header\_lbl, [5](#)