Markdown 2.10.0

LaTeX Themes & Snippets,
Two Flavors of Comments,
and LuaMetaTeX

Vít Novotný, Grandma Jane, and the Wolf

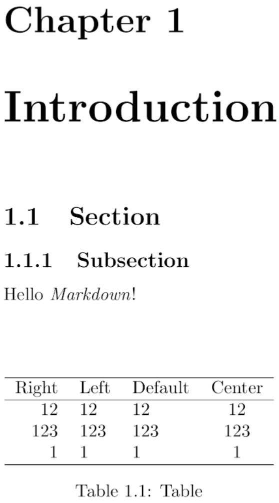
Faculty of Informatics, Masaryk University
Brno, Czech Republic

August 7, 2021

```
\documentclass{book}
\usepackage{markdown}
\markdownSetup{pipeTables,tableCaptions}
\begin{document}
\begin{markdown}
Introduction
============
## Section
### Subsection
Hello *Markdown*!

| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|   12  |  12  |    12   |    12  |
|  123  |  123 |   123   |   123  |
|    1  |    1 |     1   |     1  |

: Table
\end{markdown}
\end{document}
```



Chapter 1

Introduction

1.1 Section

1.1.1 Subsection

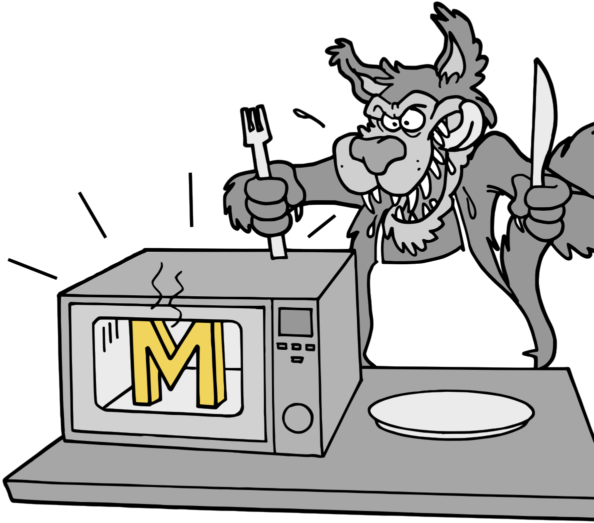
Hello *Markdown*!

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Table 1.1: Table

Contents

- LaTeX themes & setup snippets
  - Built-in themes
  - Creating your own theme
  - Setup snippets
- Two flavors of comments
  - Semantic HTML comments
  - Lexical TeX comments
- Support for LuaMetaTeX
- What's next and how do I contribute?
  - Actionable HTML attributes
  - Jekyll front matter
  - Online images using Luasocket
  - Integration with Pandoc
  - Direct mapping of elements



LaTeX themes & setup snippets

The goal of the Markdown package is to bring fire 🔥 to the users of TeX, so that they can playfully incinerate each and every element of their markdown documents. 😇

Unlike in the cathedral ⛪ of ConTeXt, where packages are few, an extraordinary bazaar 👳 of action, ferment, and innovation has sprung up in the wake of the LaTeX 2_ε kernel.

- Markdown's *themes* are LaTeX packages that specify what markdown elements *do*.
- Themes apply *globally*, but may define *setup snippets* that apply *locally*.
- We will discuss *built-in themes* and *creating your own* themes and setup snippets.

Built-in themes I

The `witiko/tilde` theme

```
\documentclass{article}
\usepackage
  [theme=witiko/tilde]
  {markdown}
\begin{document}
\begin{markdown}
Bartel~Leendert van~der~Waerden
\end{markdown}
\end{document}
```

Bartel·Leendert van·der·Waerden, where a middot (·) represents a non-breaking space.

