

# Get T<sub>E</sub>Xnical

## The Very Basic T<sub>E</sub>Xniques of L<sup>A</sup>T<sub>E</sub>X

Use L<sup>A</sup>T<sub>E</sub>X to make articles, books, formulas and even slideshows!

Zishan Rahman

King's College London  
UEL CDT Maker Club Takeover

August 2025

# Table of Contents

- 1 **Introductions**
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

# Who am I?



- PhD student at KCL (started February 2024)

# Who am I?



- PhD student at KCL (started February 2024)
- Previously completed Computer Science BSc (First Class Honours!) at KCL in 2023 (started September 2020)

# Who am I?



- PhD student at KCL (started February 2024)
- Previously completed Computer Science BSc (First Class Honours!) at KCL in 2023 (started September 2020)
- Learnt  $\text{\LaTeX}$  in 2022-2023 to write my Bachelors thesis

# Who am I?



- PhD student at KCL (started February 2024)
- Previously completed Computer Science BSc (First Class Honours!) at KCL in 2023 (started September 2020)
- Learnt  $\text{\LaTeX}$  in 2022-2023 to write my Bachelors thesis
- Have since used  $\text{\LaTeX}$  to write reports, papers, CVs and presentations

# What is $\text{\LaTeX}$ ?

$\text{\LaTeX}$  is a *typesetting* system (initially based on  $\text{\TeX}$ ).

# What is $\text{\LaTeX}$ ?

$\text{\LaTeX}$  is a *typesetting* system (initially based on  $\text{\TeX}$ ). This means it defines how text is laid out on a page or slide.



# What is $\text{\LaTeX}$ ?

$\text{\LaTeX}$  is a *typesetting* system (initially based on  $\text{\TeX}$ ). This means it defines how text is laid out on a page or slide. Unlike word processors (e.g. Microsoft Word, LibreOffice Writer), which use a “what-you-see-is-what-you-get” editing style, what you see in  $\text{\LaTeX}$  and what you get with  $\text{\LaTeX}$  typically end up looking quite different from each other.

# What is $\text{\LaTeX}$ ?

$\text{\LaTeX}$  is a *typesetting* system (initially based on  $\text{\TeX}$ ). This means it defines how text is laid out on a page or slide. Unlike word processors (e.g. Microsoft Word, LibreOffice Writer), which use a “what-you-see-is-what-you-get” editing style, what you see in  $\text{\LaTeX}$  and what you get with  $\text{\LaTeX}$  typically end up looking quite different from each other. You *could* call it some kind of “code”, but it isn’t a programming language.

# What is $\text{\LaTeX}$ ?

$\text{\LaTeX}$  is a *typesetting* system (initially based on  $\text{\TeX}$ ). This means it defines how text is laid out on a page or slide. Unlike word processors (e.g. Microsoft Word, LibreOffice Writer), which use a “what-you-see-is-what-you-get” editing style, what you see in  $\text{\LaTeX}$  and what you get with  $\text{\LaTeX}$  typically end up looking quite different from each other. You *could* call it some kind of “code”, but it isn’t a programming language. It simply *defines* how to produce the document with  $\text{\LaTeX}$ .

# What is L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X is a *typesetting* system (initially based on T<sub>E</sub>X). This means it defines how text is laid out on a page or slide. Unlike word processors (e.g. Microsoft Word, LibreOffice Writer), which use a “what-you-see-is-what-you-get” editing style, what you see in L<sup>A</sup>T<sub>E</sub>X and what you get with L<sup>A</sup>T<sub>E</sub>X typically end up looking quite different from each other. You *could* call it some kind of “code”, but it isn’t a programming language. It simply *defines* how to produce the document with L<sup>A</sup>T<sub>E</sub>X. The L<sup>A</sup>T<sub>E</sub>X compiler *produces* the document for you. You make and save your changes, then you run the compiler each time to get an updated version of your document.

# What is L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X is a *typesetting* system (initially based on T<sub>E</sub>X). This means it defines how text is laid out on a page or slide. Unlike word processors (e.g. Microsoft Word, LibreOffice Writer), which use a “what-you-see-is-what-you-get” editing style, what you see in L<sup>A</sup>T<sub>E</sub>X and what you get with L<sup>A</sup>T<sub>E</sub>X typically end up looking quite different from each other. You *could* call it some kind of “code”, but it isn’t a programming language. It simply *defines* how to produce the document with L<sup>A</sup>T<sub>E</sub>X. The L<sup>A</sup>T<sub>E</sub>X compiler *produces* the document for you. You make and save your changes, then you run the compiler each time to get an updated version of your document.

While L<sup>A</sup>T<sub>E</sub>X sees greater use in academia, particularly with conference and journal papers (it’s especially good for mathematical formulae), it’s so vast and flexible that it can effectively be used for most documents, whether you’re in academia or not, including books, CVs and even presentations!

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdf1latex`)!



# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations
- Nicer to type out commands mathematical and other symbols than to have to fetch their Unicode representations and/or ALT codes (on Windows).

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations
- Nicer to type out commands mathematical and other symbols than to have to fetch their Unicode representations and/or ALT codes (on Windows).
- (Leading on from above) Handles mathematical formulae and tables cleanly in compiled output.

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations
- Nicer to type out commands mathematical and other symbols than to have to fetch their Unicode representations and/or ALT codes (on Windows).
- (Leading on from above) Handles mathematical formulae and tables cleanly in compiled output.
- **$\text{\LaTeX}$  neatly separates the content of the document from how it's formatted**

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations
- Nicer to type out commands mathematical and other symbols than to have to fetch their Unicode representations and/or ALT codes (on Windows).
- (Leading on from above) Handles mathematical formulae and tables cleanly in compiled output.
- **$\text{\LaTeX}$  neatly separates the content of the document from how it's formatted**
- Easier to handle bibliographies and citations using  $\text{BibTeX}$

# Why use $\text{\LaTeX}$ ?

- **Multiple reasons!**
- Produces high-quality, professional-looking output when compiled
- Compiles straight to PDF (using `pdflatex`)!
- Can be easier and faster to “code” in certain situations
- Nicer to type out commands mathematical and other symbols than to have to fetch their Unicode representations and/or ALT codes (on Windows).
- (Leading on from above) Handles mathematical formulae and tables cleanly in compiled output.
- **$\text{\LaTeX}$  neatly separates the content of the document from how it's formatted**
- Easier to handle bibliographies and citations using  $\text{BibTeX}$
- Can handle multiple languages (including right-to-left written ones such as Arabic)

