

# Getting started with LaTeX

Duncan Hull

2020-11-30



# Contents

<b>1</b>	<b>Written communication with LaTeX</b>	<b>5</b>
1.1	Not a word processor . . . . .	6
1.2	Acknowledgements . . . . .	6
<b>2</b>	<b>Simple LaTeX documents</b>	<b>7</b>
2.1	The shortest document possible . . . . .	7
2.2	A longer LaTeX document . . . . .	8
2.3	Exercise one: document . . . . .	9
2.4	Bold, italic, lists and quotes . . . . .	9
2.5	Conclusion . . . . .	10
<b>3</b>	<b>Cross referencing, illustrating and citing</b>	<b>11</b>
3.1	Cross referencing . . . . .	11
3.2	Illustrating your documents . . . . .	12
3.3	Exercise two: figures . . . . .	13
3.4	Citations and footnotes . . . . .	13
3.5	Conclusion . . . . .	15
<b>4</b>	<b>Doing the math</b>	<b>17</b>
4.1	Equations . . . . .	17
4.2	Matrices . . . . .	18
4.3	Exercise 3: maths . . . . .	19
4.4	Conclusions . . . . .	19

<b>5 Collaborative editing with overleaf</b>	<b>21</b>
5.1 Exercise four: overleaf . . . . .	22
5.2 Conclusion . . . . .	22
<b>6 Your curriculum vitae</b>	<b>23</b>
6.1 Exercise five: your CV . . . . .	25
6.2 Debug your CV . . . . .	25
6.3 Conclusion . . . . .	25

# Chapter 1

## Written communication with LaTeX

This document is designed to help you get started with using LaTeX, it supports the labs for the first year undergraduate course in Computer Science, COMP101.



Figure 1.1: The LaTeX project logo, image by Jonas Jacek CC BY 4.0 via Wikimedia Commons [w.wiki/oZs](<https://w.wiki/oZs>)

LaTeX is a document preparation system for high-quality typesetting. (Anon, 2020b) It is most often used for technical or scientific documents but it can be used for almost any form of publishing, including writing CVs and formal letters. Whatever documents you create with LaTeX, one of its key strengths is making them look professional in portable document format (pdf).

## 1.1 Not a word processor

LaTeX is not a word processor! Instead, LaTeX encourages you not to worry too much about the appearance of your documents but to concentrate on getting the right content. This is similar to the approach you've been using for creating web pages where the style (CSS) is separated from the contents (HTML).

Reading this LaTeX guide and doing the exercises will enable you to:

1. Create a simple document in pdf using LaTeX
2. Illustrate a document with figures and cross references
3. Typeset some mathematics
4. Save and share LaTeX documents using overleaf
5. Create a CV using LaTeX templates in overleaf
6. Extend your knowledge of LaTeX and know where to find out more

## 1.2 Acknowledgements

This document is a revised version of earlier LaTeX lab manuals created by Ulrike Sattler, Graham Gough, Paul Waring, Toby Howard and Steve Pettifer.

## Chapter 2

# Simple LaTeX documents

We'll start by creating a simple LaTeX document.

### 2.1 The shortest document possible

Open up your favourite editor, enter the following text, then save it as a file called `turing.tex`

```
\documentclass[a4paper]{article}
\begin{document}
Computational excursions
\end{document}
```

To turn this into a PDF we need to use a LaTeX compiler, we're going to use `pdflatex`, though several other compilers are available.<sup>1</sup> The `pdflatex` compiler is already installed on the Linux VM image for VirtualBox.<sup>2</sup> For other operating systems you'll need to explore the options at [latex-project.org/get](https://www.latex-project.org/get).

```
# compile turing.tex
pdflatex turing.tex
# open the pdf output
xdg-open turing.pdf
```

The first command outputs a pdf file from your TeX input using the `pdflatex` compiler. The second command opens the file you've created. If you list the directory contents, you'll see that the compiler has also created an auxiliary

