Delivering Code-Heavy Presentations with Markdown

Ramses van Zon

October 30, 2024

ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Imagine...

You need to give a talk/presentation/lecture:

- with code blocks with syntax highlighting
- with images, columns, logos, select fonts, etc.
- with one version for presenting and another for publishing
- under version control (e.g. github, gitlab)

Slides!

But plain-text based (so version control works);

No jupyter notebooks (too messy);

No reveal.js (requires a browser and has poor pdf results);

No powerpoint (proprietary and hard to reproduce);

No libreoffice (hard to reproduce)

So make slides with LaTeX, of course!

Why?

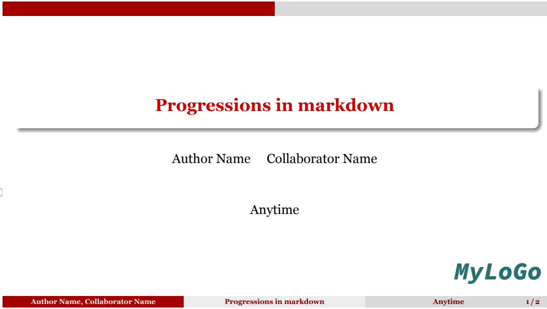
- There’s the beamer package.
- Can use overlays.
- Manual syntax highlighting
- Processed with pdflatex or xelatex
- If you’re in Physics, Math, or CS, you probably already know LaTeX.

Results can be good, but:

- Tedious and error-prone process
- Typically much LaTeX ‘hacking’ to get what you want
- Mixes style and content
- Lots of boilerplate
- Need to find LaTeX packages to use
- LaTeX’s learning curve is fairly high.

LaTeX example

```
\documentclass[13pt,aspectratio=169]{beamer}
\beamertemplatenavigationsymbolsempty
\usetheme[]{CambridgeUS}
\usefonttheme{structurebold}
\useinnertheme{rounded}
\usepackage{fontspec}
\setsansfont{Georgia}
\logo{\includegraphics[height=0.1\textheight]{MyLoGo
\usepackage{bookmark}
\title{Progressions in markdown}
\author{Author Name \and Collaborator Name}
\date{Anytime}
\begin{document}
\frame{\titlepage}
\begin{frame}{This is my first slide}
It has some text.\par
And a list:
\begin{itemize}
\item Item 1
\item Item 2
\end{itemize}
\end{frame}
\end{document}
```



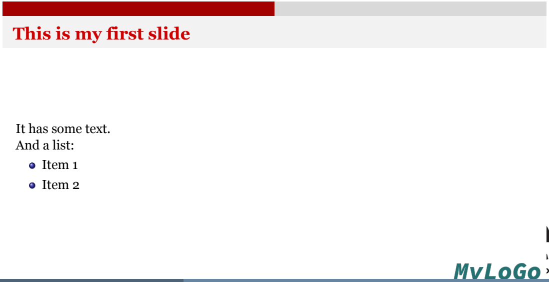
Progressions in markdown

Author Name   Collaborator Name

Anytime

MyLoGo

Author Name, Collaborator Name | Progressions in markdown | Anytime | 1 / 2



This is my first slide

It has some text.
And a list:

- Item 1
- Item 2

MyLoGo

Enter Pandoc

- Pandoc is a utility to transform one markup language into another.
- A markup language is a plain-text document format with tags that describe how it should be displayed.
- Examples of markup languages:
  - LaTeX
  - HTML
  - XML
  - Markdown
- Of these, markdown usually has the least boilerplate.
  It was designed to look closely like old-style email text.
  But there are various versions.

Markdown basics 1/3

https://www.markdownguide.org/basic-syntax

Headings

```
# Level 1 Heading

## Level 2 Heading

### Level 3 Heading
```

etc.

Alternative for level 1 and 2:

```
Level 1 Heading
===============


Level 2 Heading
---------------
```

Paragraphs

Subsequent lines are part of the same paragraph.

You need an empty line to end a paragraphs.

