
InfPALS L^AT_EX Activity 1

Beyond Text

Lists, Tables and Graphics

Art begins in imitation and ends in
innovation.

Mason Cooley

1 Introduction

By now you should be able to write basic L^AT_EX documents with headings and formatted text. While there is no doubt that you can do rather interesting things with just that, you'd probably like to add lists, tables and graphics at one point or another. The next couple of sections will let you explore your options for these.

2 Packages

Before we move on to the promised material there's a little something that hasn't been covered yet: packages. While L^AT_EX delivers a substantial range of functionality by default, you'll discover certain things which can't be done with our current toolset. This is where packages come into play. Packages extend the basic set of commands you can use and are essential for more advanced tasks like writing mathematics.

As we will need them for almost everything from here on out we need to be able to import arbitrary packages into our documents before we move on. To import a package called `pkg` (just as an example) we use the command:

```
\usepackage{pkg}
```

in the preamble of our document. Remember that this is the part after the `\documentclass` and before the `\begin{document}`.

3 Comments

Just like in any other computer language, you can, and should add comments to your code, whenever you think it may need an explanation. Additionally, even more in L^AT_EX than other cases, this visually helps you find sections within the document.

Inline comments start with %. There is no basic multi-line comment symbol, but if you want to exclude a section, you can use a little hack:

```
\iffalse
    This is ignored by the compiler!
\fi
```

4 Lists

A list of bullet points is arguably the simplest kind of list you could come up with. Luckily, it is also very simple to produce one in L^AT_EX:

```
\begin{itemize}
    \item One entry \\\
    \item And another \\\
\end{itemize}
```

There is of course [a lot more to explore](#). Click on the link in the previous sentence and see if you can reproduce the following:

- I. First level
 - (i) Second level
 - (ii) Second level
 - * Third level
 - * Third level
- II. First level
 - (i) Second level
 - (ii) Second level
 - * Third level
 - * Third level

Hint: For the spacing between the items try `\setlength{\parskip}{0pt}` before the first `\item`.

5 Tables

Next we'll explore tables a little bit. The basic idea is simple: you provide L^AT_EX with a data set where you separate columns with & and rows with \\:

```
cell 1 & cell 2 & cell 3 \\\
cell 3 & cell 4 & cell 5 \\\
cell 7 & cell 8 & cell 9 \\\
```

Of course this is not the whole story. Your table needs a little bit of set-up. Try putting the above code within the following environment:

```
\begin{tabular}{c c c}
\end{tabular}
```

The funny `{c c c}` after the `\begin{tabular}` is there to tell the compiler how many columns your table is meant to have. Try doing the table without that part and see what happens.

Now the first thing you should notice in the above example when it's compiled is that there are no border lines like you'd usually expect them in a table. They don't come by default, but we can tell \LaTeX to add them in for us. For vertical border lines replace the `{c c c}` with the following

```
{| c | c | c |}
```

Remove a couple of vertical bars (`|`) and see how this changes the look of your table. After the vertical borders we'll do horizontal ones. This requires a bit more work. Everywhere you want a horizontal border line to appear you need to put in a `\hline`. To have them before and after each row you could change your table data set to look like this:

```
\hline
cell 1 & cell 2 & cell 3 \\
\hline
cell 3 & cell 4 & cell 5 \\
\hline
cell 7 & cell 8 & cell 9 \\
\hline
```

As with the lists, there is a [a lot more to tweak](#). Have a look at the link in the previous sentence and try to replicate the following table:

cell 1	cell 2	cell 3
cell 3		cell 5
cell 7	cell 8	cell 9

Hint: Some magic numbers: 2cm - 3cm - 2cm, `lightgray!40`, `gray!40`.

6 Graphics

The last topic that this activity will cover is that of graphics. Getting started is actually quite easy. First, import the package `graphicx` and once that is done there is one main command you'll need:

```
\includegraphics{<filename>}
```

where `<filename>` is meant to be a placeholder for the actual filename (plus file extension) of the image that you'd like to insert at that location. If, however, your image is stored as a PDF you can omit the `.pdf`.

The picture on the right is a file called `good-question.jpg` which you should hopefully have access to by now. You will need to upload this picture to Overleaf. To do this, go to the menu on the left and click on the "upload" button (third one at the top of the menu). Choose the image that you want to upload, and you will see it appear in the menu with the rest of the project.

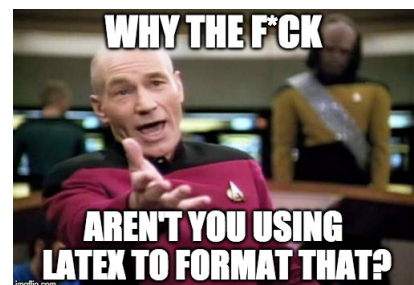


Figure 1: The Ghost of Assignment Yet to Come

You'll notice that the result will look a bit different from what you see in this document. One of the reasons is that the picture in here has been scaled, which brings us to options for commands. If you want to give a command additional information (which the command must of course accept in the first place), you can do so by putting that information in square brackets as follows:

```
\includegraphics[scale=0.3]{good-question.jpg}
```

This will scale the image to 30% of its original size. This option syntax by the way is a general pattern:

```
\command[option1=value1,option2=value2,...]{...}
```

Again, there is vastly [more to be fine-tuned](#). Go ahead and click the link and see if you can figure out how to position the image in your document like it's done here. Try also to give it a caption similar to the one in this document.

As a last treat we'll briefly talk about cross-referencing. If you're writing your document around a certain graph illustrating your data for instance, it would be a bit tedious to constantly have to refer to it as "the graph on page such and such", making sure your reader knows what you're talking about. It would be rather pleasant if you could simply say something like: figure [1](#). Click on the number in the previous sentence and see what happens.

The way we achieve this in L^AT_EX is with the help of a little command called `\ref`:

```
\ref{<label>}
```

where `<label>` is the label of whatever we're referencing. Now go ahead and [find out how to add labels](#) and then refer to your figure in your document just like we did here in this one. Note that labeling isn't restricted to graphics only. You could label and [reference mathematical equations, tables and so on](#).