

# Instructions

A reference guide for version 2.0

Pedro Henrique Mattos

## Contents

<b>Introduction</b>	<b>2</b>
<b>Get Started</b>	<b>2</b>
Installation . . . . .	2
The Project Directory . . . . .	3
Your First Document! . . . . .	3
<b>Working with T<sub>E</sub>X Files</b>	<b>4</b>
Parts of a Document . . . . .	4
The Header pt.1 . . . . .	5
The Header pt.2 . . . . .	6

## Introduction

This project provides a template to create a curriculum vitae, or résumé. It is intended as a ‘fast-introduction’ to typesetting with  $\text{\LaTeX}$ , for someone with little, or no experience in the subject.

Inspiration came about by the realisation that students might be better-motivated to learn  $\text{\LaTeX}$  if the benefits for doing so were more obvious. Students, inundated with assignments and (oh no!) exams, are 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’; instead, this template rewards the user with a CV, or résumé.

This project introduces the class file `articleCV.cls` and  $\text{\TeX}$  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  $\text{\LaTeX}$  user.

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](https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes).

## Get Started

If you haven’t already, begin by following the instructions located at `Pedro-h-mattos/LaTeXCVTemplate`.

They instruct you to (i) install a  $\text{\TeX}$  distribution, (ii) download a copy of this projects’ files onto your local device, and then (iii) compile `CVexample.tex` to produce a PDF output. This is a good litmus test to see if your software has been correctly installed.

For completion, those instructions are repeated here.

## Installation

A  $\text{\TeX}$  distribution, such as MikTeX, TeXLive or MacTeX, is a prerequisite for working with `.tex` files. They are free and easily available for download online.

$\text{\TeX}$  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  $\text{\TeX}$  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.

## The Project Directory

With the appropriate software now installed, download the package repository as a ZIP file onto your local device.

**Step 2.**

Navigate to <https://github.com/Pedro-h-mattos/LaTeXCVTemplate>, find and click on Download ZIP.

Remember to unpackage the ZIP file before continuing.

The project directory consists of the class file `articleCV.cls` and the main document file `CVexample.tex`. This guide includes `instructions.tex`, `.sty/instructions.sty` and the derived file `instructions.pdf`.

Other files, including `README.md` and the license, do not need to be retained.

## Your First Document!

Compiling `CVexample.tex` should output the document `CVexample.pdf`.

You can always compile documents from the command line, by running the command `pdflatex CVexample.tex`.

If you're using a TeX editor instead, first locate and open the file `CVexample.tex` from within your project directory. Then compile it using the `pdfLaTeX` option.

**Step 3.**

Try compiling your first document; then, open the output file `CVexample.pdf`.

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

Although very neatly typeset, 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.

Look again at your PDF. How might you describe its structure? Can you identify elements that are repeated?

This template has been designed with the standard structure of a résumé in mind; where 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. The header can be divided into three parts: the title, the author's name, and personal/-contact information. The latter is really separate and is arranged as a table with two columns and three rows.

Three sections are labelled as *Education/Qualifications*, *Technical Skills* and *Experience*. They are followed by 'subheadings', comprising two lines of text; one **bold and larger**, and the other *italicized and smaller*, which can extend across the page. Optionally following subheadings are bullet point(s).

Keep this structure in mind, when navigating `CVexample.tex` later.

## Working with T<sub>E</sub>X Files

In most cases, PDFs cannot be directly edited. Instead the original T<sub>E</sub>X file must be recompiled with changes; the corresponding PDF is then updated to match.

The following sections will explain how to navigate and edit T<sub>E</sub>X files.

### Parts of a Document

Returning to `CVexample.tex`, can you identify structural similarities between your T<sub>E</sub>X file and the template?

T<sub>E</sub>X 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 are contained between the declarations `\begin{document}` and `\end{document}`. This includes all the text that is output when the document is compiled.

Like the template, the T<sub>E</sub>X file contains three sections, which are delimited by the command `\section{}` and followed by elements 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 blocks of code. Although it may look intimidating, this is where we'll start editing.

## The Header pt.1

The first part of the document header describes the title, Résumé/CV and the author's name. It looks like:

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

When a  $\text{\TeX}$  document is compiled, only plain text is output. Special characters, e.g. `{}` or commands (prefixed 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 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 created by 25. `\begin{centering}` and 28. `\end{centering}` center the title on the page. Removing these lines would automatically left-adjust the text, which may be preferred.

Lines 26. and 27. contain the text that is output when the document is compiled.

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

Notice that the title and the author are styled differently. Each line is enclosed by a pair of braces, so that the styles don't overlap.

The command `\large` and `\scshape` increase the font size and set SMALL CAPS, respectively. Compare this with the following line, where the font size is set to `\Huge`.

$\text{\TeX}$  has many different font styles, which are worth exploring.

## The Header pt.2

The second part of the header looks more complicated. The output file contains a table with two columns, but these aren't immediately obvious within our TeX file.

```
32. \begin{table}[ht]
33.   \centering
34.   \begin{tabular}{>\small 1 >\footnotesize 1}
35.     <Undergraduate, 3rd Year>, & Email: \Email{\email} \\
36.     \dept, & Tel: \phone \\
37.     \org & \url{\website}
38.   \end{tabular}
39. \end{table}
```

Fortunately, we can once again edit the contents of our header, before worrying about the formatting. The following 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; by rewriting them, we can change what is output in the table. This is a useful trick for describing information that is often repeated and 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 feels like it's above your pay-grade).

The commands `34.\begin{tabular}` and `38.\end{tabular}` initialise a *tabular* environment, which produces a table. Confusingly, the environment created by `32.\begin{table}` and `39.\end{table}` is a float, which controls the position of the table on the page.

Look at `34.\begin{tabular}{>\small 1 >\footnotesize 1}`, which initializes the table.

The letter `l`, repeated twice, indicated that the table contains two left-adjusted columns. Then, `>\small` and `>\footnotesize` prepend the respective styling to either column. So, text is small within the first column and `footnotesize` in the second.

Lastly, `33.\centering` centers everything within the *table* environment on the page.