You should put an empty line before a header, or the start of a list, or other special markup, else they become part of the preceeding paragraph.

Text formatting

`*Italic*`	*Italic*
`**Bold**`	**Bold**
`***Bold Italic***`	***Bold Italic***
`~Subscript~`	Subscript
`^Superscript^`	Subscript
`~~Strike-through~~`	~~Strike-through~~
‘code‘	`code`

Markdown basics 2/3

https://www.markdownguide.org/basic-syntax

Unnumbered Lists

```
- Item 1
- Item 2
```

Gives:

- Item 1
- Item 2

+ and * can also be used.

Numbered Lists

```
1. Item 1
1. Item 2
```

Gives:

1. Item 1
2. Item 2

Code Blocks

Indenting a block with at least 4 spaces, makes that a code block, i.e., a verbatim text typeset in a monospace font.

```
    #!/bin/bash          #!/bin/bash
    echo Hello!          echo Hello!
```

Fenced Code Blocks

This is a **markdown extension** in pandoc.

Your code block starts with three backticks:

~~~~
```
#!/bin/bash              #!/bin/bash
echo Hello!              echo Hello!
```
~~~~

Markdown basics 3/3

https://www.markdownguide.org/basic-syntax

Including images

Syntax:

```
![CAPTION](FILELOCATION)
```

Example:

```
![A logo](MyLoGo.png)
```

Figure 1: A logo

You can leave out the caption then no text will appear.

Links

Syntax:

```
[LINK TEXT](URL)
```

Example:

```
[course](https://scinet.courses/1377)
```

If the LINK TEXT should be the URL, you can use

```
<URL>
```

Blockquotes

Can be make by prepending lines with ">".

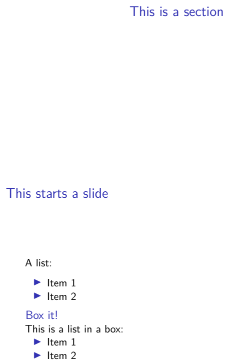
Horizontal rule

Can be made with three dashes "---".

Markdown for pandoc+beamer

- The level 1 headers become sections
- The level 2 headers become slides
- The level 3 headers become boxes

```
# This is a section

## This starts a slide

A list:

 - Item 1
 - Item 2

### Box it!

This is a list in a box:

 - Item 1
 - Item 2
```



This is a section

This starts a slide

A list:

- Item 1
- Item 2

Box it!

This is a list in a box:

- Item 1
- Item 2

```
$ pandoc -t beamer pandocbeamer.md -o pandocbeamer.pdf
```

Code syntax highlighting

You can enter add a language after the triple backticks in a fenched code block:

~~~
```
#!/bin/bash
echo "this is code"
```
~~~

```
#!/bin/bash
echo "this is code"
```

~~~
```bash
#!/bin/bash
echo "this is highlighted code"
```
~~~

```bash
#!/bin/bash
echo "this is highlighted code"
```

~~~
```{.bash}
#!/bin/bash
echo "this is highlighted code"
```
~~~

```bash
#!/bin/bash
echo "this is highlighted code"
```

This latter form will allow options (later). The period is not a mistake.

Which languages?

```
$ pandoc --list-highlight-languages | column
```

```
abc           cpp           email         isocpp        markdown      opencl        relaxngcompa  swift
actionscript  crystal       erlang        j             mathematica   orgmode       rest          systemverilog
ada           cs            fasm          java          matlab        pascal        rhtml         tcl
agda          css           fortranfixed  javadoc       maxima        perl          roff          tcsh
apache        curry         fortranfree   javascript    mediawiki     php           ruby          terraform
asn1          d             fsharp        javascriptre  metafont      pike          rust          texinfo
asp           dart          gap           json          mips          postscript    sass          toml
ats           debiancontro  gcc           jsp           modelines     povray        scala         typescript
awk           default       glsl          julia         modula2       powershell    scheme        verilog
bash          diff          gnuassembler  kotlin        modula3       prolog        sci           vhdl
bibtex        djangotempla  go            latex         monobasic     protobuf      scss          xml
boo           dockerfile    gpr           lex           mustache      pure          sed           xorg
c             dosbat        graphql       lilypond      nasm          purebasic     sgml          xslt
changelog     dot           groovy        literatecurry nim           purescript    sml           xul
clojure       doxygen       hamlet        literatehaske nix           python        spdxcomments  yacc
cmake         doxygenlua    haskell       llvm          noweb         qml           sql           yaml
coffee        dtd           haxe          lua           objectivec    r             sqlmysql      zig
coldfusion    eiffel        html          m4            objectivecpp  racket        sqlpostgresq  zsh
comments      elixir        idris         makefile      ocaml         raku          stan
commonlisp    elm           ini           mandoc        octave        relaxng       stata
```

