`toc2md` User Guide

Version 0.9.2

https://pypi.org/project/toc2md

Carlo Alessandro Verre

carlo.alessandro.verre@gmail.com

july 27, 2020

Contents

1. Before Usage

1.1. Introduction

`toc2md` is a small CLI utility written in Python 3, aimed to insert or update a nested chapter numbering and a linked table of contents (TOC) into a Markdown (MD) file.

"Nested chapter numbering" means level 1 chapters gets 1-item numbering (1. 2. ...), level 2 chapters gets 2-item numbering (1.1. 1.2. ...) and so on.

"Linked table of contents" means that if we click on a chapter title in the table of contents then (under given

conditions) we jump to chapter body.

In order to read this document you only need a minimum knowledge of MD and HTML.

In 2. Usage we show by rules and examples how to use toc2md, alone or as a coroutine together with ReText MD editor.

In 3. After Usage we discuss three ways to get a PDF document from a MD file.

1.2. Installation

Be sure you got last `pip` version typing at terminal (without `sudo`):

```
$ python3 -m pip install --user --upgrade pip
```

Now you can install `toc2md` typing (without `sudo`):

```
$ python3 -m pip install toc2md
```

2. Usage

2.1. Format

Usage: `toc2md [-h] [-H] [-V] [file]`

Positional argument:

- `file` - MD file to be backed up, read, processed and rewritten

Optional arguments:

- `-h, --help` - show a help message and exit
- `-H, --user-guide` - open User Guide in PDF format and exit
- `-V, --version` - show program's version number and exit

2.2. Logic

If we type at terminal:

```
$ toc2md example.md
```

toc2md program:

- copies `example.md` into a timestamped backup file `example-YYYY.mm.dd-HH:MM:SS.md`
- writes on standard output something like:

```
file '/home/xxxx/example.md' has been backed up
into '/home/xxxx/example-2020.06.05-18.19.07.md'
```

- reads example.md
- checks correct heading sequence (see below)
- deletes old chapter numbering (if any) and old TOC (if any)
- inserts new nested chapter numbering and new linked TOC
- rewrites the updated version into example.md
- writes on standard output something like:

```
file '/home/xxxx/example.md' has been updated
```

There is neither configuration file nor options. Logic has been kept as simple as possible, but the file to be processed must obey some rule, as follows.

toc2md processing is driven by headings in input file. A heading for us is a line which starts with a '#' character. The heading level is the number of leading '#' characters, between 1 and 6. So a heading is made by:

- a string of 1 to 6 '#' characters
- a blank
- a title

Title must start with an alphabetic character.

'#' characters after the sixth are silently discarded.

The blank after '#' characters if not present is silently inserted.

Any line not starting with '#' is processed as normal text, including MD headings as:

```
Something
===
```

or HTML headings like:

```
<h3>Something Else</h3>
```

which are anyway deprecated and **must** be avoided.

Titles in headings are:

- level 1 heading: title of the whole document
- first level 2 heading: title of the TOC
- following level 2 headings: titles of level 1 chapters (as 1. 2. ...)
- level 3 headings: titles of level 2 chapters (as 1.1. 1.2. ...)
- level 4 headings: titles of level 3 chapters (as 1.1.1. 1.1.2. ...)
- level 5 headings: titles of level 4 chapters (as 1.1.1.1. 1.1.1.2. ...)
- level 6 headings: titles of level 5 chapters (as 1.1.1.1.1. 1.1.1.1.2. ...)

So the input file **must** contain the following headings, freely intermixed with normal text, but in this order:

- one level 1 heading with title of the whole document
- one level 2 heading with title of the TOC
- one level 2 heading with title of the first level 1 chapter (1.)
- zero or more level 2..6 headings with titles of level 1..5 chapters

Beware: no heading can **ever** have a level higher than that of the previous heading plus one.

Each non-heading line in input is transcribed as it is into output, except for those between the first and second level 2 headings. These lines are supposed to be the old TOC and hence are deleted and replaced in output by the new TOC.

Chapter numbering in input headings is deleted and replaced in output by the new chapter numbering.

2.3. Example

An example will make everything clearer. As MD editor we will use ReText.

