Package ‘ymlthis’

February 4, 2020

Title Write 'YAML' for 'R Markdown', 'bookdown', 'blogdown', and More

Version 0.1.2

Description Write 'YAML' front matter for R Markdown and related documents. yml_*() functions write 'YAML' and use_*() functions let you write the resulting 'YAML' to your clipboard or to .yml files related to your project.

License MIT + file LICENSE


BugReports https://github.com/r-lib/ymlthis/issues

Depends R (>= 3.2)

Imports crayon, fs, glue, magrittr, miniUI, purrr (>= 0.3.2), rlang (>= 0.4.0), rmarkdown, rstudioapi, shiny, shinyBS, stringr, useThis (>= 1.5.0), whoami, withr, yaml

Suggests blogdown, bookdown, covr, knitr, pkgdown, prettydoc, roxygen2 (>= 7.0.0), spelling, testthat, xaringan

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.0.2

NeedsCompilation no

Author Malcolm Barrett [aut, cre] (<https://orcid.org/0000-0003-0299-5825>), Richard Iannone [aut] (<https://orcid.org/0000-0003-3925-190X>), RStudio [cph, fnd]

Maintainer Malcolm Barrett <malcolmbarrett@gmail.com>

Repository CRAN

Date/Publication 2020-02-03 23:00:02 UTC
R topics documented:

asis_yaml_output ........................................ 3
as_yml ...................................................... 3
bib2yml ...................................................... 4
blogdown_template .......................................... 5
code_chunk ................................................... 5
draw_yaml_tree ............................................ 6
gitbook_config ............................................. 7
has_field ................................................... 9
includes2 .................................................... 10
is_yaml ..................................................... 10
last_yaml ................................................... 11
pagedown_business_card_template ...................... 11
pandoc_template_types .................................... 14
pkgdown_template ......................................... 15
read_json ................................................... 15
use_yaml .................................................... 17
use_yaml_defaults ....................................... 18
use_yaml_file ............................................. 19
yaml ....................................................... 21
yaml_author ............................................... 22
yaml_blank ................................................. 24
yaml_blogdown_opts ...................................... 25
yaml_bookdown_opts ....................................... 28
yaml_citations .......................................... 30
yaml_clean ............................................... 32
yaml_code ................................................. 33
yaml_distill_opts ....................................... 34
yaml_handlers ............................................ 37
yaml_latex_opts ......................................... 38
yaml_load ................................................. 41
yaml_output .............................................. 41
yaml_pagedown_opts ..................................... 42
yaml_params .............................................. 44
yaml_pkgdown ............................................ 48
yaml_reference ......................................... 52
yaml_replace ............................................. 54
yaml_resource_files ..................................... 56
yaml Rsconnect_email ................................... 56
yaml_rtcles_opts ....................................... 58
yaml_runtime ............................................ 60
yaml_site_opts .......................................... 61
yaml_toc .................................................. 63
yaml_verbatim .......................................... 64
yaml_vignette .......................................... 65

Index 67
asis_yaml_output

Export yml object as a YAML knitr code chunk

Description

asis_yaml_output() exports a yml object as a YAML knitr code chunk instead of as an R object. Doing so adds code highlighting for YAML syntax.

Usage

asis_yaml_output(.yml, fences = TRUE)

Arguments

- **.yml**: a yml object created by yml(), as_yml(), or returned by a yml_*() function
- **fences**: Logical. Write fences (“—”) before and after YAML?

See Also

Other yml: bib2yml(), draw_yml_tree(), has_field(), read_json(), use_yaml_defaults(), use_yaml_file(), use_yaml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(), yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml_rteicles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

as_yml

Convert to yml object

Description

as_yml is a wrapper for yaml::yaml.load() that stores YAML as a yml object, which prints cleanly to the console and is easy to work with using ymlthis functions.

Usage

as_yml(x)

Arguments

- **x**: An object, either a character vector of length 1 or list, to convert to yml.

Value

a yml object
Examples

```r
x <- as_yml("author: Hadley Wickham
date: '2014-09-12'
title: Tidy Data
keywords:
- data cleaning
- data tidying
- relational databases
- R")

x

x %>%
yml_subtitle("Hadley's Tidy Data Paper")
```

### bib2yml

**Convert bib files to YAML**

**Description**

`bib2yml()` uses pandoc to convert a .bib file to YAML. It also accepts an optional `yml` object to prepend to the the YAML from the .bib file. If you want to cite several R packages, see `knitr::write_bib()` to write a bibliography file and convert it with `bib2yml()`.

**Usage**

```r
bib2yml(.yml = NULL, path)
```

**Arguments**

- `.yml` a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `path` a path to the .bib file

**Value**

a `yml` object

**See Also**

Other `yml`: `asis_yaml_output()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgsdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtidles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()

Other citations: `yml_citations()`, `yml_reference()`
blogdown_template

Create YAML based on blogdown theme archetypes

**Description**

`blogdown_template()` creates YAML based on your blogdown theme archetypes. blogdown is based on Hugo, which supports many custom themes. Each theme uses YAML in a different way. However, many come with archetypes that define the YAML or TOML. To find out which types your theme has, use `blogdown_archetypes()` to see what’s available. Use `blogdown_template()` to specify the archetype and it will convert the template to YAML that you can use in your post.

**Usage**

```r
blogdown_template(type, path = ".", theme = NULL)

blogdown_archetypes(path = ".", theme = NULL)
```

**Arguments**

- `type`: an archetype
- `path`: the path to your blogdown site
- `theme`: the theme to check for archetypes. By default, `blogdown_template()` will attempt to read your theme from your config file.

**Value**

a yml object

---

code_chunk

Write code chunks programmatically

**Description**

code_chunk() assembles a knitr code chunk as a character vector. `setup_chunk()` is a wrapper around `code_chunk()` to create setup chunks. By default it uses `include = FALSE` and inserts `knitr::opts_chunk$set(echo = TRUE)` into the chunk body. These are helper functions to write R Markdown bodies for `use_rmarkdown()`.

**Usage**

```r
code_chunk(chunk_code, chunk_name = NULL, chunk_args = NULL)

setup_chunk(chunk_code = NULL, chunk_args = list(include = FALSE))
```
draw_yml_tree

Arguments

chunk_code  An expression. Surround with {} to capture multiple lines.
chunk_name  The name of the chunk
chunk_args  A list of chunk options

Value

a character vector

Examples

setup_chunk()

code_chunk({
  yml() %>%
    yml_output(pdf_document())
}, chunk_name = "yml_example")

draw_yml_tree

Draw an tree of YAML hierarchy

Description

draw_yml_tree() draws an ASCII tree of the hierarchy of a given yml object to the console.

Usage

draw_yml_tree(.yml = last_yml(), indent = "")

Arguments

.yml  a yml object created by yml(), as_yml(), or returned by a yml_*() function
indent  a character vector used to indent the tree

Value

invisibly, .yml

See Also

Other yml: asis_yaml_output(), bib2yml(), has_field(), read_json(), use_yaml_defaults(),
use_yaml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(),
yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(),
yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(),
yml_rsconnect_email(), yml_rtile_opts(), yml_runtime(), yml_site_opts(), yml_toc(),
yml_vignette()
Examples

```r
# draw the most recently used `yml`
draw_yml_tree()

yml() %>%
  yml_output(
    pdf_document(keep_tex = TRUE),
    html_document()
  ) %>%
  draw_yml_tree()
```

---

**Description**

`gitbook_config()` is a helper function to specify the `config` argument in `bookdown::gitbook()`, as described in the `bookdown` book.

**Usage**

```r
gitbook_config(
  tocCollapse = yml_blank(),
  tocScrollHighlight = yml_blank(),
  tocBefore = yml_blank(),
  tocAfter = yml_blank(),
  toolbarPosition = yml_blank(),
  edit = yml_blank(),
  download = yml_blank(),
  search = yml_blank(),
  fontSettingsTheme = yml_blank(),
  fontSettingsFamily = yml_blank(),
  fontSettingsSize = yml_blank(),
  sharingFacebook = yml_blank(),
  sharingTwitter = yml_blank(),
  sharingGoogle = yml_blank(),
  sharingLinkedIn = yml_blank(),
  sharingWeibo = yml_blank(),
  sharingInstapaper = yml_blank(),
  sharingVk = yml_blank(),
  sharingAll = yml_blank(),
  ...
)
```
Arguments

tocCollapse: Collapse some items initially when a page is loaded via the collapse option. Its possible values are "subsection" (the default), "section", "none", or NULL.

tocScrollHighlight: Logical. Enable highlighting of TOC items as you scroll the book body? The default is TRUE.

tocBefore, tocAfter: a character vector of HTML to add more items before and after the TOC using the HTML tag <li>. These items will be separated from the TOC using a horizontal divider.

toolbarPosition: The toolbar position: "fixed" or "static." The default ("fixed") is that the toolbar will be fixed at the top of the page, whereas when set to "static" the toolbar will not scroll with the page.

edit: If not empty, an edit button will be added to the toolbar.

download: This option takes either a character vector or a list of character vectors with the length of each vector being 2. When it is a character vector, it should be either a vector of filenames or filename extensions. When you only provide the filename extensions, the filename is derived from the book filename of the configuration file _bookdown.yml

search: Include a search bar?

fontSettingsTheme: The theme. "White" (the default), "Sepia", or "Night".

fontSettingsFamily: The font family. "sans" (the default) or "serif".

fontSettingsSize: The font size. Default is 2.

sharingFacebook: Logical. Include Facebook share link? Default is TRUE.

sharingTwitter: Logical. Include Twitter share link? Default is TRUE.

sharingGoogle: Logical. Include Google share link? Default is FALSE.

sharingLinkedin: Logical. Include LinkedIn share link? Default is FALSE.

sharingWeibo: Logical. Include Weibo share link? Default is FALSE.

sharingInstapaper: Logical. Include Instapaper share link? Default is FALSE.

sharingVk: Logical. Include VK share link? Default is FALSE.

sharingAll: Logical. Include all share links? Default is FALSE.

...: additional named R objects, such as characters or lists, to transform into YAML

Value

a list to use in the config argument of bookdown::gitbook()
has_field

See Also

Other bookdown: `yml_bookdown_opts()`

---

| has_field | Check if field exists in YAML |

Description

has_field() retrieves the names of all fields (including nested fields) and checks if field is among them.

Usage

```r
has_field(.yml, field)
```

Arguments

- `yml`:
  a yaml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `field`:
  A character vector, the name of the field(s) to check for

Value

logical

See Also

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtdciles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()

Examples

```r
has_field(yml(), "author")
has_field(yml(), "toc")
```
includes2

Include content within output

Description

includes2() is a version of the includes() helper function from rmarkdown that uses yml_blank() instead of NULL as the argument defaults, as ymlthis treats NULLs as literal YAML syntax ("null").

Usage

includes2(
  in_header = yml_blank(),
  before_body = yml_blank(),
  after_body = yml_blank()
)

Arguments

in_header One or more files with content to be included in the header of the document.
before_body One or more files with content to be included before the document body.
after_body One or more files with content to be included after the document body.