# What we will cover

- 1 Setting up an Overleaf account

# What we will cover

- 1 Setting up an Overleaf account
- 2 Write a basic  $\text{\LaTeX}$  chapter



# What we will cover

- 1 Setting up an Overleaf account
- 2 Write a basic  $\text{\LaTeX}$  chapter
- 3 Lists

# What we will cover

- ① Setting up an Overleaf account
- ② Write a basic  $\text{\LaTeX}$  chapter
- ③ Lists
- ④ Figures for adding images to your documents

# What we will cover

- ① Setting up an Overleaf account
- ② Write a basic  $\text{\LaTeX}$  chapter
- ③ Lists
- ④ Figures for adding images to your documents
- ⑤ Math(s) mode

# What we will cover

- 1 Setting up an Overleaf account
- 2 Write a basic  $\text{\LaTeX}$  chapter
- 3 Lists
- 4 Figures for adding images to your documents
- 5 Math(s) mode
- 6 Special characters

# What we will cover

- 1 Setting up an Overleaf account
- 2 Write a basic  $\text{\LaTeX}$  chapter
- 3 Lists
- 4 Figures for adding images to your documents
- 5 Math(s) mode
- 6 Special characters
- 7 Little bit of Bib $\text{\TeX}$

# What we will cover

- 1 Setting up an Overleaf account
- 2 Write a basic  $\text{\LaTeX}$  chapter
- 3 Lists
- 4 Figures for adding images to your documents
- 5 Math(s) mode
- 6 Special characters
- 7 Little bit of Bib $\text{\TeX}$
- 8 Beamer for presentations

# What we will **not** cover

## 1 Graphs

# What we will **not** cover

- 1 Graphs
- 2 Other complex things



# What we will **not** cover

- ① Graphs
- ② Other complex things
- ③ BibT<sub>E</sub>X in *depth*

# What we will **not** cover

- ① Graphs
- ② Other complex things
- ③ BibT<sub>E</sub>X in *depth*
- ④ L<sup>A</sup>T<sub>E</sub>X for CVs

# What we will **not** cover

- 1 Graphs
- 2 Other complex things
- 3 BibT<sub>E</sub>X in *depth*
- 4 L<sup>A</sup>T<sub>E</sub>X for CVs
- 5 Changing/Configuring fonts and L<sup>A</sup>T<sub>E</sub>X document styles

# What we will **not** cover

- 1 Graphs
- 2 Other complex things
- 3 BibTeX in *depth*
- 4 L<sup>A</sup>T<sub>E</sub>X for CVs
- 5 Changing/Configuring fonts and L<sup>A</sup>T<sub>E</sub>X document styles
- 6 Tables, as they are too complex to type out in L<sup>A</sup>T<sub>E</sub>X; I will show you a tool you can use to design your own tables and convert to L<sup>A</sup>T<sub>E</sub>X

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

# About Overleaf and Editing $\text{\LaTeX}$

To write and compile a  $\text{\LaTeX}$  document, you need:

# About Overleaf and Editing $\text{\LaTeX}$

To write and compile a  $\text{\LaTeX}$  document, you need:

- 1 A  $\text{\LaTeX}$  compiler.

# About Overleaf and Editing $\text{\LaTeX}$

To write and compile a  $\text{\LaTeX}$  document, you need:

- 1 A  $\text{\LaTeX}$  compiler.
- 2 A  $\text{\LaTeX}$  editor.



# About Overleaf and Editing $\text{\LaTeX}$

To write and compile a  $\text{\LaTeX}$  document, you need:

- 1 A  $\text{\LaTeX}$  compiler.
- 2 A  $\text{\LaTeX}$  editor.

Overleaf is a remote/online  $\text{\LaTeX}$  editing suite, complete with a distribution (a compiler and some packages).

# About Overleaf and Editing $\text{\LaTeX}$

To write and compile a  $\text{\LaTeX}$  document, you need:

- 1 A  $\text{\LaTeX}$  compiler.
- 2 A  $\text{\LaTeX}$  editor.

Overleaf is a remote/online  $\text{\LaTeX}$  editing suite, complete with a distribution (a compiler and some packages).  $\text{\LaTeX}$  compilers are usually downloaded as part of a larger distribution with many packages (i.e.  $\text{\TeX}$  Live can be over 5GB in size), so using Overleaf takes the burden off of having to download all of that for the sake of this workshop.

# Setting up an Overleaf Account

Overleaf has an intuitive user interface that should allow you to set up a new LaTeX project and start working right away!

# Setting up an Overleaf Account

Overleaf has an intuitive user interface that should allow you to set up a new LaTeX project and start working right away!

Before you start working with Overleaf, though, you need to set up an account!

# Setting up an Overleaf Account

Overleaf has an intuitive user interface that should allow you to set up a new LaTeX project and start working right away!

Before you start working with Overleaf, though, you need to set up an account!

To do so, you can either:

- Create an account with your own username, email address and password.

# Setting up an Overleaf Account

Overleaf has an intuitive user interface that should allow you to set up a new LaTeX project and start working right away!

Before you start working with Overleaf, though, you need to set up an account!

To do so, you can either:

- Create an account with your own username, email address and password.
- Log in with your Google account.

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a `class` attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.



# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a `class` attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.

The `class` can be, for example:

# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a `class` attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.

The class can be, for example:

- `article`

# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a `class` attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.

The class can be, for example:

- `article`
- `report`

# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a class attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.

The class can be, for example:

- `article`
- `report`
- `book`

# Setting up a $\text{\LaTeX}$ document

Each  $\text{\LaTeX}$  document (file extension `.tex`) has a class attached to it. The class declaration is written like so:

```
\documentclass{class}
```

This tells the  $\text{\LaTeX}$  compiler to format the document in a certain way.

The class can be, for example:

- `article`
- `report`
- `book`
- `beamer`, for presentations

# A basic L<sup>A</sup>T<sub>E</sub>X document

We'll be using `article` for this first example, like so:

```
\documentclass{article}
```

# A basic L<sup>A</sup>T<sub>E</sub>X document

We'll be using `article` for this first example, like so:

```
\documentclass{article}
```

You also need a dedicated space to write your document within.