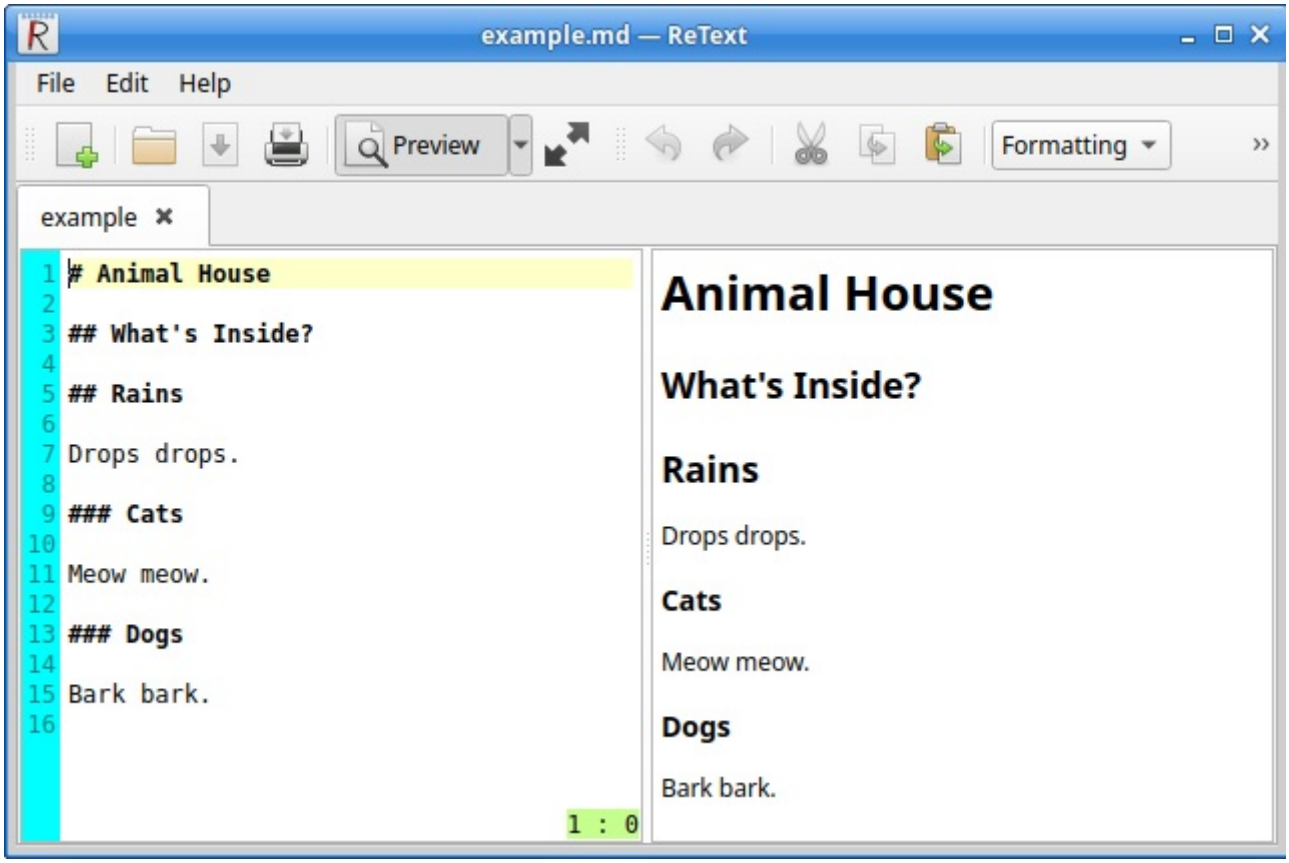
Beware: from another MD editor you could get a different behavior.

We open a terminal window and type:

```
$ retext example.md &
```

The '&' forks ReText as a separate process allowing us to use concurrently the terminal window and the ReText window.

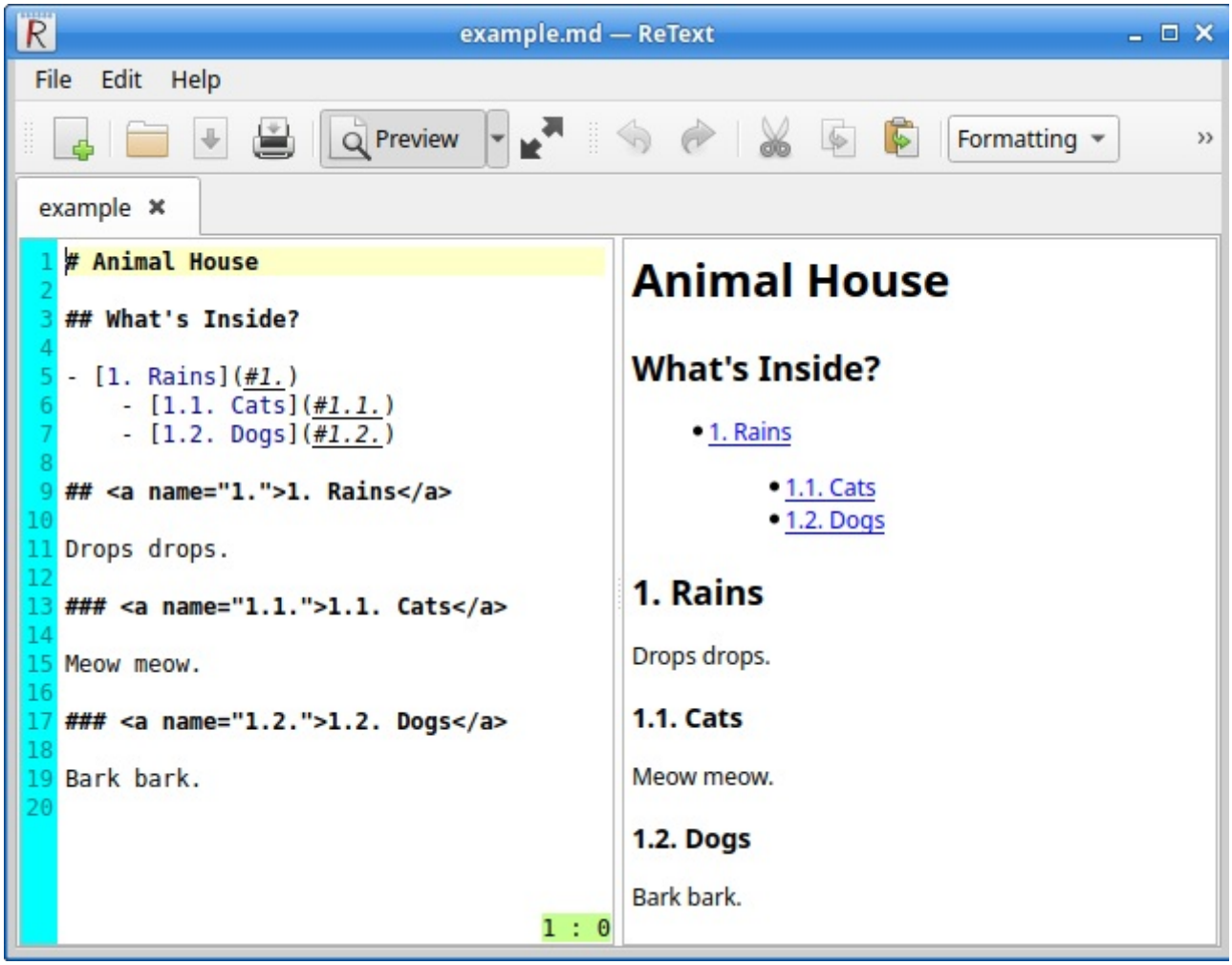
In ReText window we write:



