

Package ‘liteformats’

August 24, 2026

Title Lightweight Output Formats for 'litedown'

Version 0.2.0

Description A collection of lightweight, minimalist output formats and templates for 'litedown' by Xie (2026) <[doi:10.32614/CRAN.package.litedown](https://doi.org/10.32614/CRAN.package.litedown)>, including resumes, cover letters, and other common document types. Documents are rendered with HTML and CSS and can be printed to PDF with a 'Chromium'-based browser, without requiring 'Pandoc' or a 'LaTeX' installation.

License MIT + file LICENSE

URL <https://nanx.me/liteformats/>,
<https://github.com/nanxstats/liteformats>

BugReports <https://github.com/nanxstats/liteformats/issues>

Encoding UTF-8

Imports litedown (>= 0.10), xfun (>= 0.60)

Suggests testit

VignetteBuilder litedown

SystemRequirements Chromium, Google Chrome, or Microsoft Edge (for PDF output)

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Nan Xiao [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-0250-5673>>),
Yihui Xie [ctb, cph] (pages.js and pages.css libraries),
Ayuhito [ctb, cph] (google-font-metadata catalog)

Maintainer Nan Xiao <me@nanx.me>

Repository CRAN

Date/Publication 2026-08-24 05:20:09 UTC

Contents

cover_letter	2
cover_letter_options	3
include_graphics	5
resume	5
resume_options	7
use_resume	9
Index	10

cover_letter	<i>Render a cover letter</i>
--------------	------------------------------

Description

Render a two-column cover letter to self-contained HTML or PDF. The narrow left column holds the sender's details, while the letter remains the visual foreground in the wider right column.

Usage

```
cover_letter(
  input,
  output = NULL,
  options = cover_letter_options(),
  envir = parent.frame()
)
```

Arguments

input	Path to a .Rmd or .md source document.
output	Output path ending in .html or .pdf. If NULL, an HTML file is created beside input.
options	A cover letter configuration created by cover_letter_options() .
envir	Environment in which R code in input is evaluated.

Value

The normalized output path, invisibly.

Examples

```
source <- use_cover_letter(tempfile(fileext = ".Rmd"))
cover_letter(source, tempfile(fileext = ".html"))

## Not run:
# PDF output requires a local Chromium-based browser
cover_letter(source, "cover-letter.pdf")

## End(Not run)
```

cover_letter_options *Configure a cover letter*

Description

Create a classed options object for `cover_letter()`. Settings work in the same way as `resume_options()`, with letter-specific metadata replacing the resume-specific fields.

Usage

```
cover_letter_options(  
  author = NULL,  
  address = NULL,  
  phone = NULL,  
  email = NULL,  
  website = NULL,  
  date = NULL,  
  greeting = NULL,  
  font_family = NULL,  
  google_font = NULL,  
  font_files = NULL,  
  font_size = NULL,  
  font_scale = NULL,  
  ligatures = NULL,  
  line_height = NULL,  
  paragraph_spacing = NULL,  
  section_spacing = NULL,  
  margins = NULL,  
  paper = NULL,  
  link_color = NULL,  
  paged = NULL,  
  keep_html = FALSE,  
  browser = NULL  
)
```

Arguments

author, address, phone, email, website	Contact metadata.
date, greeting	Letter date and salutation.
font_family	A CSS font-family value. For example, "Charter, Georgia, serif".
google_font	A Google Fonts family name or a full Google Fonts CSS v2 URL. This is an explicit online resource and is never used by default. A family name loads the available weights and styles needed for regular, bold, italic, and bold italic text, preferring a variable weight range. Supply a full URL to select other weights, axes, or options.

