

LaTeX Beginner Tutorial

Zhihai He

December 8, 2024

Contents

1	Basic Page Layout	2
2	Titles and Sections	2
3	New Page, New Line, and Comments	2
4	Inserting Mathematical Formulas	3
5	Adding Figures	5
6	Creating Tables	5
7	Adding Footnotes	5
8	Using BibTeX for References	5
9	Custom: LaTeX Programming Practices	6
10	Conclusion	8

1 Basic Page Layout

The `\documentclass` command defines the type of document. For example:

- `article`: Suitable for short documents.
- `report`: For longer documents, such as theses.
- `book`: For books.

You can adjust the page layout using the `geometry` package. Example:

```
\usepackage[margin=1in]{geometry}
```

2 Titles and Sections

Define the title, author, and date of your document:

```
\title{LaTeX Beginner Tutorial}  
\author{Zhihai He}  
\date{\today}  
\maketitle
```

To structure your content, use:

- `\section{...}`: First-level heading.
- `\subsection{...}`: Second-level heading.
- `\subsubsection{...}`: Third-level heading.

To create unnumbered sections that do not appear in the Table of Contents, use:

- `\section*{...}`
- `\subsection*{...}`
- `\subsubsection*{...}`

3 New Page, New Line, and Comments

`\newpage` starts a new page. `\\` creates a new line within the same paragraph.

`%` is used for single-line comments, while multiline comments can be created using the `comment` package. Examples:

```
% This is a single-line comment.
```

```
\usepackage{comment}

\begin{comment}
  This is a multiline comment.
  This is another line in the comment.
\end{comment}
```

4 Inserting Mathematical Formulas

LaTeX provides multiple ways to insert mathematical formulas, depending on the context and formatting requirements. Below are some commonly used methods:

Inline Formulas

Inline formulas are embedded within text and written using \dots or $\langle\dots\rangle$.

- Example using \dots :
This is an inline formula: $E = mc^2$.
- Example using $\langle\dots\rangle$:
This is another inline formula: $a^2 + b^2 = c^2$.

Displayed Formulas

Displayed formulas are centered and occupy their own line. They can be written using \displaystyle or the `equation` environment (with optional numbering).

- Without numbering:

$$a^2 + b^2 = c^2$$

- With numbering:

$$E = mc^2 \tag{1}$$

The formula (1) is Einstein's energy-mass equivalence.

Aligned Formulas

Use the `align` environment to align multiple equations. The $\&$ symbol specifies the alignment point.

$$a + b = c \tag{2}$$

$$x + y = z \tag{3}$$

The equation (2) shows an example of aligned formulas.

Cases and Matrices

For piecewise functions or matrices, special environments like `cases` and `bmatrix` are used.

- Piecewise function using `cases`:

$$f(x) = \begin{cases} x^2 & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

- Matrix using `bmatrix`:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Custom Formula Tags

You can customize the numbering of equations using the `\tag` command.

$$a^2 + b^2 = c^2 \tag{Pythagoras}$$

Unnumbered Formulas

To suppress numbering for equations, use the `align*` or `equation*` environments.

$$\begin{aligned} a + b &= c \\ x + y &= z \end{aligned}$$

Some Mathematical Symbols

LaTeX provides a rich set of mathematical symbols for use in equations. Examples include:

- Operators: $+$, $-$, \times , \div
- Greek letters: α , β , γ , π
- Logical symbols: \forall , \exists , \neg , \wedge , \vee
- Set symbols: \cup , \cap , \subseteq , ∞
- Arrows: \rightarrow , \leftarrow , \iff

5 Adding Figures

To include an image, use the `graphicx` package. Example:

```
\usepackage{graphicx}

\begin{figure}[h]
  \centering
  \includegraphics[width=0.5\textwidth]{images/example.jpg}
  \caption{Sample Image}
  \label{fig:sample}
\end{figure}
```

Ensure the image is in the `images/` folder.

6 Creating Tables

Tables can be created using the `tabular` environment. Example:

```
\begin{table}[h]
  \centering
  \begin{tabular}{|c|c|c|}
    \hline
    Column 1 & Column 2 & Column 3 \\
    \hline
    Data 1   & Data 2   & Data 3   \\
    \hline
  \end{tabular}
  \caption{Sample Table}
  \label{tab:sample}
\end{table}
```

7 Adding Footnotes

Add footnotes using `\footnote{...}`. Example:

This is a sample footnote.\footnote{Here is the footnote content.}
This a sampe footnote¹.

8 Using BibTeX for References

To manage references, create a `.bib` file (e.g., `references.bib`) with the following content:

¹Here is the footnote content

```
@article{vaswani2017attention,
  title={Attention is all you need},
  author={Vaswani, A},
  journal={Advances in Neural Information Processing Systems},
  year={2017}
}
```

In your LaTeX file, use:

```
\bibliographystyle{plain}
\bibliography{references}
```

Attention is all you need[1].

9 Custom: LaTeX Programming Practices

When writing in LaTeX, it's important to follow good programming practices to ensure the document is well-structured, easy to maintain, and visually appealing. Below are some commonly recommended LaTeX coding habits:

Use Logical Sections

Organize your document into sections, subsections, and subsubsections using:

- `\section{...}`: For major sections.
- `\subsection{...}`: For subsections.
- `\subsubsection{...}`: For detailed sections.

Keep sections concise and avoid overusing `\subsubsection` unless necessary.

Break Lines Thoughtfully

LaTeX ignores single line breaks in the source file. To indicate a new paragraph, leave an empty line between two paragraphs. For example:

This is the first paragraph.

This is the second paragraph.

Do not use `\\` for creating new paragraphs. Instead, reserve it for line breaks within the same paragraph.

Use Comments for Clarity

Use `%` to add comments to your LaTeX file. Comments are ignored during compilation and can be helpful for:

- Explaining complex parts of the code.
- Temporarily disabling sections during debugging.

Example:

```
% This is a single-line comment
```

For longer comments, consider using the `comment` package:

```
\usepackage{comment}

\begin{comment}
This section is temporarily disabled.
\end{comment}
```

Keep Lines Short

Limit the length of each line in your LaTeX source to around 80 characters. This makes the file easier to read and edit, especially in version control systems.

Use Meaningful Labels

When labeling sections, figures, and tables, use clear and consistent names. For example:

```
\label{sec:introduction} % For sections
\label{fig:sample}        % For figures
\label{tab:results}       % For tables
```

Example:9 is about LaTeX programming custom. This makes cross-referencing easier and improves the readability of your code.

Avoid Hardcoding Layout

Instead of manually adjusting spaces or positions, use LaTeX commands and environments designed for layout. For example:

- Use `\vspace` and `\hspace` sparingly.
- Avoid fixed-width adjustments like `\hskip`.
- Use packages like `geometry` and `graphicx` for flexible layout control.

Test and Compile Frequently

Frequent compilation helps catch errors early. If using Overleaf, take advantage of the real-time preview feature. For large projects, compile a small subset of the document during debugging.

By following these practices, you can ensure your LaTeX document is easy to maintain and produces high-quality output.

10 Conclusion

This tutorial introduced the basics of LaTeX, including:

- Setting up the page layout.
- Inserting Mathematical Formulas
- Adding titles, figures, tables, and footnotes.
- Custom
- Managing references using BibTeX.

For more information, visit https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

References

- [1] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.