# LaTeX advanced: a Roadmap

This document contains guidelines for the advanced LaTeXtraining session. Almost all steps can be resolved using the `Documentation/RefSheets` provided in the training kit and of course a bit of Internet search !

## Contents

# 1 Setting the workspace

## 1.1 Install

1. If you haven't done it already, install a TeXdistribution and an editor (follow the instruction provided in the email). You can also use Overleaf if you prefer.
2. test your installation by typesetting the file `main.tex` located in the folder `Templates/Beginners/Article_Beginner` It should produce a `.pdf` file. If not: **call for help !**
3. I recommend creating a workbench folder where you will put all your tests. Copy and paste the templates as needed so to keep a clean version in the `LaTeXTrainingKit` folder.

## 1.2 Set the working environment

1. Install a TeX editor (VS Code recommended)
2. (Optional) Find how this editor typesets the documents
3. (Optional) Find where to customize these commands (in case you need it at some point)

# 2 Document Layout

## 2.1 Page Layout

This section is based on the template `Templates/Advanced/Article_Advanced.tex`

- Typeset the file `main.tex`
- In the `ArticlePackage.sty` file, change the layout of the page. Some features are provided by the `geometry` package.
  - Change the margins (top, left right, bottom)
  - Test the option `twoside`

  - Switch from one to two columns
  - You can also test other configuration as mentioned in the document of the `geometry` package

### 2.2 Headers and footers

- Using the package `fancyhdr`, add the title of the current section at the top right of odd pages, but not on the first page
- Recall the title (or a short version of the title) in the top left of even pages
- Draw a line below the headers
- Place the page number at the center/left/right of the footers
- Adjust the space for headers and footers using the `geometry` package

## 3 The PhD manuscript template

### 3.1 Preamble: working with multiple documents on a minimal template

Use the template `Beginners/SimpleBook`.

1. Split a document into multiple `.tex` files
2. Set a master `.tex` file
3. explore the differences between the commands `\input` and `include` to insert the content of separated `.tex` files into a master document
4. test the use of `\includeonly` in the preamble of the main document

### 3.2 Presentation of the PhD manuscript template

1. Typeset the whole document
2. Locate the front page and the last page source codes
3. Open the content of a chapter and typeset it
4. Take a look at the chapter-paper style
5. Understand the general workflow to edit a chapter independently of the others
6. Try to include an external document using the command `\includepdf`
7. Create a list of figures, a list of tables
8. Create an index using the package `imakeidx`

### 3.3 The Layout

1. Open the file `Toolbox/Layout.tex`.
2. Change the margins
3. Change the position of headers and footers
4. Change the style of the section headings
5. You can also try to customize the chapter title pages
6. Eventually explore other PhD manuscript templates

## 4 Bibliography

### 4.1 Set a reference manager

1. Install a reference manager from the web. Zotero is recommended, but alternatives like Jabref or BibDesk (only on Mac) may also be considered.
2. If you don't have any reference in your manager, import some references from the web in the `.bib`.
3. Export the list of reference as a `.bib` file in your working folder

Alternatively, you can proceed without a reference manager:

1. Create an empty file with the `.bib`
2. Add some `BibTeX` entries to the file. You can copy the reference in `.bib` format from `Google Scholar` for instance.

### 4.2 Bibliography style

You can work on this section using the `Article_Advanced` template.

1. Check the `biblatex documentation` or the `biblatex cheatsheet`.
2. configure your document so that it uses the `biblatex` package
3. make sure that your `.tex` uses `biber` instead of `bibtex` to typeset the list of references. VSCode might do it automatically.
4. Test different `style` options.
5. Test different sorting options
6. See to what extent you can change the way the citations inside the document can be customized.
7. Change the maximum number of authors displayed in the reference list
8. Try to remove some fields from the reference list

### 4.3 Organizing the references

1. Group your references in subsections by type (book, articles etc.)
2. Create collections using keywords or manual groups
3. Explore different options of the `\printbibliography` command

### 4.4 Splitting bibliography

You can work on this section using the `PhD manuscript` template

1. Create a sub-bibliography for each chapter that contain only the references of the chapter.
2. Create a general reference list at the end of the document

## 5 Drawings and plots using Ti*k*Z

This section is based on the templates located in the folder `Templates/Advanced/Figures`. If you already have an idea of a picture you would like to generate, go for it!

### 5.1 Basic drawing

Use the `Demo` template

1. Creates nodes and path between them
2. Create curved paths
3. Draw geometrical forms: circles, arcs, rectangles etc.
4. Position nodes on a path
5. Using the documentation, create a diagram: you can try to reproduce the picture in `Exercises_Advanced/BasicPicture`
6. Use options of the `standalone` class to export in different formats (`png, jpeg` etc.). You will need to install ImageMagick

### 5.2 Simple plot

Use the template `FunctionPlot`.

1. Typeset the plot
2. Change the color and the style of the line
3. Change the size of the plot
4. Change the limits of the axes
5. Add a new function
6. Create a legend

7. Change the position, font, alignment, etc. of the legend
8. Add a title to the plot
9. Change the axis layout : line axes (left right, center), with or without arrows,

### 5.3 Plotting data

Use the template `DataPlot`

1. typeset the plot
2. See how the data files are organized
3. Apply an operation on one of the curves
4. See how the coloring is done
5. See how to control the presence of extra ticks on the axes
6. Add a label on one of the curves using a relative positioning (`pos=0.5` for instance)
7. place a node inside the plot using the axis coordinates system

### 5.4 Organize multiple plots

You can try to reproduce the figure in `Figures/MultiplePlots`.