Missing features with pandoc

- Rather involved syntax to do column layout
- Cannot input a source file, or part of it.
- Theming is hard
- No centering or alert support

Enter mds…

MDS: MarkDown for Slides

github: https://github.com/vanzonr/mds

- Adds convenience that pandoc is missing;
- It’s what I use for all my slides;
- It is very much a wrapper around pandoc and pdf/xelatex, which do the hard part;
- It is only (supposed to be) working on Linux as it’s just a bunch of pre- and post-processing hacks.

To see what it can do, it’s best to look at an example.

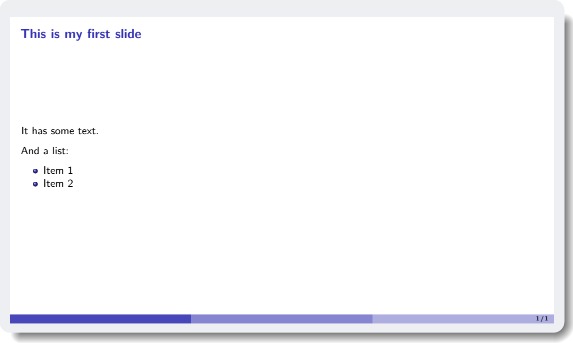
My First Slide

```
# This is my first slide
It has some text.

And a list:

  - Item 1
  - Item 2
```

Create a file with this content,
call it `progress.md`, and run

```
$ mds progress.md
```

which produces `progress.pdf`
(and the intermediate `progress.tex`).



This is my first slide

It has some text.

And a list:

- Item 1
- Item 2

1 / 1

```
% Options for packages loaded elsewhere
\PassOptionsToPackage{unicode}{hyperref}
\PassOptionsToPackage{hyphens}{url}
%
\documentclass[
  9pt,
  ignorenonframetext,
```

Now we add some style

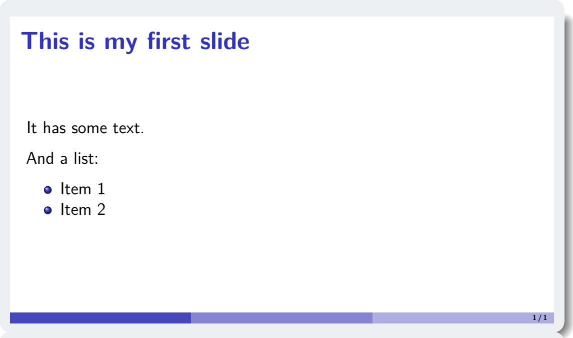
Let's start by increasing the font size, by putting, in the same directory, a file called `style.theme` or `progress.theme` containing:

```
# progress.theme
FONTSIZE=13pt
```

and run `mds progress.md`.

Now add a logo:

```
# progress.theme
FONTSIZE=13pt
LOGO=MyLoGo.png
```

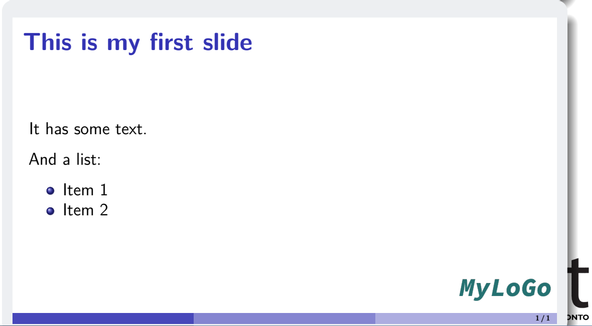


This is my first slide

It has some text.

