Readme

Pedro Henrique Mattos

Instruction manual for version 2.0

A template for creating a professional-looking, PDF-format Résumé. Aimed at users with limited or no prior experience using a typesetting software such as LaTeX.

The contents of this package (.zip folder) are designed to introduce you to typesetting with LaTeX, by providing instructions to work with .tex and .cls files, read package documentation, and manage files within a project directory.

Contents

1 Get Started

Hopefully, this is the first file that you opened. Without a TEX distribution, only *README.pdf* should be accessible to you. (We'll get to the other files later).

This document is intended to be an 'instruction manual' for creating a CV, or résumé, using the typsetting system LATEX.

It isn't intended to be a 'Hello World' style tutorial. A better introduction to the software can be found at https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

Instructions are always provided as follows:

Step 1.

This is an instruction.

Ocassionally, an extension task will prompt you with a more complicated task. Don't worry, these are optional.

Extension

This is an optional task.

2 Contents of the .zip File

- *README.tex and .pdf* — this file, containing documentation and instructions.
- */.tex* — a directory containing .tex files
- */.sty* — a directory containing the file readme.sty
- *CVexample.tex and .pdf* — 'root' or main file and corresponding PDF output
- *articleCV.cls* — class file, controls document formatting for *CVexample.tex*
- *info.sty* — contains definitions for custom commands
- *.aux, .out, .log, .toc and .synctex.gz* — auxillary files

3 Installation and Usage

3.1 Downloading Necessary Software

A prerequisite for working with .tex files is to install a TeX distribution, such as MikTex, TexLive or MacTeX — which enable your computer to use TeX code. They are free and easily available for download online.

TEX input files are plain text files, which can be written using any editor and compiled from the system shell. However for ease-of-use, you should install a dedicated graphical user interface for working with TeX files (e.g. TeXworks or TeXstudio).

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

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

Alternatively, Overleaf is an online LaTeX editor which doesn't require any local installation. Although, does require setting up an account.

3.2 Opening TeX Files

Within a TeX editor, you are able to open and work with .tex files. *CVexample.tex* is the main file for this project, which contains code that will output a document.

> **Step 2.**
>
> Within your project directory, locate and open the file *CVexample.tex*.

3.3 Compiling TeX Files

Compiling this code will output a PDF-format résumé called *CVexample.pdf*, located in your project directory. Depending on your editor, compiling may also automatically create a preview of your document.

Before compiling, take care to select the option *pdfLaTeX*, otherwise it may not work.

> **Step 3.**
>
> Try compiling your first .tex document. Then, locate and open your compiled pdf document within your project directory.

3.4 Your First Document!

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 exercise will guide you in replacing each section of this document.

Now consider the structure of the template itself. How many sections does it contain? Can you identify any elements that are repeated?

The structure of a résumé document is fairly standardized. Personal and/or contact information is usually placed in the document header and sections should correspond to a person’s education, work experience, skills, publications, etc.

Here, the document header can be subdivided into three elements: a title, the author’s name, and their personal and contact information; centered on the page. The latter of which is organised into two columns.

The three sections, labelled *Education/Qualifications*, *Technical Skills* and *Experience*, are always underlined and followed by subsections describing relevant experience. These subsections always compose two lines of text; one bold and larger, and the other italicized and smaller, that may or may not extend across the page. Optionally, a subsection can be followed by one or more bullet points.

4 Reading and Editing TeX Files

PDF files cannot be directly edited, instead *CVexample.tex* has to be recompiled with edits — which are then updated in the corresponding PDF document.

Before creating documents from scratch, we’ll learn to identify the elements within .tex files and to edit them. The result will be a completely personalised résumé, useful for sending out to employers.

4.1 Elements of the Document

> **Step 4.**
>
> Return to your .tex document. Compare elements of the *CVexample.tex* with its .pdf output — can you match each section with its corresponding code?

TeX files always begin with a class declaration, e.g. `\documentclass{}`, which determines what kind of document is produced. Our résumé is constructed from a custom class, contained within the file *articleCV.cls* (more on this later).

