

# An Introduction to L<sup>A</sup>T<sub>E</sub>X: Typesetting your Thesis or Research Paper

## Part I: The Basics

Luke Collins    Jean Paul Ebejer

Doctoral School  
University of Malta

25<sup>th</sup> November, 2019



L-Università  
ta' Malta

# Today's Plan

- 1 What is  $\text{\LaTeX}$ ?
  - What is typesetting?
  - Showcase
- 2 The Basic Document and Commands
  - Hello World!
  - Command Technicalities
- 3 Document Elements
  - Title, ToC, etc.
  - Figures and Tables
  - Tables
  - List Environments
- 4  $\text{\LaTeX}$  errors people make

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

- Pronounced lay-tek or lah-tek, NOT lay-teks!



T<sub>E</sub>X comes from the Greek word τέχνη, meaning skill/art/technique.

- A typesetting system: this means L<sup>A</sup>T<sub>E</sub>X worries about **typesetting**, so you shouldn't have to!

Typesetting means worrying about what the document *looks* like. Any respectable publisher wants these things done right!

- Spacing and kerning between words/letters

AV Wa  
No kerning

AV Wa  
Kerning applied

- Ligatures

fi ▶ fi  
fl ▶ fl  
ct ▶ ct

- Where titles are positioned, page numbering and ToC, etc.

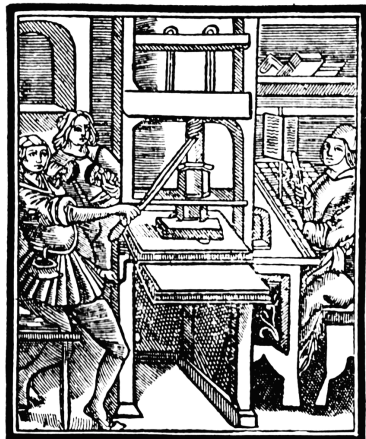


Figure: Gutenberg's press

# T<sub>E</sub>Xian Ontology

WYSWIG (read: wizzy-wig), i.e., “what you see is what you get” editors, such as Libre Office Writer or Microsoft Word, are getting better at these things, but the typographical quality of the documents they produce is still inferior to L<sup>A</sup>T<sub>E</sub>X.

The *important* thing to understand though, is that L<sup>A</sup>T<sub>E</sub>X takes care of all this for you.

## The philosophy:

*L<sup>A</sup>T<sub>E</sub>X allows you to clearly separate the content from the format of your document. As a writer (scientist, researcher or not), this gives you the opportunity to focus on the “what”, the creative part of your work, rather than the “how” is it going to look printed out in paper—that is L<sup>A</sup>T<sub>E</sub>X’s job.*

# Showcase

Here is a showcase of what people use L<sup>A</sup>T<sub>E</sub>X for:

arXiv:1906.05790v1 [math.CO] 13 Jun 2019

## ON THE WALKS AND BIPARTITE DOUBLE COVERINGS OF GRAPHS WITH THE SAME MAIN EIGENSPACE

LUKE COLLINS  
luke.collins@um.edu.mt

IRENE SCIRIHA  
irene.sciriha-aquilina@um.edu.mt

Department of Mathematics,  
University of Malta,  
Msida, Malta

13<sup>th</sup> June, 2019

### Abstract

The main eigenvalues of a graph  $G$  are those eigenvalues of the  $(0,1)$ -adjacency matrix  $\mathbf{A}$  having a corresponding eigenvector not orthogonal to  $\mathbf{j} = (1, \dots, 1)$ . The CDC of a graph  $G$  is the direct product  $G \times K_2$ . The main eigenspace of  $\mathbf{A}$  is generated by the principal main eigenvectors and is the same as the image of the walk matrix. A hierarchy of properties of pairs of graphs is established in view of their CDC's, walk matrices, main eigenvalues, eigenvectors and eigenspaces. We determine by algorithm that there are 32 pairs of non-isomorphic graphs on at most 8 vertices which have the same CDC.