---

<sup>1</sup>[https://www.overleaf.com/learn/latex/Choosing\\_a\\_LaTeX\\_Compiler](https://www.overleaf.com/learn/latex/Choosing_a_LaTeX_Compiler)

<sup>2</sup>[https://wiki.cs.manchester.ac.uk/index.php/CSImage\\_VM](https://wiki.cs.manchester.ac.uk/index.php/CSImage_VM)

file and a log file which can be helpful for debugging the compilation of more complicated documents:

```
# files created on compilation
turing.tex
turing.pdf
turing.aux
turing.log
```

## 2.2 A longer LaTeX document

Our turing.tex file is a very simple document, so let's add some sections and fill them out a bit:

```
\documentclass[a4paper]{article}
\begin{document}

\section{Algorithms: Cooking Up Programs}
A program specifies in the \textbf{exact syntax} of some programming language the comp

\section{Finite automata: The Black Box}
It occasionally happens in industrial, military or educational settings that one is pr

\section{Systems of Logic: Boolean Bases}
In an age of computers and automation, almost every electronic device one can name inc

\section{Simulation: The Monte Carlo method}
In the quest to understand the many systems that comprise the modern world we turn inc

\section{Gödel's Theorem: Limits on Logic}
In the early 1930's, Kurt Gödel, a German mathematician, attempted to show that predic

\section{Can machines think?}
Turing addressed the question ``Can machines think?'' in his 1950 paper \textit{Comput

\section{But what is LaTeX good for?}
We're using this \LaTeX\ document to demonstrate some of its key strengths that you wi

\begin{enumerate}
\item LaTeX can quickly create pdf files
\item LaTeX uses professional typesetting
\item LaTeX documents can be more legible, clear, and visually appealing to the reader
\end{enumerate}
```



```
\end{document}
```

The text here is excerpted from *The New Turing Ominbus: 66 excursions in Computer Science* (Dewdney, 2001).

## 2.3 Exercise one: document

Cut-and-paste this longer text into your document or make your own sections and text. You could use text from Lorem ipsum at [lipsum.com](http://lipsum.com) to fill out the page.

Now, at the top of your document after the `\begin{document}` line and before the command defining the first section, add the following commands, each on their own line:

```
\title{The New Turing Ominbus}
\author{A. K. Dewdney}
\maketitle
\tableofcontents
\newpage
```

The `title`, `author`, `tableofcontents` and `newpage` commands are self-explanatory. The `maketitle` automatically generates a date. Your table of contents won't be created until you run `pdflatex` **twice**. You need to process the source twice because on the first run, LaTeX gathers and stores information about what to put in the table of contents, and only creates it on the second run.

```
# remember to run pdflatex twice for the table of contents
pdflatex turing.tex
pdflatex turing.tex
```

## 2.4 Bold, italic, lists and quotes

Here's a few points to note about the text above:

- Notice how bold and italic formatting are created using `textbf` and `textit`
- Notice how lists are created using `enumerate` and `item`
- Notice that curved open quotation (“ - looks like 66) and close quotation marks (” - looks like 99) are different characters. Look carefully at the quotes around the phrase “Can machines think?” in the excerpt above in section longer.

## 2.5 Conclusion

You've created a basic document in LaTeX and we've introduced some of its advantages:

- LaTeX can quickly create pdf files
- LaTeX uses professional typesetting
- LaTeX documents can be more legible, clear, and visually appealing to the reader than those created with word processing software

Next we'll look at adding some cross-references, figures and citations.

## Chapter 3

# Cross referencing, illustrating and citing

LaTeX has simple but powerful tools to allow you to cross reference, illustrate and cite sources in your documents.