We save (by clicking **File → Save**, or typing **Ctrl+S**, or clicking the **[↓]** button), then on terminal window we run:

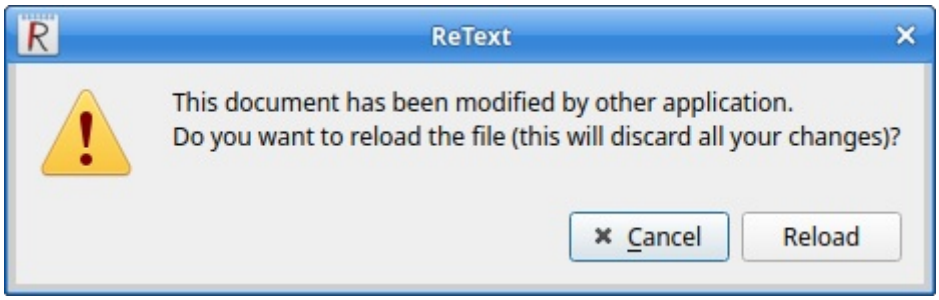
```
$ toc2md example.md
```

and the text in ReText window magically becomes:



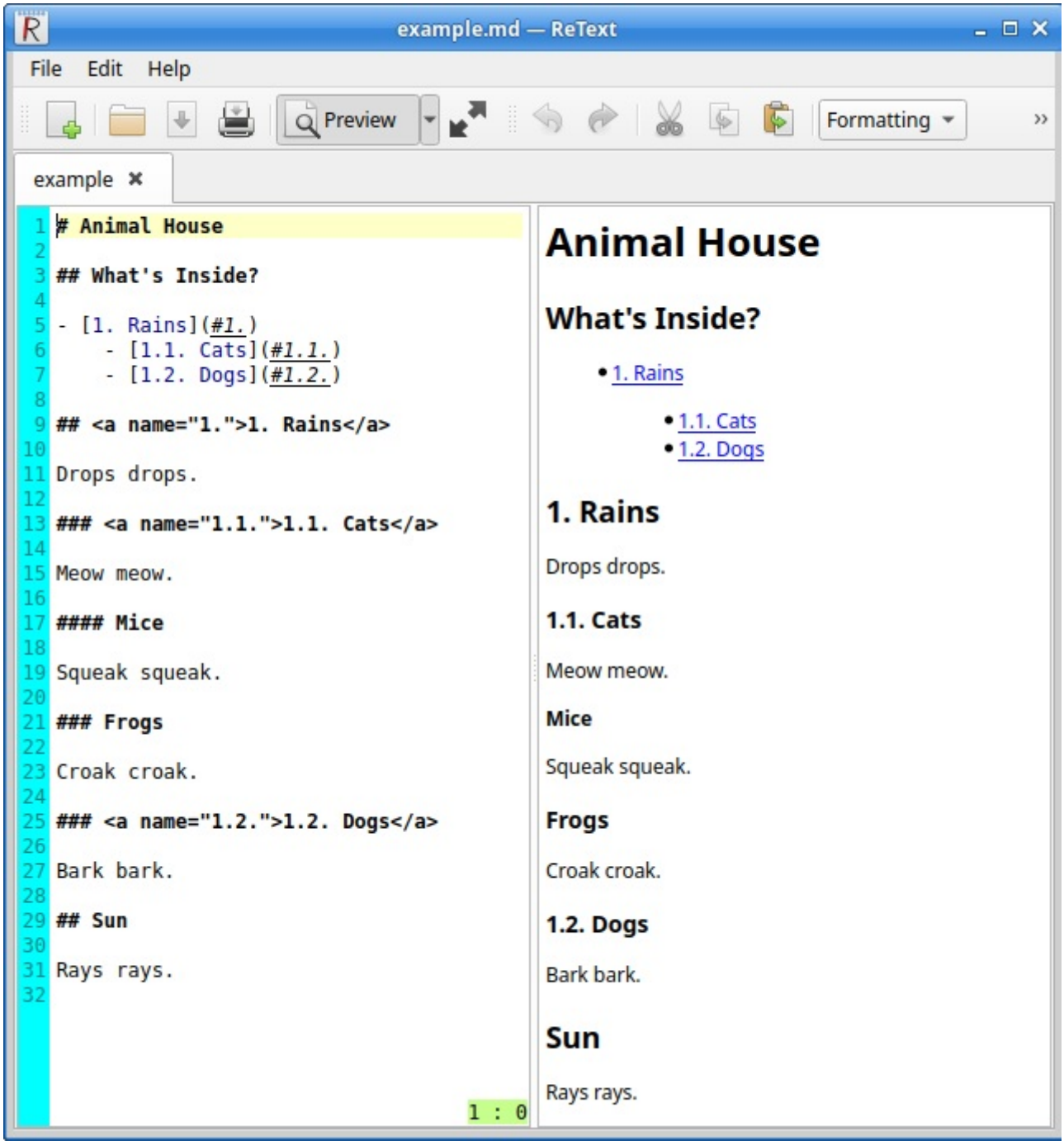
Note in the newly created TOC the links pointing to chapter headings.