**Keywords:** Eigenvalues, walks, walk matrix, main eigenspace, canonical (bipartite) double covering, TP-isomorphism.

### 1 Introduction

A graph of order  $n$  is a pair of sets  $G = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V} = \{1, \dots, n\}$  is called the set of vertices, and  $\mathcal{E} \subseteq \{(u, v) : u, v \in \mathcal{V} \text{ and } u \neq v\}$  is called the set of edges. (We consider graphs which are simple; that is, graphs which are undirected, without multiple edges or loops.) A  $k$ -walk in a graph  $G$  is a  $k$ -tuple  $(u_0, u_1, \dots, u_k) \in \mathcal{V}^{k+1}$  such that  $\{u_{i-1}, u_i\} \in \mathcal{E}$  for all  $1 \leq i \leq k$ .

The adjacency matrix of a graph  $G$ , denoted by  $\mathbf{A}(G)$ , or simply  $\mathbf{A}$  where the context is clear, is the symmetric  $n \times n$  matrix  $(a_{ij})$ , where  $a_{ij} = 1$  if  $\{i, j\} \in \mathcal{E}$ , and  $a_{ij} = 0$  otherwise. We use terminology for a graph  $G$  and its adjacency matrix  $\mathbf{A}$  interchangeably, since the graph  $G$  is determined, up to relabelling of the vertices, by  $\mathbf{A}$ . For example, the eigenvalues and eigenvectors of a graph  $G$  are respectively those of the matrix  $\mathbf{A}$ . The spectrum  $\text{spec}(G)$  of a graph  $G$  is the multiset consisting of the  $s$  distinct eigenvalues

1

Figure: Research papers

# Showcase

Here is a showcase of what people use L<sup>A</sup>T<sub>E</sub>X for:

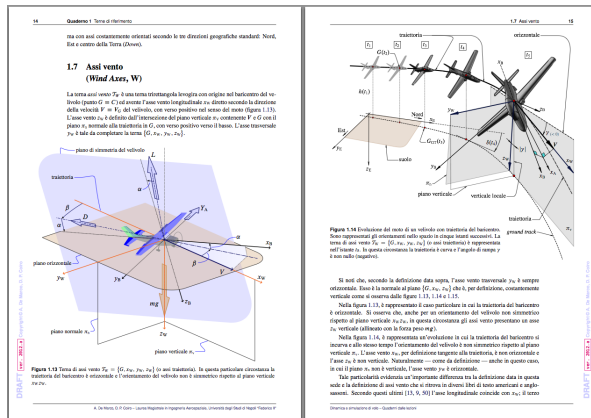


Figure: Notes and textbooks

# Showcase

Here is a showcase of what people use L<sup>A</sup>T<sub>E</sub>X for:



Figure: Curriculum vitae



# Showcase

Here is a showcase of what people use L<sup>A</sup>T<sub>E</sub>X for:



Figure: Statistical reports



Here is a showcase of what people use  $\text{\LaTeX}$  for:



Figure: Magazines

# Showcase

Here is a showcase of what people use  $\text{\LaTeX}$  for:

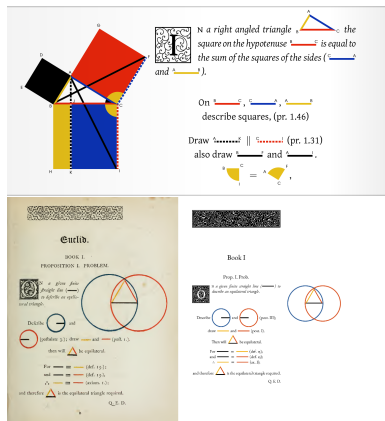


Figure: Recreating French Bibles and Euclid's Elements

# Showcase

Here is a showcase of what people use  $\text{\LaTeX}$  for:



Here is a showcase of what people use  $\text{\LaTeX}$  for:



Figure: Making presentations!



Figure: Making presentations!

Now let's get our hands dirty.

