

# Introduction to LaTeX

---

PG software skills training

Author: Tina Zhou

Date: 06/17/2023



## Table of contents

---

1. [Introduction](#)
  - i. [Why learning LaTeX](#)
2. [Online set-up](#)
  - i. [Overleaf](#)
3. [Practice](#)
  - i. [Create a document](#)
  - ii. [Title pages](#)
  - iii. [Chapters, sections and paragraphs](#)
  - iv. [Tables of contents](#)
  - v. [Lists](#)
  - vi. [Equations](#)
  - vii. [Figures](#)
4. [Other useful links](#)

## Introduction

---

LaTeX (/ˈləːtek/ LAH-tech or /ˈleɪtek/ LAY-tech) is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. When writing, the writer uses plain text as opposed to the formatted text found in WYSIWYG word processors like Microsoft Word, LibreOffice Writer and Apple Pages. The writer uses markup tagging conventions to define the general structure of a document, to stylise text throughout a document (such as bold and italics), and to add citations and cross-references. A TeX distribution such as TeX Live or MiKTeX is used to produce an output file (such as PDF or DVI) suitable for printing or digital distribution. Currently, there are cloud-based LaTeX editors such as Overleaf.

LaTeX is widely used in academia for the communication and publication of scientific documents in many fields, including mathematics, computer science, engineering, physics, chemistry, economics. It has a prominent role in the preparation and publication of books and articles that contain complex multilingual materials, such as Arabic and Greek.[6] LaTeX uses the TeX typesetting program for formatting its output, and is itself written in the TeX macro language.

## Why learn LaTeX?

Various arguments can be proposed for, or against, learning to use LATEX instead of other document-authoring applications; but, ultimately, it is a personal choice based on preferences, affinities, and documentation requirements.

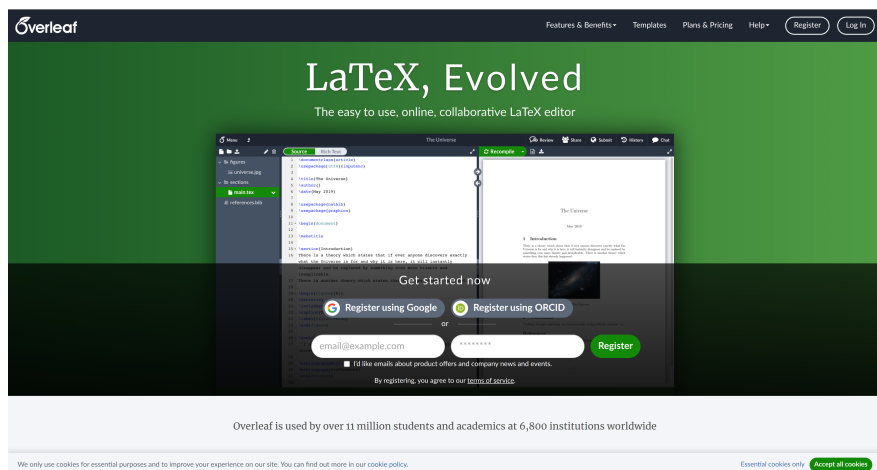
Arguments in favour of LATEX include:

- support for typesetting extremely complex mathematics, tables and technical content for the physical sciences;
- facilities for footnotes, cross-referencing and management of bibliographies;
- ease of producing complicated, or tedious, document elements such as indexes, glossaries, table of contents, lists of figures;
- being highly customizable for bespoke document production due to its intrinsic programmability and extensibility through thousands of free add-on packages.

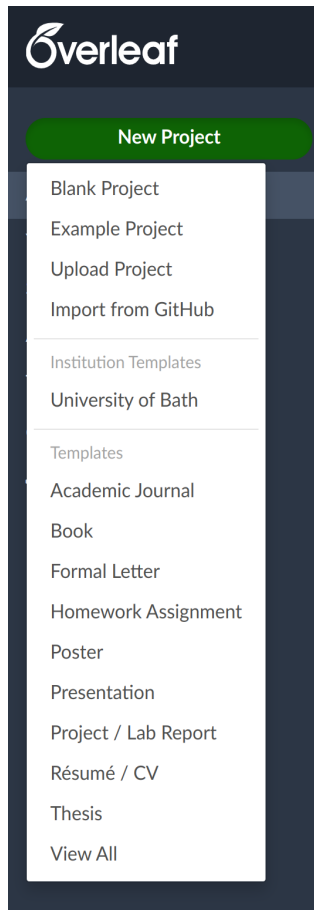
Overall, LATEX provides users with a great deal of control over the production of documents which are typeset to extremely high standards. Of course, there are types of documents or publications where LATEX doesn't shine, including many "free form" page designs typically found in magazine-type publications.

