User Guide

Maintained by Pedro H. Mattos

Last update: 2025-07-14

Contents

Introduction

This project provides a template to create a curriculum vitae, or résumé. It is intended as a 'fast-introduction' to typsetting with LaTeX, for someone with little, or no experience in the subject.

Inspiration came about by the realisation that students might be better-motivated to learn LaTeX if the benefits for doing so were more obvious. Students, inundated with assignments and (oh no!) exams, are often unwilling to waste precious time grappling with new software. Especially when the results are comparable to what can be achieved using Microsoft Word. Most tutorials, intended for beginners, start gently with 'Hello World'; helpful, but not very practical.

So instead, this template rewards the user with a CV, or résumé. Both the project and this moderately short user guide are intended to get you using LaTeX more quickly. It is not exhaustive, but complimentary to other tutorials. A more in-depth introduction to the software, or useful reference material can be found at https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

This project introduces the class file `articleCV.cls` and TeX file `CVexample.tex`, which together construct the template. The main parts of this document are concerned with reading and editing .tex and .cls files, parsing documentation and navigating a project directory. More attention is given to the construction of the class file in the appendix; which I imagine would be of more interest to the experienced LaTeX user.

Get Started

If you haven't already, begin by following the instructions located on the GitHub repository at https://github.com/Pedro-h-mattos/LaTeXCVTemplate.

They instruct you to:

- Install a TEX distribution.
- Download a copy of this projects' files onto your local device.
- Compile `CVexample.tex` to produce a PDF output.

The instructions are also repeated here in more depth.

Installation

A TEX distribution, such as MikTeX, TeXLive or MacTeX, is a prerequisite for working with .tex files. They free and easily available for download online.

TEX files can be written using any editor, and compiled from the system shell with the command `pdflatex <myfile>.tex`. However, for ease-of-use, you should probably install a dedicated graphical user interface for working with TEX files (e.g. TeXworks or TeXstudio).

Instructions for downloading the appropriate software can be found by following the link: https://www.latex-project.org/get/.

> **Step 1.**
>
> Install an appropriate TeX distribution and editor for your system.

Alternatively, Overleaf is an online LaTeX editor which doesn't require any local installation; but will ask you to register an account.

Download from GitHub

With the necessary software now installed; you must download the project files from the GitHub repository onto your local device.

This project consists of the files `articleCV.cls`, `CVexample.tex`, `info.sty` and this guide. You can download the entire repository as a ZIP file or install the required files directly into a new folder.

> **Step 2.**
>
> Within the repository, find and click on **Download ZIP**. Then, unpackage your zip file.

Alternatively, download each file by clicking on its name and then **Download raw file**. Save them in a folder with an appropriate name (e.g. myresume).

Your First Document!

Compiling your document now is a good litmus test to see if your software has been correctly installed. You can compile documents directly from the command line, by typing the command:

```
pdflatex CVexample.tex
```

Alternatively you can open `CVexample.tex` within a TEX editor, then compile it using the `pdfLaTeX` option.

> **Step 3.**
>
> Try compiling your document; then open the output file `CVexample.pdf`.

Take a moment to examine your document. What do you notice?

Although very neatly typset, your 'personalised' résumé lacks any personal information at all! `CVexample.tex` is designed as a customisable template; the rest of this guide will help you to fill it out, step-by-step.

Layout

The Document

This template has been been designed to accomodate a standard layout for a CV or résumé. Personal and/or contact information is located in the document header and below that, sections correspond to a person's education, work experience, skills, publications, etc.

With this template, the header can be divided into three parts; the title, the author's name and their personal/contact information.

The body of the document comprises three sections, labelled 'Education/Qualifications', 'Technical Skills' and 'Experience'. Sections are followed by 'subheadings'; each one delimits a single topic, such as a qualification. Optionally, bullet points(s) can follow subheadings to add extra information.

The TEX File

Returning to `CVexample.tex`, can you identify structural similarities between your TEX file and the template?

TEX files always begin with a class declaration, which determines the kind of document that is produced. The expression `\documentclass{articleCV}` loads the file `articleCV.cls`, which is an example of an *inherited class* (more on this later).

The 'body', or contents of a text document must be contained between the declarations `\begin{document}` and `\end{document}`. This includes all the text that is output when the document is compiled.

The contents between the class declaration and `\begin{document}` is called the *preamble*. Here we can define the document's configuration settings and load packages. For simplicity, our custom document class already takes care of this.