# What you need

- T<sub>E</sub>X distribution (such as T<sub>E</sub>X Live or MiK<sub>T</sub><sub>E</sub>X),
- A text editor or a T<sub>E</sub>X IDE such as T<sub>E</sub>X Studio.

L<sup>A</sup>T<sub>E</sub>X is a language, which is 'compiled'.

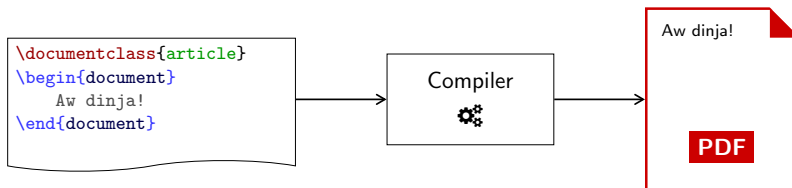


Figure: What a L<sup>A</sup>T<sub>E</sub>X compiler does

# The “Hello World” example

```
\documentclass{article}
\begin{document}
  Aw dinja!
\end{document}
```

Aw dinja!

- L<sup>A</sup>T<sub>E</sub>X commands start with backslashes: `\command`.
- The `documentclass` is the “type” of document we are producing. All this does is define structural things like whether there are chapters, or sections. We chose `article`, others include `book`, `report` and `letter`.
- The text of the document itself is enclosed between `\begin{document}` and `\end{document}`. The two commands `\begin{...}` and `\end{...}` enclose what’s called an *environment*.



# The “Hello World” example: modified

```

\documentclass{article}
\begin{document}
  Aw   dinja!
  Caw

  dinja!  A b c d e f g h
  i j k l m n o p q r s t
  u v w x y z
\end{document}

```

```

Aw dinja! Caw
  dinja! A b c d e f g h i
j k l m n o p q r s t u v
w x y z

```

- L<sup>A</sup>T<sub>E</sub>X ignores multiple white spaces in the source code (Aw\_\_\_dinja!) as well as single new-lines.
- A blank line starts a new paragraph. The first line of a paragraph is indented.

# Some common commands

`\textbf{bold text}`

**bold text**

`\textit{italic text}`

*italic text*

`\emph{emphasised text}`

*emphasised text*

`\textsl{slanting text}`

*slanting text*

`\textsc{Small Caps}`

SMALL CAPS

`\texttt{Typewriter}`

Typewriter

`\textsuperscript{superscript}`

superscript

`\textsubscript{subscript}`

subscript

`\underline{underlined}`

underlined

`\TeX` and `\LaTeX`

TeX and L<sup>A</sup>T<sub>E</sub>X

# An Example

```

\documentclass{article}
\begin{document}
  The \textbf{quick} \textit{brown}
  \textsc{fox} \underline{jumps}
  \textsuperscript{over} the
  \textsubscript{lazy} \textsl{dog}.

  \emph{This text is \emph{very}
  important}.
\end{document}

```

The **quick** brown FOX jumps  
<sup>over</sup> the <sub>lazy</sub> dog.  
*This text is very important.*

- Notice nesting `\emph` alternates between italic and non-italic text. It is better practice to use `\emph` instead of `\textit` when your goal is to *emphasise* what you are saying.

# Reserved Characters

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

These symbols have special meaning to the L<sup>A</sup>T<sub>E</sub>X compiler, and cannot be used in a T<sub>E</sub>X file as part of your text. Instead, we use the following:

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

- Notice that the corresponding commands for `^` and `~` are followed by “`{}`”. This is because they are *commands* which usually take arguments, similar to `\documentclass`. In particular, they are *accents*:

tajj`\^{}{a}`r or just tajj`\^{}ar` becomes tajjâr<sup>1</sup>  
 Jalape`\~{}{n}`o or just Jalape`\~{}no` becomes Jalapeño

---

<sup>1</sup>as of 2011 (Deċiżjonijiet 1), it is technically incorrect to spell tajjar (as in cotton wool) this way, we just use it here for illustration.

