



FACULTY  
OF INFORMATICS  
Masaryk University

# Using inside T<sub>E</sub>X Documents

---

TUG@BachoT<sub>E</sub>X 2017

<https://github.com/witiko/markdown>

Vít Novotný

# Contents

1. Introduction
  - 1.1 The Case for Lightweight Markup
  - 1.2 Existing Solutions
2. The `markdown.tex` Package
  - 2.1 Building a Parser
  - 2.2 Quick Overview of the Package
3. Conclusion
4. Q&A
5. Bibliography

# Section 1

## **Introduction**

# The Case for Lightweight Markup

## *T<sub>E</sub>X as a Content Creation Language*

### 1. High Markup to Text Ratio

- *The T<sub>E</sub>Xbook* (Knuth, 1986) is 22 % markup (plain T<sub>E</sub>X).
- *Think Java* (Downey et al., 2016) is 21 % markup (L<sup>A</sup>T<sub>E</sub>X).

### 2. Zero Sandboxing Support

- The document you are typesetting may not compile.

```
\texttt{innocent_looking_underscores.tex}
```

- The document you are typesetting may halt.

```
\def\whiletrue{\whiletrue} \whiletrue
```

- The document you are typesetting may access the system shell.

```
\immediate\write18{sudo rm -rf /}
```

### 3. Steep Learning Curve

# The Case for Lightweight Markup

## *Comparison of $\LaTeX$ and Markdown*

```
\section{This is a level one heading}
```

This is a text paragraph with `\emph{emphasis}`.

```
\begin{quotation}This paragraph will show as a quote.\end{quotation}
```

```
\begin{verbatim}
```

This is is a source code example.

```
\end{verbatim}
```

```
\begin{itemize}
```

```
  \item First item with \alert{strong emphasis}
```

```
  \item Second item with a link%
```

```
    \footnote{See \url{http://link.com} (Title)}
```

```
\end{itemize}
```

```
\begin{enumerate}
```

```
  \item First item with \verb`inline code`.
```

```
  \item Second item with an \includegraphics{image.png}
```

```
\end{enumerate}
```

# The Case for Lightweight Markup

## Comparison of $\text{\LaTeX}$ and Markdown

### # This is a level one heading

This is a text paragraph with *\_emphasis\_*.

> This paragraph will show as a quote.

```
~~~~~This is is a source code example.
```

\* First item with **\*\*strong emphasis\*\***

\* Second item with a [link](http://link.com/ "Title")

1. First item with ``inline code``.

2. Second item with an ![image](image.png "Title")

# The Case for Lightweight Markup

## *Markdown as a Content Creation Language*

### 1. Minimal Markup to Text Ratio

- Recall: Knuth (1986) and Downey et al. (2016) are **~22% markup**.
- *Efficient R programming* (Gillespie et al., 2016) is 5.5% markup.
- *R for Data Science* (Grolemund et al., 2016) is 3.8% markup.

### 2. *Either* Sandboxing Support ...

- A Markdown document converted to  $\text{T}_{\text{E}}\text{X}$  will always compile.
- The document may neither halt nor access the shell.

### 3. ... or Hybrid Markup Support

- Structurally simple sections can use pure Markdown, complex sections may combine Markdown and the host markup.

### 4. Mild Learning Curve

## Existing Solutions

### *The Swiss Army Knife of Pandoc*

*If you need to **convert files from one markup format into another**, Pandoc is your swiss-army knife.*

— MacFarlane (2016b), emphasis mine

- A multi-target publishing software.
- Supports tens of markup languages (Markdown,  $\text{\LaTeX}$ , HTML, XML Docbook) and output formats (ODF, OOXML, PDF).
- The use of Pandoc for the preparation of  $\text{\LaTeX}$  documents has been described in TUGBoat by Dominici (2014).



## Existing Solutions

### Why Is Pandoc Not Ideal?

#### 3. Half-hybrid, Half-sandboxed

- The input is heuristically parsed and sanitized:

This `{will} 2^n \begin{get} s~nitized and \this{will}`  
not `\begin{equation}2^n\end{equation} $2^n$`.

↓↓

This `\{will\} 2^{n} \textbackslashbegin\{get\}`  
`s\textasciitilde\}nitized and \this{will}` not  
`\begin{equation}2^n\end{equation} \((2^n\)`.

- Malicious input such as