As in the template, the TEX file contains three sections, helpfully prefaced by the `\section{}` command. Following these are elements beginning with the

command `\tab{}`, representing the subheadings. This is a custom command and one we'll return to later. For now, remember that these elements format the body of your résumé.

Near the top of your TEX file is the code that will format your document's header. Although it may look intimidating, this is where we'll start editing.

Building a Document

The Header pt.1

The document header includes content describing the optional title, 'Résumé/CV' and the author's name. This looks like:

```
24.  \begin{centering}
25.    {\large \scshape R\'esum\'e/CV \par}
26.    {\Huge Pedro Henrique Mattos \par}
27.  \end{centering}
```

When a TEX document is compiled, only plain text is output. Special characters, e.g. `{}` or commands (prepended by the `\` symbol) are generally ignored — as they control how your document looks.

So, we can safely change the contents and worry about the formatting later.

> **Step 4.**
>
> Rewrite the title and replace my name with your own. Then, recompile your document.

Now, we can consider how these lines are formatted.

The environment `24.\begin{centering}` and `27.\end{centering}` center the title on the page. Removing these lines would automtically left-adjust the text, which may be preferred.

The following describes the text that is output when the document is compiled.

```
26.    {\large \scshape R\'esum\'e/CV \par}
27.    {\Huge Pedro Henrique Mattos \par}
```

Commands (prepended by `\`) can change the style of text. Each line is enclosed by a pair of braces, so that the styles don't overlap. `\par` at the end of each line indicates the start of a new paragraph.

Notice how both lines have different styles. In the first line, the commands `\large` and `\scshape` make the font large and SMALL CAPS, respectively. Compare this with the following line, where the font size is set to `\Huge`.

LaTeX has many different font styles, which are worth exploring.

The Header pt.2

The header also describes personal and/or contact information; formatted as a table with two columns and three rows.

Although this is not immediately obvious within the TeX file, the following code constructs the table:

```
31.  \begin{table}[ht]
32.    \centering
33.    \begin{tabular}{>{\small} l >{\footnotesize} l}
34.    <Undergraduate, 3rd Year>, & Email: \Email{\email} \\
35.    \dept, & Tel: \phone \\
36.    \org & \url{\website}
37.    \end{tabular}
38.  \end{table}
```

Fortunately, we can once again edit the contents of our header, before worrying about the formatting. These three lines describe the contents of the table.

```
35.    <Undergraduate, 3rd Year>, & Email: \Email{\email} \\
36.    \dept, & Tel: \phone \\
37.    \org & \url{\website}
```

Each line describes a single row of the table. Separations between columns, and linebreaks are indicated by the ampersand `&` and double backslash `\\` symbols, respectively.

Excluding `<Undergraduate, 3rd Year>`, each entry of the table is actually a custom command that references your department , organisation, email, phone number and website.

These commands are defined within the file `.sty/info.sty`, which is loaded in the preamble. By rewriting their definitions, we can change what is output in the table. This is a useful trick for describing information that is often repeated, which can be applied across multiple files.

> **Step 5.**
>
> Navigate to the file `.sty/info.sty` in your project directory.

Commands are defined as `\newcommand{\<tag>}{<definition>}`. The definition is output whenever a command is referenced by its tag.

> **Step 6.**
>
> Rewrite the commands within `.sty/info.sty`. Then, save your changes and recompile `CVexample.tex`.

Don't forget to also change the expression `<Undergraduate, 3rd Year>` with your own title or position, in the case that you are not a 3rd year undergraduate student.

Now, let's look at the construction of the table. (Skip ahead to the next section if this one feels like it's above your pay-grade).

The commands `33.\begin{tabular}` and `37.\end{tabular}` create a *tabular* environment, which produces a table. Confusingly, the environment created by `31.\begin{table}` and `38.\end{table}` is called a *float*, which only controls the position of the table on the page.

To match the first header, `32.\centering` centers the table on the page. Although the table itself is centered, text within it is left-adjusted.

`\begin{tabular}{l l}` is sufficient to initialize a table with two left-adjusted columns, indicated by the two letters `l`. The commands `>{\small}` and `>{\footnotesize}` prepend the respective styling to either column; so that text is small within the first column and footnotesize in the second.

Creating Subheadings

The body of a CV or résumé is generally devoted to short 'subheadings' that describe a relevant qualification and optionally, information such as a timeframe, organisation and location. Often, these subheadings are followed by bullet-points.