# Other Accents

Grave	<code>responsabbilt\`a</code>	responsabbiltà
Acute	<code>caf\'e</code>	café
Circumflex	<code>tajj\^ar</code>	tajjâr
Umlaut	<code>F\"ur Elise</code>	Für Elise
Hungarian umlaut	<code>Erd\H os</code>	Erdős
Tilde	<code>Jalape\~no</code>	Jalapeño
Cedilla	<code>fa\c cade</code>	façade
Dot	<code>\.ci\.cri</code>	çicri

# What is a command?

Notice that those accents whose commands contained letters were followed by a space, e.g., `Erd\H` os for Erdős. If we want, we could have done `Erd\H{o}`s.

The reason for this is commands are made up of all the alphabetical characters following the `\` character. Commands do not usually contain numbers or other symbols (things like `\.`, `\^` etc. are exceptions).

So if we do `Erd\Hos`, the compiler thinks `\Hos` is the command we are invoking, and gets confused. On the other hand, we can do things like `Erd\H0s` (Erdős) or `Erd\H.s` (Erdős) without spaces, since these are not alphabetical characters and are therefore not treated as part of the command.

# What is a command?

Spaces immediately after a command are ignored, so we could equally do `tajj\^` ar for `tajjâr`.

Commands consume the object which immediately follows them, e.g.

`\textbf` abc becomes **abc**

If we want a group of things to be treated as one, we *group* them using curly brackets:

`\textbf` {abc} becomes **abc**

We can also feed `\textbf` nothing:

`\textbf` {} abc becomes **abc**

Notice that the space immediately after the command is ignored, but the space after the curly brackets is visible.

# Examples

`\`A\c ce`

Àçe

`\`A\cce`

*Error*

`\`A\c1e`

À₁e

`\textbf xyz`

xyz

`\textsc {xy}z`

XYZ

`\LaTeX is great!`

L<sup>A</sup>T<sub>E</sub>Xis great!

`\LaTeX{} is great!`

L<sup>A</sup>T<sub>E</sub>X is great!

`\textbf\textit xyz`

*Error (left associativity)*

`\textbf{\textit xyz}`

*xyz*



# Document Structure

A document should be split into logical parts, say:

- Title
- Table of Contents
- Chapters
  - ▶ Sections
    - ★ Subsections
      - ◇ Subsubsections
- Appendices

The exact structure and how customisable it is will depend on the choice of document class. For example, an `article` does not have chapters, but sections as the top-level object.

# Which document class should I use?

**article** is ideal for a paper in scientific journals (divided into sections, subsections, etc.). Usually has an abstract.

**report** is ideal for longer reports, containing chapters, say a thesis, small book, etc. Also usually has an abstract.

**book** is for actual books.

**letter** is for writing letters.

**beamer** is for making slideshows (like this one).

# The Typical Document Skeleton

```
\documentclass{article}

% Document Info
\title{How to Write a \LaTeX{} Document}
\author{Luke Collins}
\date{25\textsuperscript{th} November, 2019}

\begin{document}
    \maketitle % title

    The first step is to download a \TeX{} distribution.
\end{document}
```

# The Typical Document Skeleton

```
\documentclass{article}

% Document Info
\title{How to Write a \LaTeX{} Do
\author{Luke Collins}
\date{25\textsuperscript{th} Nove

\begin{document}
    \maketitle % title

    The first step is to download

\end{document}
```

## How to Write a L<sup>A</sup>T<sub>E</sub>X Document

Luke Collins

25<sup>th</sup> November, 2019

The first step is to download a T<sub>E</sub>X distribution.

# The Typical Document Skeleton

```

\documentclass{article}

% Document Info
\title{How to Write a \LaTeX{} Do
\author{Luke Collins}
\date{25\textsuperscript{th} Nove

\begin{document}
    \maketitle % title

    The first step is to download

\end{document}

```

## How to Write a L<sup>A</sup>T<sub>E</sub>X Document

Luke Collins

25<sup>th</sup> November, 2019

The first step is to download a T<sub>E</sub>X distribution.