Beware: if after running '`toc2md example.md`' ReText opens a dialog sub-window like this:

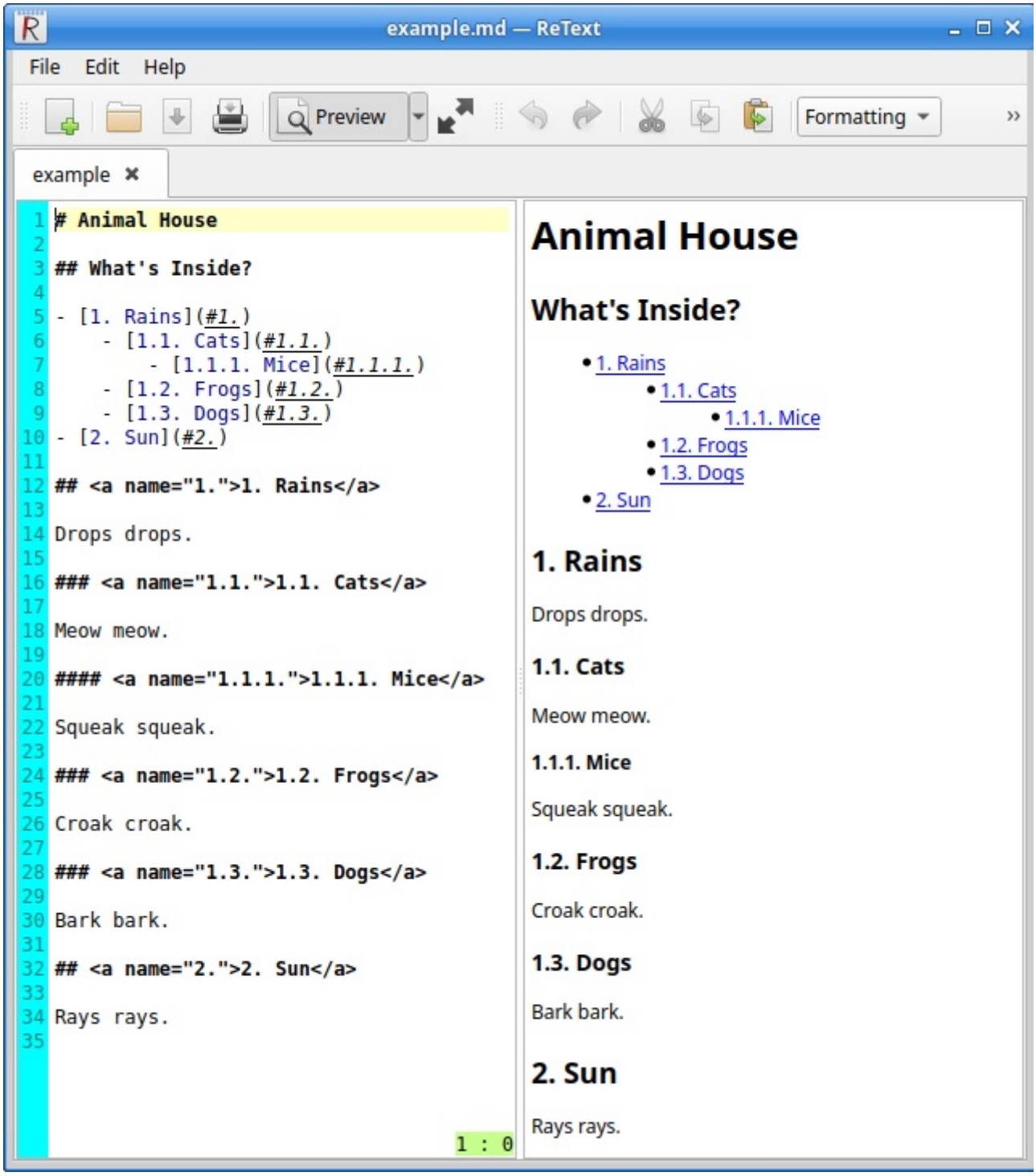


don't click **[Reload]**! The sub-window means you forgot to save! Click **[X Cancel]** (maybe twice) on sub-window, save ReText window, and run '`toc2md example.md`' in the terminal window again.

Now we realize Cats eat Mice, Frogs like Rains, and after Rains the Sun comes, so we insert Mice in Cats, Frogs between Cats and Dogs, and Sun after Rains.



As before, we save ReText window and we run '`toc2md example.md`' in terminal window, so the text in ReText window becomes:



Done!