Value

a list

Examples

yml() %>%
yml_output(
  pdf_document(includes = includes2(after_body = "footer.tex"))
)

is_yml

Is object a yml object?

Description

Is object a yml object?

Usage

is_yml(x)
### last_yml

**Arguments**

- `x` An object to test

**Value**

A logical vector

---

**Description**

This stores the most recently printed YAML object; you can use `last_yml()` to retrieve it to modify, pass to use_*() functions, and so on.

**Usage**

`last_yml()`

**Examples**

```r
yml() %>%
  yml_author("Yihui Xie")

last_yml()
```

---

### pagedown_business_card_template

*Generate a full YAML template for your pagedown business card*

**Description**

pagedown has a unique output type to make business cards: `pagedown::business_card()`. `pagedown_business_card_template()` creates a YAML template to use for this output. What’s unique about this output type is that almost all of the contents are supplied through YAML. An R Markdown file that only contains YAML related to the business card is enough to produce the output, although you can also customize the output in the body of the document (see the [pagedown vignette](https://bookdown.org/yihui/pagedown/)). A good workflow to write a business card is to use `pagedown_business_card_template()` to specify the YAML and pass it to `use_rmarkdown()`, which you can then knitted into business cards.
Usage

pagedown_business_card_template(
    name = yml_blank(),
    person = yml_blank(),
    title = yml_blank(),
    phone = yml_blank(),
    email = yml_blank(),
    url = yml_blank(),
    address = yml_blank(),
    logo = yml_blank(),
    repeat = yml_blank(),
    paperwidth = yml_blank(),
    paperheight = yml_blank(),
    cardwidth = yml_blank(),
    cardheight = yml_blank(),
    cols = yml_blank(),
    rows = yml_blank(),
    mainfont = yml_blank(),
    googlefonts = yml_blank(),
    ...
)

pagedown_person(...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name</td>
</tr>
<tr>
<td>person</td>
<td>When you are creating business cards for numerous people with shared information, passing values to the person field can override the default values, which can be any of the values accepted by this function. Use pagedown_person() to do so or manually provide them using list(field = value).</td>
</tr>
<tr>
<td>title</td>
<td>The title of the person</td>
</tr>
<tr>
<td>phone</td>
<td>A phone number</td>
</tr>
<tr>
<td>email</td>
<td>An email address</td>
</tr>
<tr>
<td>url</td>
<td>A website URL</td>
</tr>
<tr>
<td>address</td>
<td>The address</td>
</tr>
<tr>
<td>logo</td>
<td>A path to a logo file</td>
</tr>
<tr>
<td>.repeat</td>
<td>The number of cards to repeat. Note that the actual YAML field is repeat.</td>
</tr>
<tr>
<td>paperwidth</td>
<td>The paper width</td>
</tr>
<tr>
<td>paperheight</td>
<td>The paper height</td>
</tr>
<tr>
<td>cardwidth</td>
<td>The width of the card</td>
</tr>
<tr>
<td>cardheight</td>
<td>The height of the card</td>
</tr>
<tr>
<td>cols</td>
<td>The number of columns in the card grid</td>
</tr>
<tr>
<td>rows</td>
<td>The rows of columns in the card grid</td>
</tr>
</tbody>
</table>
pagedown_business_card_template

mainfont The font
googlefonts A character vector of Google Fonts
... additional named R objects, such as characters or lists, to transform into YAML

Value

a yml object

See Also

use_rmarkdown()

Other pagedown: yml_pagedown_opts()

Examples

pagedown_business_card_template(
  name = "Jane Doe",
  title = "Miss Nobody",
  phone = "+1 123-456-7890",
  email = "jane.doe@example.com",
  url = "www.example.com",
  address = "2020 South Street, Sunshine, CA 90000",
  logo = "logo.png",
  .repeat = 12
)

pagedown_business_card_template(
  phone = "+1 123-456-7890",
  url = "www.example.com",
  address = "2020 South Street, Sunshine, CA 90000",
  logo = "logo.png",
  person = list(
    pagedown_person(
      name = "Jane Doe",
      title = "Miss Nobody",
      email = "jane.doe@example.com",
      .repeat = 6
    ),
    pagedown_person(
      name = "John Doe",
      title = "Mister Nobody",
      phone = "+1 777-777-7777", # overrides the default phone
      email = "john.doe@example.com",
      .repeat = 6
    )
  ),
  paperwidth = "8.5in",
  paperheight = "11in",
  cols = 4,
Use pandoc templates and custom highlight themes

Description

Pandoc has several built-in templates and code highlighting themes that can be customized and included in the template and highlight-style YAML fields, respectively. `pandoc_template_types()` and `pandoc_highlight_styles()` return the available templates and highlight styles in pandoc, respectively. `use_pandoc_template()` creates a new file based on a template from pandoc or R Markdown and `use_pandoc_highlight_style()` creates a new highlight theme file based on an existing pandoc theme.

Usage

```
pandoc_template_types()
pandoc_highlight_styles()
use_pandoc_template(type, path, source = c("rmarkdown", "pandoc"))
use_pandoc_highlight_style(theme, path)
```

Arguments

- `type` The template type
- `path` The path to write the file to
- `source` The template source ("pandoc" or "rmarkdown")
- `theme` The name of the theme

Value

- a character vector
pkgdown_template

Generate a full YAML template for your pkgdown site

Description

pkgdown includes three helpful pkgdown::template_*() functions to generate the navbar, reference, and article YAML for the _pkgdown.yml file. pkgdown_template() is a wrapper function that runs all three, combines them, and converts them to a yml object. You may also pass pkgdown::template_*() functions to as_yml() to convert the individual sections. pkgdown_template() is particularly useful with use_pkgdown_yml() to write directly to the _pkgdown.yml file.

Usage

pkgdown_template(path = ".\")

Arguments

path The path to your package directory

Value

a yml object

See Also

use_pkgdown_yml()

Other pkgdown: yml_pkgdown()

Examples

# requires this to be a package directory
pkgdown_template() %>%
  use_pkgdown_yml()

read_json

Read and write to JSON and TOML

Description

Read JSON and TOML files in as yml objects with read_*(). Write yml objects out as JSON and YAML files with write_as_*(). You can also provide write_as_*() a path to an existing .yml file to translate to JSON or TOML. These functions rely on Hugo and blogdown, so you must have blogdown installed.
Usage

read_json(path)
read_toml(path)

write_as_json(
  .yml = NULL,
  path = NULL,
  out = NULL,
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

write_as_toml(
  .yml = NULL,
  path = NULL,
  out = NULL,
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

Arguments

path a path to a JSON or TOML file
.yml a yml object created by yml(), as_yml(), or returned by a yml_*() function
.out The path to write out to. If NULL, will write to the path but change the file extension to .toml or .json.
.build_ignore Logical. Should the file be added to the .Rbuildignore file?
.git_ignore Logical. Should the file be added to the .gitignore file?
.quiet Logical. Whether to message about what is happening.

Value

a yml object (if reading) or the path (if writing)

See Also

Other yml: asis_yml_output(), bib2yml(), draw_yml_tree(), has_field(), use_yml_defaults(),
use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(),
yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(),
yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(),
yml_rsconnect_email(), yml_sections_opts(), yml_runtime(), yml_site_opts(), yml_toc(),
yml_vignette()
Description

use_yml() takes a yml object and puts the resulting YAML on your clipboard to paste into an R Markdown or YAML file. use_rmarkdown() takes the yml object and writes it to a new R Markdown file. You can add text to include in the body of the file. If it’s not specified, use_rmarkdown() will use set_up_chunk() by default. You can also set a default for body using options(ymlthis.rmd_body = "(your text)"); see use_rmd_defaults(). Together with specifying default YAML (see use_yml_defaults()), use_rmarkdown() also serves as an ad-hoc way to make R Markdown templates. You may also supply use_rmarkdown() with an existing R Markdown file from which to read the YAML header; the YAML header from the template is then combined with .yml, if it’s supplied, and written to a new file. use_index_rmd() is a wrapper around use_rmarkdown() that specifically writes to a file called index.Rmd. By default, use_yml() and use_rmarkdown() use the most recently printed YAML via last_yml().

Usage

use_yml(.yml = last_yml())

use_rmarkdown(
  .yml = last_yml(),
  path,
  template = NULL,
  include_yaml = TRUE,
  include_body = TRUE,
  body = NULL,
  quiet = FALSE,
  open_doc = interactive()
)

use_index_rmd(
  .yml = last_yml(),
  path,
  template = NULL,
  include_yaml = TRUE,
  include_body = TRUE,
  body = NULL,
  quiet = FALSE,
  open_doc = interactive()
)

Arguments

.yml a yml object created by yml(), as_yml(), or returned by a yml_*() function
path A file path to write R Markdown file to
**template**  
An existing R Markdown file to read YAML from

**include_yaml**  
Logical. Include the template YAML?

**include_body**  
Logical. Include the template body?

**body**  
A character vector to use in the body of the R Markdown file. If no template is set, checks `getOption("yamlthis.rmd_body")` (see `use_rmd_defaults()`) and otherwise uses `setup_chunk()`.

**quiet**  
Logical. Whether to message about what is happening.

**open_doc**  
Logical. Open the document after it's created? By default, this is TRUE if it is an interactive session and FALSE if not. Also checks that RStudio is available.

**Value**

`use_yml()` invisibly returns the input `yml` object

**See Also**

code_chunk() setup_chunk()

Other yml: asis_yaml_output(), bib2yml(), draw_yaml_tree(), has_field(), read_json(), use_yml_defaults(), use_yml_file(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(), yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_pkgdwn(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml_rtrics_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

---

**use_yml_defaults**  
*Set up default YAML*

**Description**

`use_yml_defaults()` takes a `yml` object and places code on the clipboard that will save the resulting YAML as the default for `yml()`. The code that is placed on the clipboard is raw YAML passed to `ymlthis.default_yml` via `options()`. Saving this code to your `.Rprofile` (see `usethis::edit_r_profile()`) will allow `yml()` or `get_yml_defaults()` to return the saved YAML. `use_rmd_defaults()` does the same for `ymlthis.rmd_body`, which is used in `use_rmarkdown()` as the body text of the created R Markdown file.

**Usage**

`use_yml_defaults(.yml)`

`use_rmd_defaults(x)`

`get_yml_defaults()`

`get_rmd_defaults()`
use_yml_file

Arguments

- `.yml` a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `x` a character vector to use as the body text in `use_rmarkdown()`.

See Also

- `yml()`, `get_yml_defaults()`
- Other `yml`: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rictures_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

use_yml_file  Write YAML to file

Description

Write `yml` objects to a file. `use_yml_file()` writes to any given file name. `use_output_yml()` creates file `_output.yml`, which can be used by multiple R Markdown documents. All documents located in the same directory as `_output.yml` will inherit its output options. Options defined within document YAML headers will override those specified in `_output.yml`. Note that `use_output_yml()` plucks the output field from `yml`; any other YAML top-level fields will be ignored. `use_site_yml()` creates `_site.yml` for use with R Markdown websites and third-party tools like the distill package (see the R Markdown book for more). `use_navbar_yml()` is a special type of site YAML that only specifies the navbar in `_navbar.yml`; `use_pkgdown_yml()` and `use_bookdown_yml()` write YAML files specific to those packages; see the pkgdown and blogdown documentation for more.