# A basic L<sup>A</sup>T<sub>E</sub>X document

We'll be using `article` for this first example, like so:

```
\documentclass{article}
```

You also need a dedicated space to write your document within. We set it like so (the commands *before* all of this is called the **preamble**):

```
\begin{document}
```

```
\end{document}
```



# A basic L<sup>A</sup>T<sub>E</sub>X document

We'll be using `article` for this first example, like so:

```
\documentclass{article}
```

You also need a dedicated space to write your document within. We set it like so (the commands *before* all of this is called the **preamble**):

```
\begin{document}
```

```
\end{document}
```

We can then write anything we want in that document:

```
\documentclass{article}
```

```
\begin{document}
```

The quick brown fox jumps over the lazy dog.

```
\end{document}
```

# Some basic formatting

I won't go over *everything* to do with formatting text in  $\text{\LaTeX}$ , but I *will* go over some of the common formatting options you may use:

<code>\textbf{text}</code>	→	<b>text</b>
<code>\textit{text}</code>	→	<i>text</i>
<code>\underline{text}</code>	→	<u>text</u>
<code>\sout{text}</code>	→	<del>text</del>
<code>\TeX{}</code>	→	$\text{\TeX}$
<code>\LaTeX{}</code>	→	$\text{\LaTeX}$
<code>\newline{}</code>	→	Line break
<code>\\</code>	→	Line break
<code>4\textsuperscript{th}</code>	→	4 <sup>th</sup>
<code>4\$\text{\textsuperscript{th}}\$</code>	→	4 <sup>th</sup>

More on those dollar signs in that last one later!

A useful thing you can do in  $\text{\LaTeX}$  is write *comments*.

A useful thing you can do in  $\text{\LaTeX}$  is write *comments*.

These don't get passed into the compiler, so you can use them to write, for example, explanations for what certain commands do that you don't want to include in your main text.

A useful thing you can do in  $\text{\LaTeX}$  is write *comments*.

These don't get passed into the compiler, so you can use them to write, for example, explanations for what certain commands do that you don't want to include in your main text.

Comments begin with a percentage sign (%) and end at the very end of the line where they began.

A useful thing you can do in  $\text{\LaTeX}$  is write *comments*.

These don't get passed into the compiler, so you can use them to write, for example, explanations for what certain commands do that you don't want to include in your main text.

Comments begin with a percentage sign (%) and end at the very end of the line where they began.

For example:

```
\textbf{Bold} % This command bolds the text.
```

# Escaping special symbols

Want to print out a symbol that  $\text{\LaTeX}$  uses for its syntax? Sure, just escape it!

# Escaping special symbols

Want to print out a symbol that  $\text{\LaTeX}$  uses for it's syntax? Sure, just escape it!

For example:

<code>\textbackslash</code>	→	<code>\</code>
<code>\%</code>	→	<code>%</code>
<code>\&amp;</code>	→	<code>&amp;</code>
<code>\\$</code>	→	<code>\$</code>
<code>\#</code>	→	<code>#</code>



# Setting your own custom commands

Using `\newcommand{cmd}{def}` in your preamble, you can define your own commands, for anything from easy shortcuts to larger stuff.

# Setting your own custom commands

Using `\newcommand{cmd}{def}` in your preamble, you can define your own commands, for anything from easy shortcuts to larger stuff.

For example, here's a command this set of slides uses to create big gaps in-between paragraphs:

```
\newcommand{\biggap}{\newline{}\newline{}}
```

# Setting your own custom commands

Using `\newcommand{cmd}{def}` in your preamble, you can define your own commands, for anything from easy shortcuts to larger stuff.

For example, here's a command this set of slides uses to create big gaps in-between paragraphs:

```
\newcommand{\biggap}{\newline{}\newline{}}
```

Your new command can also take in data that gets used by it:

```
\newcommand{command}[number_of_arguments]{definition}
```

For example:

```
\newcommand{\namething}[1]{My name is {#1}!}
```

```
\namething{Zishan} → My name is Zishan!
```

# Headings

You can also divide your text into chapters, sections, subsections and subsubsections, and  $\text{\LaTeX}$  will format them accordingly:

```
\chapter{This is a chapter}  
\section{This is a section}  
\subsection{This is a subsection}  
\subsubsection{This is a subsubsection}
```

# Creative writing and $\text{\LaTeX}$ testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*.

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time.



# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

Once you're done, you'll feed back to me on:

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

Once you're done, you'll feed back to me on:

- How the L<sup>A</sup>T<sub>E</sub>X writing experience was

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

Once you're done, you'll feed back to me on:

- How the L<sup>A</sup>T<sub>E</sub>X writing experience was
- Any quirks you saw in the formatting and/or compilation of your L<sup>A</sup>T<sub>E</sub>X document

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

Once you're done, you'll feed back to me on:

- How the L<sup>A</sup>T<sub>E</sub>X writing experience was
- Any quirks you saw in the formatting and/or compilation of your L<sup>A</sup>T<sub>E</sub>X document
- Any compilation errors and/or warnings you got

# Creative writing and L<sup>A</sup>T<sub>E</sub>X testing exercise

Now that you know how to set up a basic LaTeX document, it's time to start writing!

Spend the next few minutes writing a number of sentences on *anything*. It can be a story, a journal, a recount, fiction, non-fiction, you name it! Just type it and compile your document regularly (Overleaf will recompile it every time you save it)!

While you type and compile, each time, make notes of any quirks and compilation errors you see and/or get during this time. You *can* try and fix them if you want, but it's neither compulsory nor do I require you to understand them at this time.

Once you're done, you'll feed back to me on:

- How the L<sup>A</sup>T<sub>E</sub>X writing experience was
- Any quirks you saw in the formatting and/or compilation of your L<sup>A</sup>T<sub>E</sub>X document
- Any compilation errors and/or warnings you got
- *Optionally*, what you wrote about

So... how did you get on?

So... how did you get on?

- How was it like to write and compile a  $\text{\LaTeX}$  document for the first time?



So... how did you get on?

- How was it like to write and compile a  $\text{\LaTeX}$  document for the first time?
- Did the  $\text{\LaTeX}$  compiler format your document nicely?

So... how did you get on?

- How was it like to write and compile a L<sup>A</sup>T<sub>E</sub>X document for the first time?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format your document nicely?
- Were there any *quirks* in the formatting?

So... how did you get on?

- How was it like to write and compile a L<sup>A</sup>T<sub>E</sub>X document for the first time?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format your document nicely?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?

So... how did you get on?

- How was it like to write and compile a L<sup>A</sup>T<sub>E</sub>X document for the first time?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format your document nicely?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

So... how did you get on?

- How was it like to write and compile a L<sup>A</sup>T<sub>E</sub>X document for the first time?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format your document nicely?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

# Beware of quirks!

Unfortunately, just like word processors,  $\text{\LaTeX}$  has its own quirks. Be aware that this can happen, but don't let it put you off of using it.