And a list:

- Item 1
- Item 2



This is my first slide

It has some text.

And a list:

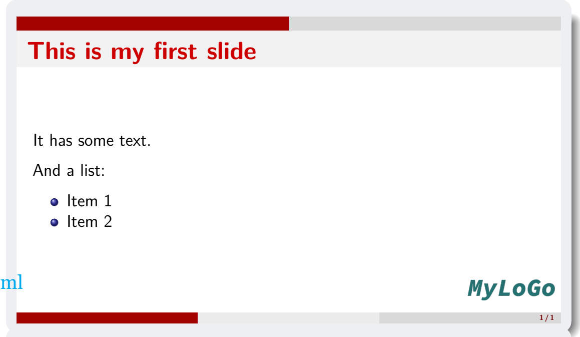
- Item 1
- Item 2

Let’s add more style

We can use any of the beamer themes.

```
# progress.theme
FONTSIZE=13pt
LOGO=MyLoGo.png
THEME=CambridgeUS
```

https://deic.uab.cat/~iblanes/beamer_gallery/index_by_theme.html



This is my first slide

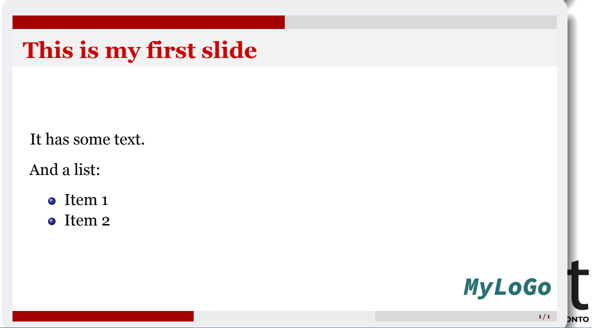
It has some text.

And a list:

- Item 1
- Item 2

Now specify a(n installed) font:

```
# progress.theme
FONTSIZE=13pt
LOGO=MyLoGo.png
THEME=CambridgeUS
FONTFAMILY=Georgia
```



This is my first slide

It has some text.

And a list:

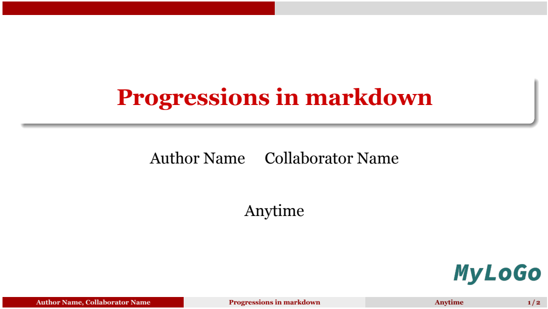
- Item 1
- Item 2

Now let's add a title slide

You need a yaml header:

```
---
title: Progressions in markdown
author:
   - Author Name
   - Collaborator Name
date: Anytime
---
# This is my first slide
It has some text.

And a list:

  - Item 1
  - Item 2
```

Behind the scenes, `mds` appends to the yaml header to affect the theme.