font_files	A named character vector or list of local font files. Names may be regular, italic, bold, and bold_italic. Relative paths are resolved from the source document. Fonts are embedded in the HTML.
font_size	A CSS length such as "11pt".
font_scale	A positive number used to scale font_size. This is useful when replacing a font with different metrics.
ligatures	Whether to enable common and discretionary ligatures. The template default is FALSE for more predictable text extraction from PDFs.
line_height	A positive unitless CSS line height or CSS length.
paragraph_spacing, section_spacing	CSS lengths controlling vertical rhythm.
margins	Page margins. Supply one to four CSS lengths using standard CSS shorthand order, or numeric values interpreted as inches.
paper	A paper name ("letter", "a4", or "legal") or a two-element vector containing width and height as CSS lengths.
link_color	Any valid CSS color.
paged	Whether the HTML preview should open in a paginated layout. Printing always invokes the packaged pages.js pagination.
keep_html	For PDF output, whether to retain the intermediate HTML beside the PDF. Defaults to FALSE.
browser	Path to Chromium, Google Chrome, or Microsoft Edge. NULL lets <code>xfun::browser_print()</code> discover it.

Value

A `liteformats_cover_letter_options` object for the options argument of `cover_letter()`.

YAML configuration

Use the same flat, kebab-case `liteformats` mapping documented in `resume_options()`. Place cover-letter metadata such as author, address, date, and greeting at the top level of the YAML header. Unknown `liteformats` YAML names are rejected.

Examples

```
cover_letter_options(
  greeting = "Dear Search Committee:",
  paragraph_spacing = "0.9em"
)
```

include_graphics	<i>Include local graphics in a liteformat</i>
------------------	---

Description

Embed one or more local graphic files as base64 data URIs and return raw HTML that `litedown::fuse()` can include directly in a document. This keeps the rendered document self-contained without copying image files beside the source or output.

Usage

```
include_graphics(path, alt = "", class = NULL)
```

Arguments

<code>path</code>	A character vector of paths to local graphic files.
<code>alt</code>	Alternative text. Supply one string for all graphics or one string per path.
<code>class</code>	An optional CSS class string added to every image.

Value

Raw HTML output for use in a litedown code chunk.

Examples

```
signature <- system.file(  
  "liteformats/skeletons/signature.png",  
  package = "liteformats"  
)  
include_graphics(  
  signature,  
  alt = "Handwritten signature",  
  class = "signature"  
)
```

resume	<i>Render a resume</i>
--------	------------------------

Description

Render a Markdown or R Markdown resume to a self-contained HTML document or to PDF through a headless Chromium-based browser. The output format is inferred from the extension of output.

Usage

```
resume(  
  input,  
  output = NULL,  
  options = resume_options(),  
  envir = parent.frame()  
)
```

Arguments

input	Path to a .Rmd or .md source document.
output	Output path ending in .html or .pdf. If NULL, an HTML file is created beside input.
options	A resume configuration created by resume_options() .
envir	Environment in which R code in input is evaluated.

Details

Document metadata and typographic settings may be supplied in the YAML header. Values in options take precedence over YAML. See [resume_options\(\)](#) for function-call configuration and [use_resume\(\)](#) for a complete starter.

Value

The normalized output path, invisibly.

Examples

```
source <- use_resume(tempfile(fileext = ".Rmd"))  
resume(source, tempfile(fileext = ".html"))  
  
## Not run:  
# PDF output requires a local Chromium-based browser  
resume(source, "resume.pdf")  
  
# Embed local font files  
resume(  
  source,  
  options = resume_options(  
    font_files = c(  
      regular = "fonts/MySerif-Regular.woff2",  
      italic = "fonts/MySerif-Italic.woff2",  
      bold = "fonts/MySerif-Bold.woff2"  
    ),  
    font_scale = 0.96,  
    ligatures = FALSE  
  )  
)  
  
## End(Not run)
```

resume_options*Configure a resume*

Description

Create a classed options object for `resume()`. The helper keeps secondary metadata, typography, page, and PDF controls out of the main rendering function while retaining argument-name checking and autocomplete.

