Pandoc Markdown Example

Contents

This is a heading

The headings appear as sections in the table of content. If your pdf viewer supports document outlines, the headings will also appear there. The table of contents is automatically generated from the headings by Pandoc. You can generate it in VSCode or write it manually instead if you want to.

This is a heading 2

This is a paragraph.

markdown text can have *italics*, **bold**, `monospace` and ~~strikethrough~~ in it. underlined text isn't a thing in markdown, but u can do it in LaTex, and in HTML too, LaTex has dunderline text as well.

This is a new paragraph, separated from the previous by two newlines.
In order to have a line break in a paragraph, you need to end the line before it with two or three spaces.

BibTex references can be added in pandoc, and zotero references can be saved as BibTex. [1]

This is an a formatted c++ code block:

```cpp
int main() {
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

A lot of other languages are supported, as well as console output, and even LaTeX code.

```
$ echo "Hello, world!"
Hello, world!
```

This heading is hidden from the table of contents

1. This is a list
2. It numbers itself
3. automagically

this	is	a table
the	first	row
the	second	one

- unnumbered
- lists
- are a thing
- too

heading 3

Headings > 3 are inlined into paragraphs

heading 4 Like this

heading 5 And this too,
but LateX doesn't like headings 5 and 6

heading 6 looks like regular text, btw

Version Control??

Yep! You can use git to version control your markdown files, and vscode has a built in git interface. You can also use github to host your markdown files, kinda as a backup and a way to share them with others. This paragraph is in a new commit from the previous. Every commit is a point in the project's time that

you decided to save and assign a name to. You can always go back to any commit, so you should create a commit for every significant addition or change to your project. Git can be used to collaborate with others, but that's a bit complicated for a text editing workflow.

AI assisted text editing

You can use your university email address or your student ID card to get a free github pro subscription that allows to use GitHub Copilot in VSCode. GitHub copilot suggests predictions of what you're going to type, and you can ask it questions in the document or in the comments. It can help with git commands, md and latex syntax or even answer general trivia.

In order to ask a question you need to precede it by "q:" or it might not suggest a response to it. Here's an example:

- q: Who is the queen of England?
- a: Elizabeth II

References[1] [2]

[1]   D. Hasselquist, C. Lindström, N. Korzhitskii, N. Carlsson, and A. Gurtov, "QUIC Throughput
       and Fairness over Dual Connectivity," *Computer Networks*, vol. 219, p. 109431, Dec. 2022, doi:
       10.1016/j.comnet.2022.109431.

---

[1] Inserting refs is optional, if you don't insert them in the doc, pandoc adds them to the last page.
[2] This is a footnote, btw.