Scientific Writing with LaTeX

Part 1: Foundations

Paulo Fagandini

Course Overview — 4 Hours

Part 1: Foundations (this deck)

1. Setup and first document
2. Document basics
3. References and bibliography
4. Lists and code

Part 2: Layout and Visuals

5. Columns, colors, boxes
6. Mathematics
7. Tables
8. Images and figures
9. TOC and `\input`

Part 3: Graphics

10. Ti*k*Z basics
11. PGFplots

Part 4: Special Classes

12. The `exam` class
13. Beamer presentations

Why LaTeX?

Advantages

- Professional typesetting (especially math)
- Separation of content and style
- Stable across decades
- Reference management built-in
- Required by many journals
- Free and open source

The learning curve

- Steeper than Word at first
- But flatter for complex documents
- You describe *what*, not *how*
- Errors can be cryptic

Mindset

Think of LaTeX as *programming* a document, not *drawing* one.

Overleaf — The Easy Start

What is Overleaf?

- Online LaTeX editor — nothing to install
- Real-time collaboration (like Google Docs)
- Thousands of templates
- Free tier is enough for this course

Recommendation

Use Overleaf for this course even if you plan to work locally later.

Free tier limits

Limited compile time and collaborators. For large theses, consider a local setup.

Getting started

1. Go to `https://www.overleaf.com`
2. Create a free account
3. New Project → Blank Project
4. You are ready

Local Installation — Step by Step

Step 1: Install a TeX distribution

- **Windows:** MiKTeX (`https://miktex.org`) or TeX Live
- **macOS:** MacTeX (`https://tug.org/mactex/`)
- **Linux:** `sudo apt install texlive-full`

Step 2: Install VSCodium or VSCode

- VSCodium: `https://vscodium.com` (open-source, no telemetry)
- VSCode: `https://code.visualstudio.com`

Step 3: Install the `LaTeX Workshop` extension

- Extensions panel → search "LaTeX Workshop" → Install
- Gives you: syntax highlighting, auto-compile, PDF preview

Your First Document

Code:

```
\documentclass{article}

\begin{document}

Hello, \LaTeX{}!

This is my first document.

\end{document}
```

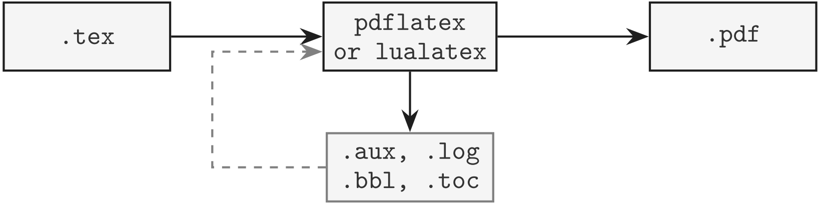
Key concepts

- `\documentclass{...}` chooses the document type
- Everything before `\begin{document}` is the **preamble** (settings, packages)
- Content goes between `\begin{document}` and `\end{document}`
- Commands start with \

Common classes

`article`, `report`, `book`, `letter`, `beamer`, `exam`

How Compilation Works



- The engine reads `.tex` and produces `.pdf`
- Auxiliary files store cross-references

- Multiple compilations often needed
- Overleaf and LaTeX Workshop handle this

Document Class Options

```
\documentclass[a4paper, 12pt]{article}
```

Paper size

- a4paper — standard in Europe
- letterpaper — standard in US

Font size

- 10pt (default), 11pt, 12pt

Other useful options

- twocolumn — two-column layout
- twoside — different odd/even margins
- draft — shows overfull boxes
- landscape — rotates the page

Tip

Most journals provide their own class file (e.g. elsarticle.cls).

Title, Author, Date

```
\title{My First Paper}
\author{Alice Smith\thanks{Univ.\ of
 Lisbon} \and Bob Jones\thanks{MIT}}
\date{February 2026}

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

Key commands

- `\title{...}` — paper title
- `\author{...}` — author name(s)
- `\and` — separates authors
- `\thanks{...}` — footnote for affiliations
- `\date{...}` — date (`\today` works)
- `\maketitle` — renders it all

Common mistake

Forgetting `\maketitle` inside the document body.

Abstract

```latex
\begin{document}
\maketitle

\begin{abstract}
  This paper presents a novel
  approach to writing documents
  using \LaTeX{}.
\end{abstract}

\section{Introduction}
...
```

The `abstract` environment

- Placed right after `\maketitle`
- Auto-formatted (indented, smaller font)
- Available in `article` and `report`
- Not in `book` class

Keywords

No built-in command. Add manually after abstract:

`\textbf{Keywords:} latex, typesetting`

Sections and Subsections

```latex
\section{Introduction}
Text of the introduction...

