

# Package ‘typr’

April 22, 2025

**Title** Write and Render 'Typst' Documents

**Version** 0.0.1

**Description** Compile 'Typst' files using the 'typst-cli' (<<https://typst.app>>) command line tool. Automatically falls back to rendering via embedded 'Typst' from 'Quarto' (<<https://quarto.org>>) if 'Typst' is not installed. Includes utilities to check for 'typst-cli' availability and run 'Typst' commands.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** cli, fs, processx, rlang

**Depends** R (>= 4.1.0)

**Suggests** jsonlite, rstudioapi, testthat (>= 3.0.0)

**SystemRequirements** Typst command line tool (<<https://github.com/typst/typst>>) or Quarto command line tool (<<https://github.com/quarto-dev/quarto-cli>>).

**Config/testthat/edition** 3

**URL** <http://christophertkenny.com/typr/>

**NeedsCompilation** no

**Author** Christopher T. Kenny [aut, cre]  
(<<https://orcid.org/0000-0002-9386-6860>>)

**Maintainer** Christopher T. Kenny <[christopherkenny@fas.harvard.edu](mailto:christopherkenny@fas.harvard.edu)>

**Repository** CRAN

**Date/Publication** 2025-04-22 14:20:02 UTC

## Contents

has_any_typst . . . . .	2
typr_compile . . . . .	3
typr_fonts . . . . .	3
typr_help . . . . .	4

typr_init . . . . .	4
typr_query . . . . .	5
typr_version . . . . .	6
typr_watch . . . . .	6
typr_which_typst . . . . .	7
<b>Index</b>	<b>8</b>

---

has_any_typst	<i>Find if Typst is available</i>
---------------	-----------------------------------

---

## Description

Find if Typst is available

## Usage

has\_any\_typst()

has\_typst\_cli()

has\_quarto\_typst()

has\_quarto\_cli()

typr\_has\_typst()

typr\_has\_quarto()

typr\_has\_quarto\_typst()

## Value

a logical if Typst or Quarto are found

## Examples

```
has_any_typst()
has_typst_cli()
has_quarto_cli()
has_quarto_typst()
```

---

typr_compile	<i>Compile a Typst document</i>
--------------	---------------------------------

---

**Description**

Compile a Typst document

**Usage**

```
typr_compile(  
  input = NULL,  
  output_file = NULL,  
  output_format = c("pdf", "png", "svg", "html"),  
  typst_args = NULL  
)
```

**Arguments**

`input` either a path to a .typ file or text to use as the file  
`output_file` file to output to  
`output_format` format to use. One of c('pdf', 'png', 'svg', 'html'),  
`typst_args` Additional arguments to pass to Typst. Can be listed with `typr_help('compile')`

**Value**

a path to the created file

**Examples**

```
typr_compile('hello world')  
typr_compile(  
  input = 'Hello world', output_format = 'png',  
  typst_args = c('--ppi', '300')  
)
```

---

typr_fonts	<i>List Available Typst Fonts</i>
------------	-----------------------------------

---

**Description**

List Available Typst Fonts

**Usage**

```
typr_fonts()
```

**Value**

a character vector of font names

**Examples**

```
typr_fonts()
```

---

 typr\_help

*Get Help Information for a Typst command*


---

**Description**

Prints a message to the terminal with help information. Silently returns the content as character which can be printed with `cat()`.

**Usage**

```
typr_help(command = NULL)
```

**Arguments**

command            name of a command to get help for, e.g. 'compile'

**Value**

a character vector of information, invisibly

**Examples**

```
typr_help('compile')
```

---

 typr\_init

*Create a project from a template*


---

**Description**

Create a project from a template

**Usage**

```
typr_init(template = NULL, version = NULL, dir = NULL, typst_args = NULL)
```

**Arguments**

template	a template to use, can be local or on Typst Universe. If the string does not specify @preview or @local, '@preview' will be prepended.
version	a version string, e.g. 0.1.0
dir	The project directory
typst_args	Additional arguments to pass to Typst. Can be listed with typr_help('init')

**Value**

a directory, invisibly

**Examples**

```
typr_init('charged-ieee', dir = paste0(tempdir(), '/typr_example'))
```

---

typr_query	<i>Query within a Typst file</i>
------------	----------------------------------

---

**Description**

Query within a Typst file

**Usage**

```
typr_query(
  input = NULL,
  selector = NULL,
  typst_args = NULL,
  as_json = !rlang::is_installed("jsonlite")
)
```

**Arguments**

input	a path to a .typ file
selector	a string to search for, e.g. a "<label>"
typst_args	Additional arguments to pass to Typst. Can be listed with typr_help('query')
as_json	if TRUE, return a list, otherwise a character string

**Value**

a length one character with json syntax if !as\_json, otherwise a list

**Examples**

```
tf <- fs::file_temp(ext = 'typ')
writeLines(text = '= test <lbl>', con = tf)
typr_query(input = tf, selector = '<lbl>')
```

---

typr_version	<i>Get installed Typst version</i>
--------------	------------------------------------

---

**Description**

Get installed Typst version

**Usage**

```
typr_version(keep_hash = FALSE)
```

**Arguments**

keep_hash	description
-----------	-------------

**Value**

a version, numeric if !keep\_hash

**Examples**

```
typr_version()
```

---

typr_watch	<i>Watch a Typst document</i>
------------	-------------------------------

---

**Description**

Watch a Typst document

**Usage**

```
typr_watch(
  input = NULL,
  output_file = NULL,
  output_format = c("pdf", "png", "svg", "html"),
  typst_args = NULL
)

typr_watch_stop()
```

**Arguments**

`input` either a path to a .typ file or text to use as the file  
`output_file` file to output to  
`output_format` format to use. One of c('pdf', 'png', 'svg', 'html'),  
`typst_args` Additional arguments to pass to Typst. Can be listed with `typr_help('watch')`

**Value**

a path to the created file

**Examples**

```
tf <- fs::file_temp(ext = 'typ')
writeLines(text = '= test', con = tf)
typr_watch(tf)
writeLines(text = c('= test', 'hello'), con = tf)
```

---

typr\_which\_typst      *Identify Install Typst to Use*

---

**Description**

Identify Install Typst to Use

**Usage**

```
typr_which_typst()
```

**Value**

a character vector or NULL if no Typst is found

**Examples**

```
typr_which_typst()
```

# Index

`has_any_typst`, [2](#)  
`has_quarto_cli` (`has_any_typst`), [2](#)  
`has_quarto_typst` (`has_any_typst`), [2](#)  
`has_typst_cli` (`has_any_typst`), [2](#)

`typr_compile`, [3](#)  
`typr_fonts`, [3](#)  
`typr_has_quarto` (`has_any_typst`), [2](#)  
`typr_has_quarto_typst` (`has_any_typst`), [2](#)  
`typr_has_typst` (`has_any_typst`), [2](#)  
`typr_help`, [4](#)  
`typr_init`, [4](#)  
`typr_query`, [5](#)  
`typr_version`, [6](#)  
`typr_watch`, [6](#)  
`typr_watch_stop` (`typr_watch`), [6](#)  
`typr_which_typst`, [7](#)