Usage

```
resume_options(  
  author = NULL,  
  job_title = NULL,  
  address = NULL,  
  phone = NULL,  
  email = NULL,  
  website = NULL,  
  linkedin = NULL,  
  font_family = NULL,  
  google_font = NULL,  
  font_files = NULL,  
  font_size = NULL,  
  font_scale = NULL,  
  ligatures = NULL,  
  line_height = NULL,  
  paragraph_spacing = NULL,  
  section_spacing = NULL,  
  margins = NULL,  
  paper = NULL,  
  link_color = NULL,  
  paged = NULL,  
  keep_html = FALSE,  
  browser = NULL  
)
```

Arguments

author, address, phone, email, website

Contact metadata.

job_title, linkedin

Resume-specific professional and contact metadata.

font_family

A CSS font-family value. For example, "Charter, Georgia, serif".

google_font

A Google Fonts family name or a full Google Fonts CSS v2 URL. This is an explicit online resource and is never used by default. A family name loads the available weights and styles needed for regular, bold, italic, and bold italic text,

	preferring a variable weight range. Supply a full URL to select other weights, axes, or options.
font_files	A named character vector or list of local font files. Names may be regular, italic, bold, and bold_italic. Relative paths are resolved from the source document. Fonts are embedded in the HTML.
font_size	A CSS length such as "11pt".
font_scale	A positive number used to scale font_size. This is useful when replacing a font with different metrics.
ligatures	Whether to enable common and discretionary ligatures. The template default is FALSE for more predictable text extraction from PDFs.
line_height	A positive unitless CSS line height or CSS length.
paragraph_spacing, section_spacing	CSS lengths controlling vertical rhythm.
margins	Page margins. Supply one to four CSS lengths using standard CSS shorthand order, or numeric values interpreted as inches.
paper	A paper name ("letter", "a4", or "legal") or a two-element vector containing width and height as CSS lengths.
link_color	Any valid CSS color.
paged	Whether the HTML preview should open in a paginated layout. Printing always invokes the packaged pages.js pagination.
keep_html	For PDF output, whether to retain the intermediate HTML beside the PDF. Defaults to FALSE.
browser	Path to Chromium, Google Chrome, or Microsoft Edge. NULL lets <code>xfun::browser_print()</code> discover it.

Details

Every argument defaults to NULL unless stated otherwise. A NULL metadata or appearance value falls back to the source document's YAML header and then to the template default. Values supplied here take precedence over YAML.

Value

A `liteformats_resume_options` object for the options argument of `resume()`.

YAML configuration

In a source document, place metadata at the top level of the YAML header and appearance controls in one flat `liteformats` mapping. YAML setting names are the kebab-case equivalents of the helper arguments:

```
author: "Jane Doe"
job-title: "Research Scientist"
liteformats:
  paper: letter
```

```

  margins: ["0.65in", "0.8in"]
  font-family: "Charter, Georgia, serif"
  font-size: 11pt
  font-scale: 0.94
  ligatures: false
  line-height: 1.15
  paragraph-spacing: 0.05em
  section-spacing: 1em
  link-color: "#385898"
  paged: true

```

Unknown liteformats YAML names are rejected to catch misspellings.

Examples

```

resume_options(
  font_size = "10.5pt",
  margins = c("0.6in", "0.75in"),
  paged = FALSE
)

```

use_resume

Create a resume or cover letter from a starter

Description

Copy an editable R Markdown starter document to path. Static assets used by the rendered formats remain inside the installed package and are referenced from the copied source instead of being copied beside it.

Usage

```

use_resume(path = "resume.Rmd", overwrite = FALSE)

use_cover_letter(path = "cover-letter.Rmd", overwrite = FALSE)

```

Arguments

path	Destination path.
overwrite	Whether to replace an existing file.

Value

The normalized destination path, invisibly.

Examples

```

use_resume(tempfile(fileext = ".Rmd"))
use_cover_letter(tempfile(fileext = ".Rmd"))

```

Index

`cover_letter`, [2](#)
`cover_letter()`, [3](#), [4](#)
`cover_letter_options`, [3](#)
`cover_letter_options()`, [2](#)

`include_graphics`, [5](#)

`litedown::fuse()`, [5](#)

`resume`, [5](#)
`resume()`, [7](#), [8](#)
`resume_options`, [7](#)
`resume_options()`, [3](#), [4](#), [6](#)

`use_cover_letter (use_resume)`, [9](#)
`use_resume`, [9](#)
`use_resume()`, [6](#)

`xfun::browser_print()`, [4](#), [8](#)