General Introduction to and Academic Applications of LaTeX

Session 1

Jad Kanaan and Joe Germany and Mouhammad Somaili

American University of Beirut

Fall 2026

SECTION 1

What LaTeX Is

The core distinction

- In Word, you select text and click **B**. The file records that those characters are bold.

The core distinction

- In Word, you select text and click **B**. The file records that those characters are bold.
- In LaTeX, you write `\textbf{like this}`. You record that this is something you want the software to make bold.

The core distinction

- In Word, you select text and click **B**. The file records that those characters are bold.
- In LaTeX, you write `\textbf{like this}`. You record that this is something you want the software to make bold.
- The typesetter then decides what bold looks like.

Think of it as a recipe

The easiest way to describe it is as a cookbook: in LaTeX you give the program the instructions and it does the actual cooking whereas in Microsoft Word or other similar software you actually do the cooking.

A little history

- **1978.** Donald Knuth writes TeX, appalled at the typesetting of his own book's second edition.

A little history

- **1978.** Donald Knuth writes TeX, appalled at the typesetting of his own book's second edition.
- **1980s.** Leslie Lamport builds LaTeX on top of it, so ordinary authors don't have to think like typesetters.

A little history

- **1978.** Donald Knuth writes TeX, appalled at the typesetting of his own book's second edition.
- **1980s.** Leslie Lamport builds LaTeX on top of it, so ordinary authors don't have to think like typesetters.
- **1994.** LaTeX2e which is still what people use today.

A little history

- **1978.** Donald Knuth writes TeX, appalled at the typesetting of his own book's second edition.
- **1980s.** Leslie Lamport builds LaTeX on top of it, so ordinary authors don't have to think like typesetters.
- **1994.** LaTeX2e which is still what people use today.
- A useful consequence of this is that a file written 20 years ago runs just as well as one written today.

Why bother

- **Mathematics.** Nothing else is as reliable for formulae and proofs. For most STEM majors this is reason enough on its own.

Why bother

- **Mathematics.** Nothing else is as reliable for formulae and proofs. For most STEM majors this is reason enough on its own.
- **Consistency.** You cannot accidentally have three subtly different heading styles.

Why bother

- **Mathematics.** Nothing else is as reliable for formulae and proofs. For most STEM majors this is reason enough on its own.
- **Consistency.** You cannot accidentally have three subtly different heading styles.
- **Plain text.** Works with Git, diffs, and every editor ever written.

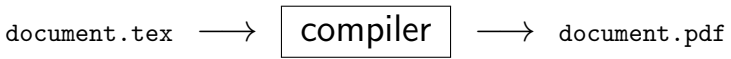
Why bother

- **Mathematics.** Nothing else is as reliable for formulae and proofs. For most STEM majors this is reason enough on its own.
- **Consistency.** You cannot accidentally have three subtly different heading styles.
- **Plain text.** Works with Git, diffs, and every editor ever written.
- **Journals want it.** Physics, maths, CS, engineering, chemistry and biology journals publish LaTeX templates, with most reputable journals even mandating that submissions be done in LaTeX.

Why bother

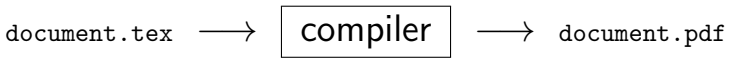
- **Mathematics.** Nothing else is as reliable for formulae and proofs. For most STEM majors this is reason enough on its own.
- **Consistency.** You cannot accidentally have three subtly different heading styles.
- **Plain text.** Works with Git, diffs, and every editor ever written.
- **Journals want it.** Physics, maths, CS, engineering, chemistry and biology journals publish LaTeX templates, with most reputable journals even mandating that submissions be done in LaTeX.
- **Portable and free.** Same output on every operating system, no version mismatches, no licence.

The pipeline



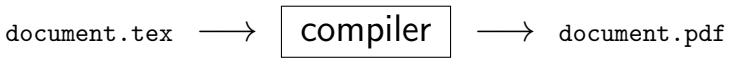
- You write the left-hand side.

The pipeline



- You write the left-hand side.
- `pdflatex` — or `xelatex`, or `lualatex` — reads it and produces the right.

The pipeline



- You write the left-hand side.
- `pdflatex` — or `xelatex`, or `lualatex` — reads it and produces the right.
- Overleaf runs that program for you, which is why nobody installs anything today.

