Introduction to Linux: **Markup Languages**

Philipp Fruck & Pius Walter

Markup Languages

Goal of this Lecture

- Introduce markup languages and show their importance
- Understand how tables are written in Markdown, Typst, and LaTeX
- Learn how bibliographies work in LaTeX (incl. Zotero & Better BibTeX)
- Explore Pandoc and its document-conversion capabilities
- Know the role of LaTeX distributions (e.g., TeX Live)
- Compare the differences between pdfLaTeX and XeLaTeX

Why?

- Tables and citations are essential for academic writing
- Pandoc is a key tool in modern documentation workflows
- Understanding LaTeX distributions helps with installing and debugging
- Choosing the right LaTeX engine improves compatibility and output quality

Overview

What Are Markup Languages?

- Text-based systems for structuring and formatting documents
- Human-readable
- Convert into other formats (HTML, PDF, slides, ...)
- Today taking a look on
  - Markdown
  - Typst
  - LaTeX

Advantages

- Easier consistent formatting compared to Word, etc.
- Similar to programming languages
- Version control using Git

When To Use What?

- **Markdown**: documentation, blogs, lightweight notes
- **Typst**: structured documents, reports, modern workflows
- **LaTeX**: heavy mathematics, academic papers, journals

Markdown

- Simple & lightweight
- Great for documentation, READMEs, notes

Typst

- New declarative typesetting
- Fast, modern, clear syntax

LaTeX

- Powerful scientific typesetting
- Industry standard for math-heavy work

LaTeX vs. Typst

LaTeX: Classic Typesetting

LaTeX is extremely powerful but more verbose

- 40+ years of tooling
- Academic gold standard
- Mature packages
- Verbose syntax
- Large installation

A "simple" LaTeX document snippet -->

```latex
\documentclass{article} % You can also use report, book, etc.

% Optional packages
\usepackage[utf8]{inputenc}   % For UTF-8 encoding
\usepackage{amsmath, amssymb} % For math symbols

\title{My LaTeX Document}
\author{Your Name}
\date{December 2, 2025}  % Or use \today for current date

\begin{document}
...
\end{document}
```

Typst and Markdown

Typst: A Modern Alternative

Typst is very fast and modern

- Engine is written in Rust
- Consists of a single static binary
- Much easier setup than LaTeX
- Syntax is a wild mix - some like it, we don't
- Built-in styling system
- Still a young ecosystem

A simple Typst document setup

```
#set page(size: "A4", margin: 2cm)
#set title["My Typst Document"]
#set author["Your Name"]
#set date[2025-12-02]
```

Markdown

- Simple syntax with minimal markup
- Raw text is clean and easy to read without rendering
- Can be converted to HTML, PDF, slides, and many other formats
- Widely supported, e.g., in GitHub, documentation, blogs, notes, etc.
- Writers can concentrate on writing instead of formatting
- Can include tables, code blocks, math, images, links, and more with minimal effort

Markdown Structure

```
# Level 1 Heading

This is an introductory paragraph under a level-1
heading.

## Level 2 Heading

More detailed section text goes here.

Here is a simple unordered list
- Apples
- Bananas

Here is an ordered list
1. First step
2. Second step

And finally, a nested list
- Programming languages
  - Rust
  - Python
```

Level 1 Heading

This is an introductory paragraph under a level-1 heading.

Level 2 Heading

More detailed section text goes here.

Here is a simple unordered list

- Apples
- Bananas

Here is an ordered list

1. First step
2. Second step

And finally, a nested list

- Programming languages
  - Rust
  - Python

Typst Structure

```
= Level 1 Heading

This is an introductory paragraph under a level-1
heading.

== Level 2 Heading

More detailed section text goes here.

Here is a simple unordered list
- Apples
- Bananas

Here is an ordered list
1. First step
2. Second step

And finally, a nested list
- Programming languages
  - Rust
  - Python
```

Level 1 Heading

This is an introductory paragraph under a level-1 heading.

Level 2 Heading

More detailed section text goes here.

Here is a simple unordered list

- Apples
- Bananas

Here is an ordered list

1. First step
2. Second step

And finally, a nested list

- Programming languages
  - Rust
  - Python

LaTeX Structure

```
\section{Level 1 Heading}
Introductory paragraph under a level-1 heading.

\subsection{Level 2 Heading}
More detailed section text goes here.

Here is a simple unordered list
\begin{itemize}
  \item Apples
  \item Bananas
\end{itemize}

Here is an ordered list
\begin{enumerate}
  \item First step
  \item Second step
\end{enumerate}

And finally, a nested list
\begin{itemize}
  \item Programming Languages
    \begin{itemize}
      \item Rust
      \item Python
    \end{itemize}
\end{itemize}
```

Level 1 Heading

Introductory paragraph under a level-1 heading.

Level 2 Heading

More detailed section text goes here.