Usage

```r
use_yml_file(
  .yml = NULL,
  path = NULL,
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

use_output_yml(
  .yml = NULL,
  path = getwd(),
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)
```
use_site_yml(
  .yml = NULL,
  path = ".",
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

use_navbar_yml(
  .yml = NULL,
  path = ".",
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

use_pkgdown_yml(
  .yml = NULL,
  path = ".",
  build_ignore = TRUE,
  git_ignore = FALSE,
  quiet = FALSE
)

use_bookdown_yml(
  .yml = NULL,
  path = ".",
  build_ignore = FALSE,
  git_ignore = FALSE,
  quiet = FALSE
)

Arguments

.yml a yml object created by yml(), as_yml(), or returned by a yml_*() function
path a file path to write the file to
build_ignore Logical. Should the file be added to the .Rbuildignore file?
git_ignore Logical. Should the file be added to the .gitignore file?
quiet Logical. Whether to message about what is happening.

Details

By default, the yaml package adds a new line to the end of files. Some environments, such as RStudio Projects, allow you to append new lines automatically. Thus, you may end up with 2 new lines at the end of your file. If you’d like to automatically remove the last new line in the file, set options(ymlthis.remove_blank_line = TRUE).
**yml**

Create a new yml object

---

**Description**

*yml()* initializes a yml object. yml objects create valid YAML and print it cleanly to the console. By default, *yml()* looks for your name (using `getOption("usethis.full_name"), getOption("devtools.name"), and whoami::fullname()`) and uses today’s date to use in the author and date fields, respectively. If you’ve set default YAML in `getOption("ymlthis.default_option")` (see `use_yml_defaults()`), *yml()* will also use include those fields by default. *yml_empty()* is a wrapper that doesn’t use any of these default YAML fields. *yml()* and all related `yml_*()` functions validate that the results are indeed valid YAML syntax, although not every function is able to check that the input fields are valid for the setting they are used in.

**Usage**

```r
yml(.yml = NULL, get_yml = TRUE, author = TRUE, date = TRUE)

yml_empty()
```

**Arguments**

- `.yml` a character vector, yml object, or YAML-like list. See details.
- `get_yml` logical. Use YAML stored in `getOption("ymlthis.default_option")`? By default, *yml()* includes if it exists.
- `author` logical. Get default author name?
- `date` logical. Get default date?

**Details**

* .yml accepts a character vector of YAML, such as "author: Hadley Wickham", an object returned by ymlthis functions that start with `yml_*()`, or a list object (e.g. `list(author = "Hadley Wickham")`). *yml* objects are processed with `as_yml()`, a wrapper around `yaml::yaml.load()`. See that function for more details.
yml_author

Value

a yml object

Examples

yml()

yml(date = FALSE)

"author: Hadley Wickham\ndate: 2014-09-12" %>%
yml() %>%
yml_title("Tidy Data") %>%
yml_keywords(
  c("data cleaning", "data tidying", "relational databases", "R")
)

yml() %>%
yml_author(
  c("Yihui Xie", "Hadley Wickham"),
  affiliation = rep("RStudio", 2)
) %>%
yml_date("07/04/2019") %>%
yml_output(
  pdf_document(
    keep_tex = TRUE,
    includes = includes2(after_body = "footer.tex")
  )
) %>%
yml_latex_opts(biblio_style = "apalike")

---

yml_author

Set Top-level R Markdown YAML Fields

Description

These functions add common top-level YAML fields for R Markdown documents, such as author, date, and title. Each takes a yml object and adds fields related to the function, as well as checking for duplicate fields and (where possible) checking for valid entries. yml_toplevel() is a catch-all function that will take any named R object and put in the top level of the YAML; it checks for duplicate fields but is unable to validate the input beyond that it is valid YAML syntax. Some R Markdown templates allow for additional variations of the YAML here. For instance, the distill package adds url and affiliation_url to the author field (see yml_distill_author, which wraps yml_author). Several yml_*() functions also contain ... which allow for these unique fields.
Usage

\begin{verbatim}
 yml_author(.yml, name = NULL, affiliation = NULL, email = NULL, ...)
 yml_date(.yml, date = NULL, format = "")
 yml_title(.yml, title)
 yml_subtitle(.yml, subtitle)
 yml_abstract(.yml, abstract)
 yml_keywords(.yml, keywords)
 yml_subject(.yml, subject)
 yml_description(.yml, description)
 yml_category(.yml, category)
 yml_lang(.yml, lang)
 yml_toplevel(.yml, ...)
\end{verbatim}

Arguments

<table>
<thead>
<tr>
<th>.yml</th>
<th>a yaml object created by yml(), as_yml(), or returned by a yml_*() function</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A character vector, name of the author(s)</td>
</tr>
<tr>
<td>affiliation</td>
<td>The author's affiliation; must match length of name, e.g. if name has length of</td>
</tr>
<tr>
<td></td>
<td>two, affiliation must as well; use NA if you don't want to include an affiliation</td>
</tr>
<tr>
<td></td>
<td>for a given author. Note that not all formats support the affiliation field.</td>
</tr>
<tr>
<td>email</td>
<td>The author email address. Note that not all formats support the email field.</td>
</tr>
<tr>
<td>...</td>
<td>additional named R objects, such as characters or lists, to transform into YAML</td>
</tr>
<tr>
<td>date</td>
<td>The date; by default this is &quot;\r format(Sys.Date())&quot;&quot;, which will populate</td>
</tr>
<tr>
<td></td>
<td>the date automatically.</td>
</tr>
<tr>
<td>format</td>
<td>When the default date is used, the format passed to format.Date().</td>
</tr>
<tr>
<td>title</td>
<td>A character vector, the title of the document</td>
</tr>
<tr>
<td>subtitle</td>
<td>A character vector, the subtitle of the document. Not all R Markdown formats</td>
</tr>
<tr>
<td></td>
<td>use subtitles, so it may depend on what you use in the output field (see yml_output()).</td>
</tr>
<tr>
<td></td>
<td>It is available in pdf_document(), html_document(), and word_document() by default.</td>
</tr>
<tr>
<td>abstract</td>
<td>A character vector, the abstract. Long character vectors are automatically wrapped</td>
</tr>
<tr>
<td></td>
<td>using valid YAML syntax. This field is not available in all output formats; it is available in pdf_document() and html_document() by default.</td>
</tr>
<tr>
<td>keywords</td>
<td>A character vector of keywords. This field is not available in all output formats; it is available in pdf_document(), html_document(), word_document(), odt_document(), and powerpoint_presentation() by default.</td>
</tr>
</tbody>
</table>
subject A character vector, the subject of the document. This field is not available in all output formats; it is available in pdf_document(), html_document(), word_document(), odt_document(), and powerpoint_presentation() by default.

description A character vector, a description of the document. This field is not available in all output formats; it is available in word_document(), odt_document(), and powerpoint_presentation() by default.

category A character vector, the category of the document. This field is not available in all output formats; it is available in word_document() and powerpoint_presentation() by default.

lang The document language using IETF language tags such as "en" or "en-US". The Language subtag lookup tool can help find the appropriate tag.

Value

a yml object

See Also

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(), use_yaml_defaults(), use_yaml_file(), use_yml(), yml_blogdown_opts(), yml_bookdown_opts(), yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml_rarticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Examples

```r
yml_empty() %>%
  yml_author("Yihui Xie") %>%
  yml_date("02-02-2002") %>%
  yml_title("R Markdown: An Introduction") %>%
  yml_subtitle("Introducing ymlthis") %>%
  yml_abstract("This paper will discuss a very important topic") %>%
  yml_keywords(c("r", "reproducible research")) %>%
  yml_subject("R Markdown") %>%
  yml_description("An R Markdown reader") %>%
  yml_category("r") %>%
  yml_lang("en-US")
```

yml_blank Return a blank object to be discarded from YAML
Description

YAML in blogdown comes from a variety of sources. Most YAML will be for your posts, as described in the blogdown book. Common R Markdown fields can be used, but there are two other main sources for YAML fields: Hugo itself and the Hugo theme you are using. Hugo has numerous top-level YAML to control the output (see the Hugo documentation). `yml_blogdown_opts()` supports Hugo YAML. Your Hugo theme may also add fields to use. To find YAML specific to your theme, see `blogdown_template()`. In addition to these sources of YAML, the configuration file for your blog can also be in YAML, but this is not very common; most use a config.toml file, based on TOML (see the blogdown book for more).
Usage

```r
yml_blogdown_opts(
  .yml,
  draft = yml_blank(),
  publishdate = yml_blank(),
  weight = yml_blank(),
  slug = yml_blank(),
  aliases = yml_blank(),
  audio = yml_blank(),
  date = yml_blank(),
  description = yml_blank(),
  expiration_date = yml_blank(),
  headless = yml_blank(),
  images = yml_blank(),
  keywords = yml_blank(),
  layout = yml_blank(),
  lastmod = yml_blank(),
  link_title = yml_blank(),
  resources = yml_blank(),
  series = yml_blank(),
  summary = yml_blank(),
  title = yml_blank(),
  type = yml_blank(),
  url = yml_blank(),
  videos = yml_blank(),
  ...
)
```

Arguments

- `.yml`: a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `draft`: Logical. Set post as a draft? Draft posts will not be rendered if the site is built via `blogdown::build_site()` or `blogdown::hugo_build()` but will be rendered in the local preview mode. See Section D.3 of the blogdown book.
- `publishdate`: A future date to publish the post. Future posts are only rendered in the local preview mode
- `weight`: This field can take a numeric value to tell Hugo the order of pages when sorting them, e.g., when you generate a list of all pages under a directory, and two posts have the same date, you may assign different weights to them to get your desired order on the list
- `slug`: A character string used as the tail of the post URL. It is particularly useful when you define custom rules for permanent URLs. See Section 2.2.2 of the blogdown book.
- `aliases`: A character vector of one or more aliases (e.g., old published paths of renamed content) that will be created in the output directory structure
- `audio`: A character vector of paths to audio files related to the page
**date**
The date assigned to this page. This is usually fetched from the date field in front matter, but this behavior is configurable.

**description**
The description for the content.

**expiration_date**
The date at which the content should no longer be published by Hugo. Note that the actual YAML field is `expiryDate`.

**headless**
If TRUE, sets a leaf bundle to be headless.

**images**
A character vector of paths to images related to the page.

**keywords**
A character vector of the keywords for the content.

**layout**
The layout Hugo should use while rendering the content. By default, layout matches type and is thus based on the directory. However, it’s possible to use additional layouts within a type. See Hugo’s Defining a Content Type documentation.