\subsection{Background}
Some background...

\subsubsection{Detail}
Even more detail...

\paragraph{A specific point.}
Named paragraph text...
```

Hierarchy in `article`

Command	Level
`\section{}`	1
`\subsection{}`	2
`\subsubsection{}`	3
`\paragraph{}`	4
`\subparagraph{}`	5

In `report`/`book`: `\chapter{}` sits above `\section{}`.

Starred: `\section*{Intro}` — no number, not in TOC.

Footnotes and Text Formatting

Footnotes

```
This is important\footnote{See
  Smith (2020) for details.}.
```

Formatting commands

```
\textbf{bold}
\textit{italic}
\underline{underlined}
\emph{emphasis}
\texttt{monospaced}
\textsc{Small Caps}
```

Rendered output

- **bold**
- *italic*
- underlined
- *emphasis*
- `monospaced`
- SMALL CAPS

Why `\emph`?

It toggles: in normal text it italicizes, inside italic it un-italicizes. Conveys *meaning*, not appearance.

Font Sizes

Size commands

Command	Example
`\tiny`	tiny text
`\scriptsize`	scriptsize
`\footnotesize`	footnotesize
`\small`	small text
`\normalsize`	normal
`\large`	large
`\Large`	Large
`\LARGE`	LARGE

Spacing

```
% Vertical space
\vspace{1cm}
\bigskip  \medskip  \smallskip

% Horizontal space
\hspace{2cm}
\quad  \qquad

% New paragraph
\par  % or a blank line
```

Avoid abusing \\

Use blank lines for paragraphs. The \\\\ command is a line break, not a paragraph break.

Special Characters and Comments

Reserved characters

These have special meaning:

```
# $ % ^ & _ { } ~ \
```

To print them literally:

```
\# \$ \% \^{} \& \_ \{ \}
\~{} \textbackslash
```

Comments

```
% This line is a comment.
Visible text. % rest is hidden
```

Useful symbols

```
\ldots          % ...
\LaTeX          % LaTeX logo
``smart quotes''
-- (en dash)  --- (em dash)
```

Packages — Extending LaTeX

```
% In the preamble:
\usepackage[utf8]{inputenc}    % character encoding
\usepackage[T1]{fontenc}       % font encoding
\usepackage{amsmath}           % advanced math
\usepackage{graphicx}          % images
\usepackage{hyperref}          % clickable links
```

How it works

- Packages add commands and features
- Options in `[brackets]`, name in `{braces}`
- Overleaf installs automatically
- Locally: MiKTeX auto-installs

Essential packages

`amsmath` — math
`graphicx` — images
`hyperref` — links (load **last**)
`booktabs` — nice tables
`biblatex` — bibliography
`xcolor` — colors
`geometry` — margins

The Modern Way: biblatex + biber

Why biblatex?

- Modern replacement for BibTeX/natbib
- Very flexible and customizable
- Uses biber backend (full Unicode)
- Actively maintained

Setup in preamble:

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

In the document body:

```
As shown by
\textcite{smith2020}, the
results are significant
\parencite{smith2020}.

% At the end of the document:
\printbibliography
```

Compilation order

pdflatex → biber → pdflatex ×2

Overleaf handles this automatically.

The .bib File

```
% references.bib
@article{smith2020,
  author  = {Smith, John and
             Doe, Jane},
  title   = {A Great Paper},
  journal = {J. of Science},
  year    = {2020},
  volume  = {42},
  pages   = {1--15},
  doi     = {10.1234/example}
}

@book{knuth1984,
  author    = {Knuth, Donald E.},
  title     = {The \TeX{}book},
  publisher = {Addison-Wesley},
  year      = {1984}
}
```

```
@inproceedings{lee2019,
  author    = {Lee, A.},
  title     = {Deep Learning},
  booktitle = {Proc. of NeurIPS},
  year      = {2019},
  pages     = {100--110}
}