### 3.1 Cross referencing

LaTeX allows you to cross-reference almost anything in the document, which includes sections, sub-sections and figures. To use the cross-referencing feature you simply insert the command:

```
\label{mymarker}
```

at the point in the document you want to refer to, and then use the command:

```
\ref{mymarker}
```

when you want to use the reference. Obviously you replace the text **mymarker** with something more meaningful. An important tip here is to call the marker something that refers to the content of that part of the document, and to avoid the temptation to use numbers in case you re-order your document. For example, lets say we wanted to cross reference between sections:

```
\documentclass[a4paper]{article}  
\begin{document}  
\section{Turing Machines: The Simplest Computers}
```

```

\label{sec:simplest}
Turing machines are the simplest and most widely used theoretical models of computing.

\section{Alan Turing}
The Turing machines described in section \ref{sec:simplest} are named after Alan Turing.
\end{document}

```

## 3.2 Illustrating your documents

A document without figures, images, picture and graphs is pretty dull, so you'll want to illustrate your document with figures such as the one in Figure 3.1, for example.



Figure 3.1: Alan Turing at the age of sixteen, picture by unknown author, public domain, via Wikimedia Commons [w.wiki/oZx](https://w.wiki/oZx)

Including figures and pictures in your document is straightforward. You do it like this:

```

\begin{figure}
\includegraphics[width=10cm]{images/turing.jpg}
\caption{Turing machines are named after Alan Turing}
\label{figure:turing}
\end{figure}

```

The command `\includegraphics{}` gets your image, which can be PDF, PNG, JPG, GIF or PostScript. The `\begin{figure}` and `\end{figure}` code wraps

up whatever picture you're including, and allows LaTeX to treat it as an unbreakable floating thing that it will position for you as best it can in the document, while maintaining an overall nice typographical layout. This floating of figures can sometimes result in the figure ending up in a place you didn't expect, but in most cases LaTeX will make the most sensible choice. It's possible to employ finer control over figure placement, but that's beyond the scope of this guide.

The `\includegraphics` command is not built in to core LaTeX but is in an additional package, which needs to be explicitly loaded. You can load this package by using the command `\usepackage` in the document preamble, in between the `\documentclass` and the `\begin{document}`.

```
\documentclass[a4paper]{article}
\usepackage{graphicx}
\begin{document}
```

### 3.3 Exercise two: figures

Create a document `image.tex`, that contains some text (maybe from Lorem Ipsum), together with a figure containing an image of your choice. Create a cross reference to the figure in the text.

### 3.4 Citations and footnotes

Footnotes<sup>1</sup> can be added to a document with `\footnote{}`

```
\footnote{This is a footnote}
```

You can cite sources such as websites, books or journal articles in your document using the `\cite` command.

```
\cite{turing}
```

The citations themselves can be stored in a separate `*.bib` file using a format called BibTeX. BibTeX gives provides citation types so books<sup>2</sup> look like this:

---

<sup>1</sup>Like this one which is a footnote about footnotes. Very meta.

<sup>2</sup>Using tools like <https://www.ottobib.com> can help you to quickly generate BibTeX entries from a given ISBN number

```
@Book{turingomnibus,
  title = {The New Turing Omnibus: Sixty-six excursions in Computer Science},
  author = {A. K. Dewdney},
  publisher = {Henry Holt},
  address = {New York},
  year = {2001},
  isbn = {9780805071665},
  url = {https://en.wikipedia.org/wiki/Special:BookSources?isbn=978-0805071665}
}
```

and articles<sup>3</sup> look like this:

```
@article{turing,
  doi = {10.1112/plms/s2-42.1.230},
  url = {https://doi.org/10.1112/plms/s2-42.1.230},
  year = {1937},
  publisher = {Wiley},
  volume = {s2-42},
  number = {1},
  pages = {230--265},
  author = {Alan Turing},
  title = {On Computable Numbers, with an Application to the Entscheidungsproblem},
  journal = {Proceedings of the London Mathematical Society}
}
```

For web references you can use misc

```
@misc{googlescholar,
  author = {Anon},
  title = {Alan Turing Google Scholar page},
  url = {https://scholar.google.co.uk/citations?user=VWCHlwAAAAJ},
  year = {2020},
}
```

If we wanted to create the text:

“You can find Alan Turing’s publications in Google scholar. (Anon, 2020a) This includes his paper on the Entscheidungsproblem published in 1937. (Turing, 1937)”

We would need the following in our tex file:

---

<sup>3</sup>BibTeX entries for journal articles can be automatically generated from unique identifiers known as digital object identifiers, see <https://doi2bib.org> for example. This saves you the hassle of typing them in by hand.