**lastmod**
The date the content was last modified at.

**link_title**
Used for creating links to content. Note that the actual YAML field is `linkTitle`.

**resources**
A named list. Used for configuring page bundle resources. See Hugo’s Page Resources documentation.

**series**
A character vector of series this page belongs to.

**summary**
A summary of the content in the `.Summary` Hugo page variable; see the content-summaries section of Hugo’s documentation.

**title**
The title for the content.

**type**
The type of the content, which is based on the from the directory of the content if not specified.

**url**
The full path to the content from the web root.

**videos**
A character vector of paths to videos related to the page.

... additional named R objects, such as characters or lists, to transform into YAML

### Value

A yml object

### See Also

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtables_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`
**Examples**

```r
yml() %>%
  yml_blogdown_opts(
    draft = TRUE,
    slug = "blog-post"
  )
```

**Description**
bookdown uses YAML in three main places, as described in the bookdown book: index.Rmd, _output.yml, and _bookdown.yml. index.Rmd can take most YAML. _output.yml is intended for output-related YAML, such as that produced by `yml() %>% yml_output(bookdown::pdf_book())`. _bookdown.yml is intended for configuring the build of the book. Pass the results of the `yml_*()` functions to `use_index_rmd()`, `use_bookdown_yml()`, `use_output_yml()` to write them to these files. `yml_bookdown_site()` adds the site: "bookdown::bookdown_site" to the YAML metadata.

**Usage**

```r
yml_bookdown_opts(
  .yml,
  book_filename = yml_blank(),
  delete_merged_file = yml_blank(),
  before_chapter_script = yml_blank(),
  after_chapter_script = yml_blank(),
  edit = yml_blank(),
  history = yml_blank(),
  rmd_files = yml_blank(),
  rmd_subdir = yml_blank(),
  output_dir = yml_blank(),
  clean = yml_blank(),
  ...
)
```

**Arguments**

- `.yml` a yaml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `book_filename` A character vector, the filename of the main .Rmd file, the .Rmd file that is created by merging all chapters. By default, it is called "_main.Rmd".
- `delete_merged_file` Logical. Delete the main .Rmd file if it exists?
before_chapter_script, after_chapter_script
   A character vector of one or more R scripts to be executed before or after each chapter

edit
   A URL that collaborators can click to edit the .Rmd source document of the current page, usually a link to a GitHub repository. This link should have %s where the actual .Rmd filename for each page will go.

history
   Similar to edit, a link to the edit/commit history of the current page.

rmd_files
   A character vector, the order order of .Rmd files for the book. rmd_files can also be a named list where each element of the list is named for the output type, e.g. "html" or "latex". By default, bookdown merges all .Rmd files by the order of filenames.

rmd_subdir
   whether to search for book source .Rmd files in subdirectories (by default, only the root directory is searched). This may be either a boolean (e.g. TRUE will search for book source .Rmd files in the project directory and all subdirectories) or vector of paths if you want to search for book source .Rmd files in a subset of subdirectories.

output_dir
   the output directory of the book ("_book" by default)

clean
   a character vector of files and directories to be cleaned by the bookdown::clean_book() function.

... additional named R objects, such as characters or lists, to transform into YAML

Value
   a yml object

See Also

use_index_rmd() use_bookdown_yml() use_output_yml()

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(), use_yaml_defaults(), use_yaml_file(), use_yaml(), yml_author(), yml_blogdown_opts(), yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml_rticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Other bookdown: gitbook_config()

Examples

yml_empty() %>%
  yml_bookdown_opts(
    before_chapter_script = c("script1.R", "script2.R"),
    after_chapter_script = "script3.R",
    edit = "https://github.com/rstudio/bookdown-demo/edit/master/%s",
    output_dir = "book-output",
    clean = c("my-book.bbl", "R-packages.bib")
)
Set citation-related YAML options

**Description**

`yml_citations()` sets citation-related YAML fields, such as specifying a bibliography file or style.
For controlling the citation engine in PDF documents, see the `citation_package` argument in `rmarkdown::pdf_document()`.

Usage

```r
yml_citations(
  .yml,
  bibliography = yml_blank(),
  biblio_style = yml_blank(),
  biblio_title = yml_blank(),
  csl = yml_blank(),
  citation_abbreviations = yml_blank(),
  link_citations = yml_blank(),
  nocite = yml_blank(),
  suppress_bibliography = yml_blank()
)
```

Arguments

- `.yml` a yml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `bibliography` a path to a bibliography file, such as a `.bib` file
- `biblio_style` bibliography style, when used with `natbib` and `biblatex`. Note that the actual YAML field is `biblio-style`
- `biblio_title` bibliography title, when used with `natbib` and `biblatex`. Note that the actual YAML field is `biblio-title`
- `csl` a path to a Citation Style Language (CSL) file. CSL files are used to specify the citation style; see the CSL repository for the CSL files of dozens of journals.
- `citation_abbreviations` Path to a CSL abbreviations JSON file. See the `pandoc-citeproc` documentation.
- `link_citations` Logical. Add citations hyperlinks to the corresponding bibliography entries?
- `nocite` Citation IDs ("@item1") to include in the bibliography even if they are not cited in the document. Including the wildcard pattern "@*" will include all citations in the bibliography regardless of if they’re cited in the document.
- `suppress_bibliography` Logical. Suppress bibliography?

Value

- a yml object

See Also

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`,
yml_clean

yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml.rticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Other citations: bib2yml(), yml_reference()

Examples

```r
yml() %>%
  yml_citations(bibliography = "references.bib", csl = "aje.csl")
```

---

**yml_clean**

*Remove intermediate rendering files*

**Description**

R Markdown may create many documents while rendering the final product, for instance by using knitr to turn the R Markdown file to a Markdown file and then using Pandoc to convert to the final output. The clean field tells R Markdown whether or not to remove these files.

**Usage**

```r
yml_clean(.yml, clean)
```

**Arguments**

- `.yml` a yml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `clean` Logical. Remove intermediate files that are created while making the R Markdown document?

**Value**

a yml object

**See Also**

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(), use_yml_defaults(), use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(), yml_citations(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_pkgdwn(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml.rticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Other R Markdown: yml_params(), yml_runtime(), yml_site_opts(), yml_vignette()
**Examples**

```r
yml() %>%
  # keep intermediate files
  yml_clean(FALSE)
```

---

**Take code and write it as valid YAML**

**Description**

`yml_code()` takes R code and writes it as valid YAML to be evaluated during knitting. Note that `yml_code()` does not evaluate or validate the R code but only captures it to use in the YAML field. R code needs to be formatted differently when using in the `params` field for parameterized reports; `yml_params_code` will format this correctly for you.

**Usage**

```r
yml_code(x)

yml_params_code(x)
```

**Arguments**

- `x` valid R code

**Value**

a character vector with class `verbatim`

**See Also**

`yml_verbatim()`

**Examples**

```r
yml_empty() %>%
  yml_date(yml_code(sys.Date()))

yml_empty() %>%
  yml_params(date = yml_params_code(sys.Date()))
```
yml_distill_opts  

Set Top-level YAML options for distill

Description

distill uses many custom YAML fields to create some of its unique features, such as article metadata and citations. In addition to the arguments in yml_distill_opts(), distill supports in a number of other ways. yml_distill_author() wraps yml_author() to include these extra used in distill. For a distill blog, you can specify the listings page a post belongs to, including an optional vector of other posts to list with it; distill_listing() is a helper function to pass to the listing argument to specify such pages. distill uses the same approach to navbars as R Markdown. yml_navbar() and friends will help you write the YAML for that. YAML specifying the site build, like the output field and navbars, can also be placed in _site.yml; see yml_site_opts() for further R Markdown website build options and use_site_yml() for creating that file based on a yml object. distill’s YAML options are discussed in greater detail in the articles on the distill website.

Usage

yml_distill_opts(
  .yml,
  draft = yml_blank(),
  slug = yml_blank(),
  categories = yml_blank(),
  listing = yml_blank(),
  collection = yml_blank(),
  citation_url = yml_blank(),
  preview = yml_blank(),
  repository_url = yml_blank(),
  base_url = yml_blank(),
  compare_updates_url = yml_blank(),
  creative_commons = yml_blank(),
  twitter_site = yml_blank(),
  twitter_creator = yml_blank(),
  journal_title = yml_blank(),
  journal_issn = yml_blank(),
  journal_publisher = yml_blank(),
  volume = yml_blank(),
  issue = yml_blank(),
  doi = yml_blank(),
  resources = yml_blank(),
  ...
)

yml_distill_author(
  .yml,
  name = yml_blank(),
  url = yml_blank(),
  ...
affiliation = yml_blank(),
affiliation_url = yml_blank()
)

distill_listing(listing_name = "posts", slugs = NULL)

distill_collection(
  collection_name = "post",
  feed_items_max = yml_blank(),
  disqus_name = yml_blank(),
  disqus_hidden = yml_blank(),
  share = yml_blank(),
  citations = yml_blank(),
  subscribe = yml_blank()
)

distill_resources(include = yml_blank(), exclude = yml_blank())

Arguments

- .yml: a yml object created by yml(), as_yml(), or returned by a yml_*() function
- draft: Logical. Set the post to be a draft? Draft posts won’t be published.
- slug: The abbreviated version of the citation included in the BibTeX entry. If you don’t provide a slug then one will be automatically generated.
- categories: A character vector, the post categories
- listing: The listing a post is under; either a character vector, the output of distill_listing(), or a named list.
- collection: Specify the RSS, sharing, and other settings of a listing; use distill_collection() or a named list.
- citation_url: A URL to the article; automatically generated for blog articles
- preview: a path or link to the preview image for your article. You can also set this by including preview = TRUE in an R Markdown code chunk in your document.
- repository_url: A URL where the source code for your article can be found
- base_url: Base (root) URL for the location where the website will be deployed (used for providing preview images for Open Graph and Twitter Card)
- compare_updates_url: a URL that will show the differences between the article’s current version and the version that was initially published
- creative_commons: Designate articles that you create as Creative Commons licensed by specifying one of the standard Creative Commons licenses. Common options include "CC BY", "CC BY-SA", "CC BY-ND", and "CC BY-NC". See the distill vignette for more details.
- twitter_site: The Twitter handle for the site
- twitter_creator: The Twitter handle for the creator
yml_distill_opts

**journal_title**  The title of the journal

**journal_issn**  The issn of the journal

**journal_publisher**  The publisher of the journal

**volume**  The volume the article is on

**issue**  The issue the article is on

**doi**  The article Digital Object Identifier (DOI)

**resources**  Files to include or exclude while publishing. Use `distill_resources()` or a named list to specify.

...  additional named R objects, such as characters or lists, to transform into YAML

**name**  A character vector, name of the author(s)

**url**  the author URL

**affiliation**  The author’s affiliation; must match length of name, e.g. if name has length of two, affiliation must as well; use `NA` if you don’t want to include an affiliation for a given author. Note that not all formats support the affiliation field.

**affiliation_url**  the affiliation URL

**listing_name**  A character vector, the name of the listing

**slugs**  A character vector of the posts to include in the listing

**collection_name**  A character vector, the name of the collection

**feed_items_max**  Number of articles to include in the RSS feed (default: 20). Specify `FALSE` to have no limit on the number of items included in the feed.