SECTION 2

Overleaf

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`
2. Register **with your AUB email address**

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`
2. Register **with your AUB email address**
3. Confirm the verification email — check spam *now*

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`
2. Register **with your AUB email address**
3. Confirm the verification email — check spam *now*
4. `Menu` → `Account Settings` → `Emails and Affiliations`, confirm the AUB affiliation

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`
2. Register **with your AUB email address**
3. Confirm the verification email — check spam *now*
4. `Menu` → `Account Settings` → `Emails and Affiliations`, confirm the AUB affiliation
5. Look for the *Premium* badge before moving on

Registering on AUB's licence

TYPE ALONG

1. Go to `https://www.overleaf.com`
2. Register **with your AUB email address**
3. Confirm the verification email — check spam *now*
4. `Menu` → `Account Settings` → `Emails and Affiliations`, confirm the AUB affiliation
5. Look for the *Premium* badge before moving on
6. `New Project` → `Blank Project`, name it `session-01`

What the AUB licence gets you

- **Unlimited collaborators** per project, instead of one

What the AUB licence gets you

- **Unlimited collaborators** per project, instead of one
- **A much longer compile timeout** — the difference between a thesis compiling and not

What the AUB licence gets you

- **Unlimited collaborators** per project, instead of one
- **A much longer compile timeout** — the difference between a thesis compiling and not
- **Full version history**, rather than the last 24 hours

What the AUB licence gets you

- **Unlimited collaborators** per project, instead of one
- **A much longer compile timeout** — the difference between a thesis compiling and not
- **Full version history**, rather than the last 24 hours
- **Track changes and comments** — how your supervisor will want to give feedback

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.
- **Editor (middle).** Where the code goes.

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.
- **Editor (middle).** Where the code goes.
- **PDF preview (right).** Does exactly as the name says: shows you what your PDF will look like before you download it.

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.
- **Editor (middle).** Where the code goes.
- **PDF preview (right).** Does exactly as the name says: shows you what your PDF will look like before you download it.
- **Recompile.** The green button.

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.
- **Editor (middle).** Where the code goes.
- **PDF preview (right).** Does exactly as the name says: shows you what your PDF will look like before you download it.
- **Recompile.** The green button.
- **Logs.** Beside Recompile. Where errors and warnings live. You will use this extensively to troubleshoot any issues that come up.

The interface

- **File tree (left).** Your `.tex` files, images, `.bib` files. Upload lives here.
- **Editor (middle).** Where the code goes.
- **PDF preview (right).** Does exactly as the name says: shows you what your PDF will look like before you download it.
- **Recompile.** The green button.
- **Logs.** Beside Recompile. Where errors and warnings live. You will use this extensively to troubleshoot any issues that come up.
- **Share.** Collaborators by email or link.

SECTION 3

Your First Document

The smallest possible document

TYPE ALONG

```
\documentclass{article}

\begin{document}

Hello, world!

\end{document}
```

Press **Recompile**. A PDF appears.

What each part does

- `\documentclass{article}` is you telling LaTeX what kind of document this is so it sets page size, headings, font, and available commands.

What each part does

- `\documentclass{article}` is you telling LaTeX what kind of document this is so it sets page size, headings, font, and available commands.
- Other classes: `report` (adds chapters), `book`, `beamer` (these slides), plus journal-supplied classes.

What each part does

- `\documentclass{article}` is you telling LaTeX what kind of document this is so it sets page size, headings, font, and available commands.
- Other classes: `report` (adds chapters), `book`, `beamer` (these slides), plus journal-supplied classes.
- **Preamble** is a term you will see often if you try to learn more about LaTeX. It is everything that you add before beginning your document in preparation like creating custom commands or importing outside packages which will be talked about extensively later.

What each part does

- `\documentclass{article}` is you telling LaTeX what kind of document this is so it sets page size, headings, font, and available commands.
- Other classes: `report` (adds chapters), `book`, `beamer` (these slides), plus journal-supplied classes.
- **Preamble** is a term you will see often if you try to learn more about LaTeX. It is everything that you add before beginning your document in preparation like creating custom commands or importing outside packages which will be talked about extensively later.
- **Body** is what's between `\begin{document}` and `\end{document}`. This is what appears on the page.

Brackets and braces

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

- Square brackets mean *optional*.

Brackets and braces

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

- Square brackets mean *optional*.
- Curly braces mean *required*.

Brackets and braces

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

- Square brackets mean *optional*.
- Curly braces mean *required*.
- That holds for every command in the language, not just this one.