One important benefit of LATEX is the separation of document content from document style: once you have written the content of your document, its appearance can be changed with ease. Similarly, you can create a LATEX file which defines the layout/style of a particular document type and that file can be used as a template to standardise authorship/production of additional documents of that type; for example, this allows scientific publishers to create article templates, in LATEX, which authors use to write papers for submission to journals. Overleaf has a gallery containing thousands of templates, covering an enormous range of document types—everything from scientific articles, reports and books to CVs and presentations. Because these templates define the layout and style of the document, authors need only to open them in Overleaf—creating a new project—and commence writing to add their content.

## Overleaf setting-up



Create a free account in . To start using, click on creating a new Project and select the type of project you need.



## Practice

---

### Create a document

#### Document type

The first step is to create a new LATEX project. You can do this on your own computer by creating a new .tex file; alternatively, you can start a new project in Overleaf.

```
\documentclass{article}
\begin{document}
First document. This is a simple example, with no
extra parameters or packages included.
\end{document}
```

You can see that LATEX has automatically indented the first line of the paragraph, taking care of that formatting for you. Let's have a closer look at what each part of our code does.

The first line of code:

```
\documentclass{article}
```

declares the document type known as its class, which controls the overall appearance of the document. Different types of documents require different classes; i.e., a CV/resume will require a different class than a scientific paper which might use the standard LATEX article class. Other types of documents you may be working on may require different classes such as book or report.

## Preamble

The previous example showed how document content was entered after the `''\latex \begin{document}''` command; however, everything in your .tex file appearing before that point is called the preamble, which acts as the document's "setup" section. Within the preamble you define the document class (type) together with specifics such as languages to be used when writing the document; loading packages you would like to use (more on this later), and it is where you'd apply other types of configuration.

A minimal document preamble might look like this:

```
\documentclass[12pt, letterpaper]{article}
\usepackage{graphicx}
```

where `''\latex \documentclass[12pt, letterpaper]{article}''` defines the overall class (type) of document. Additional parameters, which must be separated by commas, are included in square brackets ([...]) and used to configure this instance of the article class; i.e., settings we wish to use for this particular article-class-based document.

In this example, the two parameters do the following:

- 12pt sets the font size;
- letterpaper sets the paper size;

Of course other font sizes, 9pt, 11pt, 12pt, can be used, but if none is specified, the default size is 10pt. As for the paper size, other possible values are a4paper and legalpaper. For further information see the article about page size and margins.

The preamble line

```
\usepackage{graphicx}
```

is an example of loading an external package (here, graphicx) to extend LATEX's capabilities, enabling it to import external graphics files.

## Title, author and date

Adding a title, author and date to our document requires three more lines in the preamble (not the main body of the document). Those lines are:

```
\title{My first LaTeX document}
```

is the document title

```
\author{Tina Zh}
```

here you write the name of the author(s) and, optionally,

the \thanks command within the curly braces:

```
\thanks{Funded by the ... team.}
```

can be added after the name of the author, inside the braces of the author command. It will add a superscript and a footnote with the text inside the braces. Useful if you need to thank an institution in your article.

```
\date{August 2022}
```

you can enter the date manually or use the command `\today` to typeset the current date every time the document is compiled. With these lines added, your preamble should look something like this:

```
\documentclass[12pt, letterpaper]{article}
\title{My first LaTeX document}
\author{Hubert Farnsworth\thanks{Funded by the Overleaf team.}}
\date{August 2022}
To typeset the title, author and date use the \maketitle command within the body of the document:

\begin{document}
\maketitle
We have now added a title, author and date to our first \LaTeX{} document!
\end{document}
The preamble and body can now be combined to produce a complete document which can be opened in Overleaf:

\documentclass[12pt, letterpaper]{article}
\title{My first LaTeX document}
\author{Hubert Farnsworth\thanks{Funded by the Overleaf team.}}
\date{August 2022}
\begin{document}
\maketitle
We have now added a title, author and date to our first \LaTeX{} document!
\end{document}
```

