

# YAML metadata for R Markdown with examples

Hao Liang

2021-04-19

## Contents

About the project	1
Useful websites	1
useful tools	1
Terminology and norms	2
Basic	2
Syntax . . . . .	2
to be continued...	4

## About the project

Rmd makes it possible to use a YAML header to specify certain parameters right at the beginning of the document. Built-in YAML parameters make it easier to create more organized and informative reports. However, there are few tutorials or summarized articles to display all the settings and parameters for `YAML metadata` in R Markdown. This readme file give you clear and enough materials for that.

## Useful websites

- <https://rmarkdown.rstudio.com/formats.html> — R Markdown formats from RStudio
- <https://www.r-bloggers.com/2020/08/useful-yaml-options-for-generating-html-reports-in-r/> — Useful YAML options for generating HTML reports in R

## useful tools

- <https://yamlthis.r-lib.org/index.html> — `yamlthis`: a package for writing YAML for R Markdown
- <https://github.com/kamapu/yamlme> — This package aims to save documents with their respective settings (`yaml-head`) in R-objects.

## Terminology and norms

- R lang – use upper case letter R
- Rmd – capitalizes the first letter for Rmd files
- R Markdown – the R Markdown module or notebook of Rstudio
- rmarkdown – the package of **rmarkdown**
- YAML – / jæməl/ a recursive acronym for “YAML Ain’t Markup Language”
- YAML header/metadata – the settings data written by YAML in Rmd header
- All the YAML metadata are lower case letters except file paths or file names

## Basic

### Syntax

#### Data Structure

Source: ymlthis package(Barrett and Iannone 2021) vignette

A YAML code block should be fenced in with `---` before and after (you can also use `...` to end the YAML block, but this is not very common in R Markdown).

1. Scalars, or variables, are defined using a colon and a **space**. A dictionary is represented in a simple **key: value** form (the colon must be followed by a **space**)

```
---
title: "YAML metadata for R Markdown with examples"
author: Hao Liang
fontsize: 12pt
---
```

2. All members of a list are lines beginning at the same indentation level starting with a `'- '` (a dash and a space):

```
# A list of tasty fruits
- Apple
- Orange
- Strawberry
- Mango

# OR
[Apple, Orange, Strawberry, Mango]
```

3. More complicated data structures are possible, such as lists of dictionaries, dictionaries whose values are lists or a mix of both:

```
author:
- Name_1    # can be indented or not
- Name_2    # but be consistent among different entries
```

4. Dictionaries and lists can also be represented in an abbreviated form if you really want to:

```
author: [Name_1, Name_2]
```

5. Strings can be denoted with a `|` character, which preserves newlines, or a `>` character, which folds newlines.

```
abstract: |
  One or two sentences providing a basic introduction to the field, comprehensible to a scientist
  Two to three sentences of more detailed background, comprehensible to scientists in related disc

abstract: >
  One or two sentences providing a basic introduction to the field, comprehensible to a scientist
  Two to three sentences of more detailed background, comprehensible to scientists in related disc
```

Logical values in YAML are unusual: `true/false`, `yes/no`, and `on/off` are all equivalent to `TRUE/FALSE` in R. Any of these turn on the table of contents:

```
toc: true
toc: yes
toc: on
```

## Indent

In YAML, spaces(indent) are used to indicate nesting. When we want to specify the output function `pdf_document(toc = TRUE)`, we need to nest it under the `output` field. We also need to nest `toc` under `pdf_document` so that it gets passed to that function correctly.

```
---
output:
  pdf_document:
    toc: true
---
```

In R, the equivalent structure is a nested list, each with a name: `list(output = list(pdf_document = list(toc = TRUE)))`. Similarly, you can call this in R Markdown using the metadata object, e.g. `metadata$output$pdf_document$toc`. The hierarchical structure (which you can see with `draw_yaml_tree()`) looks like this:

```
output:
  pdf_document:
    toc: true
```

Without the extra indents, YAML doesn't know `toc` is connected to `pdf_document` and thinks the value of `pdf_document` is `NULL`. YAML that looks like this:

```
output:
  pdf_document:
toc: true

output:
  pdf_document: null
toc: true
```

Some YAML fields take unnamed vectors as their value. You can specify an element of the vector by adding a new line and - (note that the values are not indented below category here).

```
category:
- R
- Reproducible Research
```

## quote and R code

You may have noticed that strings in YAML don't always need to be quoted. However, it can be useful to explicitly wrap strings in quotes when they contain special characters like : and @.

```
title: 'R Markdown: An Introduction'
```

R code can be written as inline expressions . `yml_code()` will capture R code for you and put it in a valid format. R code in `params` needs to be slightly different: use `!r`(e.g. `!r expr`) to call an R object.

```
author: 'liang'
params:
  date: !r Sys.Date()
```

## to be continued...

Barrett, Malcolm, and Richard Iannone. 2021. *Ymlthis: Write 'YAML' for 'r Markdown', 'Bookdown', 'Blogdown', and More*. <https://CRAN.R-project.org/package=ymlthis>.