**disqus_name**  A shortname for the disqus comments section (base_url field is required in order to use Disqus)

**disqus_hidden**  Logical. Show full text of disqus comments? By default, this is `FALSE` so as not to obscure the bibliography and other appendices.

**share**  Share buttons to include. Choices: "twitter", "linkedin", "facebook", "google-plus", and "pinterest". (base_url field is required in order to use sharing links)

**citations**  Logical. If your _site.yml file provides a base_url field, an article citation appendix and related metadata will be included automatically within all published posts. Set to `FALSE` to disable this behavior.

**subscribe**  a path to a HTML file enabling readers to subscribe. See the distill vignette on blog posts for more details.

**include**, **exclude**  a character vector of files to explicitly include or exclude when publishing a post. Can use wild cards, such as "*.csv".

**Value**

a yml object
**yml_handlers**

*Set handlers to process the way YAML is printed*

**Description**

`yml` this uses the yaml package to process and validate YAML; this package also lets you specify how fields and values are printed using a list of handler functions. `yml_handlers()` specifies defaults for the package used in the print statement. See `yaml::yaml.load()` for more on specifying handlers.

**Examples**

```r
post_listing <- distill_listing(
  slugs = c(
    "2016-11-08-sharpe-ratio",
    "2017-11-09-visualizing-asset-returns",
    "2017-09-13-asset-volatility"
  )
)

yml() %>%
  yml_title("Gallery of featured posts") %>%
  yml_empty()

yml() %>%
  yml_title("Reproducible Finance with R") %>%
  yml_description("Exploring reproducible finance with the R statistical,
  computing environment.") %>%
  yml_site_opts(name = "reproducible-finance-with-r") %>%
  yml_distill_opts(
    base_url = "https://beta.rstudioconnect.com/content/3776/",
    feed_items_max = 30,
    disqus_name = "reproducible-finance-with-r",
    disqus_hidden = FALSE,
    share = c("twitter", "linkedin")
  )
)```

See Also

- `use_site_yml()` `use_rmarkdown()`
- Other `yml`: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rfticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`
- Other websites: `yml_pkgdown()`, `yml_site_opts()`
Usage

```r
yml_handlers()
```

---

**yml_latex_opts**  
*Set LaTeX YAML options for PDF output*

Description

`yml_latex_opts()` sets top-level YAML fields for LaTeX options used by pandoc (see the documentation, from which these descriptions were derived), as when making a PDF document with `pdf_document()`.

Usage

```r
yml_latex_opts(  
  .yml,  
  block_headings = yml_blank(),  
  classoption = yml_blank(),  
  documentclass = yml_blank(),  
  geometry = yml_blank(),  
  indent = yml_blank(),  
  linestretch = yml_blank(),  
  margin_left = yml_blank(),  
  margin_right = yml_blank(),  
  margin_top = yml_blank(),  
  margin_bottom = yml_blank(),  
  pagestyle = yml_blank(),  
  papersize = yml_blank(),  
  secnumdepth = yml_blank(),  
  fontenc = yml_blank(),  
  fontfamily = yml_blank(),  
  fontfamilyoptions = yml_blank(),  
  fontsize = yml_blank(),  
  mainfont = yml_blank(),  
  sansfont = yml_blank(),  
  monofont = yml_blank(),  
  mathfont = yml_blank(),  
  CJKmainfont = yml_blank(),  
  mainfontoptions = yml_blank(),  
  sansfontoptions = yml_blank(),  
  monofontoptions = yml_blank(),  
  mathfontoptions = yml_blank(),  
  CJKoptions = yml_blank(),  
  microtypeoptions = yml_blank(),  
  colorlinks = yml_blank(),  
  linkcolor = yml_blank(),
```

filecolor = yml_blank(),
citecolor = yml_blank(),
urlcolor = yml_blank(),
toccolor = yml_blank(),
links_as_notes = yml_blank(),
lof = yml_blank(),
lot = yml_blank(),
thanks = yml_blank(),
toc = yml_blank(),
toc_depth = yml_blank(),
biblatexoptions = yml_blank(),
biblio_style = yml_blank(),
biblio_title = yml_blank(),
bibliography = yml_blank(),
natbiboptions = yml_blank()
)

Arguments

.yml a yml object created by yml(), as_yml(), or returned by a yml_*() function
block_headings make paragraph and subparagraph (fourth- and fifth-level headings, or fifth- and sixth-level with book classes) free-standing rather than run-in; requires further formatting to distinguish from subsubsection (third- or fourth-level headings). Note that the YAML field is actually called block_headings.
classoption a character vector of options for document class, e.g. "oneside"
documentclass the document class usually "article", "book", or "report"
geometry a character vector of options for the geometry LaTeX package, e.g. "margin=1in"
indent Logical. Use document class settings for indentation? The default LaTeX template otherwise removes indentation and adds space between paragraphs.
linestretch adjusts line spacing using the setspace LaTeX package, e.g. 1.25, 1.5
margin_left, margin_right, margin_top, margin_bottom sets margins if geometry is not used, otherwise geometry overrides these. Note that the actual YAML fields use - instead of _, e.g. margin_left.
pagestyle control the pagestyle LaTeX command: the default article class supports "plain" (default), "empty" (no running heads or page numbers), and "headings" (section titles in running heads)
papersize paper size, e.g. letter, a4
secnumdepth numbering depth for sections (with --number-sections pandoc)
fontenc allows font encoding to be specified through fontenc LaTeX package (with pdflatex); default is "T1" (see LaTeX font encodings guide)
fontfamily font package for use with pdflatex: TeX Live includes many options, documented in the LaTeX Font Catalogue. The default is "Latin Modern".
fontfamilyoptions a character vector of options for fontfamily.
\begin{itemize}
  \item \texttt{latexopts} \texttt{\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore}\texttt{latex opts} \texttt{\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore}
  \item \texttt{\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore\textunderscore}
\end{itemize}

- \texttt{latexopts} is a \texttt{yml} object.

- \texttt{latexopts} includes the following fields:
  - \texttt{latexopts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex.latex opts.latex.latex opts} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex.latex opts.latex.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex.latex opts.latex.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex opts.latex opts.latex} is a \texttt{yml} object.
  - \texttt{latexopts.latex} is a \texttt{yml} object.

\section*{Value}

A \texttt{yml} object.

\section*{See Also}

Other \texttt{yml}: \texttt{asis\_yml\_output()}, \texttt{bib2yml()}, \texttt{draw\_yml\_tree()}, \texttt{has\_field()}, \texttt{read\_json()}, \texttt{use\_yml\_defaults()}, \texttt{use\_yml\_file()}, \texttt{use\_yml()}, \texttt{yml\_author()}, \texttt{yml\_blogdown\_opts()}, \texttt{yml\_bookdown\_opts()}, \texttt{yml\_citations()}, \texttt{yml\_clean()}, \texttt{yml\_distill\_opts()}, \texttt{yml\_output()}, \texttt{yml\_pagedown\_opts()}, \texttt{yml\_params()}, \texttt{yml\_pkgdown()}, \texttt{yml\_reference()}, \texttt{yml\_replace()}, \texttt{yml\_resource\_files()}, \texttt{yml\_rsconnect\_email()}, \texttt{yml\_rticles\_opts()}, \texttt{yml\_runtime()}, \texttt{yml\_site\_opts()}, \texttt{yml\_toc()}, \texttt{yml\_vignette()
yml_load

Load YAML from string

Description

yml_load() is a wrapper for `yaml::yaml.load()` that also converts the object to the `yml` class.

Usage

yml_load(x)

Arguments

x

an object to pass to `yaml::yaml.load()`

Examples

c("title: my title", "author: Malcolm Barrett") %>%
  yml_load()

yml_output

Capture, validate, and write output YAML

Description

yml_output() writes valid YAML for the output field of R Markdown YAML. `yml_output()` captures the actual output functions, such as `pdf_document()`, and translates them to YAML. This function accepts multiple output formats (separated by commas) and validates each by evaluating the function internally. The YAML fields in `under output` come from arguments in their respective R functions. If you wanted to see the available fields in `pdf_document()`, for instance, you would read the documentation for that function using `?pdf_document`.

Usage

yml_output(.yml, ...)

Examples

yml() %>%
  yml_output(pdf_document()) %>%
  yml_latex_opts(
    fontfamily = "Fira Sans Thin",
    fontsize = "11pt",
    links_as_notes = TRUE
  )
Arguments

- `.yml`  
a yml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function

- `...`  
valid R code calling functions that return objects of class `rmarkdown_output_format`, such as the `*_document()` functions in `rmarkdown`.

Value

- a yml object

See Also

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

Examples

```r
yml() %>%
yml_output(html_document())
```

```r
yml() %>%
yml_output(
  pdf_document(keep_tex = TRUE, includes = includes2(after_body = "footer.tex"),
              bookdown::html_document2()
)
```

---

**yml_pagedown_opts**  
*Top-level YAML options for pagedown*

Description

pagedown offers several output functions for paginated output, resumes, business cards, theses, and morem as described in the `pagedown vignette`. pagedown also accepts a few custom top-level YAML. See `pagedown_business_card_template()` for more on setting up the YAML for a business card.

Usage

```r
yml_pagedown_opts(
  .yml,
  toc = yml_blank(),
  toc_title = yml_blank(),
  lot = yml_blank(),
)```
Arguments

- `.yml` a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `toc` Logical. Use a table of contents?
- `toc_title` The title for the table of contents. Note that the actual YAML field is `toc-title`
- `lot` Logical. Use a list of figures?
- `lot_title` The title for the list of figures. Note that the actual YAML field is `lot-title`
- `chapter_name` The chapter title prefix
- `links_to_footnotes` Logical. Transform all the URLs to footnotes? Note that the actual YAML field is `links-to-footnotes`
- `paged_footnotes` Logical. Render notes as footnotes? Note that the actual YAML field is `paged-footnotes`

Value

a `yml` object

See Also

Other `yml`: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rarticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

Other `pagedown`: `pagedown_business_card_template()`

Examples

```r
yml() %>%
yml_pagedown_opts(
  toc = TRUE,
  toc_title = "TOC",
  chapter_name = c("CHAPTER\", ", "."),
  links_to_footnotes = TRUE
)
```
Parameterize an R Markdown report using Shiny components

**Description**

R Markdown lets you add dynamic parameters to your report using the `params` YAML field (see the [R Markdown book](https://rmarkdown.rstudio.com) for examples); parameterized reports are also used in RStudio Connect. The values of these variables can be called inside your R Markdown document using `params$field_name`. There are several ways to change the values of the parameters: manually change the YAML, use the `params` argument in `rmarkdown::render()`, or knit with parameters, which launches a Shiny app to select values for each. `yml_params()` accepts any number of named R objects to set as YAML fields. You can also pass arguments to the underlying Shiny functions using YAML. To set a shiny component, use the `shiny_*()` helper functions. `shiny_params()` captures a Shiny output function and transforms it to YAML. However, R Markdown supports only a limited number of components; each of these is included as a function starting with `shiny_*()`, e.g. `shiny_checkbox()`

