Showcasing The Tome Class

Bring Your Focus Back on Writing

Nicklas Vraa

01/01/2022

ISBN: 978-0201529838

Indhold

Preface

This is a showcase of a LaTeX template for academic writing. The template uses custom commands and environments, which allows a higher level of abstraction, to speed up writing. This short pdf is the result of compiling a TeX-document, which was written using the Academic document class. This PDF by itself is not particularly interesting. Instead look at its source-code.

Introduction

This document-class aims to simplify the process of writing beautiful and minimalistic academic papers. Academic extends the ever-popular `article` class, but greatly simplifies the syntax with which you typeset your document. In the following sections, we showcase this syntax by actually using it to define the document, you are looking at now. To use the `Academic` document class for your document, simply add `\usepackage[]{academic}`. It will pass through any optional arguments to the basic `article` class, e.g. `[twocolumn]`.

1.1. Motivation

While LaTeX is the indisputable king for typesetting academic papers, it does have a steep learning curve and is very syntax-heavy. To ease the burden of typesetting and bring the focus back on the content, the syntax should be as light as possible - hence this humble project.

The Syntax

This section will describe the syntax of the `tome` class.

2.1. Formatting

This is **bold font**. Here is some *italic font*. Maybe you need to underline, or ~~strike~~ it. You've already seen some `inline code`, and of course you can write inline $math^2$. In order, the commands for these are `\b`, `\i`, `\u`, `\s`, `\c` and `\m`. These were chosen to be easily remembered.

2.2. Metadata

To add titles, use `\titles{title...}` and `\subtitle{subtitle...}`. Similarly, there are the `\author {Name Lastname}` and the `\date{...}` commands. If you wish to have the author and date on one line, simply write the date in the `\author` command and disregard the `\date` command. To add a table of contents, use `\toc`. You can add a header using `\header{left...}{center...}{right ...}`. All parameters can of course be left blank. To include this frontmatter, use `\front` after `\begin{document}`.

If you supply an ISBN-code, and/or a creative commons copyright license, they will show up on the page immediately after the title page. These are defined by the `\isbn{1234...}` command and the `license{type...}{modifier...}{version...}` command. To see what types of licenses are possible, see the `doclicense` package [1].

2.3. Headings

To make a heading, simple use the `\h` command. For a subheading, just add another `h`, i.e. `\hh`. If you have ever used Markdown, this should be familier. For a non-numbered heading, use `\H`. These will also be included in the table of contents.

2.4. Environments

This section showcases the various environments which are available in the Academic class. The available environments are `bullets, numbers, code` and `math`. Begin and end them with `\begin{}` and `\end{}`.

2.4.1. Code Blocks

Here is a code block. The declaration always follow `\begin{code}{label...}{lang...}{caption...} ... \end{code}`. The fencing horizontal lines are separate entities and will position themselves vertically, such that they appear natural.

```python
import numpy as np
def add(x,y):
    return x+y

# This is a long comment. Note that lines will wrap naturally.
```

Snippet 2.1: This is some python code.

2.4.2. Lists

This is an unordered list, using the environment `\begin{bullets}...\end{bullets}`

- This is a very long item to test the wrapping of text in the unordered environment.
  - Another item, but indented.
    - Yet another item.
    - An item on the same level.

This is a numbered list, using the environment `\begin{numbers}...\end{numbers}`

1. This is an item.
   1.1. Another item, but indented.
      1.1.1. Yet another item.
      1.1.2. An item on the same level.
2. Last item.

2.4.3. Mathematics

Here is a cool equation. The declaration always follow `\begin{math}{label} ... \end{math}`

$$e^{i\pi} + 1 = 0 \tag{2.1}$$

2.5. Tables

There are three types, namely the `\cols`, `\rows` and `\grid` tables. The declaration always follow `\<type>{label}{caption}{...}`. Take a look at the source-files to see how simple it is to create a decent-looking table. These three tables will cover 90% of your table-needs, but since this is just the article-class extended, you have access to the `tabularray` [2] package for more tables.

This	is	a	cool	table
1	2	3	4	5
a	b	c	d	e

Tabel 2.1: This is a column-table. Notice that the caption lines up.

Another	1	2	3	4
One	a	b	c	d

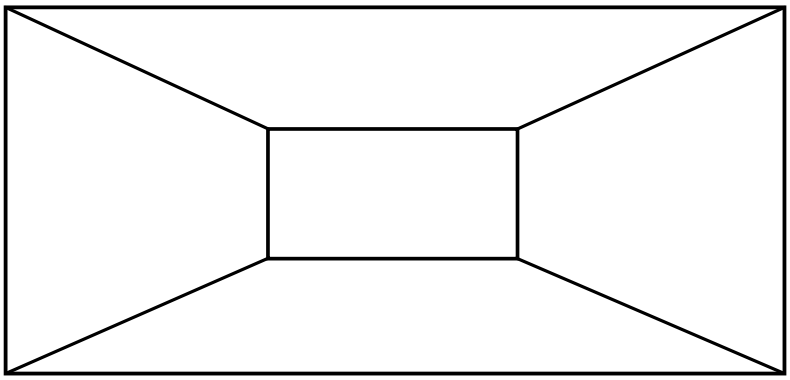
Tabel 2.2: This is a row-table.

