

# Pandoc Markdown Example

## Contents

<b>This is a heading</b>	<b>2</b>
This is a heading 2 . . . . .	2
heading 3 . . . . .	3
<b>Version Control??</b>	<b>4</b>
<b>AI-assisted text editing</b>	<b>4</b>
<b>References</b>	<b>5</b>

## 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

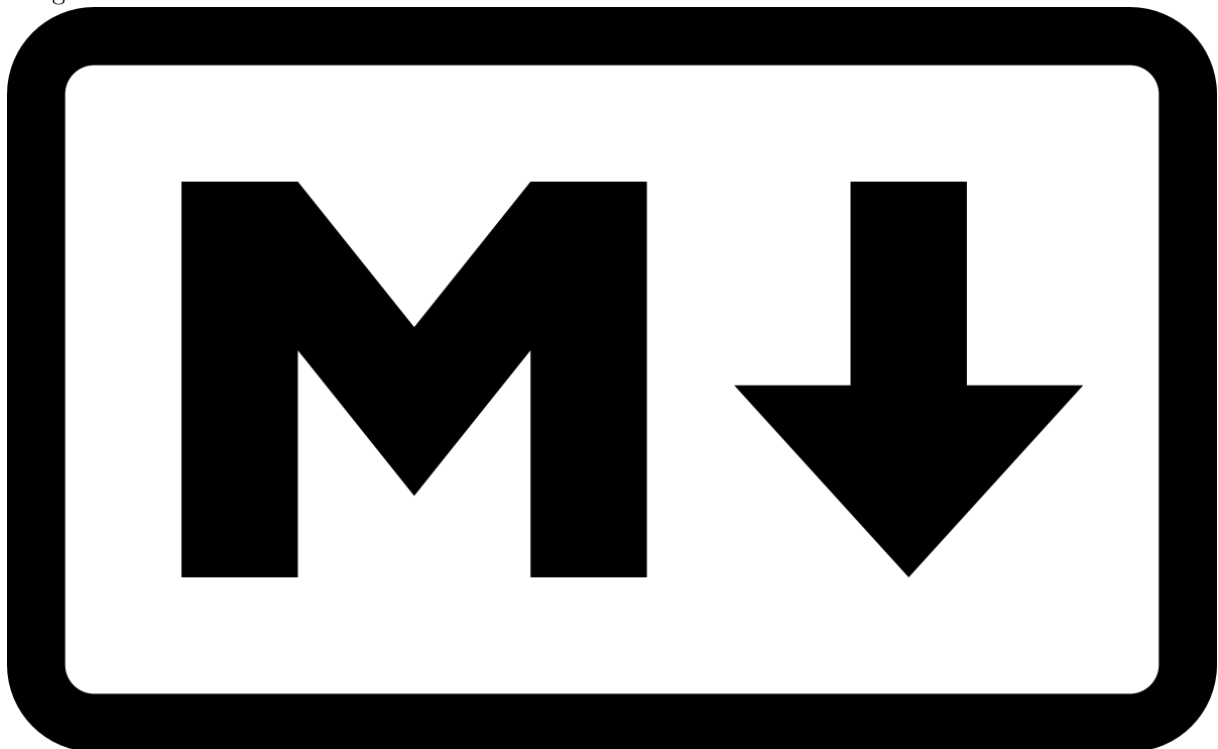
This is a paragraph.

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. To have a line break in a paragraph, you need to end the line before it with two or three spaces.

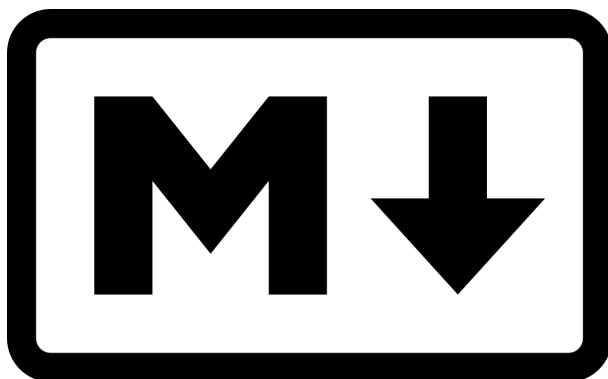
BibTex references can be added in pandoc, and zotero references can be saved as BibTex. [1]

Images can be inserted in markdown like this:



but this feels so out of control, right?

Using LaTeX, images can be scaled and positioned in the document.



Here is a bunch of text that we can put around an image. The image will be shown on the right side of the paper, but our text can go all around it. There will be spacing above and below the image where text can go. This spacing can actually be controlled by using a negative value with `vskip`, but we will not be going into that here. It will be just our little secret: you, me, and the compiler. I hope this is enough dummy text for this example. If not, I will have to type up some more. Oops, I just tried compiling it. Apparently I need to include a bit more. I could go on a bit and tell you more secrets of image placement. Did you know that you can use the `center` environment around the `\includeimage` to make things look even prettier? It's true. The control which  $\text{\LaTeX}$  provides with it is nice. Figures are automatically numbered and can be referenced in the text using their label.



Figure 1: This lion is the TeX mascot

Example: Figure (1) is TeX's picture.

Links can be added as follows this is a link to google, you can also link to other sections in the paper using the heading's id, which is usually the heading's text all lowercase with `-` instead of spaces, if you have duplicate heading names the IDs will be suffixed with `-1`, `-2`, etc.

Example:

this is a link to the heading 2. `#` This is a heading This is a link to this heading

Heading links can also be copied from the table of contents, or the markdown preview.

This is a formatted c++ code block:

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

A lot of other languages are supported, as well as console output, and even  $\text{\LaTeX}$ code.

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

```
Hello, world!
```

This is a blockquote. It can span multiple lines,  
and has the same rules as a paragraph.

## This heading is hidden from the table of contents

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

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 one. 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 u 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 the comments. It can help with git commands, md and latex syntax or even answer general trivia.

To ask a question you need to precede it with "q:" or copilot might not suggest a response to it. Here's an example:

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

## References<sup>1 2</sup>

- [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.

---

<sup>1</sup>Inserting refs is optional, if you don't insert them in the doc, pandoc adds them to the last page.

<sup>2</sup>This is a footnote, btw.