**Usage**

```r
yml_params(.yml, ...)

shiny_params(.shiny)

shiny_checkbox(label, value = FALSE, width = NULL)

shiny_numeric(label, value, min = NA, max = NA, step = NA, width = NULL)

shiny_slider(
  label,
  min,
  max,
  value,
  step = NULL,
  round = FALSE,
  format = NULL,
  locale = NULL,
  ticks = TRUE,
  animate = FALSE,
  width = NULL,
  sep = ",",
  pre = NULL,
  post = NULL,
  timeFormat = NULL,
  timezone = NULL,
  dragRange = TRUE
)

shiny_date(
```


label,
value = NULL,
min = NULL,
max = NULL,
format = "yyyy-mm-dd",
startview = "month",
weekstart = 0,
language = "en",
width = NULL,
autoclose = TRUE,
datesdisabled = NULL,
daysofweekdisabled = NULL )

shiny_text(label, value = "", width = NULL, placeholder = NULL)

shiny_file(
  label,
multiple = FALSE,
accept = NULL,
width = NULL,
buttonLabel = "Browse...",
placeholder = "No file selected"
)

shiny_radio(
  label,
choices = NULL,
selected = NULL,
inline = FALSE,
width = NULL,
choiceNames = NULL,
choiceValues = NULL
)

shiny_select(
  label,
choices,
selected = NULL,
multiple = FALSE,
selectize = TRUE,
width = NULL,
size = NULL
)

shiny_password(label, value = "", width = NULL, placeholder = NULL)
Arguments

.yml  a `yml()` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function

... additional named R objects, such as characters or lists, to transform into YAML

.shiny  a Shiny function call to capture and convert to YAML

label  Display label for the control, or NULL for no label.

value  Initial value (TRUE or FALSE).

width  The width of the input, e.g. '400px', or '100%'; see `shiny::validateCssUnit()`

min  Minimum allowed value

max  Maximum allowed value

step  Interval to use when stepping between min and max

round  TRUE to round all values to the nearest integer; FALSE if no rounding is desired; or an integer to round to that number of digits (for example, 1 will round to the nearest 10, and -2 will round to the nearest .01). Any rounding will be applied after snapping to the nearest step.

format  Deprecated.

locale  Deprecated.

ticks  FALSE to hide tick marks, TRUE to show them according to some simple heuristics.

animate  TRUE to show simple animation controls with default settings; FALSE not to; or a custom settings list, such as those created using `shiny::animationOptions()`

sep  Separator between thousands places in numbers.

pre  A prefix string to put in front of the value.

post  A suffix string to put after the value.

timeFormat  Only used if the values are Date or POSIXt objects. A time format string, to be passed to the Javascript strftime library. See `https://github.com/samsonjs/strfttime` for more details. The allowed format specifications are very similar, but not identical, to those for R’s `base::strftime()` function. For Dates, the default is "%F" (like "2015-07-01"), and for POSIXt, the default is "%F %T" (like "2015-07-01 15:32:10").

timezone  Only used if the values are POSIXt objects. A string specifying the time zone offset for the displayed times, in the format "+HHMM" or "-HHMM". If NULL (the default), times will be displayed in the browser’s time zone. The value "+0000" will result in UTC time.

dragRange  This option is used only if it is a range slider (with two values). If TRUE (the default), the range can be dragged. In other words, the min and max can be dragged together. If FALSE, the range cannot be dragged.

startview  The date range shown when the input object is first clicked. Can be "month" (the default), "year", or "decade".

weekstart  Which day is the start of the week. Should be an integer from 0 (Sunday) to 6 (Saturday).

autoclose Whether or not to close the datepicker immediately when a date is selected.

datesdisabled Which dates should be disabled. Either a Date object, or a string in yyyy-mm-dd format.

daysofweekdisabled Days of the week that should be disabled. Should be an integer vector with values from 0 (Sunday) to 6 (Saturday).

placeholder A character string giving the user a hint as to what can be entered into the control. Internet Explorer 8 and 9 do not support this option.

multiple Whether the user should be allowed to select and upload multiple files at once. **Does not work on older browsers, including Internet Explorer 9 and earlier.**

accept A character vector of MIME types; gives the browser a hint of what kind of files the server is expecting.

buttonLabel The label used on the button. Can be text or an HTML tag object.

choices List of values to select from (if elements of the list are named then that name rather than the value is displayed to the user). If this argument is provided, then choiceNames and choiceValues must not be provided, and vice-versa. The values should be strings; other types (such as logicals and numbers) will be coerced to strings.

selected The initially selected value (if not specified then defaults to the first value)

inline If TRUE, render the choices inline (i.e. horizontally)

choiceNames List of names and values, respectively, that are displayed to the user in the app and correspond to the each choice (for this reason, choiceNames and choiceValues must have the same length). If either of these arguments is provided, then the other must be provided and choices must not be provided. The advantage of using both of these over a named list for choices is that choiceNames allows any type of UI object to be passed through (tag objects, icons, HTML code, ...), instead of just simple text. See Examples.

choiceValues List of names and values, respectively, that are displayed to the user in the app and correspond to the each choice (for this reason, choiceNames and choiceValues must have the same length). If either of these arguments is provided, then the other must be provided and choices must not be provided. The advantage of using both of these over a named list for choices is that choiceNames allows any type of UI object to be passed through (tag objects, icons, HTML code, ...), instead of just simple text. See Examples.

selectize Whether to use `selectize.js` or not.

size Number of items to show in the selection box; a larger number will result in a taller box. Not compatible with `selectize=TRUE`. Normally, when `multiple=FALSE`, a select input will be a drop-down list, but when `size` is set, it will be a box instead.
Value

a yml object

See Also

yml_params_code()

Other yml: asis_yamll_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(),
use_yml_defaults(), use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(),
yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(),
yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(),
yml_rticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

yml_citations(), yml_clean(), yml_discard(), yml_distill_opts(), yml_latex_opts(), yml_output(),
yml_pagedown_opts(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(),
yml_rticles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Other R Markdown: yml_clean(), yml_runtime(), yml_site_opts(), yml_vignette()

Other shiny: yml_runtime()

Examples

```r
yml() %>%
  yml_params(
    z = "z",
    x = shiny_numeric("Starting value", 23),
    no = shiny_checkbox("No option?"),
    y = shiny_slider("Data range", 0, 1, .5, round = TRUE)
  )
```

---

**yml_pkgdown**

*Set Top-level YAML options for pkgdown*

Description

These functions set YAML for various pkgdown options to be used in _pkgdown.yml_. The options are described in greater depth in the pkgdown vignette and in the help pages for pkgdown::build_site(), pkgdown::build_articles(), pkgdown::build_reference(), and pkgdown::build_tutorials(). Essentially, they control the build of vignettes and function references. pkgdown also uses the same approach to navbars as R Markdown. `yml_navbar()` and friends will help you write the YAML for that. A useful approach to writing pkgdown YAML might be to use pkgdown_template() to build a template based on your package directory, modify with yml_pkgdown_*() and pkgdown_*() functions or yml_replace() and yml_discard(), then pass the results to use_pkgdown_yml() to write to _pkgdown.yml_

Usage

```r
yml_pkgdown(.yml, as_is = yml_blank(), extension = yml_blank())

yml_pkgdown_opts(
  .yml,
```
site_title = yml_blank(),
destination = yml_blank(),
url = yml_blank(),
toc_depth = yml_blank()
)

yml_pkgdown_development(
  .yml,
  mode = yml_blank(),
  dev_destination = yml_blank(),
  version_label = yml_blank(),
  version_tooltip = yml_blank()
)

yml_pkgdown_template(
  .yml,
  bootswatch = yml_blank(),
  ganalytics = yml_blank(),
  noindex = yml_blank(),
  package = yml_blank(),
  path = yml_blank(),
  assets = yml_blank(),
  default_assets = yml_blank()
)

yml_pkgdown_reference(.yml, ...)

pkgdown_ref(
  title = yml_blank(),
  desc = yml_blank(),
  contents = yml_blank(),
  exclude = yml_blank(),
  ...
)

yml_pkgdown_news(.yml, one_page = yml_blank())

yml_pkgdown_articles(.yml, ...)

pkgdown_article(
  title = yml_blank(),
  desc = yml_blank(),
  contents = yml_blank(),
  exclude = yml_blank(),
  ...
)

yml_pkgdown_tutorial(.yml, ...)

pkgdown_tutorial(
  name = yml_blank(),
  title = yml_blank(),
  tutorial_url = yml_blank(),
  source = yml_blank(),
  ...
)

yml_pkgdown_figures(
  .yml,
  dev = yml_blank(),
  dpi = yml_blank(),
  dev.args = yml_blank(),
  fig.ext = yml_blank(),
  fig.width = yml_blank(),
  fig.height = yml_blank(),
  fig.retina = yml_blank(),
  fig.asp = yml_blank(),
  ...
)

yml_pkgdown_docsearch(
  .yml,
  api_key = yml_blank(),
  index_name = yml_blank(),
  doc_url = yml_blank()
)

Arguments

.yml a yml object created by yml(), as_yml(), or returned by a yml_*() function
as_is Logical. Use the output_format and options that you have specified?
extension The output extension, e.g. "pdf".
site_title The title of the website (by default, this is the package name). Note that the actual YAML is title (specified as site_title to avoid duplication with content titles).
destination The path where the site should be rendered ("docs/" by default)
url URL where the site will be published; setting the URL will allow other pkgdown sites to link to your site when needed, generate a sitemap.xml to increase the searchability of your site, and generate a CNAME.
toc_depth The depth of the headers included in the Table of Contents. Note that the actual YAML is depth and is nested under toc.
mode The development mode of the site, one of: "auto", "release", "development", or "unreleased". development controls where the site is built; the color of the
package version; the optional tooltip associated with the version; and the indexing of the site by search engines. See ?pkgdown::build_site() for more details.

dev_destination
The subdirectory used for the development site, which defaults to "dev/". Note that the actual YAML is destination and is nested under development.

version_label
Label to display for "development" and "unreleased" mode. One of: "danger" (the default), "default", "info", or "warning".

version_tooltip
A custom message to include in the version tooltip

bootswatch
A bootswatch theme for the site. See the options at https://rstudio.github.io/shinythemes/.

ganalytics
A Google Analytics tracking id

noindex
Logical. Suppress indexing of your pages by web robots?

package
an R package with with directories inst/pkgdown/assets and inst/pkgdown/templates to override the default templates and add additional assets; alternatively, you can specify this in path and assets

path
A path to templates with which to override the default pkgdown templates

assets
A path to additional assets to include

default_assets
Logical. Include default assets?