The whole grammar

- **Commands** start with a backslash:
  `\textbf{bold}`, `\LaTeX`

The whole grammar

- **Commands** start with a backslash:
  `\textbf{bold}`, `\LaTeX`
- **Environments** come in pairs:
  `\begin{something} ... \end{something}`

Packages and comments

TYPE ALONG

```
\documentclass[12pt]{article}
\usepackage{amsmath}    % better mathematics
\usepackage{graphicx}   % lets us include images

\begin{document}

Hello, world! % everything after a percent sign is ignored

\end{document}
```

Packages and comments

TYPE ALONG

```
\documentclass[12pt]{article}
\usepackage{amsmath}    % better mathematics
\usepackage{graphicx}   % lets us include images

\begin{document}

Hello, world! % everything after a percent sign is ignored

\end{document}
```

LaTeX alone is minimal. Packages teach it new tricks — thousands of them, all free, all already installed on Overleaf.

The percent sign

% starts a comment.

Write "a 40% increase" and the rest of that line vanishes with the paragraph running into the next one.

You need `40\%`

SECTION 4

Ordinary Text

Paragraphs

```
This is one paragraph. It doesn't matter how you break the lines in your
    source file, LaTeX interprets it all as one block.

A blank line starts a new paragraph.
```

- Line breaks in the source are just spaces. Multiple spaces collapse into one.
- A blank line means "new paragraph."
- Ten blank lines mean exactly the same as one.

The ten characters that need escaping

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

- Seven take a backslash: `\# \$ \% \& \_ \{ \}`
- Three are special: `\textbackslash`, `\textasciitilde`, `\textasciicircum`
- You don't need to memorise the list, just recognise that if you get an error while using non-standard characters

Emphasis

```
\textbf{bold}, \textit{italic}, \texttt{monospaced},
\textsc{Small Caps}, and \emph{semantic emphasis}.
```

- Prefer `\emph` over `\textit`.

Emphasis

```
\textbf{bold}, \textit{italic}, \texttt{monospaced},
\textsc{Small Caps}, and \emph{semantic emphasis}.
```

- Prefer `\emph` over `\textit`.
- It means "emphasise this" and if used in text that is already in italics it turns it back into normal text.

One more character

A tilde ~ is a space that never breaks across a line.

One more character

A tilde ~ is a space that never breaks across a line.

```
Table~\ref{tab:yields}
```

One more character

A tilde ~ is a space that never breaks across a line.

```
Table~\ref{tab:yields}
```

So "Table" and its number can never end up on different lines. You will use this constantly from Session 2 onward.

SECTION 5

Structure

Sections, title, contents

TYPE ALONG

```
\title{A Very Short Report}
\author{Jad Kanaan \\ Department of Biology}
\date{\today}
\begin{document}
\maketitle
\tableofcontents
\section{Introduction}
\subsection{Background}
\section{Methods}
\section*{Acknowledgements}
\end{document}
```

Things to notice

- You never type a section number. LaTeX counts.
- Hierarchy: `\section` → `\subsection` → `\subsubsection`. `report` and `book` add `\chapter` as a tool above `\section` in the hierarchy.
- A starred `\section*` is unnumbered and stays out of the contents. Right for acknowledgements.
- `\today` inserts the compilation date.
- The contents page fills itself in on the first compile in Overleaf.

Exercise

Build a one-page document about yourself

1. A title block with your name and today's date
2. A table of contents
3. Three numbered sections and at least one subsection
4. One unnumbered section
5. Two paragraphs, with something bold and something emphasised
6. Somewhere: three of the special characters we discussed earlier.

Exercise

Build a one-page document about yourself

1. A title block with your name and today's date
2. A table of contents
3. Three numbered sections and at least one subsection
4. One unnumbered section
5. Two paragraphs, with something bold and something emphasised
6. Somewhere: three of the special characters we discussed earlier.

Then download the PDF to your own machine.

SECTION 6

Reading Errors

Break it on purpose

Delete a closing brace. Recompile. Open the log.

1. **Read the first error, not the last.** One mistake causes a dozen more down the line.
2. **The line number is where LaTeX noticed**, not necessarily where you made your mistake, so you may need to look a little above it.
3. **"Undefined control sequence"** means that you have a misspelled command, or a missing `\usepackage`.
4. **"Missing $ inserted"** — a maths-only character in ordinary text, usually a stray `_` or `^`.