## Chapters, sections and paragraphs

### Chapters and sections

Longer documents, irrespective of authoring software, are usually partitioned into parts, chapters, sections, subsections and so forth. LaTeX also provides document-structuring commands but the available commands, and their implementations (what they do), can depend on the document class being used. By way of example, documents created using the book class can be split into parts, chapters, sections, subsections and so forth but the letter class does not provide (support) any commands to do that.

This next example demonstrates commands used to structure a document based on the book class:

```
\documentclass{book}
\begin{document}

\chapter{First Chapter}

\section{Introduction}

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam lobortis facilisis sem. Nullam nec mi et
neque pharetra sollicitudin. Praesent imperdiet mi nec ante.
Donec ullamcorper, felis non sodales...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi nec ante...

\subsection{First Subsection}
Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...
```

```
\section*{Unnumbered Section}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem...
\end{document}
```

The names of sectioning commands are mostly self-explanatory; for example,

```
\chapter{First Chapter}
```

creates a new chapter titled First Chapter,

```
\section{Introduction}
```

produces a section titled Introduction, and so forth. Sections can be further divided into

```
\subsection{...} and even \subsubsection{...}.
```

The numbering of sections, subsections etc. is automatic but can be disabled by using the so-called starred version of the appropriate command which has an asterisk (\*) at the end, such as

```
\section*{...} and \subsubsection*{...}.
```

Collectively, LaTeX document classes provide the following sectioning commands, with specific classes each supporting a relevant subset:

```
\part{part}
\chapter{chapter} % the \part and \chapter commands are only available in %the report and book document class
\section{section}
\subsection{subsection}
\subsubsection{subsubsection}
\paragraph{paragraph}
\subparagraph{subparagraph}
```

## Paragraphs and new lines

With the abstract in place, we can begin writing our first paragraph. The next example demonstrates:

how a new paragraph is created by pressing the "enter" key twice, ending the current line and inserting a subsequent blank line;  
how to start a new line without starting a new paragraph by inserting a manual line break using the

```
\\ command
```

, which is a double backslash; alternatively, use the

```
\newline command.
```

The third paragraph in this example demonstrates use of the commands

`\\` and `\newline`:

```
\documentclass{article}
\begin{document}
```

```
\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}
```

After our abstract we can begin the first paragraph, then press ``enter'' twice to start the second one.

This line will start a second paragraph.

I will start the third paragraph and then add `\\` a manual line break which causes this text to start on a new  
`\end{document}`

## Tables of contents

Creating a table of contents is straightforward because the command

```
\tableofcontents
```

does almost all the work for you:

```
\documentclass{article}
\title{Sections and Chapters}
\author{Tina Zh}
\date{June 2023}
\begin{document}
```

```
\maketitle
```

```
\tableofcontents
```

```
\section{Introduction}
```

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...

```
\section*{Unnumbered Section}
\addcontentsline{toc}{section}{Unnumbered Section}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...

```
\section{Second Section}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...  
`\end{document}`

## Lists

You can create different types of list using environments, which are used to encapsulate the LATEX code required to implement a specific typesetting feature. An environment starts with

```
\begin{environment-name}
```

and ends with

```
\end{environment-name}
```

where environment-name might be figure, tabular or one of the list types: itemize for unordered lists or enumerate for ordered lists.

### Unordered lists

Unordered lists are produced by the itemize environment. Each list entry must be preceded by the

```
\item
```

command, as shown below:

```
\documentclass{article}
\begin{document}
\begin{itemize}
  \item The individual entries are indicated with a black dot, a so-called bullet.
  \item The text in the entries may be of any length.
\end{itemize}
\end{document}
```

Put this code in your LaTeX project to see the output.

You can also open this larger Overleaf project which demonstrates various types of LATEX list.

