Awesome Note taking Workflow for a PhD

This is a workflow highly inspired by the Simple Research Journal by Julian Straub during my stay at Oculus Research. I liked the simplicity of his approach and saw its potential to combine it with sophisticated but still simple tools for searching or formatting so that not only taking notes is simple but also sharing and refering back to them.

Requirements

- Take daily notes.
- Review papers.
- Summarize meetings.

The Philosophy

In the end, this philosophy is implemented with the following toolchain:

- Markdown as format.
- VIM for text editing.
- Pandoc for document conversion.
- Evince for PDF viewing.
- TeX Live as LaTeX distribution.
- Jekyll for websites.

The Toolchain

Markdown

VIM

Evince

TeX Live

Jekyll

Pandoc

The Workflow

Setting up variables and metadata

Remember to execute `source ~/bashrc.` or reopen the terminal to apply the changes and be able to use new functions, variables, and aliases.

```
export PHD_WORKFLOW_HOME="$HOME/phd"
export PHD_WORKFLOW_AUTHOR="Alberto Garcia-Garcia"
```

Creating the directory structure

Before starting to use this workflow, you must create the following directory structure in your `$PHD_WORKFLOW_HOME` folder.

```
. `$PHD_WORKFLOW_HOME`
+-- README.md
+-- entries
+-- papers
+-- meetings
```

Taking daily notes

Add an alias to your `~/.bashrc` file – namely `j` – which will automatically open a file named after the current date with VIM in the notes folder. That alias will call `open_jotdown()` which will create the entry for the current date if it does not exists or reopen it if it does. The entry will be created with a YAML header containing metadata about the title, author, date, and tags.

```
alias j=open_jotdown

open_jotdown()
{
    local FILE="$PHD_WORKFLOW_HOME/entries/$(date +%F).md"

    if [ ! -f ${FILE} ]; then
        echo "Creating new jotdown file!"
        echo ${FILE}
        touch ${FILE}
        echo "---" >> ${FILE}
        echo "title: \"$(date +%F)\"" >> ${FILE}
        echo "author: [$PHD_WORKFLOW_AUTHOR]" >> ${FILE}
        echo "date: $(date +%F)" >> ${FILE}
        echo "tags: [Entry]" >> ${FILE}
        echo "..." >> ${FILE}
        vim ${FILE}
    else
        vim ${FILE}
    fi
}
```

Converting entries to PDF or TeX

Sometimes it can be useful to convert any entry into a PDF file for sharing, printing, or just for the sake of reading it in a beautiful format. It can also come in handy to convert it to LaTeX to reuse it in a paper or any other LaTeX document. To do this easily, add the following function to the `~/.bashrc` file:

```
pandoc_convert()
{
    local FILE="$1"
    echo $FILE
    local OUTPUT_FILE="${FILE%%.*}.$2"
    echo $OUTPUT_FILE
    pandoc -s $FILE -o $OUTPUT_FILE
}
```

After reopening the terminal or executing `source ~/.bashrc` you will be able to call `pandoc_convert [entry.md] pdf` which will generate a PDF file with the same file name as the entry you provided or `pandoc_convert [entry.md] tex` to generate a LaTeX source file.