Built-in themes II

The witiko/dot theme

~~~
\documentclass{article}
\usepackage[theme=witiko/dot]{markdown}
\begin{document}
\begin{markdown}
```dot A parse tree of “Let's eat grandma!”
digraph tree {
    graph [margin = 0]; node [shape = none]
    edge  [arrowhead = none]
    {rank=same; VP1[label = VP]}
    {rank=same; Let; NP1[label = NP];
                VP2[label = VP]}
    {rank=same; us; eat; NP2[label = NP]}
    {rank=same; S}; {rank=same; grandma}
    S -> VP1; VP1 -> Let; VP1 -> NP1
    VP1 -> VP2; NP1 -> us; VP2 -> eat
    VP2 -> NP2; NP2 -> grandma }
\end{markdown}
\end{document}
~~~

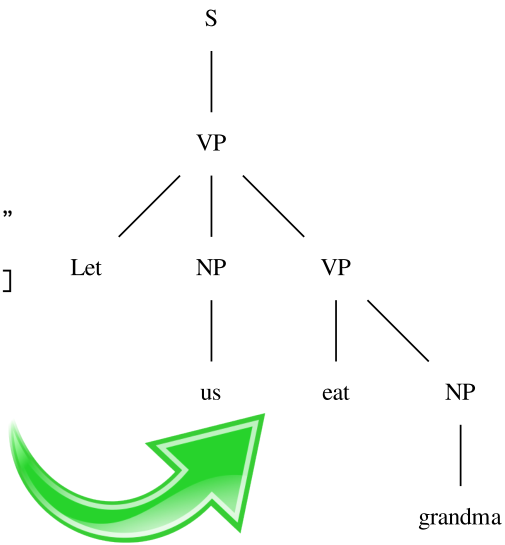
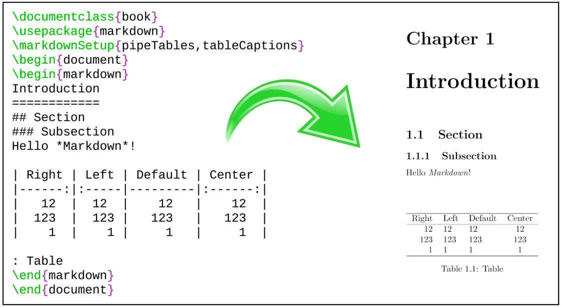


Figure: A parse tree of “Let's eat grandma!”

Built-in themes III

The witiko/graphicx/http theme

```
\documentclass{article}
\usepackage
  [texComments, contentBlocks,
   theme=witiko/graphicx/http]
  {markdown}
\begin{document}
\begin{markdown}
https://github.com/witiko%
       /markdown/raw/master%
       /banner.png
(Banner of the Markdown package)
\end{markdown}
\end{document}
```



```
\documentclass{book}
\usepackage{markdown}
\markdownSetup{pipeTables,tableCaptions}
\begin{document}
\begin{markdown}
Introduction
============
## Section
### Subsection
Hello *Markdown*!

| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|   12  |  12  |    12   |    12  |
|  123  |  123 |   123   |   123  |
|    1  |    1 |     1   |     1  |

: Table
\end{markdown}
\end{document}
```

Chapter 1

Introduction

1.1 Section

1.1.1 Subsection

Hello *Markdown*!

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Table 1.1: Table

Figure: Banner of the Markdown package

Creating your own theme I

1. Decide on a name in the following form:

   ⟨*theme author*⟩ / ⟨*target package*⟩ / ⟨*private naming*⟩

   For example: `jdoe/beamer/headings`
2. *Munge* the theme name by substituting slashes (/) with underscores (_).

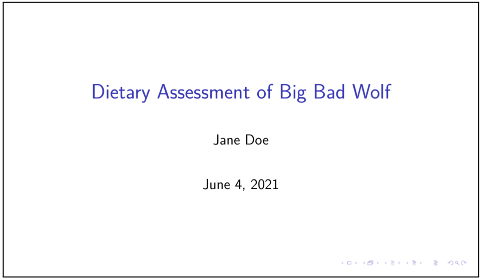
   For example: `jdoe_beamer_headings`
3. Create a text file named `markdowntheme`⟨*munged theme name*⟩`.sty`.

   For example `markdownthemejdoe_beamer_headings.sty` with the following:

```
\ProvidesPackage{markdownthemejdoe_beamer_headings}[2021/06/04]
\markdownSetup{
    rendererPrototypes = {
        headingOne = {\frametitle{#1}},
        headingTwo = {\framesubtitle{#1}}
    }
}
```

Creating your own theme II

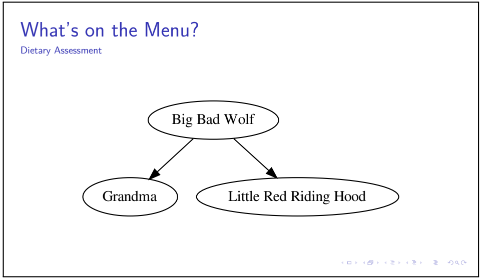
~~~~
\documentclass{beamer}
\usepackage[theme = witiko/dot,
  theme = jdoe/beamer/headings]{markdown}
\title{Dietary Assessment of Big Bad Wolf}
\author{Jane Doe}
\begin{document}
\maketitle
\begin{frame}[fragile]
\begin{markdown}
# What's on the Menu?
## Dietary Assessment
``` dot
digraph tree {
  Wolf -> Grandma; Wolf -> Hood
  Wolf [label = "Big Bad Wolf"]
  Hood [label = "Little Red Riding Hood"] }
\end{markdown}
\end{frame}
\end{document}
~~~~



Dietary Assessment of Big Bad Wolf

Jane Doe

June 4, 2021



What's on the Menu?

