Pandoc

https://pandoc.org/

You can use Pandoc to generate PDF's directly from other **document formats**, like Markdown, wikitext, Libre Office, slides, InDesign ICML files, or PDF.

Pandoc is described as an "universal document converter": it **converts documents** from one markup language into another.

Extensive documentation: Pandoc's Manual or `man pandoc`

Pandoc common arguments

-f or **--from** - option standing for "from", is followed by the input format

-t or **--to** - option standing for "to", is followed by the output format

-S or **--standalone** - option standing for "standalone", produces output with an appropriate header and footer

-s or **--stylesheet** - option to use a CSS stylesheet

-o or **--output** - option for file output

Pandoc stdin and stdout

By default, Pandoc can take **`stdin`** as an **input**.

Likewise, Pandoc also writes its **output** to **`stdout`** by default.

An example of using the **`stdin`**: if you use a pipeline to get your text from a pad, you can use `curl` to download this text and *pipe* it into Pandoc:

```
$ curl https://pad.xpub.nl/p/collaborations/export/txt | pandoc --from markdown --to html
```

This will output HTML in your terminal.

An example of using the **`stdout`**: if you want to turn this HTML page into a PDF, for example using Weasyprint, you can *pipe* the output of Pandoc into Weasyprint:

```
$ curl https://pad.xpub.nl/p/collaborations/export/txt | pandoc --from markdown --to html | weasyprint - pad.pdf
```

Note that Weasyprint (and many other programs) use a **dash** `-` as a special symbol to use `stdin` as input.

Changing the default template

```
$ pandoc --from markdown --to html --print-default-template=html5 > template.html
$ pandoc --from markdown --to html --template template.html input.md -o output.html
```

PDF

A range of **PDF engines** are supported at the moment, including `Paged.js`, `weasyprint` and `LaTeX`. You need to select the one of choice using the `--pdf-engine` option, and have the PDF engine installed on your computer.

You can follow this page for instructions: https://pandoc.org/MANUAL.html#creating-a-pdf

Examples

STRING to MARKDOWN

```
$ echo "Hello Pandoc from html to markdown" | pandoc -f html -t markdown
```

WIKI to HTML

- Save the content of a wiki page on to a plain-text file, example: `page.wiki`
- convert mediawiki to html:

```
$ pandoc page.wiki -f mediawiki -t html -o page.html
```

WIKI to HTML to PDF to BOOKLET PDF

MediaWiki provides a way to get the content of a page in wikitext or HTML:

- https://pzwiki.wdka.nl/mediadesign/Wordhole?action=raw (wikitext)
- https://pzwiki.wdka.nl/mediadesign/Wordhole?action=render (HTML)

You can use Weasyprint to generate a PDF from an URL!

```
$ weasyprint -s stylesheet.css https://pzwiki.wdka.nl/mediadesign/Wordhole?action=render wordhole.pdf
```

Another example:

```
$ curl https://pzwiki.wdka.nl/mediadesign/Voting_by_show_of_hands?action=raw > Voting_by_show_of_hands.mediawiki
```

```
$ pandoc --from mediawiki --to html Voting_by_show_of_hands.mediawiki --output Voting_by_show_of_hands.html
```

```
$ weasyprint Voting_by_show_of_hands.html Voting_by_show_of_hands.pdf
```

And to make a booklet PDF:

```
$ pdfbook2 --paper=a4paper --short-edge --no-crop Voting_by_show_of_hands.pdf
```

PAD to MARKDOWN to HTML to PDF to BOOKLET PDF

In one pipeline:

```
$ curl https://pad.xpub.nl/p/collaborations/export/txt | pandoc --from markdown --to html | weasyprint -s stylesheet.css - pad.pdf
```

or in multiple lines, and with in between moments of saving the content to files:

```
$ curl https://pad.xpub.nl/p/collaborations/export/txt > pad.md
```

```
$ pandoc --from markdown --to html pad.md --output pad.html
```

```
$ weasyprint -s stylesheet.css pad.html pad.pdf
```

And to make a booklet PDF:

```
$ pdfbook2 --paper=a4paper pad.pdf
$ pdfbook2 --paper=a4paper --short-edge --no-crop pad.pdf
```