# Beware of quirks!

Unfortunately, just like word processors,  $\text{\LaTeX}$  has its own quirks. Be aware that this can happen, but don't let it put you off of using it.

I'll show you one quirk right now

# Beware of quirks!

Unfortunately, just like word processors,  $\text{\LaTeX}$  has its own quirks. Be aware that this can happen, but don't let it put you off of using it.

I'll show you one quirk right now:

$$\begin{array}{ccc} \text{“} \text{“} & \rightarrow & \text{“} \\ \text{”} & \rightarrow & \text{”} \end{array}$$

(`“` = two backticks)

That's why you're quotes ended up like `”this”` and not `“this”`. Keep this in mind as you continue working with  $\text{\LaTeX}$ .



# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images**
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

To add a bullet point list into your document, you set up a new `itemize` environment and add `\items` to it

To add a bullet point list into your document, you set up a new `itemize` environment and add `\items` to it, like so:

```
\begin{itemize}
    \item I am an item in a list!
    \item I am another item in the same list!
\end{itemize}
```

To add a bullet point list into your document, you set up a new `itemize` environment and add `\items` to it, like so:

```
\begin{itemize}
    \item I am an item in a list!
    \item I am another item in the same list!
\end{itemize}
```

This will render the following list:

- I am an item in a list!
- I am another item in the same list!

# List exercise

Let's put this to action!

# List exercise

Let's put this to action!

Spend the next minute or so writing a list of things you like doing.

# List exercise

Let's put this to action!

Spend the next minute or so writing a list of things you like doing. Hopefully, no weird stuff should happen (i.e. compilation warnings, errors etc.), but if anything weird *does* happen, make a note of it and get back to me afterwards.

# Numbered lists

To have your lists *numbered* instead of bulleted, replace `itemize` in your list environment with `enumerate` (in both your `begin` and `end` declarations)



# Numbered lists

To have your lists *numbered* instead of bulleted, replace `itemize` in your list environment with `enumerate` (in both your `begin` and `end` declarations), like so:

```
\begin{enumerate}  
    \item I am the first item in the list!  
    \item I am the second item in the list!  
\end{enumerate}
```

# Numbered lists

To have your lists *numbered* instead of bulleted, replace `itemize` in your list environment with `enumerate` (in both your `begin` and `end` declarations), like so:

```
\begin{enumerate}  
    \item I am the first item in the list!  
    \item I am the second item in the list!  
\end{enumerate}
```

This will render the following list:

- 1 I am the first item in the list!
- 2 I am the second item in the list!

# Numbered list exercise

Now, let's put *this* into action as well!

# Numbered list exercise

Now, let's put *this* into action as well!

Spend the next few minutes writing either:

- A process described in order (i.e. a recipe, steps for doing something etc.)
- A ranking of anything that won't cause offence (from, e.g., best to worst, tallest to highest etc.)

# Numbered list exercise

Now, let's put *this* into action as well!

Spend the next few minutes writing either:

- A process described in order (i.e. a recipe, steps for doing something etc.)
- A ranking of anything that won't cause offence (from, e.g., best to worst, tallest to highest etc.)

Again, if anything weird happens when you try to compile it, make a note of it and get back to me afterwards. We will be feeding back on both this and the previous list exercise in the next slide.

# Feedback on list and numbered list exercises

So... how did you get on?

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?



# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?
- What changed about *how* you wrote?

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?
- What changed about *how* you wrote?
- Were there any *quirks* in the formatting?

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?
- What changed about *how* you wrote?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?
- What changed about *how* you wrote?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

# Feedback on list and numbered list exercises

So... how did you get on?

- Did your  $\text{\LaTeX}$  experience change at all?
- Did the  $\text{\LaTeX}$  compiler format your lists nicely?
- What changed about *how* you wrote?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

# (Very) Basic figures

You start and end a figure like so (notice a pattern here):

```
\begin{figure}[h]  
    \textbf{Stuff$\ldots$}  
\end{figure}
```

# (Very) Basic figures

You start and end a figure like so (notice a pattern here):

```
\begin{figure}[h]  
    \textbf{Stuff$\ldots$}  
\end{figure}
```

Which produces:

**Stuff...**

# (Very) Basic figures

You start and end a figure like so (notice a pattern here):

```
\begin{figure}[h]
    \textbf{Stuff$\ldots$}
\end{figure}
```

Which produces:

**Stuff...**

A figure can be centred using the `\centering` command within the figure itself:

```
\begin{figure}[h]
    \centering
    \textbf{Stuff$\ldots$}
\end{figure}
```

**Stuff...**



# Figure placement

Notice that  $h$ ?

# Figure placement

Notice that `\h?`  $\text{\LaTeX}$  can place your figure within your document depending on how you want it:

- `[h]` → As it was placed in the `.tex` file
- `[t]` → Top of page
- `[b]` → Bottom of page
- `[p]` → A separate page for figures (and tables)  
Try placing it as it was placed in the `.tex` file,  
otherwise place at bottom of page. The above four  
figure placement options can likewise be combined  
in multiple ways. `[hb]` is an example of just one way.

# Captions on figures

A figure can also have a `\caption{text}`:

```
\begin{figure}[h]
  \centering
  \textbf{Stuff$\ldots$}
  \caption{This is stuff!}
\end{figure}
```

**Stuff...**

Figure: This is stuff!

# Labels for figures

You can even use a `\label{fig:label}` to easily refer to it using `\ref{fig:label}` (like this: [2](#)). Make sure the `\label{}` is placed *after* the `\caption{}`.

```
\begin{figure}[h]
    \centering
    \textbf{Stuff$\ldots$}
    \caption{This is stuff!}
    \label{fig:stuff}
\end{figure}
```

**Stuff...**

**Figure:** This is stuff!

In articles, you can use `\autoref{fig:label}` to have the text for that reference render to, for example, “Figure 3”.

# One more thing on basic figures

There's just *one* more thing that a figure can take great advantage of...

# One more thing on basic figures

There's just *one* more thing that a figure can take great advantage of...

...**Images!**

# One more thing on basic figures

There's just *one* more thing that a figure can take great advantage of...

...**Images!**

To replace our placeholder text (**Stuff...**) in our figure with an image, we will use the `graphicsx` package.

# A note about packages

$\text{\LaTeX}$ , by itself, is very “bare bones”, so for things like images, hyperlinks and other things that are contained in many documents, we often import “packages” that give us additional commands to use.