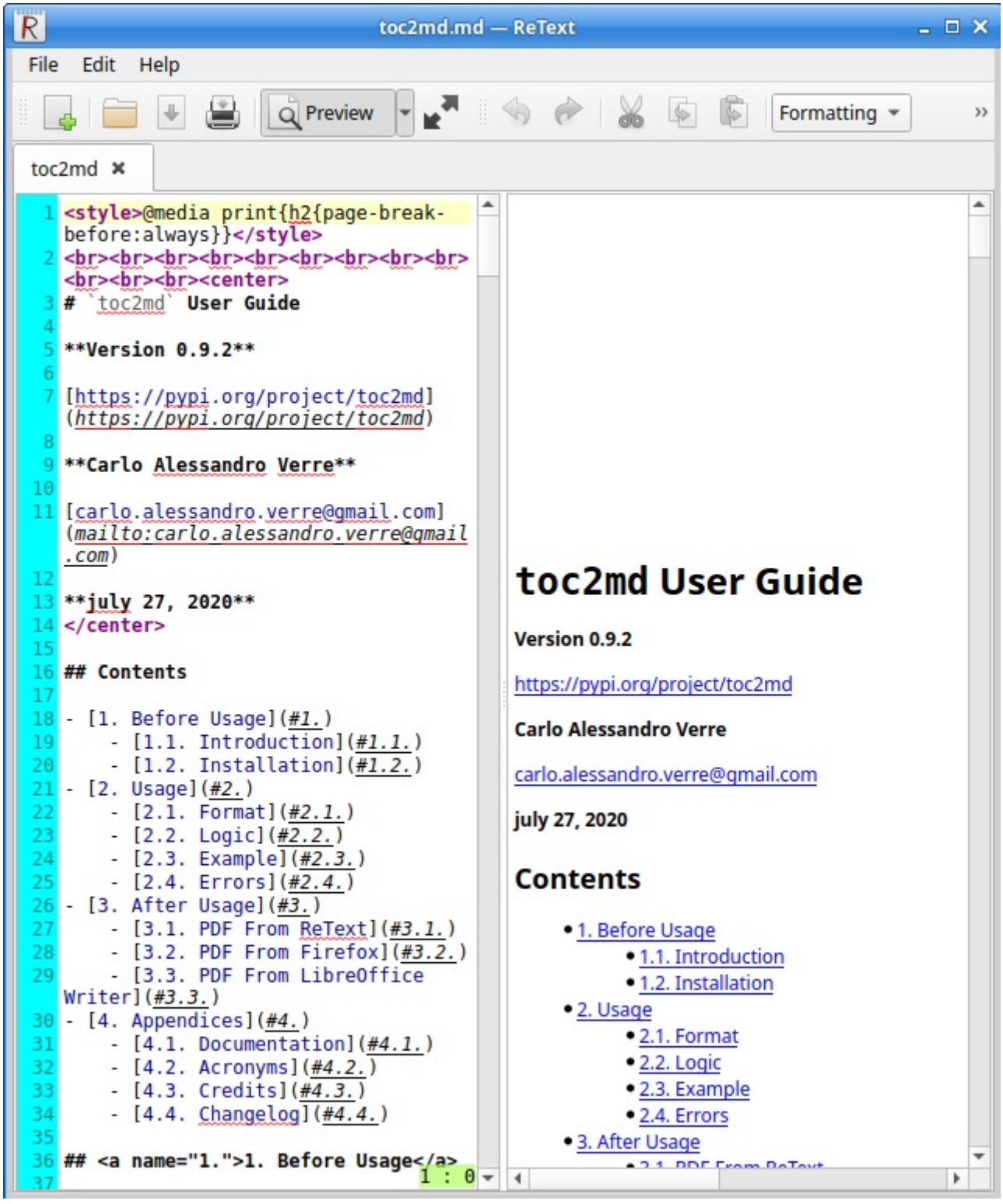
2.4. Errors

toc2md can fail with one of the following error messages:

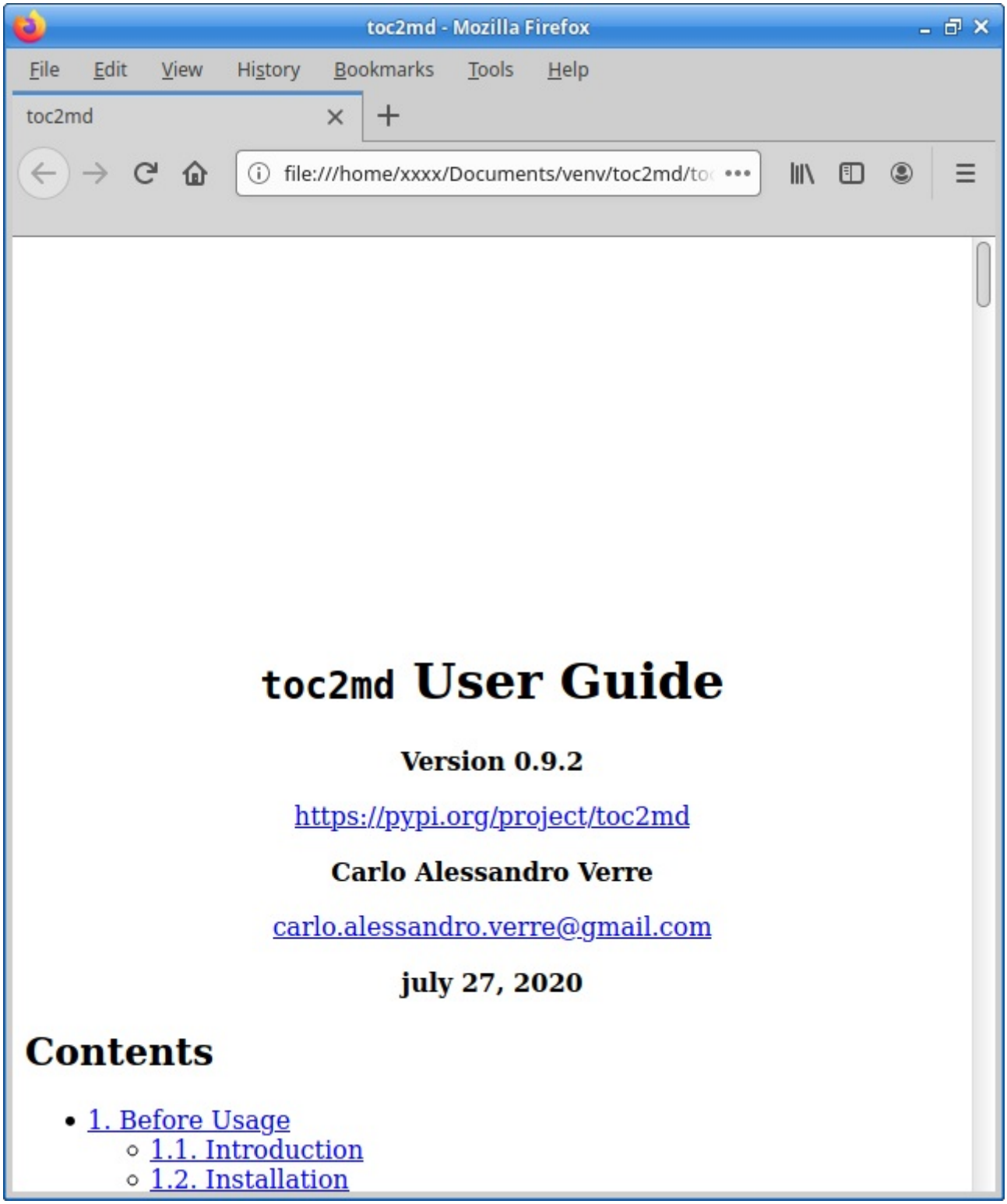
```
ERROR: no file
ERROR: file '...' has no '.md' extension
ERROR: file '...' not found
ERROR: heading sequence error in line ... '#...'
```

3. After Usage

As an example, we will use this same User Manual that you are reading, born as `toc2md.md`, edited and processed concurrently by Retext and toc2md.