You can find Alan Turing's publications in Google scholar. `\cite{googlescholar}`  
This includes his paper on the Entscheidungsproblem published in 1937. `\cite{turing}`

To generate the bibliography you'll need to add these two lines

```
\bibliographystyle{natbib}  
\bibliography{latex}
```

and when you compile you need to run `pdflatex` before and after running `bibtex`.

```
pdflatex turing.tex  
bibtex turing.bib  
pdflatex turing.tex  
pdflatex turing.tex
```

## 3.5 Conclusion

You have cross referenced, illustrated and added citations and footnotes to your document. Next we'll look at doing some maths.





## Chapter 4

# Doing the math

LaTeX provides powerful tools for typesetting mathematics, such as this equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

To render the equation above, you include this in your tex file:

```
\[ x = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \]
```

### 4.1 Equations

You can probably work out how most of this creates the formula, but it won't be obvious that the `\[` and `\]` symbols that enclose the formula mean typeset this as a displayed formula, giving it some vertical space from the surrounding text. If we'd used `\(` and `\)` instead to enclose the formula it would appear in-line, like this:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ . It still looks very nice, and observe how it's automatically been resized to fit, and that the lines of text have had their spacing changed a bit. This all looks simple, but the implementation inside LaTeX and TeX is complex. It involves parsing the description of the formula to create a corresponding tree data structure, which is then recursively walked to work out the horizontal and vertical typographical spacings needed. You'll meet these ideas in COMP11120 Mathematical Techniques for Computer Science in your first year and COMP26120 Algorithms and Imperative Programming in your second year.

Here's another example, taken from computer graphics, it's a local illumination model incorporating ambient, diffuse and specular reflection by multiple lights:

$$I = k_a I_a + \sum_{i=1}^M \frac{I_{p_i}}{d'_i} [k_d(\hat{N} \cdot \hat{L}_i) + k_s(\hat{R}_i \cdot \hat{V})^n]$$

We write this in LaTeX as follows:

```
\[ I = k_a I_a + \sum_{i=1}^M { \frac{{I_p}_i}{{d'}_i} }
[ k_d(\hat{N} \cdot \hat{L}_i)
+ k_s(\hat{R}_i \cdot \hat{V})^n] \]
```

Try to match the LaTeX commands with the formula displayed above. You'll see lots of curly brackets, and this example illustrates their two uses in LaTeX. The first is to provide an argument to a command; for example `\hat{N}` means apply the `\hat` command to `N`, which creates  $\hat{N}$ , the vector  $N$  with a little hat on.

The second use of curly brackets is to group things together to avoid ambiguities. In the example you can see  $\sum_{i=1}^M$ , which creates a summation sign and its lower and upper limits:  $\sum_{i=1}^M$ . We wrap the lower bound, `i=1` in curly brackets to group it into an indivisible unit. If we were to omit the brackets, writing `\sum_i=1^M`, LaTeX would then produce  $\sum_i = 1^M$ , which is not at all what we want (even LaTeX can't always know what we really want).

## 4.2 Matrices

If we need to use matrix, such as the one below which expresses a particular 3D geometrical transformation

$$T_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & \delta \\ \sin \theta & \cos \theta & 0 & \epsilon \\ 0 & 0 & 1 & \eta \\ \alpha & \beta & \gamma & 1 \end{bmatrix}$$

we can use the following

```
\[ T_1 = \left[
\begin{array}{cccc}
\cos \theta & -\sin \theta & 0 & \delta \\
\sin \theta & \cos \theta & 0 & \epsilon \\
0 & 0 & 1 & \eta \\
\alpha & \beta & \gamma & 1
\end{array}
\right] \]
```

In the LaTeX code, `\left[` means ‘big opening square bracket please’; `{cccc}` means ‘an array with 4 columns please, with the items in each column centred’; `&` means ‘start a new column’; and `\\` means ‘start a new row’. You’ll notice that LaTeX knows about greek letters; it knows about most standard maths symbols too, and also the ways they’re usually used.