# A note about packages

$\LaTeX$ , by itself, is very “bare bones”, so for things like images, hyperlinks and other things that are contained in many documents, we often import “packages” that give us additional commands to use.

Importing a package in your  $\LaTeX$  document is as simple as writing `\usepackage{package}` before your `\begin{document}` statement, and you usually do **not** need to install the package separately, as it will be included in your  $\LaTeX$  distribution (such as  $\TeX$  Live and the one that Overleaf uses). That’s *why* they’re so huge!

# A note about packages

$\text{\LaTeX}$ , by itself, is very “bare bones”, so for things like images, hyperlinks and other things that are contained in many documents, we often import “packages” that give us additional commands to use.

Importing a package in your  $\text{\LaTeX}$  document is as simple as writing `\usepackage{package}` before your `\begin{document}` statement, and you usually do **not** need to install the package separately, as it will be included in your  $\text{\LaTeX}$  distribution (such as  $\text{\TeX}$  Live and the one that Overleaf uses). That’s *why* they’re so huge!

We’ll be using a number of different packages to add things to our  $\text{\LaTeX}$  documents, so pay attention and keep your eyes on them!

# Adding an image to our figure

First, add the following statement before your `\begin{document}` declaration: `\usepackage{graphicsx}`.

# Adding an image to our figure

First, add the following statement before your `\begin{document}` declaration: `\usepackage{graphicsx}`.

Then, replace the placeholder text with your image (either use one of your own or download one from the web), like so:

`\includegraphics{your_image}`. You don't need to explicitly define its file type; `graphicsx` accepts most common image types, i.e. PNG, JPG et cetera.

# Adding an image to our figure

First, add the following statement before your `\begin{document}` declaration: `\usepackage{graphicsx}`.

Then, replace the placeholder text with your image (either use one of your own or download one from the web), like so:

`\includegraphics{your_image}`. You don't need to explicitly define its file type; `graphicsx` accepts most common image types, i.e. PNG, JPG et cetera.

For example, the image placement in Figure 3 was done like so:

```
\begin{figure}[h]
    \includegraphics[scale=0.13]{me}
    \centering
    \label{fig:me}
\end{figure}
```

# Scaling images

Noticed the `[scale=0.13]`?

Noticed the `[scale=0.13]`? Some  $\text{\LaTeX}$  commands come with additional configuration options that can be added within a pair of square brackets before the curly ones.

# Scaling images

Noticed the `[scale=0.13]`? Some  $\text{\LaTeX}$  commands come with additional configuration options that can be added within a pair of square brackets before the curly ones. The image file I used for Figure 3 is too large to be added to the slide without taking over everything, so I used the `scale` argument to control its size (it takes a multiplier value which it applies to the image size).



# Scaling images

Noticed the `[scale=0.13]`? Some  $\text{\LaTeX}$  commands come with additional configuration options that can be added within a pair of square brackets before the curly ones. The image file I used for Figure 3 is too large to be added to the slide without taking over everything, so I used the `scale` argument to control its size (it takes a multiplier value which it applies to the image size). If *your* image is too large, I'd advise you do the same!

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments**
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

# Math mode

Remember those dollar signs?

# Math mode

Remember those dollar signs? Those dollar signs put  $\text{\LaTeX}$  in **Math mode** for the things within them!

# Math mode

Remember those dollar signs? Those dollar signs put  $\text{\LaTeX}$  in **Math mode** for the things within them!

Math mode is for rendering simple and complex mathematical formulae.

# Math mode

Remember those dollar signs? Those dollar signs put  $\text{\LaTeX}$  in **Math mode** for the things within them!

Math mode is for rendering simple and complex mathematical formulae. For example,  $ax^2 + bx + c = 0$  renders to  $ax^2 + bx + c = 0$ . You can use single dollar signs to easily place formulae in math mode “inline” within your paragraphs.

# Math mode

Remember those dollar signs? Those dollar signs put  $\text{\LaTeX}$  in **Math mode** for the things within them!

Math mode is for rendering simple and complex mathematical formulae. For example,  $ax^2 + bx + c = 0$  renders to  $ax^2 + bx + c = 0$ . You can use single dollar signs to easily place formulae in math mode “inline” within your paragraphs.

To create *dedicated* math mode placements, you can use double dollar signs ( $\text{\$}$  $\text{\$}$ ) at both ends.

For example,  $\text{\$}$  $\text{\$}$  $ax^2 + bx + c = 0$  $\text{\$}$  $\text{\$}$  produces the following:

$$ax^2 + bx + c = 0$$

You can also define a dedicated maths environment as you'd define (e.g.) a document, which will require the `mathtools` package:

```
\usepackage{mathtools}
...
\begin{math}
    ax^{2} + bx + c = 0
\end{math}
```



You can also define a dedicated maths environment as you'd define (e.g.) a document, which will require the `mathtools` package:

```
\usepackage{mathtools}
...
\begin{math}
    ax^{2} + bx + c = 0
\end{math}
```

This will produce:

$$ax^2 + bx + c = 0$$

# Special characters within math mode

L<sup>A</sup>T<sub>E</sub>X allows you to use special symbols using dedicated commands.

# Special characters within math mode

$\text{\LaTeX}$  allows you to use special symbols using dedicated commands. For example, instead of having to fetch out the “therefore” symbol from the web and copy-pasting it into your document (or entering a Unicode value), you can type in  $\text{\$}\backslash\text{therefore}\text{\$}$  and  $\text{\LaTeX}$  will render it easily:  $\therefore$ .

# Special characters within math mode

L<sup>A</sup>T<sub>E</sub>X allows you to use special symbols using dedicated commands. For example, instead of having to fetch out the “therefore” symbol from the web and copy-pasting it into your document (or entering a Unicode value), you can type in  `$\therefore$`  and L<sup>A</sup>T<sub>E</sub>X will render it easily:  $\therefore$ .

You can then use it within your formulae. For example:

```
\begin{math}  
      8 + 9 = 17\newline  
      \therefore 17 - 9 = 8  
\end{math}
```

$$8 + 9 = 17$$

$$\therefore 17 - 9 = 8$$

# Special characters within math mode

$\LaTeX$  allows you to use special symbols using dedicated commands. For example, instead of having to fetch out the “therefore” symbol from the web and copy-pasting it into your document (or entering a Unicode value), you can type in `\therefore` and  $\LaTeX$  will render it easily:  $\therefore$ .

You can then use it within your formulae. For example:

```
\begin{math}
      8 + 9 = 17\newline
      \therefore 17 - 9 = 8
\end{math}
```

$8 + 9 = 17$   
 $\therefore 17 - 9 = 8$

Some symbols can also be entered in *normal* mode (for typing text). For example, `\copyright` renders to ©, in both normal and math mode.

# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters?

# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters? Like

`this?`

# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters? Like

`this?`

Sometimes, you want to print certain characters and words *as* they are without any of L<sup>A</sup>T<sub>E</sub>X's formatting.



# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters? Like

`t h i s ?`

Sometimes, you want to print certain characters and words *as* they are without any of L<sup>A</sup>T<sub>E</sub>X's formatting. To do that *inline*, use the `\verb|` command (note the vertical bars we use instead of curly braces here; use math mode to type in the bars). For *dedicated* environments, use `\begin{verbatim}` and remember to end it.

# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters? Like

`t h i s ?`

Sometimes, you want to print certain characters and words *as* they are without any of  $\text{\LaTeX}$ 's formatting. To do that *inline*, use the `\verb|` command (note the vertical bars we use instead of curly braces here; use math mode to type in the bars). For *dedicated* environments, use `\begin{verbatim}` and remember to end it. Verbatim environments are also useful for code samples, for which we need a dedicated `lstlisting` environment from the `listings` package (which you create just as you'd create documents, dedicated math environments et cetera).

# Verbatim Environments

Notice the different styles of text throughout this presentation for certain words and characters? Like

this?

Sometimes, you want to print certain characters and words *as* they are without any of L<sup>A</sup>T<sub>E</sub>X's formatting. To do that *inline*, use the `\verb|` command (note the vertical bars we use instead of curly braces here; use math mode to type in the bars). For *dedicated* environments, use `\begin{verbatim}` and remember to end it. Verbatim environments are also useful for code samples, for which we need a dedicated `lstlisting` environment from the `listings` package (which you create just as you'd create documents, dedicated math environments et cetera). Syntax highlighting for most programming languages is supported here (i.e. `\begin{lstlisting}[language=Python]`), with extensive configuration and customisability options for languages not supported out-of-the-box.

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$**
- 7 Beamer for Presentations
- 8 And that's it!

# A *tiny* bit on BibT<sub>E</sub>X

See those citations on Wikipedia articles?

# A *tiny* bit on BibT<sub>E</sub>X

See those citations on Wikipedia articles? You can do the same thing in L<sup>A</sup>T<sub>E</sub>X using BibT<sub>E</sub>X!

# A *tiny* bit on BibT<sub>E</sub>X

See those citations on Wikipedia articles? You can do the same thing in L<sup>A</sup>T<sub>E</sub>X using BibT<sub>E</sub>X!

BibT<sub>E</sub>X uses its own file with extension `.bib`. Create one such file in your Overleaf project (ideally, for this workshop, in the same folder as your main `.tex` file). Call it, for example, `references.bib`.

# A *tiny* bit on BibT<sub>E</sub>X

See those citations on Wikipedia articles? You can do the same thing in L<sup>A</sup>T<sub>E</sub>X using BibT<sub>E</sub>X!

BibT<sub>E</sub>X uses its own file with extension `.bib`. Create one such file in your Overleaf project (ideally, for this workshop, in the same folder as your main `.tex` file). Call it, for example, `references.bib`.

Then, import the `natbib` package (`biber` is another citation package; we'll use `natbib` for this workshop):

```
\usepackage[square,sort,comma,numbers]{natbib}.
```



# A *tiny* bit on BibT<sub>E</sub>X

See those citations on Wikipedia articles? You can do the same thing in L<sup>A</sup>T<sub>E</sub>X using BibT<sub>E</sub>X!

BibT<sub>E</sub>X uses its own file with extension `.bib`. Create one such file in your Overleaf project (ideally, for this workshop, in the same folder as your main `.tex` file). Call it, for example, `references.bib`.

Then, import the `natbib` package (`biber` is another citation package; we'll use `natbib` for this workshop):

```
\usepackage[square,sort,comma,numbers]{natbib}.
```

Before you end your document, point BibT<sub>E</sub>X to your file. After your text, it will generate a bibliography containing your citations:

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

Types of BibT<sub>E</sub>X entries include `article` (for journals), `book`, `manual` and `inproceedings` (for conference proceedings). There's also a catch-all `misc` type (used in the next slide) for entries that don't quite fit into the standard categories.

# Your bibliography

BibT<sub>E</sub>X files consist of one or more entries in *this* form:

```
@misc{entryid ,  
  year = {2025},  
  title = {{Get TeXNical}},  
  author = {{Rahman, Zishan}},  
  howpublished = {\url{https://www.example.com}}  
}
```

The `\url{}` command does as explained (when printing the link, it embeds the URL so it can be clicked on and opened). For it to work properly, add `\usepackage{hyperref}` to your preamble.

# Your bibliography

BibT<sub>E</sub>X files consist of one or more entries in *this* form:

```
@misc{entryid ,  
  year = {2025},  
  title = {{Get TeXNical}},  
  author = {{Rahman, Zishan}},  
  howpublished = {\url{https://www.example.com}}  
}
```

The `\url{}` command does as explained (when printing the link, it embeds the URL so it can be clicked on and opened). For it to work properly, add `\usepackage{hyperref}` to your preamble. To *cite* the resource, use the `\cite{id}` command (e.g. `\cite{entryid}`). A number enclosed with square brackets will appear right where you made your citation (like this[1]). Your bibliography at the end will show what you cited next to that number.

# Your bibliography

BibT<sub>E</sub>X files consist of one or more entries in *this* form:

```
@misc{entryid ,  
  year = {2025},  
  title = {{Get TeXNical}},  
  author = {{Rahman, Zishan}},  
  howpublished = {\url{https://www.example.com}}  
}
```

The `\url{}` command does as explained (when printing the link, it embeds the URL so it can be clicked on and opened). For it to work properly, add `\usepackage{hyperref}` to your preamble. To *cite* the resource, use the `\cite{id}` command (e.g. `\cite{entryid}`). A number enclosed with square brackets will appear right where you made your citation (like this[1]). Your bibliography at the end will show what you cited next to that number. Note that only references that *actually get cited* will show up in your L<sup>A</sup>T<sub>E</sub>X document's bibliography by default, so if you want them to show up, cite them.

# Title and Table of Contents

Every book needs a title, author, publication date and list of chapters. The title, author and date you define in your **preamble**, like so:

```
\title{Get \TeX{nical}}  
\author{Zishan Rahman}  
\date{2025}
```

# Title and Table of Contents

Every book needs a title, author, publication date and list of chapters. The title, author and date you define in your **preamble**, like so:

```
\title{Get \TeX{nical}}  
\author{Zishan Rahman}  
\date{2025}
```

As for *making* the title and table of contents show up on your L<sup>A</sup>T<sub>E</sub>X document, that couldn't be any simpler.