And	is	a	table
Another	1	2	3
one	2	4	6

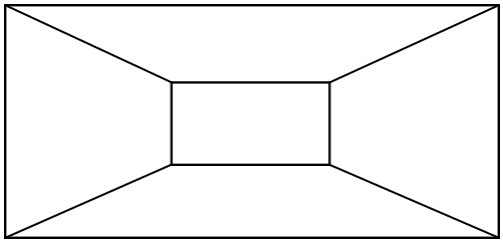
Tabel 2.3: This is a grid-table.

2.6. Figures

The `\fig[size...]{label...}{caption...}{path...}` command only needs to know an internal label for referencing, the path of your resource, and a caption. It takes care of placing it correctly and is file-format agnostic, i.e. it works the same for both regular images and vector graphics. Optionally, you can specify a scale, like `[0.5]`, which will shrink the image to half the width of the column.



Figur 2.1: This is an svg-figure scaled by 0.7, and this is a very long description of the figure to showcase how captions will automatically fit itself underneath its figure.



Figur 2.2: This is a png-figure.

2.7. Referencing

Here is a link to a webpage, made using `\url{text...}{address...}`. Internal referencing is done using `\r{label...}` These are references to the previous equation 2.1, snippet 2.1, table 2.1, figure 2.1 and figure 2.2. Equation 2.1 is an example of how to easily capitalize the reference name with `\R{label...}` when appropriate. Labels are also automatically added to sections, like chapter 1 and section 2.4.3. The internal labels are the same as the section text. For all references, both the name and number are links. Of cource, you can cite sources using `\cite{...}`, like this [3] or this [2]. Insert your bibliography using `\bib{file...}`.

Conclusion

This class offers a layer of abstraction for those who want an easy way to produce professional-looking academic papers, while not worrying too much about learning all the intricacies of LaTeX, but still having access to the full power of the standard `article` class, should it be necessary.

The plan is to formulate a markup language inspired by YAML and Markdown with a transpiler written in Python, that will translate it into TeX, and using this document class, produce PDF's. I don't like the way lists and tables are handled in LaTeX. Taking inspiration from Markdown, these two elements in particular, could be made much more intuitive. This development will be hosted in its own repository.

Don't agree with some of the stylistic choices? Feel free to change the class. Simply edit `academic.cls` to your liking. The file is not very long and fairly simple, so it should be easy for someone with rudamentary knowledge of LaTeX.

Command Overview

Commands in alphabetical order.

- `\authors{Name Lastname...}`.
  Add your information to the frontmatter.
- `\b{text...}`
  Enbolden text.
- `\begin{code}{lang...}{label...}{caption...}`
  ... `\end{code}`
  Enclose and write code.
- `\begin{math}{label...}{caption...}`
  ... `\end{math}`
  Enclose and wite math.
- `\bib{path/to/bib-file.bib}`
  Print bibliography/references.
- `\byline{Name Lastname - date...}`
  Add author and date on the same line.
- `\c{code...}`
  Format as code.
- `\cite{source...}`
  Cite a source from your bibliography.
- `\cols{label...}{caption...}{content...}`
  Table of columns.
- `\date{11-11-11...}`.
  Add a date to the frontmatter.
- `\fig{path...}{label...}{caption...}`
  Place a figure.
- `\front`
  Print frontmatter, like titles, authors and date.
- `\grid{label...}{caption...}{content...}`
  Grid-like table.
- `\h{...}`, `\hh{...}`, ... , `\hhhhh{...}`
  Add numbered headings of various levels.
- `\H{...}`
  Add non-numbered heading.
- `\header{left...}{center...}{right...}`
  Define a document header.
- `\i{text...}`
  Italize text.
- `\m{math...}`
  Format as math.

- `\r{label...}`
  Internally reference, i.e. linking to labels.
- `\R{label...}`
  Capitalize-first-letter equivalent of `\r{}`.
- `\rows{label...}{caption...}{content...}`
  Table of rows.
- `\s{text...}`
  Strike text.
- `\titles{title...}{subtitle...}`.
  Add a title and subtitle to the frontmatter.
- `\u{text...}`
  Underline text.
- `\url{text...}{https...}`
  Reference a website.

Acknowlegdements

This class leverages the hard work that went into creating all the amazing packages, which it imports. I encourage anyone finding use for this project to visit the individual package repositories. Check out the full list at the top of `academic.cls`.

Litteratur

[1] Robin Schneider. Github - ypid/doclicense: Support for putting documents under a license. https://github.com/ypid/latex-packages/tree/master/doclicense

[2] Lvjr. Github - lvjr/tabularray: Typeset tabulars and arrays with latex3 syntax. https://github.com/lvjr/tabularray

[3] Geoffrey Poore. GitHub - gpoore/minted: syntax highlighting using the Pygments library. https://github.com/gpoore/minted