```
\def\shell{18} \immediate\write\shell{sudo rm -rf /}
```

is left alone by Pandoc.

## Section 2

# The `markdown . tex` Package

## Building a Parser

*Is T<sub>E</sub>X Up to the Task?*

There exist formal language parsers written solely in T<sub>E</sub>X. These parsers recognize regular (L<sup>A</sup>T<sub>E</sub>X3 Project, 2016) and context-free LL(1) languages (Carlisle, 2000). Markdown is not context-free:

```
``There is a literal backtick (`) here.``
```

and a parser needs to be able to backtrack over the entire input:

```
[Leci n'est pas un link](http://link.com/ "Link"
```

Implementing such a parser in T<sub>E</sub>X is possible, but generally a bad idea due to the lack of efficient data structures.

## Building a Parser

*Can We Use Lua Instead of T<sub>E</sub>X?*

*Lua is a powerful, efficient, lightweight, embeddable scripting language. It supports procedural programming, object-oriented programming, functional programming, data-driven programming, and data description.*

— Lua Team (2016)

*LuaT<sub>E</sub>X is an extended version of pdfT<sub>E</sub>X using Lua as an embedded scripting language.*

— LuaT<sub>E</sub>X Team (2016)

## Building a Parser

*Can We Use Lua Instead of T<sub>E</sub>X?*

- With LuaT<sub>E</sub>X, we can directly execute Lua code:

```
1 + 2 = \directlua{ tex.sprint(1 + 2) }
```

- With pdfT<sub>E</sub>X and other modern T<sub>E</sub>X engines, we can spawn a shell and execute the Lua code in a separate process:

```
1 + 2 = \newwrite\script  
\immediate\openout\script=script.lua  
\immediate\write\script{ print(1 + 2) }%  
\immediate\closeout\script  
\immediate\write18{texlua script.lua > output.tex}%  
\input output.tex
```

# Building a Parser

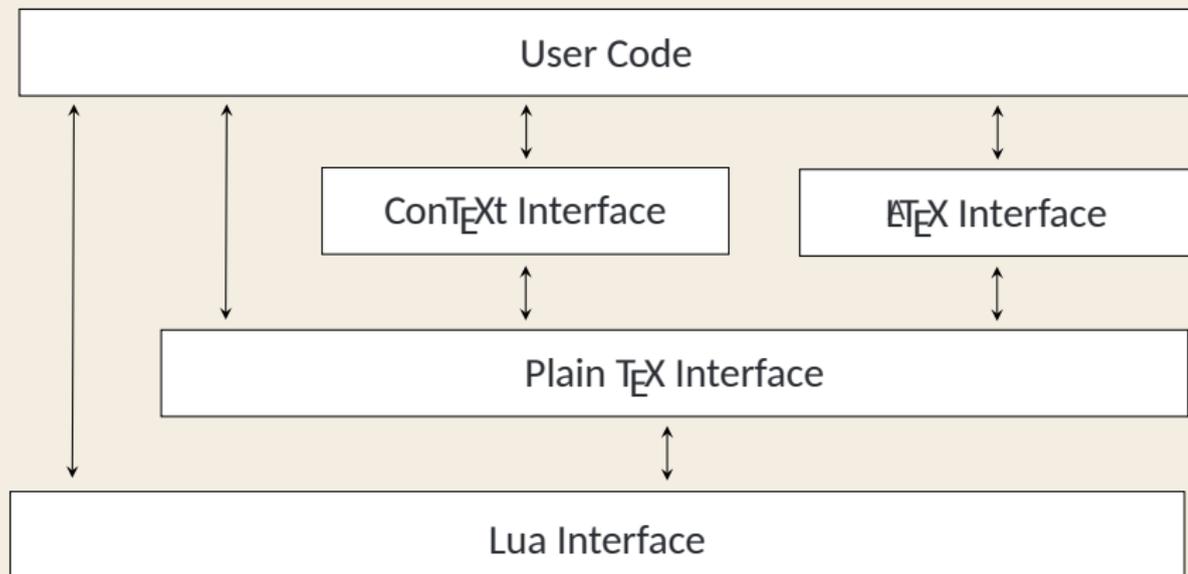
## *The Lunamark Library*

- Lunamark (MacFarlane, [2016a](#)) is a Markdown parser in Lua.
- The language is specified using a Parsing Expression Grammar (PEG) via the LPeg C library (and a bit of cheating).
- The dependencies of Lunamark were all either compiled into Lua $\TeX$  (LPeg, Slnunicode), or unnecessary (Cosmo, Alt-getopt).
- The library has been released under the Expat (MIT) License.



# Quick Overview of the Package

## *A Block Diagram of the Package*



# Quick Overview of the Package

## *The Lua Interface*