1

- Remember that % was a reserved character—it is used for *comments*.

# Chapters, Sections, Subsections, etc.

```

\documentclass{report}
\begin{document}
  \chapter{Complex Analysis}

  \section{What is Complex Analysis?}
  Complex analysis is the study of functions of complex
  variables.

  \subsection{Holomorphic Functions}
  A function is \emph{holomorphic} if it is differentiable
  at every point in some open region of the complex plane.

  \subsection{Cauchy}
  A lot of nice results in complex analysis are due to
  Augustin-Louis Cauchy.
\end{document}

```

# Chapters, Sections, Subsections, etc.

```

\documentclass{report}
\begin{document}
  \chapter{Complex Analysis}

  \section{What is Complex}
  Complex analysis is the
  variables.

  \subsection{Holomorphic}
  A function is \emph{holo}
  at every point in some o

  \subsection{Cauchy}
  A lot of nice results in
  Augustin-Louis Cauchy.
\end{document}

```

## Chapter 1

# Complex Analysis

## 1.1 What is Complex Analysis?

Complex analysis is the study of functions of complex variables.

### 1.1.1 Holomorphic Functions

A function is *holomorphic* if it is differentiable at every point in some open region of the complex plane.

### 1.1.2 Cauchy

A lot of nice results in complex analysis are due to Augustin-Louis Cauchy.

# The Abstract

```

\documentclass{article}

\title{Fermat's Last Theorem}
\author{Andrew Wiles}
\date{23\textsuperscript{rd} June, 1993}

\begin{document}
  \maketitle % title

  \begin{abstract}
    In this paper, we give a proof of Fermat's 1637
    conjecture, his so-called last theorem.
  \end{abstract}

  \section{Introduction}
  We start with some computations of cohomology groups.
\end{document}

```



# The Abstract

```

\documentclass{article}

\title{Fermat's Last Theorem}
\author{Andrew Wiles}
\date{23\textsuperscript{rd}}

\begin{document}
  \maketitle % title

  \begin{abstract}
    In this paper, we give a proof of Fermat's 1637
    conjecture, his so-called last theorem.
  \end{abstract}

  \section{Introduction}
  We start with some computations of cohomology groups.
\end{document}

```

## Fermat's Last Theorem

Andrew Wiles

23<sup>rd</sup> June, 1993

### Abstract

In this paper, we give a proof of Fermat's 1637 conjecture, his so-called last theorem.

## 1 Introduction

We start with some computations of cohomology groups.

# Table of Contents

- A table of contents can be automatically generated from chapters, sections, etc. by invoking `\tableofcontents`.
- **WARNING:** The `\tableofcontents` command depends on an auxiliary file which is generated during compilation. For this reason, you should **compile twice**, once to generate the file, and a second time to ensure the correct file is used in displaying the table of contents.

# Table of Contents

```

      :
      :
\begin{document}
  \maketitle           % title
  \tableofcontents     % contents
      :
      :
\end{document}

```

## Some notes on L<sup>A</sup>T<sub>E</sub>X

Luke Collins & JP Ebejer

December, 2019

### Contents

<b>1 Basics</b>	<b>2</b>
1.1 What is L <sup>A</sup> T <sub>E</sub> X?	2
1.2 The basic Document and Commands	3
1.3 Document Structure	4
1.4 Other Miscellaneous Tips	7
<b>2 Maths, Bibliographies and Templates</b>	<b>9</b>
2.1 Typesetting Mathematical Equations	10
2.2 AMS Theorems	12
2.3 Other Document Elements	15
2.3.1 Footnotes	17
2.3.2 Bibliographies	19
2.3.3 Indices	20
2.3.4 Cross references	21
2.4 Templates	22

# Including Images

- This is the first time we will be making use of a *package*.
- At the top of the T<sub>E</sub>X file, after `\documentclass{...}` but before `\begin{document}`, do

```
\usepackage{graphicx}
```

This will allow us to make use of extra commands in the graphicx package which are not available in L<sup>A</sup>T<sub>E</sub>X by default.