1. Change the size of the plots
2. Change the relative positioning of the plots
3. Move plot labels

You can now play with the template `Figure/GroupplotExample`

1. Change the spacing between the plots
2. Add two additional plots to create a $2\,X\,2$ array
3. Place the axis descriptions (ticks and axis titles) on the sides (left or right) of the array
4. Add or change the labels ((a), (b), etc.)

## 6 Creating a beamer presentation

### 6.1 Create a presentation starting from a template

Use a template from the `Templates/Beginners/Presentation` folder

1. Open one of the templates
2. Set a title, an author name, an institution picture etc...

### 6.2 Write some content

1. Create sections and a table of content
2. Create frames with titles and subtitles
3. Create blocks to structure your slides
4. Create columns
5. Insert references as footnotes in slides

### 6.3 Overlays

Explore the Beamer user guide and create overlays

### 6.4 Themes and look of the slides

Using the user guide

1. Change the theme of the presentation
2. change the inner/outer
3. use the second template (`presentation_1_custom.tex` )to customize the fonts and the style of specific items

## 7 Single source publishing using Markdown and Pandoc

### 7.1 Setting your Markdown workspace (for VSCode Users)

1. Install the extensions `Markdown All-in-one` and `Markdown Preview Enhanced`
2. (optional)Install the `MarkDownLint` extension
3. In `Markdown All-in-one` preferences disable `Math: Enable` option.
4. In `Markdown Preview Enhanced` settings set option `Math Rendefinr Option` to `MathJax`

**Alternative** You can use alternative Markdown editors for editing. Pandoc is not based on VSCode.

### 7.2 Write a sample Markdown document

1. Create a folder for your project and a `.md` file within
2. Reproduce the content of the exercise on Jacques Hadamard. Use the first level heading (`#`) for the title of the document. Bibliographic references can be omited at this stage.
3. Preview your document
4. Add a table of content to your document

## 7.3 Pandoc setup

1. Install Pandoc
2. Install Python (will be useful later)
3. (Optional) Install the VSCode extension `Pandoc Citer`.
4. Create a `.bib` file with the necessary references, or other random references
5. Create an empty file `pandoc-config.yaml`

## 7.4 First pandoc exports

Starting from now, you should have the Pandoc User Guide available

1. Using the `pandoc` command, export your `.md` file into other format. Try `.docx` and `.pdf` to start with.
2. Do what's necessary to display the equations
3. Play with conversions from one format to another
4. Take a look at the `.tex` ouput
5. Include a `.yaml` frontmatter to your markdown file and include the `title` and `author` fields. See the effect on the outputs

## 7.5 Adding references

1. Create a `.bib` file containing references
2. Add citations to your `.md` file
3. Convert the document to a `.tex` file using Pandoc and the appropriate filter
4. Use a `.csv` file to change the layout of the references

## 7.6 Cross-references

1. Install the pandoc filter `pandoc-xnos`. Python is needed for that filter to work
2. Implement cross-referencing in your document so that equation numbers, figures and eventually tables can be referred to throughout the document.

## 7.7 Use a configuration file to ease the conversation

1. Create a `.yaml` file containing all the above mentioned configurations
2. Test the export in various formats using this this `.yaml` file as a so called `defaults file` in the pandoc command

### 7.8 Other things to consider

You may consider the following to enhance your experience with Pandoc

- Check out some Personal Knowledge Management apps like `Obsidian` or `LogSeq` or `Zettlr`. Your can alternatively try VSCode Extensions `Dendron` or `Foam`.
- `Zettlr` is a good entry point if you want to have
- You can change the default settings of the `MarkDown Preview Enhanced` VSCode extension, so that it uses the pandoc parser for rendering your `.md` files
- If you know how to program a `markefile` you can use it to refresh the output of multiple documents at once

## 8 Tips, commands and other geek stuff

### 8.1 Custom commands

You can use the `Article_Advanced` template

1. look into the `commands.tex` file and see how commands can be defined
2. 
3. Create a custom command using `\newcommand{\name_of_command}{...}`
4. Create a command with arguments
5. Create some commands, with or without arguments
6. 
7. Create a command that ease the writing of derivatives
8. Improve your command for derivative including an optional argument setting the order of the derivative
9. 
10. ... then take a look at the physics and diffcoeff packages.

### 8.2 Units in LaTeX

1. Use the `siunitx` package
2. Write numbers with exponents like $10 \times 10^3$ without using the math environment. Use the command `\num`
3. Write a number range
4. Write a number with a unit using `\qty{}{}` or `\SI{}{}`
5. Write complex units
6. Create a new unit
7. Change the way number range are shown

### 8.3 For VSCode users

1. Create snippets to accelerate your writing
2. Create custom typesetting rules

## A (Incomplete) List of useful packages

- `babel`
- `graphix`
- `amsmath`
- `geometry`
- `hyperref`
- `fontspec` and `mathspec`
- `tikz` and `pgfplot`
- `siunitx` to handle units
- `esvect` to draw nice arrow vectors
- `erewhon` to have a nice very complete and rather compact font
- `physics` providing current operator commands
- `booktabs` for nice looking tables
- ... (if you have any idea to increase that list, all suggestions are welcomed)