```
#!/usr/bin/env texlua
local kpse = require"kpse"
kpse.set_program_name"kpsewhich"
local markdown = require"markdown"
local convert = markdown.new({ something = true })
local input = "Some _Markdown_ text."
print(convert(input))
```

# Quick Overview of the Package

## *The Plain T<sub>E</sub>X Interface*

```
\input markdown %  
\def\markdownOptionSomething{true}%  
\def\markdownRendererSomethingElse#1#2#3{\foo{#1}}%  
\markdownInput{bar.md}  
\markdownBegin  
Some _Markdown_ text.  
\markdownEnd  
\bye
```

# Quick Overview of the Package

## *The ConT<sub>E</sub>Xt Interface*

```
\usemodule[t][markdown]
\def\markdownOptionSomething{true}
\def\markdownRendererSomethingElse#1#2#3{\foo{#1}}
\starttext
\markdownInput{bar.md}
\startmarkdown
Some Markdown text.
\stopmarkdown
\stoptext
```

# Quick Overview of the Package

## *The L<sup>A</sup>T<sub>E</sub>X2e Interface*

```
\documentclass{article}
\usepackage[something]{markdown}
\markdownSetup{renderers = {somethingElse = {\foo{#1}}}}
\begin{document}
\markdownInput{bar.md}
\begin{markdown}
Some Markdown text.
\end{markdown}
\end{document}
```

# Quick Overview of the Package

## *The Sandbox and Hybrid Modes*

```
\documentclass{article}
\usepackage{markdown}
\begin{document}
\begin{markdown}
  Foo bar \TeX{}  $2^n$ .
\end{markdown}
\begin{markdown*}{hybrid}
  Foo bar \TeX{}  $2^n$ .
\end{markdown*}
\end{document}
```

Foo bar \TeX{}  $2^n$ . Foo bar  $\TeX 2^n$ .

## Quick Overview of the Package

*Mapping Markdown Tokens to T<sub>E</sub>X Macros*

```
\documentclass{article}
\usepackage{markdown}
\markdownSetup{renderers = {
  link = {#1\footnote{See \url{#3} (#4)}},
}}
\begin{document}
\begin{markdown}
  Foo [bar](http://link.com "Link").
\end{markdown}
\end{document}
```

Foo bar<sup>1</sup>.

---

<sup>1</sup>See <http://link.com> (Link)

# Quick Overview of the Package

## *Syntax Extensions*

- Some syntax extensions were already supported by Lunamark:
  - HTML,
  - footnotes,
  - definition lists,
- New syntax extensions were added as a part of the project:
  - citations,
  - fenced code blocks,
  - IA Writer content blocks.

## Quick Overview of the Package

*Syntax Extensions – \markdownSetup{html}*

HTML `<b>tags</b>` such as `&lt;b&gt;` are recognized  
*<!-- and comments are just ignored-->*.

HTML tags such as `<b>` are recognized.

## Quick Overview of the Package

*Syntax Extensions* – `\markdownSetup{footnotes}`

Here is a footnote reference, `[^1]` and another. `[^long]`

`[^1]`: Here is the footnote.

`[^long]`: Here's one with multiple blocks.

Subsequent paragraphs are indented to show that they belong to the footnote.

Here is a footnote reference,<sup>2</sup> and another.<sup>3</sup>

---

<sup>2</sup>Here is the footnote.

<sup>3</sup>Here's one with multiple paragraphs.

Subsequent paragraphs are indented to show that they belong to the footnote.

## Quick Overview of the Package

*Syntax Extensions* – `\markdownSetup{definitionLists}`

Term 1

: Definition

Term 2

: Definition with

multiple paragraphs

**Term 1** Definition 1

**Term 2** Definition  
with multiple paragraphs

## Quick Overview of the Package

*Syntax Extensions* – `\markdownSetup{citations}`

Here is a parenthetical citation [`@knuth86`] and a string of several [`see @knuth86, pp. 33-35; also @gruber04, chap. 1`].

Here is a text citation `@knuth86` and a string of several `@knuth86 [pp. 33-35; @gruber04, chap. 1]`.

Here is a parenthetical citation (Knuth, [1986](#)) and a string of several (see Knuth, [1986](#), pp. 33-35; also Gruber, [2004](#), chap. 1).

Here is a text citation Knuth ([1986](#)) and a string of several Knuth ([1986](#), pp. 33-35) and Gruber ([2004](#), chap. 1).

## Quick Overview of the Package

*Syntax Extensions – \markdownSetup{fencedCode}*