- Images can be the usual .jpeg, .png, etc., but also .pdf (for vectorised images). Place the image file (say, image.jpg) in the same directory as the T<sub>E</sub>X file, and do

```
\includegraphics{image}
```

at the place in the text you want the image to appear (the file extension is optional).

# Including Images

```

\documentclass{article}
\usepackage{graphics}
\begin{document}
  Aw dinja!
  \includegraphics{dinja}
\end{document}

```

Aw dinja!



# Including Images — Optional Arguments

- Remember, command arguments are placed immediately after a command, often in curly brackets.

`\command{argument}`

Some commands take multiple arguments, we'll see some of those later on:

`\command{argument1}{argument2}{argument3}`

- Other times, commands have *optional* arguments (usually just called *options*.) Options are placed in square brackets, usually before the mandatory arguments.

`\command[option]{argument}`

`\command[option1,option2]{argument1}{argument2}`

# Including Images — Optional Arguments

For example, say we want to make an included image smaller/larger. There are optional arguments we can pass to `\includegraphics` for this.

```
\documentclass{article}
\usepackage{graphics}
\begin{document}
  Aw dinja!
  \includegraphics[scale=0.5]{dinja}
\end{document}
```



Other options for `\includegraphics` are width, height and angle.

Try experimenting with multiple options at the same time.

# Including Images — Figures

If you do not want to include an image among the text (as is often the case), it is best to make use of the `figure` environment. Place the environment in the document where it should logically be, and L<sup>A</sup>T<sub>E</sub>X will determine an ideal position for it. All figures will be numbered automatically.

```
\documentclass{article}
\begin{document}
  Aw dinja! \.Caw dinja!
  \begin{figure}
    \centering % to centre stuff
    \includegraphics{dinja}
    \caption{The world}
  \end{figure}
\end{document}
```



Figure 1: The world

Aw dinja! Ćaw dinja!



# Including Images — Figures

Environments also sometimes have options and arguments.

You can suggest to L<sup>A</sup>T<sub>E</sub>X where you would like the figure to be placed by doing `\begin{figure}[option]`, where `option` is one of:

- `t`, top of the page,
- `b`, bottom of the page,
- `h`, *here*, i.e., where it appears in the text,
- `p`, on a page where other figures are present.

These are only suggestions to L<sup>A</sup>T<sub>E</sub>X, they are not definite. If you really want to insist, you can do `!h` to insist that L<sup>A</sup>T<sub>E</sub>X put the figure `h` (also `!t`, `!b`, `!p`), but this is still not guaranteed.

# Tables

- First choose a layout for the columns by combining the symbols

`l c r | || p{x}`

where  $x$  is a length, for example, 3.5cm or 50pt.

For example, `l c | c | | p{2.4cm}` corresponds to a table with a left aligned column, a centred column, a vertical line, a centred column, two vertical lines, and a “paragraph” column of width 2.4 cm.

- The `&` character is used as a column separator, `\\` starts a new line, and `\hline` creates a horizontal line.

# Tables

```

\begin{tabular}{lc|c|lp{2.4cm}}
  Name      & ID      & Grade & Description & \\
\hline \hline
A Cauchy   & 555555F & 10    & Bad \LaTeX{} skills. & \\
JP Ebejer  & 123456M & 75    & OK \LaTeX{} skills.  & \\
L Collins  & 987654M & 100   & God-tier \LaTeX{} skills. & \\
\end{tabular}

```

Name	ID	Grade	Description
A Cauchy	555555F	10	Bad L <sup>A</sup> T <sub>E</sub> X skills.
JP Ebejer	123456M	75	OK L <sup>A</sup> T <sub>E</sub> X skills.
L Collins	987654M	100	God-tier L <sup>A</sup> T <sub>E</sub> X skills.

