Documentation for using Atom and Pandoc

Packages and atom settings

- The command palette can be opened by CTRL+SHIFT+P (called CP from now on)
- pdf-view allows to see pdfs inside platformio
- Settings > Editor set tabtype to soft and two spaces.
- Packages > autocomplete plus > autocomplete only on tab (otherwise also completes on enter)
- To see invisibles, press window: toggle invisibles in CP
- Markdown document outline package is helpful to nagivate quickly. Needs a vertical split though
- vim-mode-plus to get vim into atom

Pandoc

To use Pandoc to convert to pdf, you need the package platformio-ide-terminal. This allows to use the terminal inside of Atom. Once installed, it can be called with the command Ctrl-‘.

Note you need to install latex separately and add it to your path.

These are the commands for using pandoc. Call the commands in the terminal.

pandoc markdownfilename.md -s -o view.pdf
Creates a pdf called view.pdf from the file markdownfilename.md.
Option -s for standalone and option -o for output.
Add –toc for table of contents

Markdown

Beginning a document

Between triple lines (—) write the title and author, as follows:

---
title: Something

author: myself

Lists

Do two enters and - (slash). Looks like this:

- First
- Second

To indent the lists, use a tab:

- First
    - Second

Small but useful

To do a single linebreak, use two spaces and enter. To change paragraph, use two enters. A single enter doesn't do anything.

To write taken characters, use the escape character backslash \.
Known taken characters are:

‘

Using Git

First install Git. It is required to work properly.
On Git bash, write the following commands for set up:
git config –global user.name “Napoleon Freitas Paajanen”
git config –global user.email “napoleonfreitaspaajanen@gmail.com”
currently using the following sites:

Basics of git: https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Possibly in the future use “git-control” package for easier use.

Some possible problems that can occur:

- Windows stores the username/passwords in a place called “Credential manager”. Check that they are correct!

- To set up SSH, go to git bash and check `~/.ssh`. If you don't have that, you need to create it, easiest probably with ubuntu and and write `ssh-keygen -t rsa -b 4096 -C "your_email@example.com"`. Then you want to copy the public key using `cat ~/.ssh/id_rsa.pub | clip` and paste that to the SSH of the git server. Then you should hopefully be done.

Initialize git

This can be done inside Atom.

- File > Open Folder > “find the folder you want to control”
- Press the arrow on the right side of editor (not visible until you put your mouse over there) of Atom to open the git panel
- Press Create Repository > init

Pushing an existing repository to github

- First initialize git on the folder (separate instruction)
- Then commit some initial files (separate instruction)
- Next start a new repo in github with the same name as the folder in your computer, but don’t initialize with a README (don’t click on that option)
- Launch git bash on the target folder in your computer and do
  - git remote add origin “name of github url”
  - git push -u origin master
  - You might need to put some passwords or something at this stage

Clone a repository

- Go to github and get the “clone url”
- Open git bash and write git clone “clone url”

Staging, adding, committing and others.

This can all be done internally in Atom so one does not need to know commands. However, if one wants to use the git bash, one can use: - Stage all: git add . - Need to add more when I feel like it. For now, use Atom.

Latex

In order to use Latex commands, use double dollars $$. E.g.,

$$y \in R(A)$$

Adding a package
In the header, set the following:
header-includes: |
\usepackage{amsthm}

Coding

```
Code block
if mood == 'happy':
  print('hello')
```