Reading the log

- Overleaf's log panel has a "Go to error location" link.
- Yellow warnings are usually safe to ignore for now.

SECTION 7

Exercise 2

Exercise 2

Four things

1. Expand your exercise document to a page and a half or rewrite the first page of a CV you already have.
2. Break your document three different ways. Read the log each time. Then comment out the mistakes and record what the log said.
3. Add a table of contents, then insert a new section between two existing ones. Watch the numbering update itself.
4. Change `article` to `report` and see the layout change without you touching anything else.

Three references

A journal article, a book, and anything else

You will build a bibliography from them.

Questions?

https://www.overleaf.com/learn • https://tex.stackexchange.com

Academic Applications

1. Tables

Academic Applications

1. Tables
2. Cross-references, footnotes, hyperlinks

Academic Applications

1. Tables
2. Cross-references, footnotes, hyperlinks
3. Bibliographies

Academic Applications

1. Tables
2. Cross-references, footnotes, hyperlinks
3. Bibliographies
4. Custom commands

Academic Applications

1. Tables
2. Cross-references, footnotes, hyperlinks
3. Bibliographies
4. Custom commands
5. Multi-file projects and collaboration

SECTION 8

Tables

Two environments, not one

`tabular`	the grid of cells itself
`table`	the float wrapper: caption, number, and it can move

- You almost always want both, one inside the other.

Two environments, not one

`tabular`	the grid of cells itself
`table`	the float wrapper: caption, number, and it can move

- You almost always want both, one inside the other.
- The same pattern returns next session: `\includegraphics` inside a `figure`.

The grid itself

TYPE ALONG

```
\begin{tabular}{lcr}
    Sample & Mass (g) & Yield (\%) \\
    A      & 1.24     & 87 \\
    B      & 0.98     & 91 \\
\end{tabular}
```

- `{lcr}` — one letter per column: **l**eft, **c**entre, **r**ight.

The grid itself

TYPE ALONG

```
\begin{tabular}{lcr}
    Sample & Mass (g) & Yield (\%) \\
    A      & 1.24     & 87 \\
    B      & 0.98     & 91 \\
\end{tabular}
```

- {lcr} — one letter per column: **l**eft, **c**entre, **r**ight.
- & separates columns. \\ ends a row.

A proper table

```
\begin{table}[htbp]
    \centering
    \caption{Yields for the three trial conditions.}
    \label{tab:yields}
    \begin{tabular}{lcr}
        \toprule
        Sample & Mass (g) & Yield (\%) \\
        \midrule
        A & 1.24 & 87 \\
        B & 0.98 & 91 \\
        \bottomrule
    \end{tabular}
\end{table}
```

What a float is

- Your table did not stay where you typed it. That is not a bug.

What a float is

- Your table did not stay where you typed it. That is not a bug.
- LaTeX will not break a table across a page, so it slides it to where it fits.

What a float is

- Your table did not stay where you typed it. That is not a bug.
- LaTeX will not break a table across a page, so it slides it to where it fits.
- This is why you write “see Table 2” and never “see the table below.”

What a float is

- Your table did not stay where you typed it. That is not a bug.
- LaTeX will not break a table across a page, so it slides it to where it fits.
- This is why you write “see Table 2” and never “see the table below.”

What a float is

- Your table did not stay where you typed it. That is not a bug.
- LaTeX will not break a table across a page, so it slides it to where it fits.
- This is why you write "see Table 2" and never "see the table below."

`[htbp]` = **h**ere, **t**op, **b**ottom, own **p**age

SECTION 9

Cross-References

The problem labels solve

- You write “as shown in Table 2”.

The problem labels solve

- You write “as shown in Table 2”.
- Someone inserts a table above it. Yours is now Table 3.

The problem labels solve

- You write “as shown in Table 2”.
- Someone inserts a table above it. Yours is now Table 3.
- Your sentence still says 2, and now points at the wrong table.

The problem labels solve

- You write “as shown in Table 2”.
- Someone inserts a table above it. Yours is now Table 3.
- Your sentence still says 2, and now points at the wrong table.
- Nobody notices until it is published.

The problem labels solve

- You write “as shown in Table 2”.
- Someone inserts a table above it. Yours is now Table 3.
- Your sentence still says 2, and now points at the wrong table.
- Nobody notices until it is published.