# Title and Table of Contents

Every book needs a title, author, publication date and list of chapters. The title, author and date you define in your **preamble**, like so:

```
\title{Get \TeX{nical}}  
\author{Zishan Rahman}  
\date{2025}
```

As for *making* the title and table of contents show up on your  $\text{\LaTeX}$  document, that couldn't be any simpler. As soon as you begin your document:

```
\maketitle  
\tableofcontents
```

There are similar commands for glossaries and indexes (which we're not covering how to make today): `\makeglossary` and `\makeindex`.

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!



# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it.

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it. It can be anything, it doesn't have to be serious. You don't even have to tell me what it is!

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it. It can be anything, it doesn't have to be serious. You don't even have to tell me what it is! Just remember to include at least two citations of some sort.

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it. It can be anything, it doesn't have to be serious. You don't even have to tell me what it is! Just remember to include at least two citations of some sort. You can cite anything from a news article to a YouTube video to even a meme (if you want)!

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it. It can be anything, it doesn't have to be serious. You don't even have to tell me what it is! Just remember to include at least two citations of some sort. You can cite anything from a news article to a YouTube video to even a meme (if you want)!

Research articles are typically quite long and wordy, but for this exercise, I only need you to write a paragraph or two.

# Exercise: Your first (very short) research article

Time to do some (very basic and not very *guided*) research!

Write about a topic that interests you so much you want to research into it. It can be anything, it doesn't have to be serious. You don't even have to tell me what it is! Just remember to include at least two citations of some sort. You can cite anything from a news article to a YouTube video to even a meme (if you want)!

Research articles are typically quite long and wordy, but for this exercise, I only need you to write a paragraph or two. Although your “article” will be short, give it a title and put yourself down as an author. A table of contents wouldn't hurt either.

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing Bib $\text{T}_\text{E}$ X citations?

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?



# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?
- What changed about *how* you wrote, now that you were writing a research “article” (well, *paragraph(s)*)?

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?
- What changed about *how* you wrote, now that you were writing a research “article” (well, *paragraph(s)*)?
- Were there any *quirks* in the formatting?

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?
- What changed about *how* you wrote, now that you were writing a research “article” (well, *paragraph(s)*)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?
- What changed about *how* you wrote, now that you were writing a research “article” (well, *paragraph(s)*)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don’t have to say if you don’t want to.

# Feedback on short research article exercise

So... how did you get on?

- How did you find making and citing BibT<sub>E</sub>X citations?
- Did the L<sup>A</sup>T<sub>E</sub>X compiler format them nicely?
- What changed about *how* you wrote, now that you were writing a research “article” (well, *paragraph(s)*)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don’t have to say if you don’t want to.

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations**
- 8 And that's it!

Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

In a new `.tex` file, set the document class to `beamer` at the very beginning of your preamble, then set the title, author and year as you did before (I'll show you later how to make a Beamer-specific table of contents).



Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

In a new `.tex` file, set the document class to `beamer` at the very beginning of your preamble, then set the title, author and year as you did before (I'll show you later how to make a Beamer-specific table of contents). Then, add the following to your document (within `\begin{document}` and `\end{document}`): `\frame{\titlepage}`. This will make a title slide for you.

Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

In a new `.tex` file, set the document class to `beamer` at the very beginning of your preamble, then set the title, author and year as you did before (I'll show you later how to make a Beamer-specific table of contents). Then, add the following to your document (within `\begin{document}` and `\end{document}`): `\frame{\titlepage}`. This will make a title slide for you. You might notice it looks a bit *plain*.

Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

In a new `.tex` file, set the document class to `beamer` at the very beginning of your preamble, then set the title, author and year as you did before (I'll show you later how to make a Beamer-specific table of contents). Then, add the following to your document (within `\begin{document}` and `\end{document}`): `\frame{\titlepage}`. This will make a title slide for you.

You might notice it looks a bit *plain*. That's why you can use one of Beamer's built-in *themes* to add a bit more style to your slides.

Beamer is a document class meant specifically for making slideshows, much like the ones you'd see on Microsoft PowerPoint and/or LibreOffice Impress (only with less animations and transitions, given that your  $\text{\LaTeX}$  slides will typically compile to PDF).

In a new `.tex` file, set the document class to `beamer` at the very beginning of your preamble, then set the title, author and year as you did before (I'll show you later how to make a Beamer-specific table of contents). Then, add the following to your document (within `\begin{document}` and `\end{document}`): `\frame{\titlepage}`. This will make a title slide for you.

You might notice it looks a bit *plain*. That's why you can use one of Beamer's built-in *themes* to add a bit more style to your slides. I use the "Madrid" theme, like so (in the preamble, after the document class declaration): `\usetheme{Madrid}`

# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any).

# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any). The slide count at the bottom refers to the number of *frames*, while the number of *overall* pages for the compiled PDF will consist of every frame and every pause in-between.

# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any). The slide count at the bottom refers to the number of *frames*, while the number of *overall* pages for the compiled PDF will consist of every frame and every pause in-between.

As I've demonstrated here, figures, math mode, verbatim environments and links can easily be embedded into your slide, and you'd do that just as you would in an article (for verbatim environments, start your frame like so: `\begin{frame}[fragile]`).

# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any). The slide count at the bottom refers to the number of *frames*, while the number of *overall* pages for the compiled PDF will consist of every frame and every pause in-between.

As I've demonstrated here, figures, math mode, verbatim environments and links can easily be embedded into your slide, and you'd do that just as you would in an article (for verbatim environments, start your frame like so: `\begin{frame}[fragile]`). You can even make citations and bibliographies on Beamer slides, though that's rarely done in practice and I'm not showing you how to do that here.



# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any). The slide count at the bottom refers to the number of *frames*, while the number of *overall* pages for the compiled PDF will consist of every frame and every pause in-between.

As I've demonstrated here, figures, math mode, verbatim environments and links can easily be embedded into your slide, and you'd do that just as you would in an article (for verbatim environments, start your frame like so: `\begin{frame}[fragile]`). You can even make citations and bibliographies on Beamer slides, though that's rarely done in practice and I'm not showing you how to do that here.

To add a pause to your Beamer frame couldn't be any simpler

# How Beamer Works

Within your singular document, Beamer works with multiple “frames”, each of which connote 1 slide, including any pauses in-between (if there are any). The slide count at the bottom refers to the number of *frames*, while the number of *overall* pages for the compiled PDF will consist of every frame and every pause in-between.