... additional named R objects, such as characters or lists, to transform into YAML

title
The title of the article, reference, tutorial, or other resource

desc
A description of the article or reference

contents
The contents, which can also be dplyr-style tidy selectors (e.g "contains('index')").

exclude
What to exclude of the what's captured by contents

one_page
Logical. Create one page per release for NEWS.md?

name
The name of the file

tutorial_url
The tutorial URL to embed in an iframe

source
A URL to the source code of the tutorial

dev
The graphics device (default: "grDevices::png")

dpi
The DPI (default: 96)

dev.args
A vector of arguments to pass to dev

fig.ext
The figure extension (default: "png")

fig.width
The figure width (default: 7.2916667)

fig.height
The figure height (default: NULL)

fig.retina
The figure retina value (default: 2)

fig.asp
The aspect ratio (default: 1.618)

api_key
The API key provided by docsearch (see the pkgdown vignette)

index_name
The index name provided by docsearch (see the pkgdown vignette)

doc_url
the URL specifying the location of your documentation. Note that the actual YAML field is url but is nested.
Value

a yml object

See Also

use_pkgdown_yml() yml_navbar()

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(), use_yml_defaults(), use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(), yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(), yml_params(), yml_reference(), yml_replace(), yml_resource_files(), yml_rsconnect_email(), yml_rtdcites_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Other pkgdown: pkgdown_template()

Other websites: yml_distill_opts(), yml_site_opts()

Examples

```r
yml_empty() %>%
yml_pkgdown(
  as_is = TRUE,
  extension = "pdf"
) %>%
yml_pkgdown_reference(
  pkgdown_ref(
    title = "pkgdown functions",
    contents = "contains('function_name')"
  )
) %>%
yml_pkgdown_articles(
  pkgdown_article(
    title = "Introduction to the package"
  )
)
```

---

**yml_reference**

Write references as YAML fields

Description

`yml_reference()` creates YAML fields for references to be used in citation. `reference()` is a simple function to add references to `yml_reference()`. The easiest way to add references to an R Markdown file is to use a bibliography file, such as `.bib`, in the bibliography field (see `yml_citations()`). For documents with very few references, however, it might be useful to make the references self-contained in the YAML. `yml_reference()` can also transform to YAML `bibentry` and citation objects created by `bibentry()` and `citation()`. To cite many R packages and convert the references to YAML, it may be better to use `knitr::write_bib()` to write a bibliography file and convert it with `bib2yml()`.
Usage

```r
yml_reference(.yml, ..., .bibentry = NULL)
```

```r
reference(id = NULL, ...)
```

Arguments

- `.yml`
  a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `...`
  Fields relevant to the citation (e.g. `bibtex` fields)
- `.bibentry`
  An object created by `bibentry()` or `citation()`. Note that this requires pandoc-citeproc to be installed.
- `id`
  a character vector to use as the reference ID

Value

a `yml` object

See Also

Other `yml`: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

Other citations: `bib2yml()`, `yml_citations()`

Examples

```r
ref <- reference(
  id = "fenner2012a",
  title = "One-click science marketing",
  author = list(
    family = "Fenner",
    given = "Martin"
  ),
  `container-title` = "Nature Materials",
  volume = 11L,
  URL = "http://dx.doi.org/10.1038/nmat3283",
  DOI = "10.1038/nmat3283",
  issue = 4L,
  publisher = "Nature Publishing Group",
  page = "261-263",
  type = "article-journal",
  issued = list(
    year = 2012,
    month = 3
  )
)
```
Replace, pluck, or discard top-level YAML fields

Description

`yml_replace()` replaces a named field with another value. As opposed to duplicating top-level fields with other functions, explicitly replacing them with `yml_replace()` will not raise a warning.
**yml_replace**

yml_discard() removes values given either a character vector of names or a purrr-style lambda with a predicate (~ predicate); see the examples. yml_pluck() and yml_chuck() are wrappers around `purrr::pluck()` and `purrr::chuck()` that return yml objects.

**Usage**

```r
yml_replace(.yml, ...)  
yml_discard(.yml, .rid)  
yml_pluck(.yml, ...)  
yml_chuck(.yml, ...)
```

**Arguments**

- `.yml` a yml object created by `yml()`, `as_yml()`, or returned by a yml_*() function
- `...` additional named R objects, such as characters or lists, to transform into YAML
- `.rid` a character vector of fields to remove or a purrr-style lambda with a predicate (~ predicate) where fields that are TRUE will be discarded

**Value**

a yml object

**See Also**

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rarticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

**Examples**

```r
yml() %>%  
  yml_clean(TRUE) %>%  
  yml_replace(clean = FALSE) %>%  
  yml_discard("author")

yml() %>%  
  yml_output(  
    pdf_document(),  
    html_document()  
  ) %>%  
  yml_discard(~ length(.x) > 1)
```
yml_resource_files  Add external resource files to R Markdown document

Description

The resource_files field specifies a character vectors of paths to external resources to include in
the output, e.g. files that are necessary for rendering. These files are handled with rmarkdown::find_external_resources().

Usage

yml_resource_files(.yml, resource_files)

Arguments

.yml  a yml object created by yml(), as_yml(), or returned by a yml_*() function
resource_files  A path to a file, directory, or a wildcard pattern (such as "data/*.csv")

Value

a yml object

See Also

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(),
use_yml_defaults(), use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(),
yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(),
yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_rsconnect_email(),
yml_rscicles_opts(), yml_runtime(), yml_site_opts(), yml_toc(), yml_vignette()

Examples

```r
yml() %>%
  yml_resource_files(c("data/mydata.csv", "images/figure.png"))
```

yml_rsconnect_email  Set YAML for Scheduled Emails in RStudio Connect

Description

RStudio Connect allows you to schedule emails to send using R Markdown. It uses a special type of
YAML using the top-level field rmd_output_metadata that tells RStudio Connect about the email output. Several rsc_* fields exist to specify different components of the email, which can be set in
the YAML header or programmatically using rmarkdown::output_metadata(). See the RStudio
Connect documentation for more. yml_output_metadata() allows you to add any type of content
to the rmd_output_metadata field.
Usage

\begin{verbatim}
yml_rsconnect_email(
  .yml,
  rsc_email_subject = yml_blank(),
  rsc_email_body_html = yml_blank(),
  rsc_email_body_text = yml_blank(),
  rsc_email_images = yml_blank(),
  rsc_output_files = yml_blank(),
  rsc_email_attachments = yml_blank(),
  rsc_email_suppress_scheduled = yml_blank(),
  rsc_email_suppress_report_attachment = yml_blank(),
  resource_files = yml_blank(),
  ...
)
\end{verbatim}

\begin{verbatim}
yml_output_metadata(.yml, ...)
\end{verbatim}

Arguments

- **.yml**
  a yml object created by yml(), as_yml(), or returned by a yml_*() function

- **rsc_email_subject**
  The subject of the email. A report without an rsc_email_subject entry uses its published document name.

- **rsc_email_body_html, rsc_email_body_text**
  The body of the email, either in plain text or HTML. A report with neither entry uses an automatically generated, plain-text body with a link to the report's URL.

- **rsc_email_images**
  Images to embed in the email. The embedded image must have a Content ID that is used in the body of the HTML and when providing the image to rsc_email_images, and the image itself must be base64-encoded, e.g. with the base64enc package.

- **rsc_output_files**
  A vector of file names that should be available after the report has rendered. If you list a file that does not exist after rendering your report, Connect will log a message but continue trying to processing the other files listed. If the output files are not generated during the rendering of your report, then you will also need to list them in resource_files when you upload your report to Connect.

- **rsc_email_attachments**
  A vector of file names that should be attached to the email.

- **rsc_email_suppress_scheduled**
  Logical. Should the email schedule be suppressed? Default is FALSE.

- **rsc_email_suppress_report_attachment**
  Logical. Should the rendered document be included as an attachment? Default is TRUE.

- **resource_files**
  A file or files to host on RStudio Connect that is not generated by your report, e.g. an existing file.

- **...**
  additional named R objects, such as characters or lists, to transform into YAML
**Value**

a yml object

**See Also**

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yaml_tree()`, `has_field()`, `read_json()`, `use_yaml_defaults()`, `use_yaml_file()`, `use_yaml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdwn()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

**Examples**

```r
yml() %>%
  yml_rsconnect_email(
    rsc_email_subject = "Quarterly report",
    rsc_output_files = "data.csv",
    rsc_email_attachments = c("attachment_1.csv", "attachment_2.csv")
  )
```

---

**yml_rticles_opts**

*Set YAML related to rticles output formats*

**Description**

The rticles package include numerous output formats specific to academic journals. All of these can take YAML similar to `pdf_document()`. Additionally, two templates include custom YAML, `rticles::sage_article()` and `rticles::sim_article()`. See the help pages for these functions for more details and the sources of the LaTeX templates used for each.

**Usage**

```r
yml_rticles_opts(
  .yml,
  title = yml_blank(),
  runninghead = yml_blank(),
  author = yml_blank(),
  authormark = yml_blank(),
  address = yml_blank(),
  corrauth = yml_blank(),
  corres = yml_blank(),
  email = yml_blank(),
  abstract = yml_blank(),
  received = yml_blank(),
  revised = yml_blank(),
  accepted = yml_blank(),
)```
```r
keywords = yml_blank(),
bibliography = yml_blank(),
longtable = yml_blank(),
classoption = yml_blank(),
header_includes = yml_blank(),
include_after = yml_blank(),
... )

articles_author(name = yml_blank(), num = yml_blank())

articles_address(name = yml_blank(), org = yml_blank())

articles_corr_author(
    name = yml_blank(),
    author = yml_blank(),
    address = yml_blank()
)
```

**Arguments**

- `.yml` a YAML object created by `yml()`, `as_yaml()`, or returned by a `yml_*()` function
- `title` Title of the manuscript
- `runninghead` A character vector, a short author list for the header (sage_article)
- `author` A list of authors, containing `name` and `num` fields (sage_article, sim_article). Use `articles_author()` or a list to specify.
- `authormark` A character vector, the short author list for the header (sim_article)
- `address` list containing `num` and `org` for defining author affiliations (sage_article, sim_article). Use `articles_address()` or a list to specify.
- `corrauth` corresponding author name and address (sage_article). Use `articles_corr_author()` or a list to specify.
- `corres` author and address for correspondence (sim_article). Use `articles_corr_author()` or a list to specify.
- `email` The email of the correspondence author (sage_article)
- `abstract` The abstract, limited to 200 words (sage_article), 250 words (sim_article)
- `received, revised, accepted` The dates of submission, revision, and acceptance of the manuscript (sim_article)
- `keywords` The keywords for the article (sage_article), up to 6 keywords (sim_article)
- `bibliography` BibTeX `.bib` file name (sage_article, sim_article)
- `longtable` Logical. Include the `longtable` package? Used by default from pandoc to convert markdown to LaTeX code (sim_article)
- `classoption` a character vector of `classoption` options for the `sagej` class (sage_article)
- `header_includes` additional LaTeX code to include in the header, before the `\begin{document}` statement (sage_article, sim_article). Note that the actual YAML field is `header-includes`


```r
yml() %>%
  yml_runtime_opts(received = "09-12-2014")
```

### Examples

#### Description

The `yml_runtime` field lets you use Shiny in your R Markdown document, making it interactive. See the R Markdown book for examples.

#### Usage