The problem labels solve

- You write “as shown in Table 2”.
- Someone inserts a table above it. Yours is now Table 3.
- Your sentence still says 2, and now points at the wrong table.
- Nobody notices until it is published.

\label and \ref

```
\begin{table}[htbp]
    \centering
    \caption{Yields for the three trial conditions.}
    \label{tab:yields}
    ... the tabular goes here ...
\end{table}

Yields are summarised in Table~\ref{tab:yields}.
```

What LaTeX actually does

1. Reaches `\caption` and adds one to the table counter. This is the first table, so the counter is now **1**.
2. Reaches `\label{tab:yields}` and records: *the name `tab:yields` means 1*. Nothing prints.
3. Later reaches `\ref{tab:yields}`, looks the name up, finds 1, and prints "1".

Three consequences

- **The name never prints.** `tab:yields` appears nowhere in the PDF. It is a private note between you and LaTeX.

Three consequences

- **The name never prints.** `tab:yields` appears nowhere in the PDF. It is a private note between you and LaTeX.
- **You invent the name.** You don't need to use a command or something specific, you can name it something you can easily remember.

Three consequences

- **The name never prints.** `tab:yields` appears nowhere in the PDF. It is a private note between you and LaTeX.
- **You invent the name.** You don't need to use a command or something specific, you can name it something you can easily remember.
- **Insert a table above and step 1 gives 2 instead.** Steps 2 and 3 follow automatically.

Order matters

`\label` **goes after** `\caption`

The label attaches to whatever was most recently numbered and `\caption` does the numbering. If you reverse the order, LaTeX will give you a wrong number.

Naming labels

`sec:`	sections
`tab:`	tables
`fig:`	figures
`eq:`	equations

- These are used by convention since they're easy to remember and identify. You can change them, though, as they're not rules encoded by LaTeX

Naming labels

`sec:`	sections
`tab:`	tables
`fig:`	figures
`eq:`	equations

- These are used by convention since they're easy to remember and identify. You can change them, though, as they're not rules encoded by LaTeX
- But every label shares one namespace, so a section called `methods` and a table called `methods` collide.

Naming labels

`sec:`	sections
`tab:`	tables
`fig:`	figures
`eq:`	equations

- These are used by convention since they're easy to remember and identify. You can change them, though, as they're not rules encoded by LaTeX
- But every label shares one namespace, so a section called `methods` and a table called `methods` collide.
- Prefixes make collisions impossible and labels searchable.

Two more

```
See Section~\ref{sec:methods} on page~\pageref{sec:methods}.

As proved by Ted Kaczynski.\footnote{Better known for other work.}
```

- `\pageref` returns the page number instead of the item number. Useful when Section 4 is sixty pages away.

Hyperlinks

```
\usepackage[hidelinks]{hyperref}   % load this LAST

Full information is at \href{https://www.overleaf.com/learn}{Overleaf's
    Documentation}.
```

- One line, and every `\ref`, `\pageref` and citation becomes clickable, plus a PDF bookmark panel.

Hyperlinks

```
\usepackage[hidelinks]{hyperref}   % load this LAST

Full information is at \href{https://www.overleaf.com/learn}{Overleaf's
    Documentation}.
```

- One line, and every `\ref`, `\pageref` and citation becomes clickable, plus a PDF bookmark panel.
- `\url` prints the address. `\href` takes destination first, visible text second.

Hyperlinks

```
\usepackage[hidelinks]{hyperref}   % load this LAST

Full information is at \href{https://www.overleaf.com/learn}{Overleaf's
    Documentation}.
```

- One line, and every `\ref`, `\pageref` and citation becomes clickable, plus a PDF bookmark panel.
- `\url` prints the address. `\href` takes destination first, visible text second.
- `hidelinks` removes the coloured boxes that look dreadful in print.

SECTION 10

Bibliographies

The pitch

- Keep your references in one file, once, in a standard format.
- Cite them by a short key.
- LaTeX works out which you actually used, formats them, sorts them, and numbers them.
- Change one word and the whole thing reformats to another style.

The pitch

- Keep your references in one file, once, in a standard format.
- Cite them by a short key.
- LaTeX works out which you actually used, formats them, sorts them, and numbers them.
- Change one word and the whole thing reformats to another style.

The promise

You will never renumber a reference list by hand again.