As I've demonstrated here, figures, math mode, verbatim environments and links can easily be embedded into your slide, and you'd do that just as you would in an article (for verbatim environments, start your frame like so: `\begin{frame}[fragile]`). You can even make citations and bibliographies on Beamer slides, though that's rarely done in practice and I'm not showing you how to do that here.

To add a pause to your Beamer frame couldn't be any simpler: `\pause`

# Your first Beamer slide (well, *frame*)

Set up your frame as described in the previous slide, then within that frame, put in:

```
\frametitle{My first frame}  
This is text in a frame! \pause  
I just paused my slide! \pause  
Did you know you could use \verb|\href{URL}{text}|  
to embed hyperlinks into text? \pause  
\href{https://www.example.com}{Like this!}
```

Don't forget to start your frame with `\begin{frame}[fragile]` so you don't get compilation errors when trying to use `\verb!`

The results are in the next slide.

# My first frame

This is text in a frame!

# My first frame

This is text in a frame! I just paused my slide!

# My first frame

This is text in a frame! I just paused my slide! Did you know you could use `\href{URL}{text}` to embed hyperlinks into text?

# My first frame

This is text in a frame! I just paused my slide! Did you know you could use `\href{URL}{text}` to embed hyperlinks into text? [Like this!](#)

# Tables of Contents in Beamer

Sections in your slideshow can be defined just as you would do in articles, books et cetera.

```
\section{title}
```

To create a table of contents based on those sections, add *this* to your preamble (replace `frim` with `frame`; I had compilation errors with `frame` for this slide):

```
\AtBeginSection []  
{  
    \begin{frim}  
        \frametitle{Table of Contents}  
        \tableofcontents[currentsection]  
    \end{frim}  
}
```



# Final exercise for the day: Your first Beamer presentation

Time for one more exercise before we end this workshop!

# Final exercise for the day: Your first Beamer presentation

Time for one more exercise before we end this workshop!

Create a short slideshow with `beamer` on a topic of your choice!

# Final exercise for the day: Your first Beamer presentation

Time for one more exercise before we end this workshop!

Create a short slideshow with `beamer` on a topic of your choice!

Have at least one frame other than the title slide, but other than that, the slideshow shouldn't be *too* long.

# Final exercise for the day: Your first Beamer presentation

Time for one more exercise before we end this workshop!

Create a short slideshow with `beamer` on a topic of your choice!

Have at least one frame other than the title slide, but other than that, the slideshow shouldn't be *too* long.

Include at least **one** figure containing an image.

# Final exercise for the day: Your first Beamer presentation

Time for one more exercise before we end this workshop!

Create a short slideshow with `beamer` on a topic of your choice!

Have at least one frame other than the title slide, but other than that, the slideshow shouldn't be *too* long.

Include at least **one** figure containing an image.

*Optionally*, include at least one formula.

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?
- What changed about *how* you wrote the content in each slide (if anything changed at all)?



# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?
- What changed about *how* you wrote the content in each slide (if anything changed at all)?
- Were there any *quirks* in the formatting?

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?
- What changed about *how* you wrote the content in each slide (if anything changed at all)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?
- What changed about *how* you wrote the content in each slide (if anything changed at all)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

# Feedback on Beamer presentation exercise

So... how did you get on?

- How did you find making your presentation in  $\text{\LaTeX}$  compared to PowerPoint or Impress et cetera?
- Did the  $\text{\LaTeX}$  compiler format your slides nicely?
- What changed about *how* you wrote the content in each slide (if anything changed at all)?
- Were there any *quirks* in the formatting?
- Any compilation warnings and/or errors?
- What did you write about? You don't have to say if you don't want to.

# Table of Contents

- 1 Introductions
- 2 Set Up
- 3 Writing your first  $\text{\LaTeX}$  document
- 4 Lists, Figures, Packages and Images
- 5 Math Mode, Special Characters and Verbatim Environments
- 6 Basic Bib $\text{\TeX}$
- 7 Beamer for Presentations
- 8 And that's it!

- $\text{\TeX}$  StackExchange forum

# Useful resources

- T<sub>E</sub>X StackExchange forum
- Overleaf's *own* L<sup>A</sup>T<sub>E</sub>X tutorials (they're how I learnt L<sup>A</sup>T<sub>E</sub>X back then)

# Useful resources

- $\text{\TeX}$  StackExchange forum
- Overleaf's *own*  $\text{\LaTeX}$  tutorials (they're how I learnt  $\text{\LaTeX}$  back then)
- $\text{\LaTeX}$  Reddit forum



# Useful resources

- $\text{\TeX}$  StackExchange forum
- Overleaf's *own*  $\text{\LaTeX}$  tutorials (they're how I learnt  $\text{\LaTeX}$  back then)
- $\text{r}/\text{\LaTeX}$  Reddit forum
- **Tables Generator for  $\text{\LaTeX}$**

- $\text{\TeX}$  StackExchange forum
- Overleaf's *own*  $\text{\LaTeX}$  tutorials (they're how I learnt  $\text{\LaTeX}$  back then)
- $\text{r}/\text{\LaTeX}$  Reddit forum
- Tables Generator for  $\text{\LaTeX}$
- Wikibooks section listing all of the mathematical symbols you can use in  $\text{\LaTeX}$ , so that you don't have to remember them all.

- $\text{\TeX}$  StackExchange forum
- Overleaf's *own*  $\text{\LaTeX}$  tutorials (they're how I learnt  $\text{\LaTeX}$  back then)
- $\text{r}/\text{\LaTeX}$  Reddit forum
- Tables Generator for  $\text{\LaTeX}$
- Wikibooks section listing all of the mathematical symbols you can use in  $\text{\LaTeX}$ , so that you don't have to remember them all.
- The Comprehensive  $\text{\LaTeX}$  Symbols List

# Thanks for attending this workshop!

I am on:

- [LinkedIn](#)
- [Mastodon](#)
- [BlueSky](#)
- [Instagram](#)
- [GitHub](#) (I put the slides up there as well)
- [My website](#) (including blog and project portfolio, currently working on updating it!)

Advice for you on  $\text{\LaTeX}$ :

- Practice, practice, practice
- Try experimenting with different types of  $\text{\LaTeX}$  documents
- Seek assistance for compilation errors and more about about specific  $\text{\LaTeX}$  and  $\text{BibTeX}$  problems and information.
- Never give up!