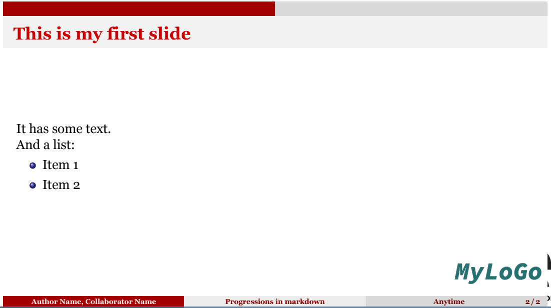
Progressions in markdown

Author Name   Collaborator Name

Anytime

MyLoGo

Author Name, Collaborator Name | Progressions in markdown | Anytime | 1 / 2



This is my first slide

It has some text.
And a list:

- Item 1
- Item 2

MyLoGo

Author Name, Collaborator Name | Progressions in markdown | Anytime | 2 / 2

MDS extensions: Easy columns

```
[[

Column1

### With a block

||
Column 2

### Another block

||
### Column 3

... in a block

]]
```

This produces the following:

Column1

With a block

Column 2

Another block

Column 3

... in a block

Note that the width is automatically evenly devided.

MDS Extensions: Handy, random markup

Centering a line

```
->This message is centered.<-
```

This message is centered.

Alert

```
****A different color as an alert****
```

A different color as an alert!

A single period on a line = Vertical space

```
[[
### H1
The quick brown fox jumps ov
||
### H2
C
.
]]
```

H1

The quick brown fox jumps over the lazy dog.

H2

C

MDS Extensions: Source file inclusion

- What if the source that you want to show is inside an existing file?
- We can simply ‘include’ it
- Syntax

~~~
```{.LANGUAGE include=FILENAME}
```
~~~

- To include a subportion, add a `startLine=` and/or a `endLine=`.

Example

~~~
```{.markdown include=cocomds.md startLine=1 endLine=5}
```
~~~

```
---
title: "Delivering Code-Heavy Presentations with Markdown"
author: "Ramses van Zon"
date: "October 30, 2024"
---
```

Incremental Reveal

A.k.a overlays

Pandoc can "pause" showing the rest of a slide by having . . . on a single line.

This slide is a good example.

```
A.k.a overlays
. . .
Pandoc can "pause" showing the rest of a slide
by having `. . .` on a single line.
. . .
This slide is a good example.
```

MDS Extension: pick on which overlays something appears.

Syntax:

@ @ @ NUMBER

That which should only appear on overlay 'NUMBER'

@ @ @ @

More theming

Available .theme Options

Option	Default value	Meaning
HIGHLIGHT	zenburn	Syntax highlight style
ASPECTRATIO	169	Slide aspect ratio (1610, 169, 149, 54, 43 or 32)
LOGO		Path to a file containing the botton-right logo
LOGOHEIGHT	0.1	Size of logo relative to the height
LOGOWIDTH		Size of logo relative to the width
LOGOUP		Relative amount to move the logo up (allowed to be negative)
LOGOONTITLE	true	Whether the logo should appear on the title page
FONTFAMILY		Name of an installed font
FONTSIZE	9pt	Size of the font
THEME	Boadilla	Beamer theme to use
INNERTHEME	rounded	Beamer ‘inner’ theme to use
FONTTHEME	structurebold	Beamer ‘font’ theme to use
COLORTHEME	orchid	Beamer ‘color’ theme to use
HANDOUT	false	When set to true, omits overlays (e.g. “. . .” pauses)
ENGINE	pdflatex	Program to use to generate the pdf from the LaTeX intermediate
THEMETEX	style.tex	Additional latex to add in the LaTeX preamble

Fonts: FONT=fontname

```
$ fc-list : family
```

If this does not give anything, you must install some fonts, or use LaTeX's default fonts.

When using a font, the engine will automatically change from pdflatex to xelatex.

Further tweaking: THEMETEX=style.tex

You can insert a file into the preamble of latex code.

If you know what you're doing, you can then modify virtually all aspects of the style that beamer and/or latex make possible.

Conclusions

What does this MDS achieve?

- Quick development of slides
- Separating style/theme from content to a large extent
- Making common slide patterns simpler, e.g.
  - Centering
  - Columns
  - Including source files
- Still tweakable through adding a style.tex file or even inline latex (but try not to)

ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Further opportunities

- Theming: we hardly touch upon this.

  Keep an eye on the github repo, I plan to make this easier.

- Running code automatically.

  Like `rmarkdown`, or `quarto`.

Thank you for your attention. Questions?