@online{wiki2024,
  author  = {{Wikipedia}},
  title   = {LaTeX},
  url     = {https://en.wikipedia
             .org/wiki/LaTeX},
  urldate = {2024-12-01}
}
```

Other types: `@phdthesis`, `@techreport`, `@misc`, `@incollection`

Citation Commands

Command	Output (authoryear)	Use case
`\textcite{key}`	Smith (2020)	Subject of sentence
`\parencite{key}`	(Smith, 2020)	Parenthetical
`\autocite{key}`	Depends on style	Let the style decide
`\citeauthor{key}`	Smith	Author only
`\citeyear{key}`	2020	Year only
`\footcite{key}`	(footnote)	Footnote citation

Multiple citations: `\parencite{smith2020, jones2021}`

With page numbers: `\parencite[p.~42]{smith2020}`

Common Bibliography Styles

Style option	Description
`authoryear`	Smith (2020) — social sciences
`numeric`	[1] — engineering, CS
`alphabetic`	[Smi20] — compact label
`apa`	APA 7th (needs `biblatex-apa`)
`ieee`	IEEE (needs `biblatex-ieee`)

Where to get `.bib` entries

Google Scholar → Cite → BibTeX | Zotero/Mendeley export | Journal DOI lookup

Labels and Cross-References

```
\section{Introduction}
\label{sec:intro}

See Section~\ref{sec:methods} for the approach.

\section{Methods}
\label{sec:methods}

The result is in Table~\ref{tab:results}
and Figure~\ref{fig:diagram}.
```

How it works

- `\label{key}` — place a marker
- `\ref{key}` — insert the number
- `\pageref{key}` — insert the page
- Needs **two compilations**

Naming convention

Use prefixes: `sec:`, `fig:`, `tab:`, `eq:`
Avoids collisions and keeps things clear.

Basic Lists

Unordered

```
\begin{itemize}
  \item First point
  \item Second point
  \item Third point
\end{itemize}
```

- First point
- Second point
- Third point

Ordered

```
\begin{enumerate}
  \item First step
  \item Second step
  \item Third step
\end{enumerate}
```

1. First step
2. Second step
3. Third step

Description lists: `\begin{description} \item[Term] Definition \end{description}`

Nested and Custom Lists

Nesting

```
\begin{enumerate}
  \item Main point
    \begin{itemize}
      \item Sub-point A
      \item Sub-point B
    \end{itemize}
  \item Another point
\end{enumerate}
```

Custom labels

```
\begin{itemize}
  \item[-] Minus sign
  \item[$\star$] Star
\end{itemize}
```

With `enumitem` package:

```
\usepackage{enumitem}
\begin{enumerate}[label=\alph*)]
  \item First     % a)
  \item Second    % b)
\end{enumerate}
```

`enumitem`

Full control over labels, spacing, and alignment.

Displaying Code — `verbatim`

The simplest way

```
\begin{verbatim}
for i in range(10):
    print(i)
\end{verbatim}
```

Simple monospace output, no highlighting.

Inline:

`\verb|x = 42|`

When to use `verbatim`

- Quick code snippets
- Showing exact text (paths, commands)
- No configuration needed

Fragile frames

In Beamer, frames with verbatim or listings need the `[fragile]` option:

```
\begin{frame}[fragile]{Title}
```

Displaying Code — `listings` Package

Setup

```
\usepackage{listings}

\begin{lstlisting}[language=Python]
for i in range(10):
    print(f"Value:  {i}")

\end{lstlisting}
```

Also available: `minted`

```
\usepackage{minted}

\begin{minted}{python}
for i in range(10):
    print(f"Value:  {i}")

\end{minted}
```

Requires `-shell-escape` and Python.

	listings	**minted**
Setup	Easy	Needs Python
Highlighting	Manual config	Automatic (Pygments)
Overleaf	Works	Works

`listings` — Customization

```
\lstdefinestyle{mystyle}{
  language=Python,
  backgroundcolor=\color{backcolour},
  basicstyle=\ttfamily\footnotesize,
  keywordstyle=\color{blue}\bfseries,
  commentstyle=\color{green!50!black},
  stringstyle=\color{red},
  numbers=left,
  numberstyle=\tiny\color{gray},
  frame=single,
  breaklines=true
}
\lstset{style=mystyle}
```

Inline code:

`\lstinline|x = 42|`

From external file:

`\lstinputlisting[language=R]{code.R}`

End of Part 1

Next: Layout and Visuals

(Let’s grab some coffee!)