# Tables – Table Environment

The table environment is identical to the figure environment, but for tables.

```
\begin{table}[b]
  \centering % centres everything within the table environment
  \begin{tabular}{l c | c | p{2.4cm}}
    Name      & ID      & Grade & Description & \\
    \hline \hline
    A Cauchy  & 555555F & 10    & Bad \LaTeX{} skills. & \\
    JP Ebejer & 123456M & 75    & OK \LaTeX{} skills. & \\
    L Collins & 987654M & 100   & God-tier \LaTeX{} skills. & \\
  \end{tabular}
  \caption{Results of the \LaTeX{} exam}
\end{table}
```

# Tables – Table Environment

The table environment  
tables.

```
\begin{table}[b]
  \centering % centered
  \begin{tabular}
    Name & ID & Grade & Description \\
    \hline \hline
    A Cauchy & 555555F & 10 & Bad LATEX skills. \\
    JP Ebejer & 123456M & 75 & OK LATEX skills. \\
    L Collins & 987654M & 100 & God-tier LATEX skills. \\
  \end{tabular}
  \caption{Results of the LATEX exam}
\end{table}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Name	ID	Grade	Description
A Cauchy	555555F	10	Bad L <sup>A</sup> T <sub>E</sub> X skills.
JP Ebejer	123456M	75	OK L <sup>A</sup> T <sub>E</sub> X skills.
L Collins	987654M	100	God-tier L <sup>A</sup> T <sub>E</sub> X skills.

Table 1: Results of the L<sup>A</sup>T<sub>E</sub>X exam

# List Environments

Another collection of useful environments is that of the *list environments*. These are:

- Unordered lists (`itemize`)
- Ordered lists (`enumerate`)
- Description lists (`description`)

```
\begin{itemize}
  \item A thing
  \item Another thing
  \item Oh and another thing
\end{itemize}
```

- A thing
- Another thing
- Oh and another thing

# List Environments

```
\begin{enumerate}
  \item A first thing
  \item A second thing
\end{enumerate}
```

1. A first thing
2. A second thing

List environments can be nested:

```
\begin{enumerate}
  \item Question 1
  \item Question 2
    \begin{enumerate}
      \item Question 2(a)
      \item Question 2(b)
    \end{enumerate}
  \item \begin{enumerate}
    \item Question 3(a)
  \end{enumerate}
\end{enumerate}
```

1. Question 1
2. Question 2
  - (a) Question 2(a)
  - (b) Question 2(b)
3. (a) Question 3(a)

# List Environments

The enumerate package allows us to customise the format of the enumerate counter, write `\usepackage{enumerate}` in your preamble.

```
\begin{enumerate}[i.]
  \item A first thing
  \item A second thing
\end{enumerate}
```

- i. A first thing
- ii. A second thing

```
\begin{enumerate}[(A)]
  \item A first thing
  \item A second thing
\end{enumerate}
```

- ((A)) A first thing
- ((B)) A second thing



# List Environments

Description lists are for scripts, dictionaries, etc.

```
\begin{description}
  \item[Gloucester] Now is the
    winter of our discontent
    made glorious summer by
    this sun of York\dots
  \item[Clarence] Because my
    name is George.
\end{description}
```

**Gloucester** Now is the  
winter of our  
discontent  
made glorious  
summer by this  
sun of York...

**Clarence** Because my  
name is George.

# Other Useful Environments

- To centre text, you can use the `center` environment.

```
\begin{center}
  Aw dinja!
\end{center}
```

Aw dinja!

If you just have one line, you can use `\centerline{Aw dinja!}` to achieve the same effect.

Do *not* confuse any of these with the `\centering` command which we used in things like the `figure` environment. That command is “dangerous”, because it centres **everything** that follows it and is in the same group. (In the case of figures, it centres everything within the `figure` environment, for example

Another example of a “dangerous” command is `\bfseries`. Compare what the following do:

- ▶ `\bfseries{Aw}` dinja, kif int?
- ▶ `\textbf{Aw}` dinja, kif int?
- ▶ `{\bfseries Aw}` dinja, kif int?

- The verbatim environment prints everything typed in literally using a typewriter font (including indentation and new lines!).

