# LaTeX Workshop

Learn how to write:

```
 Use \LaTeX{} to write your reports:
\[
      \sum_{i=1}^\infty \frac{1}{2^i} = 2.
\]
```

To obtain:

Use LaTeX to write your reports:

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 2.$$

# Organization of the workshop

- Presentation and interactive exercises
- Break
- Exercises and questions

# Outline

1. Introduction
   - How to get started?
   - Basic document structure
2. Write in LaTeX
   - Math and symbols
   - Figures and tables
   - Include code
3. Conclusion
   - Tools
   - To go further

# What is LaTeX?

LaTeX is a language used to create documents like reports, articles or presentations. That language need to be compiled to produce a PDF document.

## File format

You need to keep the file with the `.tex` extension to work on, and the `.pdf` file to view the result. When you make a report, you only submit the PDF file. Compiling also produces auxiliary files like `.aux`, `.log` or .out (among others) that you can ignore.

When you write internal references, you might need a second compilation for the links and the table of content to be correctly generated.

# How does LaTeX work?

The `.tex` file and the `.pdf` are very different. You focus on the content, not the way it is displayed.

**Example**

To write a section, you only type `\section{Title of your section}` and LaTeX will automatically generate the number corresponding to this section.

You don't want to make any hard coded reference (to a section or an equation). Instead, you should be using references.

# Where to write your LaTeX documents?

Depending on your needs choose from :

- In a dedicated LaTeX editor (TeXstudio, TeXmaker)
- In an online LaTeX editor (Overleaf)
- With a LaTeX plug-in in your favourite editor (Emacs, Vim, ...)

If you are in unsure, a dedicated editor should be the best option.

# Packages installation

LaTeX relies on the installation and the use of packages for specific features that you need.

- Online editors usually deal with the common packages automatically.
- On linux distributions, you can install packages with the packages `texlive` (refer to the documentation of your distribution for more details).
- On Windows, you can use a package manager like `MiKTeX` that has the option to install missing packages when needed.
- On Mac, `MacTeX` seems to provide a configured version of `TeX Live`

# First Latex document

```
\documentclass{article}

\begin{document}
        Hello world!
\end{document}
```

# Exercise on documentclass and structure

Download the folder with the exercises for the workshop on absalon. Start with the file `structure_and_documentclass.tex`

Look at the different comments and observe how the document is rendered when you apply each of those comments.
Try to identify the different structures of the commands in that document.

# Comments on the exercise

- All commands start with \
- Some of them can take an argument inside {}
- The environments are expressed with:

```
\begin{*environment-name*}
content...
\end{*environment-name*}
```

## Nested environments

Always remember to close each environment you open. If you have several environments, the inner most have to be closed before the outer ones, otherwise it won't compile.
The same goes for the arguments in {}.

# Structure your document

Observe in the exercise document that sections (and subsections, enumerate and itemize environments) are automatically numbered.

You can remove the numbering on one element by adding a $*$ after the declaration of the section.

You want to use labels and cross-references when you want to refer to another section (or subsection, figure, table, equation . . . )

# Math mode

To write math equations, we need to be in math mode.
There are different ways to enter math mode:

```
Not numbered equations:
\( in-line equation \)
\[ equation on its own line\]
\begin{align*} multi-line equations \end{align*}

Numbered equations:
\begin{equation} equation on its own line \end{equation}
\begin{align} multi-line equations \end{align}
```

# Exercise

Now use the `math.tex` document to observe and play with the different syntaxes in sections 1 to 4. Could you reproduce the following equation?

$$\begin{cases} f(x) = 2x \\ g(x,y) = x^2 \end{cases} \quad \Rightarrow \frac{\partial g(x,y)}{\partial x} = f(x) \tag{1}$$

# Vectors and matrices

As simple way to write a matrix (or a vector) would be to use the `bmatrix` environment. If you use the `matrix`, you need to manually insert the brackets around the matrix.

```
\[
    \begin{bmatrix}
    1 & 2 & 3 \\
    4 & 5 & 6
    \end{bmatrix}
\]
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

# Figures

A basic structure to display a figure:

```
\begin{figure}
\centering
\includegraphics[width=0.5\linewidth]{my-fig}
\caption{Don't forget to add a caption!}
\label{fig:my-fig}
\end{figure}
```

# Figures

Figures have to be displayed in a `figure` environment, but you can put several figures in the same environment. There are even packages that can provide a subcaption to each figure.

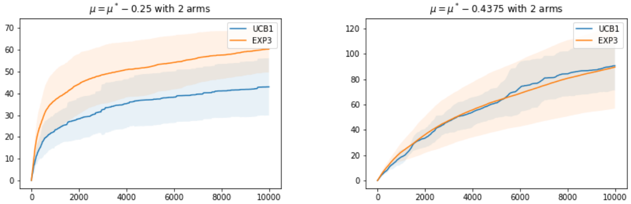

Figure 1: Don't forget to add a caption!

# Tables

Tables are constructed with the same use of & and \\ as in the align environment.

```
\begin{table}
\centering
\begin{tabular}{l|r}
Item & Quantity \\\hline
Widgets & 42
\end{tabular}
\caption{\label{tab:widgets}An example table.}
\end{table}
```

| Item | Quantity |
|---|---:|
| Widgets | 42 |

Table 1: An example table.

# Include code

Many packages can include code in LaTeX. Among others:

- `lstlisting` is rather straight forward to use
- `minted` is more diffcult to install

Only load one of those packages in a file as they can conflict. Both can take a language as a parameter to improve highlighting.
You can write code in the `lstlisting` environment, or load an entire file using the `\lstinputlisting{my-source-filename}` command.

## Indentation

In your LaTeX source file, the indentation doesn't matter unless you are inserting code. Be very careful as the tabulations might not be rendered by the same amount of white-spaces in your editor and your pdf.

# Exercise

Now look at the remaining parts of the `math.tex` file, and try to answer those questions.

By now you should be able to produce a good report. In case of doubt, at a look to the tools seconds or ask. Also, the LaTeX community is quite large online so do not hesitate to search for that too.

# Common tools to help you get started

- List containing most LaTeX symbols you will ever need: `https://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf`
- Hand-draw a symbol to obtain the list of corresponding possibles commands: `http://detexify.kirelabs.org/classify.html`
- Fill a table and generate the corresponding code in LaTeX: `https://www.tablesgenerator.com/`

# To go further

This is outside the scope of this workshop, as those elements are not necessary to produce good reports but that a more advanced LaTeX user might want to use.

- Create and use your own macros.
- Install and use a template with the university logo (in particular for projects and thesis).
- Structure large documents into several source files and load them in one main source file.
- Use the minted package to highlight source code included in your report according to its language.