Here is a simple unordered list

- Apples
- Bananas

Here is an ordered list

1. First step
2. Second step

And finally, a nested list

- Programming Languages
  - Rust
  - Python

Markdown Formatting

```
**Bold text**

__Also bold text__

*Italic text*

_Also italic text_

~~Strikethrough~~

<u>Underlined text</u>
<!-- Not standard, but supported in many Markdown
renderers -->

[Link example](https://example.com)

Line<br>break
<!-- Not standard, but supported in many Markdown
renderers -->
```

Bold text

Also bold text

Italic text

Also italic text

~~Strikethrough~~

<u>Underlined text</u>

<!-- Not standard, but supported in many Markdown
renderers -->

Link example (*https://example.com*)

Line
break

<!-- Not standard, but supported in many Markdown
renderers -->

Italic text cannot be displayed in this presentation due to the font used!

Typst Formatting

```
#text(weight: "bold")[Bold text]

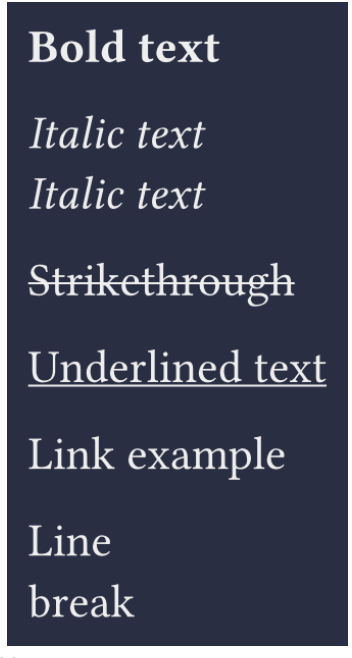
_Italic text_ \
#emph[Italic text]

#strike[Strikethrough]

#underline[Underlined text]

#link("https://example.com")[Link example]

Line\ break
```



LaTeX Formatting

```latex
\textbf{text}

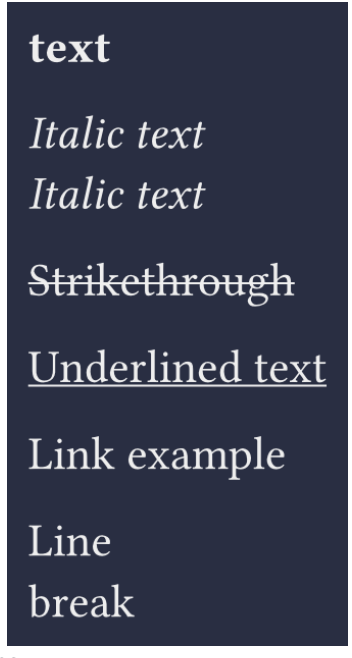
\textit{Italic text}\\
\emph{Italic text}

% requires \usepackage{ulem}
\sout{Strikethrough}

\underline{Underlined text}

% requires \usepackage{hyperref}
\href{https://example.com}{Link example}

Line\\break
```



Mathematical Formulas

Typst

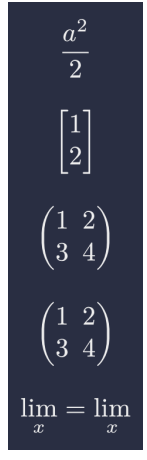
```
$frac(a^2, 2)$

$
vec(1, 2, delim: "[")
$

$
mat(1, 2; 3, 4)
$

$
mat(..#range(1, 5).chunks(2))
$

$
lim_x =
    op("lim", limits: #true)_x
$
```



LaTeX

```
\[ \frac{a^{2}}{2} \]

\[
\begin{bmatrix}
1 \\ 2
\end{bmatrix}
\]

\[
\begin{pmatrix}
1 & 2 \\3 & 4
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 2 \\3 & 4
\end{pmatrix}
\]

\[
\lim_{x} = \operatorname{lim}_{x}
\]
```

Tables

Markdown

- Very simple table syntax
- Limited styling in pure Markdown
- Extended features depend on renderer (GitHub, Pandoc, etc.)

```
| **Name** | **Age** | **City** |
|:---------|:-------:|---------:|
| Alice    | 23      | Ulm      |
| Bob      | 21      | Mannheim |
```

http://markdowntable.com

Name	**Age**	**City**
Alice	23	Ulm
Bob	21	Mannheim

Typst

- Built-in table type
- Very clean syntax
- Supports alignment, styling, table headers

```
#table(
  columns: 3,
  align: (left, center, right),
  [*Name*], [*Age*], [City],
  [Alice], [23], [Ulm],
  [Bob], [21], [Mannheim],
)
```