There are some symbols that you might need (for example `\therefore`, which produces the usual three dots symbol  $\therefore$ ) that are not part of standard LaTeX. Many such symbols are provided by the `amssymb` package of symbols compiled by the American Mathematical Society. Details of the many symbols provided by this package can be found online. To use these extra symbols, you need to have `\usepackage{amssymb}` in your document preamble.

LaTeX really shines at typesetting mathematics, we’ve really only scratched the surface here. For more have a look at [en.wikibooks.org/wiki/LaTeX/Mathematics](http://en.wikibooks.org/wiki/LaTeX/Mathematics) or books such as the *Guide to LaTeX* by Helmut Kopka and Patrick W. Daly. (Kopka and Daly, 2003)

### 4.3 Exercise 3: maths

Now for an exercise that involves some mathematics. Create a file `maths.tex` which typesets the following piece of text and mathematics:

There are many positive integer solutions to the equation

$$x^2 + y^2 = z^2$$

which can be rewritten as

$$z = \sqrt{x^2 + y^2}$$

For example (3, 4, 5) or (5, 12, 13). Such solutions are called *Pythagorean triples*.

However, for higher powers the situation is very different, and we have:-

**Theorem: Fermat-Wiles** For all natural numbers  $n \geq 3$ , there are no integers  $x, y, z$  satisfying the equation:

$$x^n + y^n = z^n$$

### 4.4 Conclusions

We’ve briefly looked at typesetting mathematics in LaTeX, in the next chapter we’ll look at collaborative authoring with overleaf.



## Chapter 5

# Collaborative editing with overleaf

If you are the sole author of a document, then compiling files on your local machine is fine. However, if you need to collaboratively co-author a document with other people, you'll need to share your TeX on a server. Overleaf (overleaf.com) allows you to do this, and is shown in the screenshot in Figure 5.1.

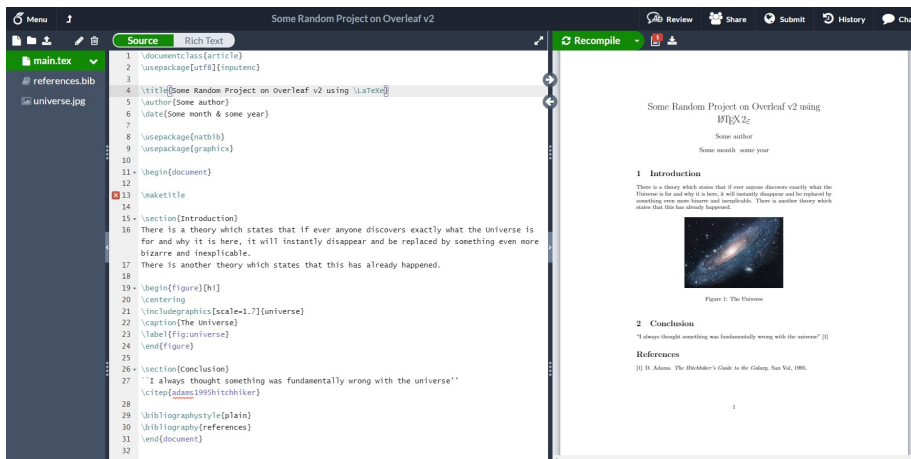


Figure 5.1: A screenshot of overleaf showing the source TeX on the left and side and the corresponding pdf on the right hand side

Overleaf has two features which are useful

- Your documents are saved the cloud

- Packages, class files and compilers are also in the cloud, saving you time installing things
- Templates are provided for common types of documents

## 5.1 Exercise four: overleaf

Overleaf exercise: 1. Login to [overleaf.com](https://overleaf.com) 2. Create and save a simple document using the tutorial at [overleaf.com/learn/how-to/Creating\\_a\\_document\\_in\\_Overleaf](https://overleaf.com/learn/how-to/Creating_a_document_in_Overleaf) 3. Note that you could store your TeX source in a git repository if you wanted to

## 5.2 Conclusion

Overleaf is a handy tool for creating LaTeX documents. While LaTeX has been around since In the next chapter we will create a CV using templates provided by overleaf.