If we export `toc2md.md` from Retext by **File → Export → HTML** into, say, `toc2md.retext.html` and we open it by Firefox (or any other HTML browser) links in our TOC work fine as we expected, we click a line in TOC and we jump on the desired chapter.



Now we will explore three different ways to obtain a PDF document from our MD file:

- exporting directly from ReText
- exporting from Firefox via HTML format
- exporting from LibreOffice Writer via HTML format

As you can expect, from a more complex procedure we will get a higher degree of control and a better final quality.

3.1. PDF From ReText

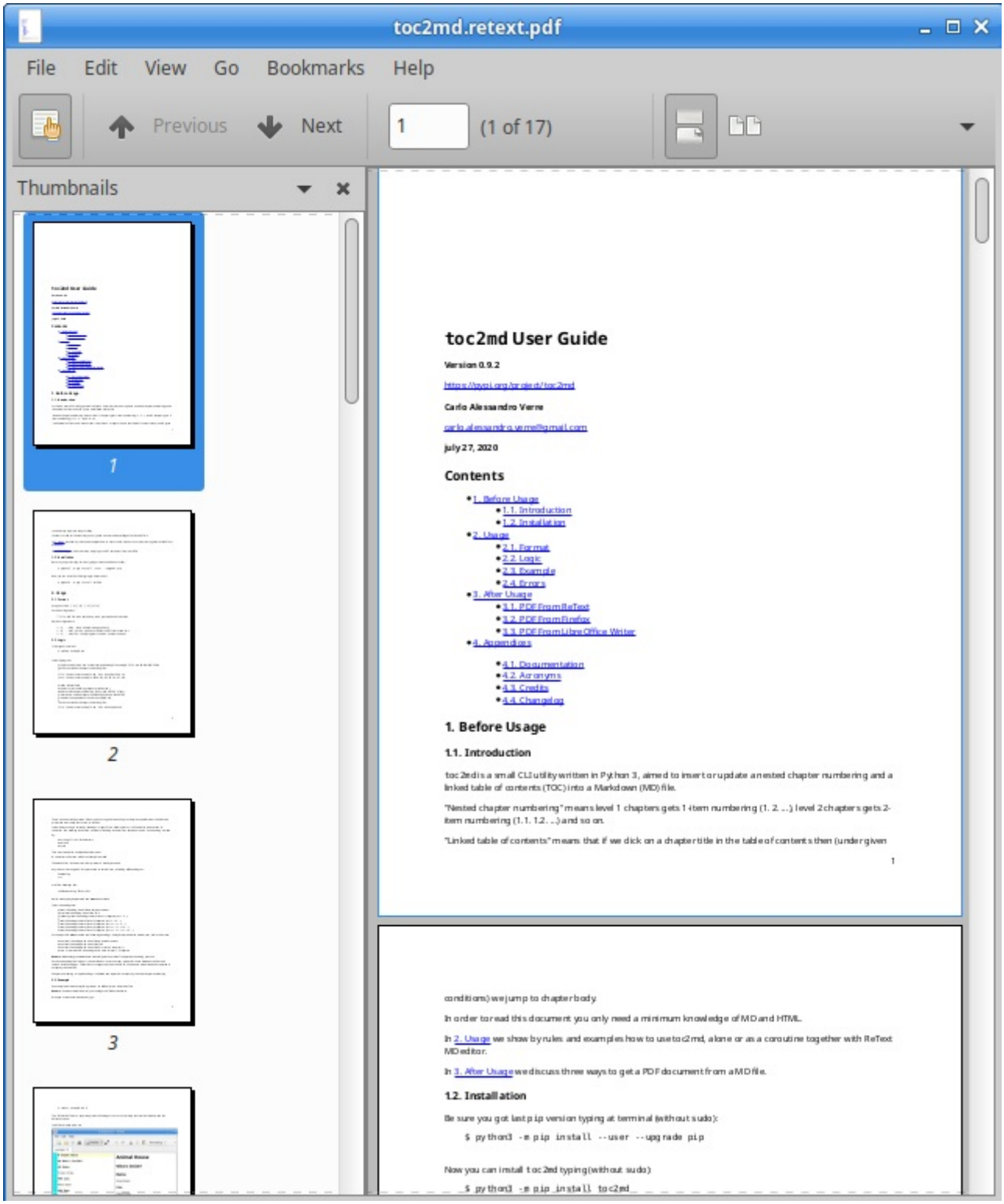
We can directly export `toc2md.md` from ReText by **File → Export → PDF** into a file, say `toc2md.retext.pdf`.

We get a not-indexed PDF file with page numbers in page bottom right corners and nothing else.

Procedure is instantaneous, control on final result is absent, the quality is only sufficient for a quick-and-dirty draft:

- there is no way to insert page breaks
- there is no way to manually format the final result
- resulting PDF file is not indexed
- links in table of contents do not work

By the way, if we print directly from ReText by **File → Print** we obtain on paper the same result.



3.2. PDF From Firefox

In order to force page breaks in given points of the final PDF document we can insert a line:

```
<p style="break-after:page"></p>
```

in all corresponding places of our MD file. Alternatively, we can force a page break for each heading of a given level by a single line, for example if we insert:

```
<style>@media print{h2{page-break-before:always}}</style>
```

as first line of our MD file, then we get a page break before each level 2 heading (level 1 chapter).

Beware: These tricks work with Firefox only, while they do not work exporting our PDF file from ReText or from LibreOffice Writer.

