InfPALS LaTeX Activity 3

The Game Changers

Macros and Biblatex

> Elegance is not a dispensable luxury but a factor that decides between success and failure.
>
> ———————————————
>
> Edsger Dijkstra

1 Introduction

By now you've seen a lot of what LaTeX has to offer you. Text formatting, tables and mathematical equations are crucial parts of many documents and LaTeX handles them particularly well. We're now going to look at two features of LaTeX that go far beyond word processing. Both of them give us a rather elegant way to make our documents modular and adaptable to change.

2 Macros

LaTeX offers you wonderful ways in which to typeset your document just the way you like it. However, there's often quite a bit of boilerplate involved with custom styling and you'll end up writing the same code over and over again. If requirements then change you're left with no other option than to go through the entire document and fix everything manually.

But of course LaTeX wouldn't be LaTeX if it wouldn't offer a neat solution to that: macros. To understand these we first have to clarify two terms we've used more or less interchangeably: command and environment.

A command is something like `\emph`. It takes one or more arguments within one or more sets of curly braces right after it, e.g. `\emph{this}`, and stands on its own. An environment is anything that you've put in the `\begin{}` and `\end{}` commands, so for instance `itemize`.

In this activity we'll limit ourselves to defining new commands and leave defining new environments for you to explore later. But don't worry, new commands will get us a long way. The command to define new commands is as follows:

```
\newcommand{<command_name>}[<num_args>]{<definition>}
```

It's best if you read up the details of the syntax yourself. The one thing to note is that in LaTeX, much like in Bash, arguments are positional. That is, they are addressed in the order in which they are are provided. The first argument is accessed via `#1`, the second via `#2` and so on.

We're now in the position to define a command to typeset, for example, quotes:

```
\newcommand{\myquote}[2]{\begin{flushright}#1\\-\emph{#2}\end{flushright}}
```

Once we've defined it (best done in the preamble) we can use it like this:

```
\myquote{Yes we can!}{Bob The Builder}
```

Note that each of the arguments is given in a separate pair of curly braces. Try the command out to see what it does and try to understand how it does it. Once you're done with that try to write a command that when called with `My Heading` as argument produces the following:

My Heading

Hint: There's [one extra command](#) you'll need. Oh and try `\textwidth` for the width.

3   BibTeX

One of the key principles in LaTeX is to separate presentation from content. This comes in particularly handy when we get to things like citations and references. If you ever had to do either of these in a regular word processor like Microsoft Word you'll probably have discovered that it can be quite a pain making sure that everything is cited correctly according to regulations and so on. As with many things, LaTeX offers us a rather pleasant way to handle such situations: Biblatex.

Biblatex is a bibliography management package and automates the task of typesetting citations and references for us based on an input bibliography file which we'll discuss shortly. In order to use Biblatex you first want to include the package `biblatex` in your preamble in the following way:

```
\usepackage[backend=bibtex]{biblatex}
```

The bibliography files are documents with a layout like the provided `good_read.bib`. They will always have the file extentions `.bib`:

```
@book{rose,
author      = "d'Antoine de Saint-Exupery",
year        = {1943},
title       = "Le Petit Prince",
publisher   = "Editions Gallimard"
}
```

This particular file defines a book labelled `rose` (this will be relevant for citations later), with appropriate author, year it was published and so on. You can have a look at [possible fields to be used in bibliography files](#).

To include the bibliography file `good-read.bib` we simply add the following line to our preamble:

```
\addbibresource{good-read.bib}
```

We are now able to cite all the books, article, etc. that are defined in our bibliography file. To do so we use the `\cite` command:

```
\cite{<label>}
```

Simply replace `<label>` with the appropriate label, in the case of the book defined in our file it would be `\cite{rose}`. What you should see is a `[1]` refering to bibliography item one. This is in itself of little use for the reader of our document as we haven't actually shown them what item one is. To print all bibliography items which you've made use of throughout your document add the following command at the position you want your bibliography to be rendered:

```
\printbibliography
```

And that is it, it's this simple: you create a file that contains relevant information for your sources, you include it, cite it and LaTeX handles the rest. Of course there is one more treat for you. Different organizations tend to use different citation styles. If for instance you're required to have author and year of the book you're referring to instead of just brackets and a number, you don't need to change every single citation manually. Simply change the style of citations by passing Biblatex the following extra option:

```
\usepackage[backend=bibtex, style=authoryear]{biblatex}
```

Have a look at your styling options. Play around a little and see if you can replicate the following bibliography for Adams 1981, Lascarides 1991 and *How to Become a Hacker*:

References

Adams, Douglas (1981). *The Hitchhiker's Guide To The Galaxy*. Pan Books.

Lascarides, Alex (1991). "The Progressive and the Imperfective Paradox". In: *Synthese* 87.6, pp. 401–447.

Raymond, Eric Steven. *How to Become a Hacker*. URL: http://www.catb.org/esr/faqs/hacker-howto.html. (accessed: 01.09.2016).

Hint: You should probably investigate the bibliography file example.