The `\tab` command, described in the class file `articleCV.cls`, creates a subheading. It is implemented as follows:

```
\tab %
  {#1}{#2}
  {#3}{#4}
```

It creates a page-width table with two columns, that are justified to the right and left margins, and two rows. Cells are given as inputs (`#1-4`), which can be null. The first line (inputs `#1` and `#2` will be **larger and bold**. The second line (inputs `#3` and `#4`) will be *smaller, italicized and sans-serif*.

Consider, for example, this excerpt from my own résumé:

```
\tab
  {Bachelor's of Science in Biochemistry}{2022--2026}
  {Queen Mary University of London}{London, England}
```

Bachelor's of Science in Biochemistry **2022–2026**
Queen Mary University of London *London, England*

Pretty snazzy, eh?

It should now be easier to spot where the `\tab` command has been implemented. If you're struggling, try looking at your TEX file and document side-by-side.

We can rewrite the arguments passed to each subheading to fill out the contents of our document. We'll do this section by section.

Our document contains three sections in total.

```
43.  \section{Education/Qualifications}
54.  \section{Technical Skills}
69.  \section{Work Experience}
```

For more detail on how the `\tab{}` command was implemented, see the appendix.

Educational Background

We'll cover your educational background first, to reflect a standard résumé layout. Later, we'll discuss how to reorder parts of the document.

> **Step 7.**
>
> Change the first section label to something like 'Education', or whatever you prefer.

Next, look at the `\tab` command, underneath the first section heading. We're going to rewrite this to describe your most recent qualification.

```
45.  \tab
46.  {Qualification}{Timeframe}
47.  {Organisation}{Location}
```

> **Step 8.**
>
> Add information to the first `\tab` command.

The default layout is only a suggestion. But, consider what information you want to stand out.

You might choose to put your degree program first, next to your period of study. Or, this is interchangeable with the name of your awarding body (i.e. university), and optionally an address.

Bachelor's of Science in Biochemistry **2022–2026**
Queen Mary University of London *London, England*

Is equivalent to:

Queen Mary University of London
Bachelor's of Science in Biochemistry *2022–2026*

Remember that inclusive dates (i.e. 2024–2025) are written using *en-dashes*. In LaTeX this is achieved by typing out two hypens `--`.

Directly below the first subheading is the following environment, that creates a bulleted list:

```
48.  \begin{itemize}
49.    \item
50.  \end{itemize}
```

A list is contained by the expressions `\begin{itemize}` and `\end{itemize}`. The command `\item`, followed by plain text, creates a bullet-point.

This is an optional space to add more detail about your qualification. You can give specific examples of your skills, projects that you have worked on, significant achievements or classes that you have taken.

> **Step 9.**
>
> Within the list environment, add some more information to your résumé.

You likely have multiple qualifications that you'd like to show off.

Luckily, the same block of code can be copied-and-pasted to create another subheading.

> **Extension**
>
> Using `\tab` and a list environment, create another subheading. Rinse and repeat for all your qualifications.

If you haven't already, recompile your document to see the changes that you made. You are more than one-third towards finishing your document.

Technical Skills

An employer will want evidence of technical or personal skills, related to the job for which you are applying. For example, an application for a research position

may ask for evidence of your ability to carry out titrations, perform western blots or qPCR analyses.

Besides laboratory techniques, remember that report-writing, experiment-design, literature-searching and delivering presentations are all examples of skills you will have gained within your degree.

Consider a position that you are interested in. What skills are the employer asking for?

> **Step 10.**
>
> Make a note of the skills you already possess.

Look at this example, again taken from my own résumé.

```
\tab
  {Software Proficiencies}{}
  {Python, R, BASH, \LaTeX}{}
```

Software Proficiencies
Python, R, BASH, LaTeX

In this case, inputs `#2` and `#4` have been left blank; because we don't need to add so much information. As before, we can use bullet-points to add more information about our skill.

> **Step 11.**
>
> Using bullet points, add information about your technical skill.

When describing technical skills, focus on clear and specific examples. A good idea is to describe projects you completed, within or outside the context of your degree.

Extension

Add a relevant skill to the third subheading and recompile your document.

As with the previous section; you can repeat this operation for as many skills as you want to share.

Work Experience

The last section of your document will describe your employment history. If you don't have one, don't worry. You can instead describe your internships, volunteer work or personal projects.

Step 12.

Rewrite the subheading with your last position. Also add bullet-points to describe your experience.

A résumé should be tailored towards the position or field you are interested in. Consider what relevant skills or experiences should be highlighted here.

Extension

Try adding the rest of your employment history to this section.