## Chapter 6

# Your curriculum vitae

You'll write many documents at University, but there is one document that is **really** seriously important in a potentially life-changing way. Your curriculum vitae<sup>1</sup> or resume. An example CV is shown in Figure 6.1.

Creating a professional document is especially important when you write your CV, because it determines if you are invited to interview for opportunities you are applying for. The decision to interview is typically based on:

- how your CV looks, it's typesetting, layout and style
- the content of your CV, what you've done and how you describe it
- the quality and clarity of your written communication
- the editing, what you've decided to leave in and out of your CV

By opportunities we mean immediate ones within the next 12 months as well as those further in the future. Those opportunities might include:

- Spring insights during easter next year<sup>2</sup>
- Summer internships next year or during subsequent summers<sup>3</sup>
- Year long placements in your penultimate year, if you're considering doing industrial experience
- Graduate jobs or graduate schemes
- Postgraduate study or research via masters or PhD

So it is well worth investing a bit of your time to create version 1.0 of your CV. It's never too early to make a start.

---

<sup>1</sup>or curriculum vitae (not vitae) if you're being pedantic

<sup>2</sup>see e.g. <https://www.ratemyplacement.co.uk/insights>

<sup>3</sup>see where can I look for jobs? <https://waggle.cs.manchester.ac.uk/waggle/search>

---

<h2 style="text-align: center;">Jake Ryan</h2> <p style="text-align: center;">123-456-7890   <a href="mailto:jake@su.edu">jake@su.edu</a>   <a href="https://www.linkedin.com/in/jake">linkedin.com/in/jake</a>   <a href="https://github.com/jake">github.com/jake</a></p>	
<b>EDUCATION</b>	
<b>Southwestern University</b> <i>Bachelor of Arts in Computer Science, Minor in Business</i>	Georgetown, TX <i>Aug. 2018 – May 2021</i>
<b>Blinn College</b> <i>Associate's in Liberal Arts</i>	Bryan, TX <i>Aug. 2014 – May 2018</i>
<b>EXPERIENCE</b>	
<b>Undergraduate Research Assistant</b> <i>Texas A&amp;M University</i>	June 2020 – Present <i>College Station, TX</i>
<ul style="list-style-type: none"> <li>Developed a REST API using FastAPI and PostgreSQL to store data from learning management systems</li> <li>Developed a full-stack web application using Flask, React, PostgreSQL and Docker to analyze GitHub data</li> <li>Explored ways to visualize GitHub collaboration in a classroom setting</li> </ul>	
<b>Information Technology Support Specialist</b> <i>Southwestern University</i>	Sep. 2018 – Present <i>Georgetown, TX</i>
<ul style="list-style-type: none"> <li>Communicate with managers to set up campus computers used on campus</li> <li>Assess and troubleshoot computer problems brought by students, faculty and staff</li> <li>Maintain upkeep of computers, classroom equipment, and 200 printers across campus</li> </ul>	
<b>Artificial Intelligence Research Assistant</b> <i>Southwestern University</i>	May 2019 – July 2019 <i>Georgetown, TX</i>
<ul style="list-style-type: none"> <li>Explored methods to generate video game dungeons based off of <i>The Legend of Zelda</i></li> <li>Developed a game in Java to test the generated dungeons</li> <li>Contributed 50K+ lines of code to an established codebase via Git</li> <li>Conducted a human subject study to determine which video game dungeon generation technique is enjoyable</li> <li>Wrote an 8-page paper and gave multiple presentations on-campus</li> <li>Presented virtually to the World Conference on Computational Intelligence</li> </ul>	
<b>PROJECTS</b>	
<b>Gitlytics</b>   <i>Python, Flask, React, PostgreSQL, Docker</i>	June 2020 – Present
<ul style="list-style-type: none"> <li>Developed a full-stack web application using with Flask serving a REST API with React as the frontend</li> <li>Implemented GitHub OAuth to get data from user's repositories</li> <li>Visualized GitHub data to show collaboration</li> <li>Used Celery and Redis for asynchronous tasks</li> </ul>	
<b>Simple Paintball</b>   <i>Spigot API, Java, Maven, TravisCI, Git</i>	May 2018 – May 2020
<ul style="list-style-type: none"> <li>Developed a Minecraft server plugin to entertain kids during free time for a previous job</li> <li>Published plugin to websites gaining 2K+ downloads and an average 4.5/5-star review</li> <li>Implemented continuous delivery using TravisCI to build the plugin upon new a release</li> <li>Collaborated with Minecraft server administrators to suggest features and get feedback about the plugin</li> </ul>	
<b>TECHNICAL SKILLS</b>	
<b>Languages:</b> Java, Python, C/C++, SQL (Postgres), JavaScript, HTML/CSS, R <b>Frameworks:</b> React, Node.js, Flask, JUnit, WordPress, Material-UI, FastAPI <b>Developer Tools:</b> Git, Docker, TravisCI, Google Cloud Platform, VS Code, Visual Studio, PyCharm, IntelliJ, Eclipse <b>Libraries:</b> pandas, NumPy, Matplotlib	