Dietary Assessment

Setup snippets I

```
\ProvidesPackage
  {markdownthemejdoe_lists_roman}
  [2021/06/04]
\markdownSetup{
  rendererPrototypes = {
    olItemWithNumber = {
      \item[\romannumeral#1\relax.]
    }}}
```

❌ Can be used only in preamble.

```
\documentclass{article}
\usepackage{markdown}
\begin{document}
\begin{markdown}
1. wahid
2. aithnayn
\end{markdown}
\begin{markdown*}% This won't work!
      {theme=jdoe/lists/roman}
3. tres
4. quattuor
\end{markdown*}
\end{document}
```

Setup snippets II

```
\ProvidesPackage
  {markdownthemejdoe_lists}
  [2021/06/04]
\markdownSetupSnippet{roman}{
  rendererPrototypes = {
    olItemWithNumber = {
      \item[\romannumeral#1\relax.]
    }}}
```

```
\documentclass{article}
\usepackage
  [theme=jdoe/lists]
  {markdown}
\begin{document}
\begin{markdown}
1. wahid
2. aithnayn
\end{markdown}
\begin{markdown*}
      {snippet=jdoe/lists/roman}
3. tres
4. quattuor
\end{markdown*}
\end{document}
```

1. wahid
2. aithnayn
iii. tres
iv. quattuor

Two flavors of comments

In TeX, comments fulfil several *distinct roles*:

1. Prevent the processing of code:

```
%\author{Authors anonymized}
\author{John Doe \and Jane Roe}
```

2. Mark up parallel documents:

```
% \cs{foo} prints ``bar'':
%  \begin{macrocode}
\def\foo{bar}
%    \end{macrocode}
```

3. Insert little side notes:

```
% Aren't we missing a comma here?
Let's eat grandma!
```

4. Consume newlines and leading spaces:

```
My parents have first met in Llan%
fairpwllgwyngyllgogerychwyrndrobw%
            llllantysiliogogogoch.
```

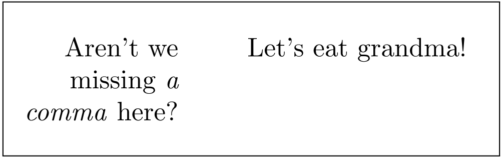
The Markdown package supports two flavors of comments:

Semantic May only appear in text. Markup is recognized. Enables roles 1, 2, and 3.

Lexical May appear anywhere. Markup is ignored. Enables roles 1, 2, and 4.

Semantic HTML comments

```
\documentclass{article}
\usepackage{marginnote}
\usepackage[html]{markdown}
\markdownSetup{
  renderers = {
    inlineHtmlComment = {
      \marginnote{#1}
    }
  }
}
\begin{document}
\begin{markdown}
<!-- Aren't we missing
     *a comma* here? -->
Let's eat grandma!
\end{markdown}
\end{document}
```



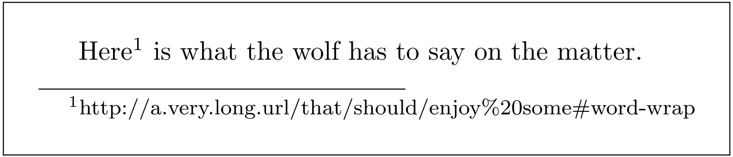
Aren't we missing *a comma* here?

Let's eat grandma!

Lexical TEX comments

```
\documentclass{article}
\usepackage[texComments]{markdown}
\begin{document}
\begin{markdown}
[Here][1] is what the wolf has
to say on the matter.

[1]: http://a.very.long.url/that/%
     should/enjoy\%20some#word-wrap
\end{markdown}
\end{document}
```



Here[1] is what the wolf has to say on the matter.

[1] http://a.very.long.url/that/should/enjoy%20some#word-wrap

Support for LuaMetaTeX

The LuaMetaTeX engine is a fast minimalist development ⚗️ version of LuaTeX, which is used in the ConTeXt LMTX format. Since the Markdown package supports ConTeXt, the time is ripe 🍇 to make Markdown play nice with LuaMetaTeX as well.

- LuaMetaTeX lacks Selene Unicode, but uses Lua 5.4 with its built-in `utf8` library.
- LuaMetaTeX lacks KPathSea, but provides an optional library interface.
- LuaMetaTeX provides Luasocket, but lacks high-level libraries such as `socket.http`.

What's next and how do I contribute?

The Markdown package would be well-served by encouraging users to lecture it 🎓 *on ever-new things and release their lecture notes* 📖 *to the whole wide world.* 🌍

There are many intriguing ideas for the future of Markdown:

- Some ideas are already *under development* by contributors.