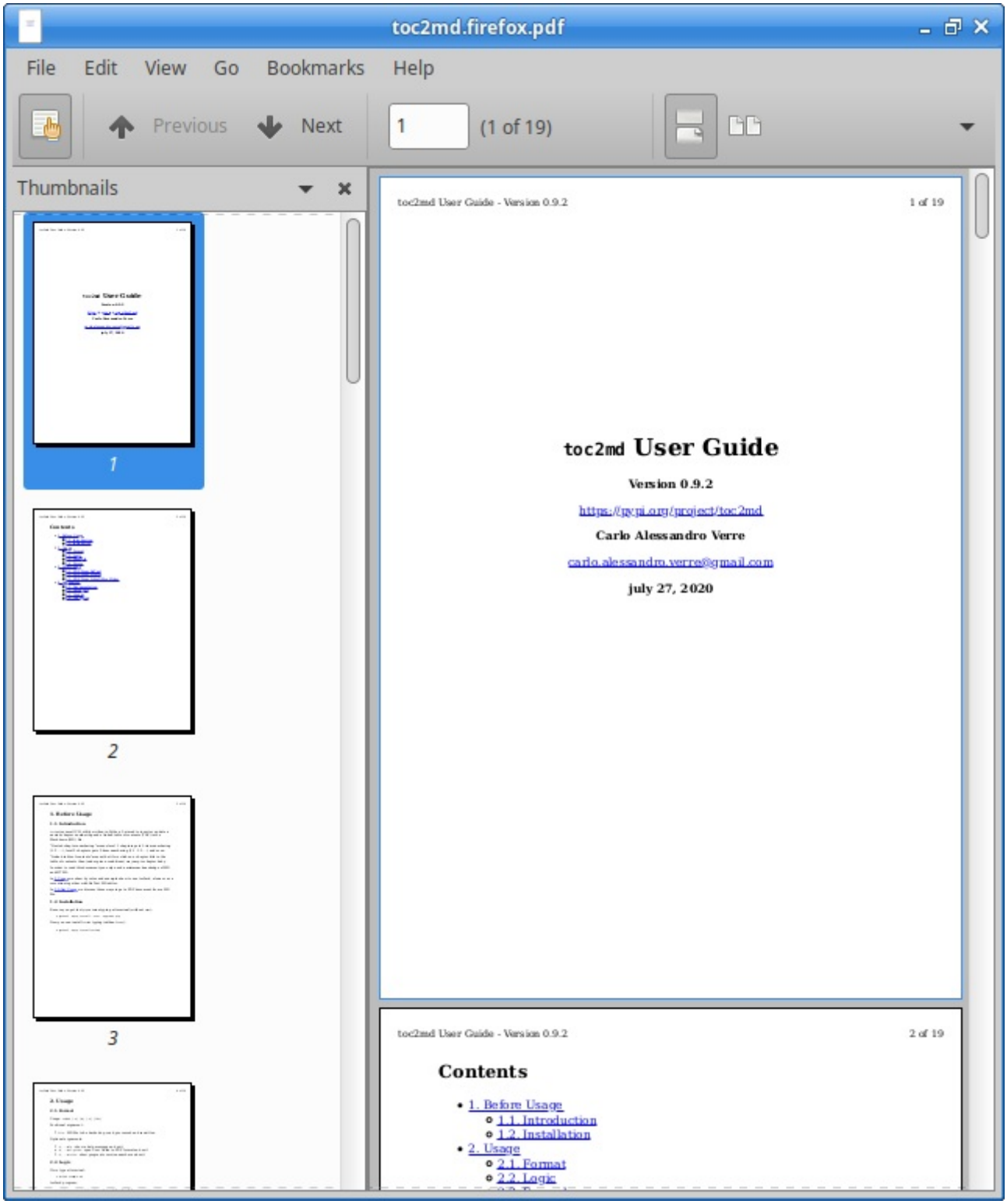
We proceed as follows:

1. we export `toc2md.md` from Retext by **File → Export → HTML** into `toc2md.retext.html`
2. we open `toc2md.retext.html` file by Firefox
3. optionally we control page headers and footers via **[≡] button → Print...→ Print... → Options → Header and Footer**
4. optionally we control paper size and other print parameters via **[≡] button → Print... → Print... → Page Setup**
5. finally we export into `toc2md.firefox.pdf` by **[≡] button → Print... → Print... → General → Print to File → Print**

Procedure is fast, control of result and final quality are good for printing but not for web publishing:

- as before, resulting PDF file is not indexed
- as before, links in table of contents do not work

By the way, if we print directly from Firefox by **File → Print... → Print... → Print** we obtain on paper the same result.



3.3. PDF From LibreOffice Writer

We proceed as follows:

1. we export `toc2md.md` from Retext by **File → Export → HTML** into `toc2md.retext.html`
2. we replace in `toc2md.retext.html` each string **`'<table>'`** with string **`'<table border=1>'`** and we write result in `toc2md.sed.html`, in order to maintain visibility of table borders (see for instance the table in 4.2. Acronyms)
3. open `toc2md.sed.html` by LibreOffice Writer
4. make changes at our taste (for instance change text font or alignment, insert page breaks by **Ctrl-**

Enter, insert headers and/or footers with page number, resize or move pictures, or whatever else...)
5. finally export from Writer into `toc2md.writer.pdf` by **File → Export As → Export Directly As PDF**

Step 2. can be performed typing at terminal:

```
$ sed 's/<table>/<table border=1>/' <toc2md.retext.html >toc2md.sed.html
```

Procedure is a bit cumbersome, control of final result is maximum, result is an indexed, professional-looking PDF file, good for printing and web publishing.