```
~~~ js
if (a > b)
  return c + 4;
else
  return d + 5;
~~~~~
```

```
if (a > b)
  return c + 4;
else
  return d + 5;
```

# Quick Overview of the Package

Syntax Extensions - `\markdownSetup{contentBlocks} I`

/Flowchart.png "Engineering Flowchart"

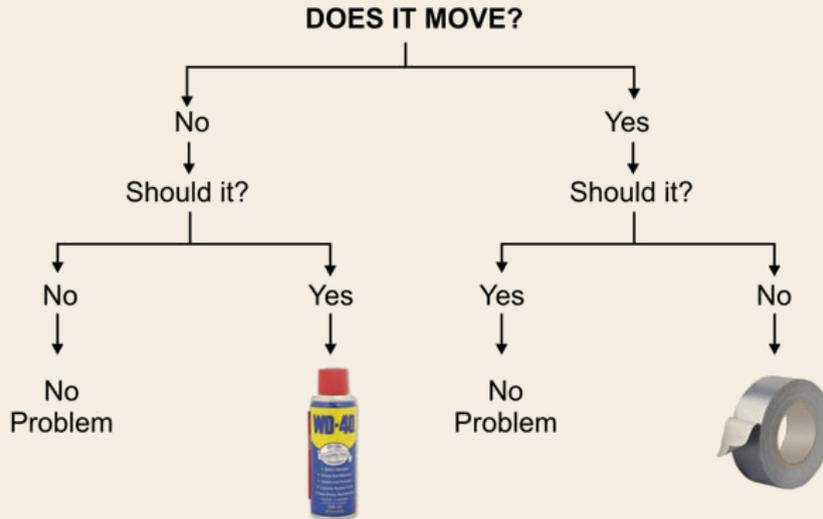


Figure: Engineering Flowchart

## Quick Overview of the Package

*Syntax Extensions* – `\markdownSetup{contentBlocks} II`

`/Scientists.csv` (Great Minds of the 19th Century)

name	surname	age
Albert	Einstein	133
Marie	Curie	145
Thomas	Edison	165

Table: Great Minds of the 19th Century

# Quick Overview of the Package

*Syntax Extensions – \markdownSetup{contentBlocks} III*

/chapters/01.txt

/chapters/02.txt

## Chapter 1

This is the first chapter.

## Chapter 2

And this is the second chapter.

## Quick Overview of the Package

*Syntax Extensions – \markdownSetup{contentBlocks} IV*

<https://tug.org/tugboat/noword.jpg>

(The Communications of the  $\TeX$  Users Group)

The logo for TUGBOAT, consisting of the text 'TUGBOAT' in a large, bold, black serif font. The letters are closely spaced and have a classic, slightly condensed appearance.

Figure: The Communications of the  $\TeX$  Users Group

(This actually does not work out-of-box at the moment.)

Section 3

**Conclusion**

## Conclusion

### *The Missing Pieces of the Puzzle*

The `markdown.tex` package

- enables the use of Markdown in environments where tools from outside T<sub>E</sub>X distributions are unavailable,
- gives the authors full control over how individual Markdown elements are rendered and how much access to T<sub>E</sub>X markup the Markdown documents have,
- exposes Lua, plain T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, and ConT<sub>E</sub>Xt interfaces.
- was released under the L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) 1.3 on the Comprehensive T<sub>E</sub>X Archive Network (CTAN) and on GitHub (<https://github.com/witiko/markdown>).

## Conclusion

### *The Missing Pieces of the Puzzle*

- The syntax extensions were backported to Lunamark and merged by MacFarlane, resulting in a new minor version release of the library (0.5.0). (Novotný, 2016a)
- The package was featured on the twitter profile and the blog of Overleaf – a major online service for preparing  $\text{\LaTeX}$  documents – along with original example documents. (Lim, 2017)
- $\text{\TeX}$  comments in hybrid code behave in an unexpected way.
- Logging should be improved, so that programs such as `latexmk` can automatically remove cache files that are no longer needed.

## Section 4

### Q&A

## Section 5

# **Bibliography**

## Bibliography I

TEX3 PROJECT, 2016. *The l3regex package: regular expressions in T<sub>E</sub>X*  
[online] [visited on 2016-11-08]. Available from:

[http://mirrors.ctan.org/macros/latex/contrib/  
l3experimental/l3regex.pdf](http://mirrors.ctan.org/macros/latex/contrib/l3experimental/l3regex.pdf).

CARLISLE, David, 2000. XMLT<sub>E</sub>X: A non-validating (and not 100%  
conforming) namespace-aware XML parser implemented in T<sub>E</sub>X.  
*TUGboat* [online]. Vol. 21, no. 3, pp. 193–199 [visited on  
2016-11-08]. ISSN 0896-3207. Available from:

<https://www.tug.org/TUGboat/tb21-3/tb68carl.pdf>.