### Ordered lists

Ordered lists use the same syntax as unordered lists but are created using the enumerate environment:

```
\documentclass{article}
\begin{document}
\begin{enumerate}
  \item This is the first entry in our list.
  \item The list numbers increase with each entry we add.
\end{enumerate}
\end{document}
Open this example in Overleaf.
```

T

As with unordered lists, each entry must be preceded by the

```
\item
```

command which, here, automatically generates the numeric ordered-list label value, starting at 1.

For further information you can open this larger Overleaf project which demonstrates various types of LATEX list or visit our dedicated help article on LATEX lists, which provides many more examples and shows how to create customized lists.



## Equations

One of the main advantages of LATEX is the ease with which mathematical expressions can be written. LATEX provides two writing modes for typesetting mathematics:

inline math mode used for writing formulas that are part of a paragraph

display math mode used to write expressions that are not part of a text or paragraph and are typeset on separate lines

Inline math mode

Let's see an example of inline math mode:

```
\documentclass[12pt, letterpaper]{article}
\begin{document}
In physics, the mass-energy equivalence is stated
by the equation  $E=mc^2$ , discovered in 1905 by Albert Einstein.
\end{document}
```

To typeset inline-mode math you can use one of these delimiter pairs:

```
\( ... \), $ ... $ or \begin{math} ... \end{math}
```

as demonstrated in the following example:

```
\documentclass[12pt, letterpaper]{article}
\begin{document}
\begin{math}
E=mc^2
\end{math} is typeset in a paragraph using inline math mode---as is  $E=mc^2$ , and so too is  $(E=mc^2)$ .
\end{document}
```

## Display math mode

Equations typeset in display mode can be numbered or unnumbered, as in the following example:

```
\documentclass[12pt, letterpaper]{article}
\begin{document}
The mass-energy equivalence is described by the famous equation

$$E=mc^2$$
 discovered in 1905 by Albert Einstein.
```

In natural units ( $c = 1$ ), the formula expresses the identity

```
\begin{equation}
E=m
\end{equation}
\end{document}
```

To typeset display-mode math you can use one of these delimiter pairs:

```
\[ ... \], \begin{displaymath} ... \end{displaymath} or \begin{equation} ... \end{equation}.
```

Historically, typesetting display-mode math required use of  $\mathbb{}$  characters delimiters, as in  $\mathbb{E} ...$  display math here  $\mathbb{E}$ , but this method is no longer recommended: use LaTeX's delimiters

```
\[ ... \]
```

instead.

## More complex example.

The following examples demonstrate a range of mathematical content typeset using LaTeX.

```
\documentclass{article}
\begin{document}
Subscripts in math mode are written as  $a_b$  and superscripts are written as  $a^b$ . These can be combined and

\[ T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1}, \dots, x^{i_p}, e_{j_1}, \dots, e_{j_q}) \]
```

We write integrals using  $\int$  and fractions using  $\frac{a}{b}$ . Limits are placed on integrals using superscripts.

```
\[ \int_0^1 \frac{dx}{e^x} = \frac{e-1}{e} \]
```

Lower case Greek letters are written as  $\omega$   $\delta$  etc. while upper case Greek letters are written as  $\Omega$   $\Delta$  etc.

Mathematical operators are prefixed with a backslash as  $\sin(\beta)$ ,  $\cos(\alpha)$ ,  $\log(x)$  etc.

```
\end{document}
```

The next example uses the `equation*` environment which is provided by the `amsmath` package, so we need to add the following line to our document preamble:

```
\usepackage{amsmath}%
```

For the `equation*` environment

For further information on using `amsmath`.

```
\documentclass{article}
\usepackage{amsmath}% For the equation* environment
\begin{document}
\section{First example}

The well-known Pythagorean theorem  $(x^2 + y^2 = z^2)$  was proved to be invalid for other exponents, meaning

\[ x^n + y^n = z^n \]
```

`\section{Second example}`

This is a simple math expression  $(\sqrt{x^2+1})$  inside text.  
And this is also the same:

```
\begin{math}
\sqrt{x^2+1}
\end{math}
```

but by using another command.

This is a simple math expression without numbering

```
\[ \sqrt{x^2+1} \]
```

separated from text.

This is also the same:

```
\begin{displaymath}
\sqrt{x^2+1}
\end{displaymath}
```

`\ldots` and this:

```
\begin{equation*}
\sqrt{x^2+1}
\end{equation*}
```

```
\end{equation*}
\end{document}
```

## Figures

In this section we will now look at how to add images to a LATEX document—note that you need to upload images to your Overleaf project.