If we open `toc2md.writer.pdf` by a PDF browser:

- the `toc2md`-generated links in TOC jump perfectly to chapters
- the file is said "indexed" because in the side pane on the left we can view the "Index", which is a Writer-generated TOC which matches one-to-one the `toc2md`-generated TOC in the text and remains visible when we scroll down along the file

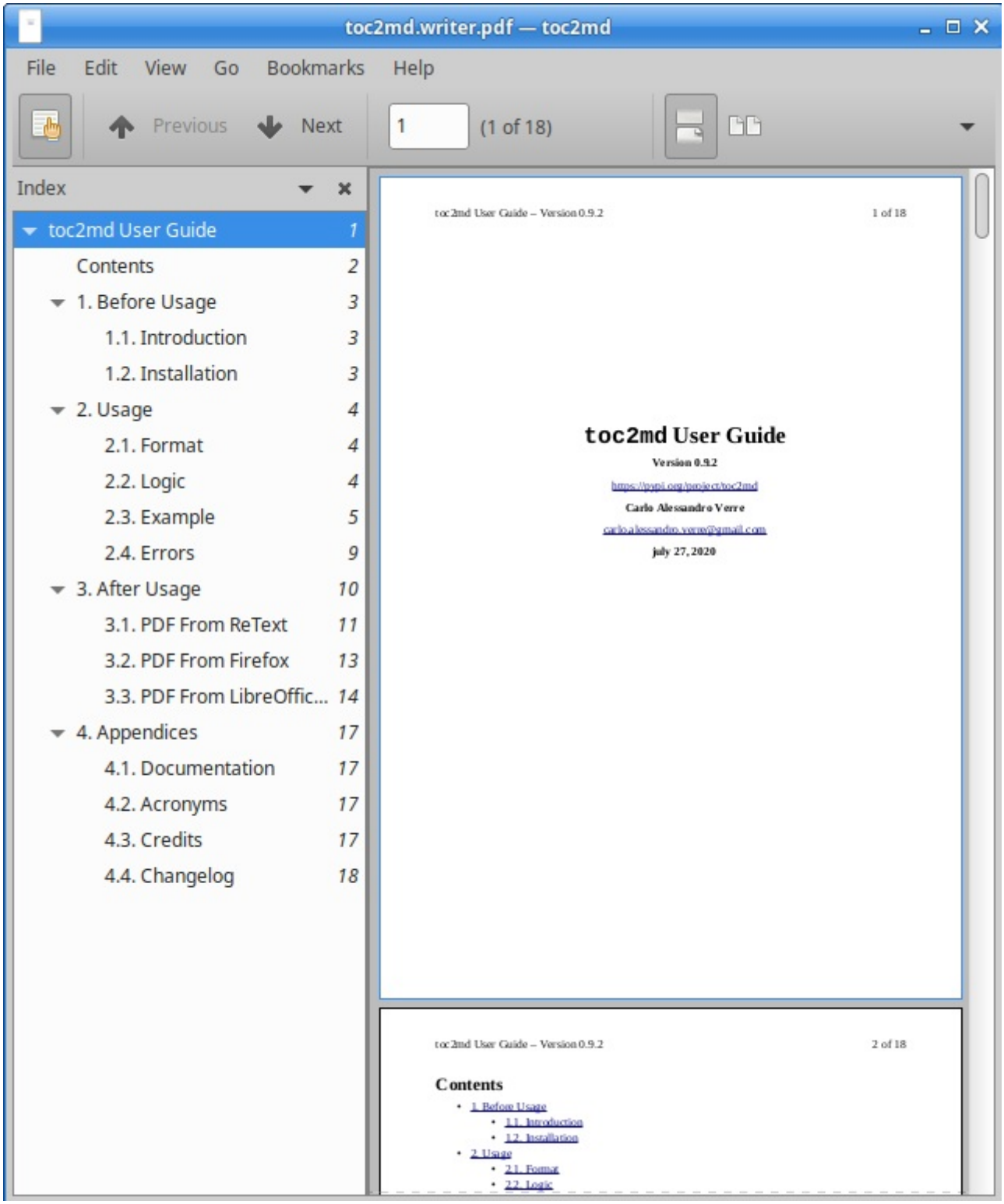
Beware: Writer can not distinguish '#'-headings from the others, so the two TOCs (the Writer-generated at left and the `toc2md`-generated in the text) they will only have a one-to-one match if we have avoided entering MD headings as:

```
Something
===
```

or HTML headings as:

```
<h3>Something Else</h3>
```

in our MD file.



4. Appendices

4.1. Documentation

If we type at terminal:

```
$ cd ~/.local/lib/python3.*/site-packages/toc2md/docs
```

we can find all versions of the `toc2md` User Guide discussed in 3. After Usage:

- `toc2md.md` - original file edited by ReText and toc2md
- `toc2md.retext.html` - exported by ReText, in input to FireFox
- `toc2md.sed.html` - `toc2md.retext.html` modified by sed, in input to Writer
- `toc2md.retext.pdf` - directly exported by ReText (low quality)
- `toc2md.firefox.pdf` - exported by FireFox from `toc2md.retext.html` (medium quality)
- `toc2md.writer.pdf` - exported by Writer from `toc2md.sed.html` (high quality)

Anyway the "official" version of the `toc2md` User Guide is `toc2md.writer.pdf`. We can view it typing at terminal:

```
$ toc2md -H
```

4.2. Acronyms

INITIALS	MEANING
CLI	Command Language Interface
HTML	HyperText Markup Language
MD	MarkDown
ODT	Open Document Text
PDF	Page Description Format
PyPI	Python Package Index
TOC	Table Of Contents
toc2md	Table Of Contents to MarkDown

4.3. Credits

The `toc2md` project has been:

- developed in Python 3.8.2
- by Idle 3.8.2
- built and published on PyPI by flit 2.3.0
- on Linux Xubuntu 20.04 LTS

The original User Guide in `toc2md.md` file has been:

- written by ReText 7.1.0
- processed by toc2md 0.9.2

During development and test we used:

- ReText 7.1.0 to write MD files and to export them into PDF and HTML format
- sed 4.7 to alter HTML files
- Firefox 76.0.1 and LibreOffice Writer 6.4.3.2 to translate HTML files into PDF format
- Atril 1.24.0 to browse PDF files

4.4. Changelog

- Version **0.9.2** - 2020-07-27
    - Added
        - This Changelog chapter to this User Guide
    - Changed
        - Rewritten (faster) the `count_hash()` function in `__main__.py` file
        - Corrected (slightly) `README.md` file and this User Guide
- Version **0.9.1** - 2020-07-06

- First version published on Pypi