Figure 6.1: An example CV using an overleaf template. Would you invite Jake to interview based on his CV? There are many other templates available, this one is by Jake Gutierrez and published under an MIT license at [\[overleaf.com/latex/templates/jakes-resume/syzfjbzwjncs\]](https://www.overleaf.com/latex/templates/jakes-resume/syzfjbzwjncs)(<https://www.overleaf.com/latex/templates/jakes-resume/syzfjbzwjncs>)



## 6.1 Exercise five: your CV

Create a basic CV which describes your story including:

- your education, including University
- your experience, voluntary, paid or otherwise
- your projects: personal, technical and non-technical

You can do this using the ready-made templates at <https://www.overleaf.com/gallery/tagged/cv>

## 6.2 Debug your CV

As you progress through University, it is a good idea to keep your CV up to date, and get feedback from as many people as you can. There will be opportunities to debug your CV later, but you'll need a "beta release" (version 1.0) of your CV to get started. The best time to start debugging is now, so that you can squash any bugs before employers see them. Here's some common bugs we've seen in CVs.

1. Is your year of graduation, degree program, University and expected (or achieved) degree classification clear?
2. Are there any spelling mistakes, typos and grammar? Don't just rely on a spellchecker, they can't detect everything
3. Does it look good, decent layout, easy on the eye?
4. Does it fit comfortably on preferably one (Résumé) or two pages (CV) only?
5. Is it in reverse chronological order? Most important (usually recent) things first?
6. Have you talked about what you have actually done using prominent verbs, rather than just what you think you know? See [git.io/verbsfirst](https://git.io/verbsfirst) for examples
7. Have you mentioned things you are studying now and throughout the year, not just courses you have finished?
8. Have you quantified and provided evidence for the claims you make?
9. Is it robot proof? Feed it through software like [careerset.io/manchester](https://careerset.io/manchester) and [resume.io](https://resume.io). What feedback do the robots give you?
10. Find out more in the *Debug your CV* guide at [git.io/mycv](https://git.io/mycv)

## 6.3 Conclusion

Your CV is a *really* important document. We recommend you start working on it sooner rather than later and get feedback from as many different people as

you reasonably can.

# Bibliography

Anon (2020a). Alan turing google scholar page.

Anon (2020b). Latex – a document preparation system.

Dewdney, A. K. (2001). *The New Turing Omnibus: Sixty-six excursions in Computer Science*. Henry Holt, New York.

Kopka, H. and Daly, P. W. (2003). *Guide to LaTeX*. Addison Wesley, Boston, Massachusetts, fourth edition edition.

Turing, A. (1937). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.