The ‘body’, or contents of a text document are contained between the declarations `\begin{document}` and `\end{document}`. This includes all the text that you see on your PDF document.

From here, the section headers are probably the easiest elements to pick out. Sections are always enclosed by the command `\section{}`. There are three within this .tex file.

Each section is followed by at least one code element, beginning with the command `\tab{}`. 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, which includes two code blocks. Although it may look intimidating, this is where we'll start editing.

4.2 The Document Header pt.1

The first element of the document header contains the title *Résumé/CV* and a name. It looks like:

```
1.  \begin{centering}
2.    {\large \scshape R\'esum\'e/CV \par}
3.    {\Huge Pedro Henrique Mattos \par}
4.  \end{centering}
```

Only plain text is output on in your document, once it is compiled. Special characters, e.g. `{}` or commands (prepended by the `\` symbol) are ignored — as they control how your document looks.

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

> **Step 5.**
>
> Rewrite the name with your own, and then choose a title for your document, or no title at all. Then, recompile your document.

Now, we can consider how these lines are formatted.

`1.\begin{centering}` and `4.\end{centering}` ensure that the document title is centered on the page. Deleting or commenting-out these lines would automtically left-adjust the text, which may be preferred.

The command `\large` increases the font size, relative to the normal font size. Compare this to the command `\Huge`, in the following line. LaTeX has several different font sizes, which are worth experimenting with.

> **Extension**
>
> Try changing `\large` to `\Huge` and recompile your document, so that both lines are the same size.

Q. Consider the change you made. Do you prefer this layout? Why might you want your name to be larger than the title?

The last command, `\scshape`, stands for *small caps*. Again, this is an optional choice. You may prefer a normal font; in which case this expression can be removed.

4.3 The Document Header pt.2

The second element that describes the section header looks more complicated. Our PDF file contains two columns, but these aren't immediately obvious within our TeX file.

```
1.  \begin{table}[ht]
2.    \centering
3.    \begin{tabular}{>{\small} l >{\footnotesize}l}
4.    <Undergraduate, 3rd Year>, & Email: \Email{\email} \\
5.    \dept, & Tel: \phone \\
6.    \org & \url{\website}
7.    \end{tabular}
8.  \end{table}
```

The above code is an example of the *tabular* environment, which creates tables. Our table is held within the commands `3.\begin{tabular}` and `7.\end{tabular}`.

Look at the expression `3.\begin{tabular}{>{\small} l >{\footnotesize}l}`. The tabular environment contains two columns, that are each left-adjusted. These are indicated by the two *'l'* symbols.

The commands `>{\small}` and `>{\footnotesize}` prepend the respective size ahead of each row in either column. So, text is small within the first column and 'footnotesize' within the second column.

The tabular itself is contained within a *table*, which is a type of *float* or 'container' that holds an entity and controls its position on the page.

The next three lines describe the contents of our table.

```
4.    <Undergraduate, 3rd Year>, & Email: \Email{\email} \\
5.    \dept, & Tel: \phone \\
6.    \org & \url{\website}
```

Each line describes a single row and ends with a `\\` symbol — indicating an end-of-line. Each column is delimited by the ampsersand `&` symbol.

These three lines contain several custom commands, `\email`, `\dept`, `\phone`, `\org` and `\website`. The definitions of these are contained within the file *info.sty*.

Personal information, that is often repeated, can be saved in a file and referenced with a short command. This is useful if you'd like to have multiple copies of your résumé.

In this case, we can change all of the information in the header, without worrying about the formatting of the table.

Step 6.

Open the file *info.sty*.

New commands are structured as follows: `\newcommand{\<tag>}{<definition>}`. Commands are referenced by their tags. Changing the definitions of the commands contained within the file *info.sty* will update them where they are referenced in your main file.

Step 7.

Rewrite each definition with your own information. Then, save *info.sty* and recompile *CVexample.tex*

In the case that you are not an undergraduate student, also replace the row *'Undergraduate, 3rd Year'* with your current title or position.