```
\begin{verbatim}
# $ % ^ & _ { } ~ \
int main(){printf("Aw
dinja!"); return 0;}
\end{verbatim}
```

```
# $ % ^ & _ { } ~ \
int main(){printf("Aw
dinja!"); return 0;}
```

In-line, the command `\verb|code|` or `\verb+code+` does the same (notice we use characters like `|` or `+` to open/close).

# Know your Hyphens and Dashes!

There are three kinds of “dashes” in the English language. The **hyphen** is used to join words such as semi-prime. The **en dash** is used to state ranges, e.g., pages 5–10, or to adjoin names together e.g., the Hardy–Littlewood conjecture. Finally, the **em dash** is used as a punctuation mark to serve as a long pause—like this. (Notice there are no spaces between the words around the dash).

These are entered differently in L<sup>A</sup>T<sub>E</sub>X:

hyphen	non-zero	non-zero
en dash	Borsuk–Ulam theorem	Borsuk--Ulam theorem
em dash	so—as I was saying—he was going to Tipperary	so---as I was saying---he was going to Tipperary

# Other Mistakes

- Using `\` to end a paragraph instead of a blank line.
- Literally typing `...` for *ellipsis* (i.e., dot dot dot). In L<sup>A</sup>T<sub>E</sub>X you should use the command `\dots` to get the correct spacing (and don't put a `.` after it!)
- Using the `"` character for quotes. In L<sup>A</sup>T<sub>E</sub>X, the single quote `'` *always* denotes a close quote (’), and the grave accent ``` *always* denotes an open quote (‘). To use “double quotes”, type them twice: ``double quotes'` and do not use the `"` character.
- L<sup>A</sup>T<sub>E</sub>X puts more space after a full-stop than it does between words, since it assumes a new sentence is starting. If you want to have a fullstop followed by a space mid-sentence, use `\_` (i.e., backslash space). An example:

✗	Prof. Borg	Prof. Borg
✓	Prof. Borg	Prof.\ Borg

# Other Mistakes

- On a similar note, honorifics such as Dr, Mr, Mrs, etc. should *not* be followed by a full-stop. The rule is: *If the last letter of the abbreviation is the same as last letter of the word, then **no** full-stop.* (Similarly, it's ABC Ltd, not ABC Ltd.)

Full-stops should also be omitted from capitalised abbreviations, so it's UK not U.K., NATO not N.A.T.O., and CV not C.V.

- Lower-case abbreviations such as e.g., i.e., etc. have a full-stop and the end of every word (etc. is an exception).

i.e.	<i>id est</i>	that is	ca.	<i>circa</i>	approximately
e.g.	<i>exempli gratia</i>	for example	etc.	<i>et cetera</i>	and so on
cf.	<i>confer</i>	see/compare	vs.	<i>versus</i>	against

It is better to always put a comma after i.e. and e.g., rather than a space (in which case do \\_,), since they should be read as “that is,” and “for example,” respectively.

# Next Time

- How to typeset maths formulæ, such as

$$f(z_0) = \frac{1}{2\pi i} \oint_{\gamma} \frac{f(z)}{z - z_0} dz,$$

and get numbered theorems.

- Other document elements such as footnotes, bibliography, index, etc.
- How to add links and cross references within a document.
- How to draw your own diagrams!
- How to use the UM L<sup>A</sup>T<sub>E</sub>X dissertation template, and where to find other L<sup>A</sup>T<sub>E</sub>X templates and resources.

# *Thank you!*

**Luke Collins**

[luke.collins@warwick.ac.uk](mailto:luke.collins@warwick.ac.uk)

<https://drmenguin.com>

**Dr Jean-Paul Ebejer**

[jean.p.ebejer@um.edu.mt](mailto:jean.p.ebejer@um.edu.mt)

<https://bitsilla.com>

**Doctoral School**  
University of Malta



**L-Università  
ta' Malta**