The `.bib` file

```
@article{knuth1984,
    author  = {Knuth, Donald E.},
    title   = {Literate Programming},
    journal = {The Computer Journal},
    year    = {1984},
    volume  = {27}
}

@book{lamport1994,
    author    = {Lamport, Leslie},
    title     = {LaTeX: A Document Preparation System},
    publisher = {Addison-Wesley},
    year      = {1994}
}
```

Anatomy of an entry

- **The entry type.** `@article`, `@book`, `@inproceedings`, `@phdthesis`, `@misc`.

You will not type these by hand

Google Scholar, publisher "Export citation" links, Zotero and Mendeley all produce them for you.

Anatomy of an entry

- **The entry type.** `@article`, `@book`, `@inproceedings`, `@phdthesis`, `@misc`.
- **The citation key.** `knuth1984` — you choose it, and it is how you refer to the entry.

You will not type these by hand

Google Scholar, publisher "Export citation" links, Zotero and Mendeley all produce them for you.

Anatomy of an entry

- **The entry type.** `@article`, `@book`, `@inproceedings`, `@phdthesis`, `@misc`.
- **The citation key.** `knuth1984` — you choose it, and it is how you refer to the entry.
- **The fields.** `field = {value}` pairs. Different entry types require different fields.

You will not type these by hand

Google Scholar, publisher "Export citation" links, Zotero and Mendeley all produce them for you.

But check what they give you

- Auto-generated entries are frequently wrong.

Braces protect capitals

Styles lowercase words in titles. To keep a proper noun, an element symbol, or a species name capitalised, wrap it:

```
title = {Studies of {DNA} in {E. coli}}
```

But check what they give you

- Auto-generated entries are frequently wrong.
- Wrong capitalisation, missing pages, inconsistent journal abbreviations, mangled accented names.

Braces protect capitals

Styles lowercase words in titles. To keep a proper noun, an element symbol, or a species name capitalised, wrap it:

```
title = {Studies of {DNA} in {E. coli}}
```

Wiring it up

```
% preamble
\usepackage[style=numeric, backend=biber, sorting=none]{biblatex}
\addbibresource{references.bib}

% body
Knuth argued for literate programming \cite{knuth1984}.
As \textcite{lamport1994} explains, ...

% at the end
\printbibliography
```

Change one word

`style=numeric`

↓

`authoryear` `alphabetic` `apa` `nature` `chem-acs`

The entire bibliography and every in-text citation reformat.

Citation commands

Command	Produces	Use when
`\cite{key}`	[1]	the reference supports the sentence
`\textcite{key}`	Lamport [2]	the author is the subject
`\parencite{key}`	[Lamport 1994]	parenthetical styles
`\cite{a,b,c}`	[1–3]	several at once

Exercise

Build a bibliography

1. Add `references.bib` with the three references you googled.
2. Cite all three — one of them with `\textcite`.
3. Compile until the bibliography appears.
4. Change the style, and recompile.

SECTION 11

Custom Commands

Define your own

```
% in the preamble
\newcommand{\eg}{e.g.\ }
\newcommand{\ecoli}{\textit{E.~coli}\xspace}
\newcommand{\pdv}[2]{\frac{\partial #1}{\partial #2}}

% in the body
Cultures of \ecoli were grown to mid-log phase.
```

Define your own

```
% in the preamble
\newcommand{\eg}{e.g.\ }
\newcommand{\ecoli}{\textit{E.~coli}\xspace}
\newcommand{\pdv}[2]{\frac{\partial #1}{\partial #2}}

% in the body
Cultures of \ecoli were grown to mid-log phase.
```

`\newcommand{\name}[arguments]{definition}`, where `#1` and `#2` stand in for what you pass.

Three reasons, in ascending order

1. **Less typing.** Obvious, and the least important.

Three reasons, in ascending order

1. **Less typing.** Obvious, and the least important.
2. **Consistency.** Every species name written `\ecoli` is italicised identically.

Three reasons, in ascending order

1. **Less typing.** Obvious, and the least important.
2. **Consistency.** Every species name written `\ecoli` is italicised identically.
3. **Global change.** If you decide to change names or styles after a meeting means that you'll only have to modify the one command instead of a hundred lines of text.

SECTION 12

Homework

Before Session 2

Three things

1. Build one table from your own work using `booktabs` — captioned, labelled, and referred to from prose.
2. Add a `.bib` file, cite three references, get the bibliography printing. Then change the style once.
3. Define one custom command for something you type repeatedly, and use it at least twice.

Bring an image file

Questions?

https://www.overleaf.com/learn • https://tex.stackexchange.com