Name	Age	City
Alice	23	Ulm
Bob	21	Mannheim

LaTeX

- Very flexible
- Requires the `tabular` environment
- Use `booktabs` for nicer tables

```
\begin{tabular}{l c r}
  \hline
  Name & Age & City \\
  \hline
  Alice & 23 & Ulm \\
  Bob   & 21 & Mannheim \\
  \hline
\end{tabular}
```

https://www.tablesgenerator.com/latex_tables

Name	Age	City
Alice	23	Ulm
Bob	21	Mannheim

Bibliography in LaTeX

Basics

- Managed using BibTeX or BibLaTeX
- Bibliography stored in `.bib` files
- Common fields:
  - `author`, `title`, `year`, `journal`, `url`, ...
- Citation commands
  - `\cite{key}`
  - `\parencite{key}` (BibLaTeX)
  - `\textcite{key}` (BibLaTeX)

The `.bib` file (bibliography database file)

```
@book{knuth1998,
  author = {Donald E. Knuth},
  year = {1998},
  title = {The {\TeX} Book},
  publisher = {Addison-Wesley Professional}
}
```

Import the package `biblatex` and add the bibliography database file

```
\usepackage[backend=biber,style=authoryear]{biblatex}
\addbibresource{refs.bib}
```

Use the references in the document

```
According to \textcite{knuth1998}, ...
```

Managing the Bibliography

Zotero

- Free reference manager
- Automatic metadata scraping from PDFs and websites
- Plugins for Word, LibreOffice and Firefox
- Stores references, notes, PDFs in a central library

Why use it?

- Easy to organise references
- Sync across devices
- Works extremely well with LaTeX

Better BibTeX (BBT)

- Zotero plugin for LaTeX users
- Adds
  - Better BibTeX keys (e.g., `AuthorYearTitle`)
  - Automatic `.bib` export
  - Auto-updating `.bib` on save
- Perfect toolchain for LaTeX writing

LaTeX Distributions

What Is a LaTeX Distribution?

- Bundle of
  - LaTeX engine (pdfTeX, XeTeX, LuaTeX, ...)
  - Thousands of packages
  - Fonts
  - Documentation tools
- Needed to compile `.tex` files

Common Distributions

- TeX Live
  - Cross-platform (Linux, macOS, Windows)
  - Very complete
  - Package manager: `tlmgr`
- MiKTeX
  - Windows-focused
  - Installs missing packages automatically
- MacTeX
  - macOS version of TeX Live
  - Large but well-integrated

TeX Live Features

- Industry-standard distribution
- Updated yearly (`TeX Live YYYY`)
- Includes:
  - LaTeX kernel
  - BibLaTeX/Biber
  - AMS packages
  - Graphics and font packages
  - Many engines: `pdftex`, `xetex`, `luatex`

Package Management

```
tlmgr install package
tlmgr update --self
tlmgr update --all
```

Why TeX Live?

- Best support on Linux
- Most reproducible environment
- Works perfectly with CI/CD pipelines

Building a LaTeX project

pdfLaTeX

- Oldest and most stable
- Produces compact PDFs
- Limited font support (only Type 1 fonts by default)
- UTF-8 works but sometimes tricky
- Best for:
  - Scientific papers
  - Documents without special typography

latexmk

- Wrapper around multiple LaTeX compilers
- Automates repeated runs (e.g. bibliography, ToC)

latexdiff

- Highlight diff between documents (with git)
- `latexdiff-vc --pdf --flatten -r <commit1> [-r <commit2>] main.tex`

XeLaTeX

- Modern engine
- Full Unicode support
- Can use system fonts (e.g., Arial, Fira Code)
- Better for multilingual documents
- Slightly slower than pdfLaTeX
- Ideal for:
  - CVs
  - Design-heavy documents
  - Anything requiring custom fonts

```
\usepackage{fontspec}
\setmainfont{Fira Sans}
```

Pandoc

What Is Pandoc?

- The Swiss-army knife of document conversion
- Converts between almost any formats
- Popular in technical documentation

What Can It Do?

- Markdown -> PDF (via LaTeX or Typst)
- Markdown -> HTML/Slides
- LaTeX -> Markdown
- Word -> Markdown
- Typst -> PDF

```
pandoc notes.md -o notes.pdf
pandoc file.tex -o file.md
```

Why It's Useful

- Automates formatting
- Great for reproducible documents
- Works nicely in CI pipelines
- Robust tool for long-term documentation workflows

Pandoc

Markdown -> PDF via LaTeX

```
pandoc report.md \
    --from markdown \
    --to pdf \
    --pdf-engine=pdflatex \
    -o report.pdf
```

Real world example using a LaTeX template!

Markdown -> Typst

```
pandoc slides.md -t typst -o slides.typ
```

Filters & Extensions

- Pandoc has **filters** for modifying AST
- Example: convert latex math to images, handle citations, code-block highlighting
- Zotero -> Pandoc works via Citeproc

```
pandoc paper.md \
    --citeproc \
    --bibliography=refs.bib \
    -o paper.pdf
```

Thank you!

Thank you for your attention!

Don't forget the feedback in Moodle please!