## Bibliography II

DOMINICI, Massimiliano, 2014. An overview of Pandoc. *TUGboat* [online]. Vol. 35, no. 1, pp. 44–50 [visited on 2016-08-15]. ISSN 0896-3207. Available from:  
<http://tug.org/TUGboat/tb35-1/tb109dominici.pdf>.

DOWNEY, Allen B.; MAYFIELD, Chris, 2016. *Think Java: How to Think Like a Computer Scientist* [online]. Green Tea Press. Version 6.1.0 [visited on 2016-11-08]. Available from:  
<http://thinkjava.org/>.

FENN, Jürgen, 2016. Neue Pakete auf CTAN. *Die T<sub>E</sub>Xnische Komödie*. No. 3/2016. ISSN 1434-5897.

## Bibliography III

FORD, Bryan, 2002. Packrat Parsing: Simple, powerful, lazy, linear time, functional pearl. In: *Packrat Parsing: Simple, powerful, lazy, linear time, functional pearl*. ACM SIGPLAN Notices [online]. Vol. 37, pp. 36–47 [visited on 2016-11-08]. No. 9. Available from DOI: [10.1145/581478.581483](https://doi.org/10.1145/581478.581483).

FORD, Bryan, 2004. Parsing expression grammars: A recognition-based syntactic foundation. In: *Parsing expression grammars: A recognition-based syntactic foundation*. ACM SIGPLAN Notices [online]. Vol. 39, pp. 111–122 [visited on 2016-08-16]. No. 1. Available from DOI: [10.1145/964001.964011](https://doi.org/10.1145/964001.964011).

## Bibliography IV

GILLESPIE, Colin; LOVELACE, Robin, 2016. *Efficient R programming* [online]. O'Reilly Media [visited on 2016-11-08]. ISBN 978-1-4919-5078-4. Available from:  
<https://github.com/hadley/r4ds/>.

GROLEMUND, Garrett; WICKHAM, Hadley, 2016. *R for Data Science* [online]. O'Reilly Media [visited on 2016-11-08]. ISBN 978-1-4919-1039-9. Available from:  
<https://github.com/hadley/r4ds/>.

GRUBER, John, 2004. *Markdown* [online] [visited on 2016-08-15]. Available from:  
<https://daringfireball.net/projects/markdown/>.

## Bibliography V

KNUTH, Donald Ervin, 1986. *The T<sub>E</sub>Xbook* [online]. 3rd ed.

Addison-Westley [visited on 2016-11-08]. ISBN 0-201-13447-0.

Available from: <https://mirrors.ctan.org/systems/knuth/dist/tex/texbook.tex>.

LIM, Lian Tze, 2017. *Markdown into L<sub>A</sub>T<sub>E</sub>X with Style* [online] [visited on 2017-04-28]. Available from:

<https://www.overleaf.com/blog/501-markdown-into-latex-with-style>.

LUA TEAM, 2016. *Lua: About* [online] [visited on 2016-08-15].

Available from: <https://www.lua.org/about.html>.

LUAT<sub>E</sub>X TEAM, 2016. *LuaT<sub>E</sub>X: Welcome* [online] [visited on 2016-08-15]. Available from: <http://luatex.org/>.

## Bibliography VI

MACFARLANE, John, 2016a. *Lunamark: Lua library for conversion between markup formats* [online] [visited on 2016-08-15]. Available from: <https://github.com/jgm/lunamark>.

MACFARLANE, John, 2016b. *Pandoc: a universal document converter* [online] [visited on 2016-08-15]. Available from: <http://pandoc.org/>.

NOVOTNÝ, Vít, 2017. *A Markdown Interpreter for T<sub>E</sub>X* [online] [visited on 2017-04-26]. Available from: <http://mirrors.ctan.org/macros/generic/markdown/markdown.pdf>.

NOVOTNÝ, Vít, 2016a. *Added support for Pandoc-style citations* [online] [visited on 2016-08-15]. Available from: <https://github.com/jgm/lunamark/pull/20>.

## Bibliography VII

NOVOTNÝ, Vít, 2016b. *Markdown: A package for converting and rendering markdown documents inside T<sub>E</sub>X* [online] [visited on 2016-08-15] Available from: <http://ctan.org/pkg/markdown>, <https://github.com/Witiko/markdown>, and <https://gitlab.fi.muni.cz/xnovot32/markdown>.