```r
yml_runtime(.yml, runtime = c("static", "shiny", "shiny_prerendered"))
```

#### Arguments

- `.yml` a `yml` object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `runtime` The runtime target for rendering. `static`, the default, renders static documents; `shiny` allows you to include use Shiny in your document. `shiny_prerendered` is a subset of the `shiny` runtime that allows pre-rendering of app components (see the R Markdown site for more)
Value

a `yml` object

See Also

Other `yml`: `asis_yaml_output()`, `bib2yml()`, `draw_yaml_tree()`, `has_field()`, `read_json()`, `use_yaml_defaults()`, `use_yaml_file()`, `use_yaml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdown()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtticles_opts()`, `yml_site_opts()`, `yml_toc()`, `yml_vignette()`

Other R Markdown: `yml_clean()`, `yml_params()`, `yml_site_opts()`, `yml_vignette()`

Other shiny: `yml_params()`

Examples

```r
yml() %>%
yml_runtime("shiny")
```

---

```r
dataset <- yml_site_opts(.yml, name = yml_blank(),
                         favicon = yml_blank(),
                         output_dir = yml_blank(),
                         include = yml_blank(),
                         exclude = yml_blank(),
                         new_session = yml_blank(),
                         ...)
```

Description

R Markdown has a simple website builder baked in (see the R Markdown book for a detailed description). An R Markdown website must have at least have an `index.Rmd` file and a `_site.yml` file (which can be empty). Including YAML in `_site.yml` will apply it to all R Markdown files for the website, e.g. setting the output format here will tell R Markdown to use that format across the website. R Markdown websites also support navbars, which you can specify with YAML (see `yml_navbar()`), as well as `?rmarkdown::render_site` and `?rmarkdown::html_document`). Pass `navbar_page()` to the `left` or `right` field to set up page tabs and use `navbar_separator()` to include a separator. In addition to writing YAML with `yml_*()` functions, `use_site_yml()` will take the a `yml` object and write it to a `_site.yml` file for you.

Usage

```r
yml_site_opts(
  .yml,
  name = yml_blank(),
  favicon = yml_blank(),
  output_dir = yml_blank(),
  include = yml_blank(),
  exclude = yml_blank(),
  new_session = yml_blank(),
  ...
)```
yml_site_opts

)

yml Navbar

  .yml,
  title = yml blank(),
  type = yml blank(),
  left = yml blank(),
  right = yml blank(),
  ...
)

navbar_page(
  text = yml blank(),
  href = yml blank(),
  icon = yml blank(),
  menu = yml blank(),
  ...
)

navbar_separator()

Arguments

  .yml  a yml object created by yml(), as_yml(), or returned by a yml_*() function
  name  The name of the website
  favicon  Path to a file to use as the favicon
  output_dir  Directory to copy site content into ("_site" is the default if none is specified)
  include, exclude  Files to include or exclude from the copied into output_dir. You can use * to indicate a wildcard selection, e.g. ".csv".
  new_session  Logical. Should each website file be rendered in a new R session?
  ...
  title  The title of the website
  type  The color scheme for the navigation bar: either "default" or "inverse".
  left, right  the side of the navbar a navbar_page() should go (see example)
  text  The link text
  href  The link URL
  icon  An icon to include
  menu  drop-down menus specified by including another navbar_page()

Value

  a yml object
**See Also**

use_site_yml() use_navbar_yml() use_index_rmd()

Other yml: asis_yaml_output(), bib2yml(), draw_yml_tree(), has_field(), read_json(),
use_yml_defaults(), use_yml_file(), use_yml(), yml_author(), yml_blogdown_opts(), yml_bookdown_opts(),
yml_citations(), yml_clean(), yml_distill_opts(), yml_latex_opts(), yml_output(), yml_pagedown_opts(),
yml_params(), yml_pkgdown(), yml_reference(), yml_replace(), yml_resource_files(),
yml_rscconnect_email(), yml_rtiicles_opts(), yml_runtime(), yml_toc(), yml_vignette()

Other R Markdown: yml_clean(), yml_params(), yml_runtime(), yml_vignette()

Other websites: yml_distill_opts(), yml_pkgdown()

**Examples**

```r
ylm_empty() %>%
  ylm_site_opts(
    name = "my-website",
    output_dir = "_site",
    include = "demo.R",
    exclude = c("docs.txt", "*.csv")
  ) %>%
  ylm Navbar(
    title = "My Website",
    left = list(
      navbar_page("Home", href = "index.html"),
      navbar_page(navbar_separator(), href = "about.html")
    )
  ) %>%
  ylm_output(html_document(toc = TRUE, highlight = "textmate"))
```

---

**ylm_toc**

*Specify Table of Contents options*

**Description**

It’s generally better to specify Table of Contents in the output function you are using so you have a clearer idea of your options (e.g. html_document(toc = TRUE, toc_float = TRUE)). However, you can also generally specify at the top level of YAML.

**Usage**

```r
ylm_toc(
  .ylm,
  toc = ylm_blank(),
  toc_depth = ylm_blank(),
  toc_title = ylm_blank(),
  ...
)
```
yml_verbatim

Arguments

- `.yml` a yml object created by `yml()`, `as_yml()`, or returned by a yml_*() function
- `toc` Logical. Use a Table of Contents?
- `toc_depth` An integer. The depth of headers to use in the TOC. Note that the actual YAML field is `toc-depth`.
- `toc_title` The title of the TOC. Note that the actual YAML field is `toc-title`.
- `...` additional named R objects, such as characters or lists, to transform into YAML

Value

a yml object

See Also

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgn()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rtrees_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_vignette()`

Examples

```r
yml() %>%
yml_toc(toc = TRUE, toc_depth = 1, toc_title = "Article Outline")
```

---

## yml_verbatim

**Write YAML field or content verbatim**

### Description

`yml_verbatim()` is a helper function to write YAML precisely as given to the yml_*() function rather than the defaults in `ymlthis` and yaml. `ymlthis` uses the yaml package to check for valid syntax; yaml and `ymlthis` together make decisions about how to write syntax, which can often be done in numerous valid ways. See `yml::as.yaml()` for more details.

### Usage

```r
yml_verbatim(x)
```

### Arguments

- `x` a character vector
**Value**

an object of class `verbatim`

**Examples**

```r
# "yes" and "no" serve as alternatives to 'true' and 'false'. This writes
# "yes" literally.
yml_verbatim("yes")
```

---

**yml_vignette**  
*Set up a package vignette*

**Description**

To use an R Markdown file as a vignette, you need to specify an output format appropriate for inclusion in a package (for example, the lightweight `html_vignette()` output function included in `rmarkdown`) and to specify the `vignette` field, which specifies the title, engine, and encoding type of the vignette. See also `usethis::use_vignette()` for setting up a package vignette.

**Usage**

```r
yml_vignette(.yml, title, engine = "knitr::rmarkdown", encoding = "UTF-8")
```

**Arguments**

- `.yml`  
a yml object created by `yml()`, `as_yml()`, or returned by a `yml_*()` function
- `title`  
The title of the vignette
- `engine`  
The rendering engine for the vignette ("knitr::rmarkdown" by default)
- `encoding`  
The character encoding for the document ("UTF-8" by default).

**Value**

a yml object

**See Also**

Other yml: `asis_yaml_output()`, `bib2yml()`, `draw_yml_tree()`, `has_field()`, `read_json()`, `use_yml_defaults()`, `use_yml_file()`, `use_yml()`, `yml_author()`, `yml_blogdown_opts()`, `yml_bookdown_opts()`, `yml_citations()`, `yml_clean()`, `yml_distill_opts()`, `yml_latex_opts()`, `yml_output()`, `yml_pagedown_opts()`, `yml_params()`, `yml_pkgdwn()`, `yml_reference()`, `yml_replace()`, `yml_resource_files()`, `yml_rsconnect_email()`, `yml_rarticles_opts()`, `yml_runtime()`, `yml_site_opts()`, `yml_toc()`

Other R Markdown: `yml_clean()`, `yml_params()`, `yml_runtime()`, `yml_site_opts()`
Examples

```r
yml() %>%
  yml_output(html_vignette()) %>%
  yml_vignette("An introduction to R Markdown")
```
Index

as_yml, 3
as_yml(), 21
asis_yaml_output, 3, 4, 6, 9, 16, 18, 19, 21, 24, 27, 29, 31, 32, 37, 40, 42, 43, 48, 52, 53, 55, 56, 58, 60, 61, 63–65

base::strftime(), 46
bib2yml, 3, 4, 6, 9, 16, 18, 19, 21, 24, 27, 29, 31, 32, 37, 40, 42, 43, 48, 52, 53, 55, 56, 58, 60, 61, 63–65
bib2yml(), 52
bibentry(), 52
blogdown_archetypes (blogdown_template), 5
blogdown_template, 5
blogdown_template(), 25
bookdown::gitbook(), 7, 8
citation(), 52
code_chunk, 5
code_chunk(), 18
distill_collection (yml_distill_opts), 34
distill_listing (yml_distill_opts), 34
distill_resources (yml_distill_opts), 34
draw_yml_tree, 3, 4, 6, 9, 16, 18, 19, 21, 24, 27, 29, 31, 32, 37, 40, 42, 43, 48, 52, 53, 55, 56, 58, 60, 61, 63–65

format.Date(), 23

get_rmd_defaults (use_yml_defaults), 18
get_yml_defaults (use_yml_defaults), 18
get_yml_defaults(), 18, 19
gitbook_config, 7, 29

has_field, 3, 4, 6, 9, 16, 18, 19, 21, 24, 27, 29, 31, 32, 37, 40, 42, 43, 48, 52, 53, 55, 56, 58, 60, 61, 63–65

includes2, 10
is_yml, 10
is_yml_blank (yml_blank), 24
knitr::write_bib(), 4, 52
last_yml, 11
last_yml(), 17
navbar_page (yml_site_opts), 61
navbar_separator (yml_site_opts), 61
pagedown_business_card_template, 11, 43
pagedown_business_card_template(), 42
pagedown_person
(pagedown_business_card_template), 11
pandoc_highlight_styles (pandoc_template_types), 14
pandoc_template_types, 14
pkgdown_article (yml_pkgdown), 48
pkgdown_ref (yml_pkgdown), 48
pkgdown_template, 15, 52
pkgdown_tutorial (yml_pkgdown), 48
purrr::chuck(), 55
purrr::pluck(), 55

read_json, 3, 4, 6, 9, 15, 18, 19, 21, 24, 27, 29, 31, 32, 37, 40, 42, 43, 48, 52, 53, 55, 56, 58, 60, 61, 63–65
read_toml (read_json), 15
reference (yml_reference), 52
rticles_address (yml_rarticles_opts), 58
rticles_author (yml_rarticles_opts), 58
rticles_corr_author (yml_rarticles_opts), 58

setup_chunk (code_chunk), 5
setup_chunk(), 17, 18
shiny::animationOptions(), 46
shiny::validateCssUnit(), 46