Pandoc Document converter

START HERE

`pandoc <in> -o <out>`
Convert; formats from extensions

`pandoc -f <fmt> -t <fmt>`
Set formats explicitly

`pandoc -s <in>.md -o <out>.html`
Full HTML page

`pandoc --toc -N -s ...`
Contents and numbered headings

`pandoc <in>.md -o <out>.pdf`
PDF via a LaTeX engine

`pandoc -C --bibliography=<f>.bib`
Process citations

`pandoc <in>.md -o <out>.docx`
Word document

`pandoc --list-output-formats`
See supported formats

CONCEPTS

Pandoc reads any input format into an internal document tree, then writes any output format. Formats are guessed from file extensions; -f/-t override that. Without -s you get only a fragment, without header or footer.

Input markdown, docx, html...
↓ reader
Document tree filters edit here
↓ writer
Output html, pdf, epub...

Filters run on the tree between reader and writer

BASIC CONVERSION

`pandoc <file> -o <out>`
Convert, formats from extensions

`pandoc -f <fmt> -t <fmt> <file>`
Choose input and output format

`pandoc -s <file> -o <out>.html`
Standalone page with head and body

`pandoc <a>.md <b>.md -o <out>`
Concatenate several inputs

`cat <file> | pandoc -f <fmt> -t <fmt>`
Read stdin, write stdout

`pandoc -o <out>.pdf <file>.md`
Make a PDF

Needs a PDF engine such as pdflatex, xelatex, typst or weasyprint on PATH

`pandoc --pdf-engine=<prog> ...`
Pick the PDF engine

`pandoc -t plain <file>`
Plain text output

`pandoc -f markdown -t gfm <file>`
Normalise to GitHub Markdown

`pandoc <file>.docx -t markdown --wrap=none`
Docx to readable Markdown

`pandoc --extract-media=<dir> <file>.docx -o <out>.md`
Pull embedded images out

`pandoc --embed-resources -s <file> -o <out>.html`
Single self-contained HTML file

`pandoc -d <defaults>.yaml`
Load options from a defaults file

BASIC CONVERSION (cont.)

`pandoc --wrap=none|auto|preserve`
Control line wrapping

FORMATS AND EXTENSIONS

`pandoc --list-input-formats`
Readable formats

`pandoc --list-output-formats`
Writable formats

`pandoc --list-extensions=<fmt>`
Extensions of a format and defaults

`-f markdown+<ext>`
Enable an extension

`-f markdown-<ext>`
Disable an extension

`-f gfm / commonmark`
Strict Markdown dialects

`-t html5 / latex / beamer`
Common output formats

`-t docx / odt / epub / pptx`
Binary document outputs

`-t revealjs / slidy`
HTML slide shows

STANDALONE, TEMPLATES AND STYLING

`pandoc -s ...`
Add header and footer

`pandoc --toc --toc-depth=<n>`
Table of contents to depth n

`pandoc -N`
Number section headings

`pandoc -V <key>=<val>`
Set a template variable

`pandoc -M <key>=<val>`
Set a metadata field

`pandoc --metadata-file=<f>.yaml`
Metadata from a YAML file

`pandoc --template=<f>`
Use a custom template

`pandoc -D <fmt>`
Print the default template

`pandoc --css=<url>`
Link a stylesheet in HTML

`pandoc -H <f> / -B <f> / -A <f>`
Include in head / before / after body

STANDALONE, TEMPLATES AND STYLING (cont.)

`pandoc --reference-doc=<f>.docx`
Style docx, odt or pptx from a file

Get a starting file with: pandoc -o custom.docx --print-default-data-file reference.docx

`pandoc --syntax-highlighting=<style>`
Code highlight style, or none

Since 3.8; replaces the deprecated --highlight-style

`-V geometry:margin=2cm`
LaTeX page margins

`-V fontsize=12pt`
LaTeX base font size

`-V mainfont=<name>`
Main font with xelatex or lualatex

`--resource-path=<dirs>`
Where to look for images and includes

A variable is set with -V or in the metadata block; -M and YAML both fill metadata. Templates read both as $name$.

MARKDOWN SYNTAX

`# Title {#id .class}`
Heading with id and class

`# Title {-}`
Unnumbered heading

`::: {.note} ... :::`
Fenced div with a class

`| a | b | with |:--|--:|`
Pipe table, colons align

`text[^1] and [^1]: note`
Footnote

`$x^2$ and $$...$$`
Inline and display math

`![alt](img.png){width=50%}`
Image with attributes

```` ```python ````
Fenced code with language

`` `<b>x</b>`{=html} ``
Raw inline content for one format

YAML metadata block

```
---
title: My report
author: [Ada, Grace]
date: 2026-10-11
toc: true
---
```

CITATIONS

`pandoc -C --bibliography=<f>.bib`
Resolve citations with citeproc

`pandoc -C --csl=<style>.csl`
Choose the citation style

`[@key]`
Parenthetical citation

`[@a; @b]`
Several citations

`[-@key]`
Suppress author name

`@key [p. 33]`
Author in text with locator

`[see @key, pp. 33-35]`
Prefix and locator

`bibliography: <f>.bib`
Set it in the metadata block

Put a final heading such as # References; the bibliography is appended at the end, or placed in a div with id refs.

FILTERS AND SLIDES

`pandoc --lua-filter=<f>.lua`
Run a Lua filter

`pandoc -F <program>`
Run a JSON filter program

`pandoc -t revealjs -s <f>.md -o <out>.html`
reveal.js slides

`pandoc -t beamer <f>.md -o <out>.pdf`
Beamer slides as PDF

`pandoc <f>.md -o <out>.pptx`
PowerPoint slides

`--slide-level=<n>`
Heading level that starts a slide

`---` **(horizontal rule)**
Start a new slide

`pandoc --sandbox ...`
Block file and network access

Does not protect against filters or PDF engines you run

`pandoc --version`
Show version and data directory

RECIPES

Markdown to a styled PDF

```
pandoc report.md -s --toc -N \
  --pdf-engine=xelatex \
  -V geometry:margin=2.5cm -V
mainfont="DejaVu Serif" \
  -o report.pdf
```

Paper with bibliography

```
pandoc paper.md -s -C \
  --bibliography=refs.bib --
csl=ieee.csl \
  -o paper.docx
```
```