4.4 A Note on Document Formatting

The main body of a résumé document is generally limited to short 'sections' or subheadings, that normally include details about a relevant position, qualification or experience and optionally a date; an organisation and a location. Often, these subheadings are followed by bullet-points providing more information.

In the PDF document, subheadings are elements of two lines, with text justified to both the right and left margins. Text in the first line is larger, and bold. Text in the second line is smaller, sans-serif and italicized.

4.5 Creating Résumé Subheadings

The class file *articleCV.cls* contains definitions for a custom command `\tab` that creates a subheading with this structure. These require loading two packages *tabularx* and array.

The `\tab{}` command is defined in the following code block:

```
\newcommand{\tab}[4]{
  \begin{tabularx}{1\textwidth}{@{} $ X ^ r}
  \rowstyle{\bfseries} {#1} & {#2} \\
  \rowstyle{\itshape\sffamily\small} {#3} & {#4} \\
  \end{tabularx}
}
```

`6.\newcommand{\tab}[4]{` creates a new command, with the name `\tab`, that accepts four arguments, which can be NULL. Arguments to a command are supplied within curly braces, which must be present.

Thus in the main file, a header is created by writing the command:

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

Excecution of the `\tab` command simply creates a table, with two columns and two rows, whose arguments are given as inputs.

Formatting of each row is controlled by `3.\rowstyle{\bfseries}` and `4.\rowstyle {\itshape\sffamily\small`, which respectively make the first row bold, and the second row italicized, sans-serif and small. Changing these parameters will change the style of your résumé's subsections.

For more detail on how the `\tab{}` command was implemented, see section 5.1.

5 Building a Résumé

Return to your main file *CVexample.tex*.

Now, you should be able to easily identify where the `\tab` command has been used.

In each case, we can modify the arguments passed to the command to fill out the contents of our résumé. We'll do this section-by-section.

5.1 Educational History

In most cases, it is appropriate to make the first section of your résumé about your educational history; unless you feel that your work experience is particularly relevant for the position you're applying to.

Step one is to rewrite the first `\tab` command, with your most recent qualification. (This will most likely to be your last, or current degree program).

> **Step 8.**
>
> Find the first instance of the `\tab` command in *CVexample.tex* and rewrite the four arguments.

Remember that in your main document, the `tab` command is formatted as follows:

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

The first line (corresponding to arguments `#1` and `#2`) will be **bold and larger**.

The second line (arguments `#3` and `#4`) will be *smaller, italicized and sans-serif*. At first glance, this text will be less noticable.

Consider then, the best way to arrange this information. What should be prioritized for an employer to see?

You probably want your degree program to stand out. This should be the first argument, next to your period of study.

On the second line, you could put the name of your awarding body (i.e. university) and optionally, the city, and country that you studied in.

Directly below the first `\tab` command is the following element:

```
\begin{itemize}
   \item
\end{itemize}
```

Which creates a bulleted list environment.

The list environment is contained within the expressions `\begin{itemize}` and `\end{itemize}`. A bullet point is created with the expression `\item`, followed by plain text.

This is an optional space to add more detail about your qualification.

You may choose to give specific examples of your skills, projects that you worked on, significant achievements or classes that you attended.

> **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.

5.2 Technical Skills

In most cases, your work experience should follow your qualifications. Unless your work experience isn't particularly relevant for the role you're applying to.

I.e. students with part-time work experience that are applying for laboratory research positions. *Sound familiar*?

So instead, try focusing on your industry-specific experience and technical skills; which employers will want to see evidence of.

For example, a lab assisstant position may ask for evidence of your ability to carry out titrations, perform western blot or qPCR analyses.

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

It is useful to research the position that you want whilst, or even before writing your application. Different employers will have different expectations and you should try tailoring your résumé to the job.

With that being said, don't be dissauded from applying for a position you may be underqualified for. Sometimes this is not a deal-breaker.

Step 10.

Make a list of skills that your employer is looking for. Which of these do you already have? Are they included in your résume?

CVexample.tex contains two examples of subheadings, underneath the section *Technical Skills*. Notice that the right-justifed cells have been left empty, because we don't need to include certain information.