The following example demonstrates how to include a picture:

```
\documentclass{article}
\usepackage{graphicx} %LaTeX package to import graphics
\graphicspath{{images/}} %configuring the graphicx package

\begin{document}
Some thing written here

% The \includegraphics command is
% provided (implemented) by the
% graphicx package
\includegraphics{<image name>}

There's a picture of .....
\end{document}
```

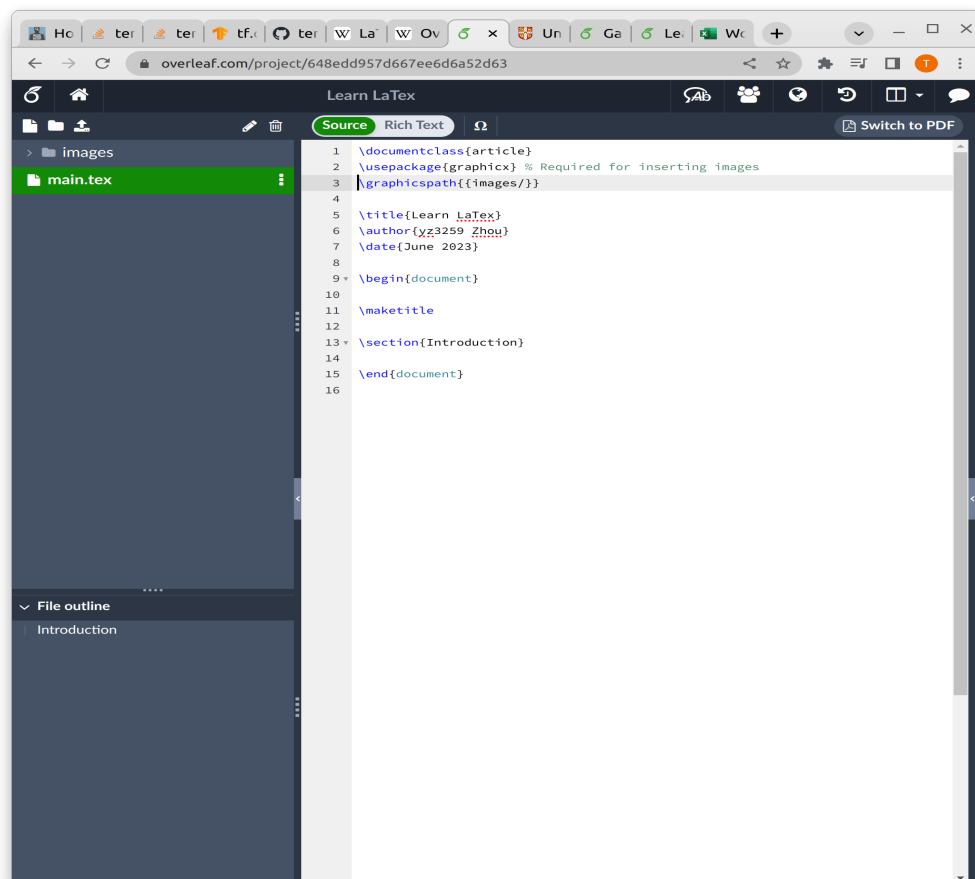
Try the code in your overleaf project and see the result. `<image name>` needs to be replaced to an image.

Importing graphics into a LATEX document needs an add-on package which provides the commands and features required to include external graphics files. The above example loads the `graphicx` package which, among many other commands, provides `\includegraphics{...}` to import graphics and `\graphicspath{...}` to advise LATEX where the graphics are located.

To use the `graphicx` package, include the following line in your Overleaf document preamble:

```
\usepackage{graphicx}
```

In our example the command `\graphicspath{{images/}}` informs LATEX that images are kept in a folder named `images`, which is contained in the current directory:



## Other useful links

---

Overleaf templates:

<https://www.overleaf.com/gallery>

List of Greek letters and math symbols

[https://www.overleaf.com/learn/latex/List\\_of\\_Greek\\_letters\\_and\\_math\\_symbols](https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols)

Use Tikz to draw graphs:

<https://www.mathcha.io/>