  `https://github.com/witiko/markdown/pulls`

- Some ideas are only now beginning to be *discussed*:

  `https://github.com/witiko/markdown/discussions` & `/issues`

- Other ideas are yet to be discovered by *you*.

Actionable HTML attributes

```
I conclude in Section~\ref{sec:conclusion}.

Conclusion
==========
\label{sec:conclusion}
In this paper, we have discovered that most
grandmas would rather eat dinner with their
grandchildren than get eaten. Begone, wolf!
```

```
I conclude in Section <#sec:conclusion>.

Conclusion {#sec:conclusion .some-snippet}
==========
In this paper, we have discovered that most
grandmas would rather eat dinner with their
grandchildren than get eaten. Begone, wolf!
```

- Recommended reading: Parsing complex data formats in LuaTEX with LPEG (Menke, 2019)

Jekyll front matter

```
\title{Of \emph{Wolves} and \emph{Grandmas}}
\author{Little Red Riding Hood
   \and Big Bad Wolf}
\maketitle
```

```
---
title: Of *Wolves* and _Grandmas_
author:
- name: Little Red Riding Hood
- name: Big Bad Wolf
---
```

- Recommended reading: Parsing complex data formats in LuaTeX with LPEG (Menke, 2019)
- For more information, see `/issues/22` and `/pull/77`.

Online images using Luasocket

Just use sockets. 🔌
—Hans Hagen during a Q&A session at BachoTeX 2017

- The `witiko/graphicx/http` theme uses either GNU Wget or cURL.
- We could remove both prerequisites by using the Luasocket library.
- In `/issues/82`, I drafted an implementation:

  ```lua
  local http = require("socket.http")
  local result, status, headers = http.request(url)
  ```

  The draft has several issues:

  1. The `http.request()` method of `socket.http` mishandles redirects.
  2. LuaMetaTeX lacks `socket.http`.
  3. The `\directlua` command needs to be replaced with a shell escape for non-Lua TeX engines.
- Lua programmers familiar with Luasocket are encouraged to help tackle points 1 & 2.

Integration with Pandoc

- Pandoc is a Haskell library for converting between dozens of document formats.
- Pandoc uses an intermediate AST format, so that every document format only needs a conversion function from the format to the AST and back. $N > 2 \implies 2N < N^2$.
- If the Markdown package understood the AST, we could typeset any of the document formats of Pandoc while maintaining full control over the formatting:

```
\usepackage[theme=jdoe/lists]{markdown}
\begin{document}
\pandocInput[snippet=jdoe/lists/roman]
            {of-wolves-and-grandmas.docx}
\end{document}
```

- Recommended reading: Parsing complex data formats in LuaTeX with LPEG (Menke, 2019)
- For more information, see /issues/25, /issues/62, and drehak/lunamark.

Direct mapping of elements I

```
\markdownSetup {
  renderers = {
    emphasis = {\textbf{#1}}
  }
}
```

```
Hello *Markdown*!
↓
Hello \markdownRendererEmphasis{Markdown}!
↓
Hello \textbf{Markdown}!
↓
```

Hello **Markdown**!

Direct mapping of elements II

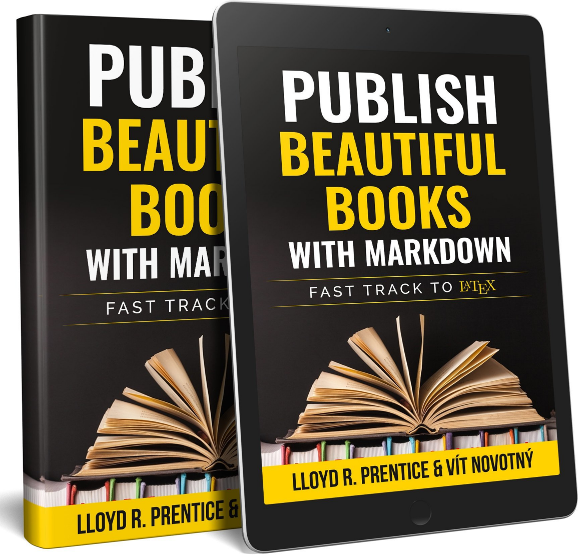
```
\markdownSetup {
  directMapping = {
    headingOne = {
      \begin{frame}
      \frametitle{#1}
    },
    horizontalRule = {
      \end{frame}
    },
  }
}
```

```
# What's on the Menu?
Grandma

***
```

↓

```
\begin{frame}
\frametitle{What's on the Menu?}
Grandma
\end{frame}
```


PUBLISH
BEAUTIFUL
BOOKS
WITH MARKDOWN
FAST TRACK TO LaTeX
LLOYD R. PRENTICE & VÍT NOVOTNÝ

MUNI
FACULTY
OF INFORMATICS