

What is pandoc?

Pandoc is a "universal" document converter.

- .md → .docx or pdf
- .html → .docx
- .tex → .html
- .docx → .EPUB

Overview

1. Installation (briefly)
2. Basic conversion: .txt, .html, .pdf, .odt
3. How to use metadata and styling variables
4. Basic templating

How to install?

You can install for Windows, Mac, Linux by following the instructions here: **pandoc installation instructions:**
<https://pandoc.org/installing.html>

Using Pandoc: Open Terminal

Use **pandoc** by running commands in terminal / CLI (e.g., cmd, powershell, bash):

- Mac: Command(⌘) + Spacebar, type "Terminal"
- Windows: Searchbar, type "cmd"

Check Pandoc is installed

```
pandoc --version
```

Using Pandoc: Simple Conversion

In the terminal, enter the directory of the file you want to convert.

```
cd Desktop  
cd text/
```

Using Pandoc: Simple Conversion

```
pandoc sample.md -f markdown -t html -s -o sample.html
```

- `-f` (from / input) markdown `-t` (to / output) html
- Since pandoc can figure out the file type from the extension, we can just write:

```
pandoc sample.md -o sample.html
```

Format

You can see the different file formats that pandoc can work with here:

pandoc general options -f -t

Example: TXT to HTML

```
pandoc sample.txt -o sample.html
```

Using Pandoc: TXT to HTML Standalone

- By default, pandoc produces document fragments.
- To produce a standalone document (valid HTML), it applies a template.
- For HTML, it will add things like a header, metadata, basic styling, etc.

```
pandoc sample.txt -s -o sample.html
```

Concatenate files

- You can combine multiple files together into a single output.

```
pandoc intro.txt body.txt conclusion.txt -s -o sample.html
```

Exercise

1. Create two plaintext files, then combine them in a conversion to an .odt (or .docx) file.
2. Make some styling changes to the .odt / .docx file, e.g., bolding, italicizing, adding a heading, color some text.
3. Convert it to .html and open it in a browser.

Pandoc variables

- When using `-s`, pandoc takes the input file and applies a template to convert the file
- Variables influence this conversion.
- Metadata variables: `-M`
- Template (style) variables: `-v`

Metadata variables

Pandoc metadata variables

- Metadata variables set certain template but also affect metadata
- title, author, date, subtitle
- abstract, abstract-title
- keywords, subject, description, category

Converting to PDFS

- To convert to PDF, you'll need $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ installed:
- **MacTeX**
- **MiKTeX**

Metadata variables

```
pandoc syllabus.md -s -o syllabus.pdf
```

Let's add a title, an author, and a date (prefix each with `-M`):

```
pandoc syllabus.md -s -M title="Logic Syllabus" -M author="Lain Iwakura" -M  
date="6/23/2025" -o syllabus.pdf
```

Styling variables

Let's add some styling with (prefix each with `-v`):

```
pandoc syllabus.md -s -M title="Logic Syllabus" -M author="Lain Iwakura, PhD, MD" -M  
date="6/23/2025" -V geometry=margin=0.5in -o syllabus.pdf
```

Set the margin to 0.5in.

Styling variables

Color links and set them to magenta.

```
pandoc syllabus.md -s -M title="Logic Syllabus" -M author="Lain Iwakura, PhD, MD" -M  
date="6/23/2025" -V geometry=margin=0.5in -V colorlinks=true, -V linkcolor=magenta -V  
urlcolor=magenta -o syllabus.pdf
```

YAML

At this point, our commands are getting long. Let's use YAML

1. In the same directory, create a file `syllabus.yaml`
2. Add the data we used with `-M` and `-v` flags there.

YAML

```
pandoc syllabus.md -s -M title="Logic Syllabus" -M author="Lain Iwakura, PhD, MD" -M  
date="6/23/2025" -V geometry=margin=0.5in -V colorlinks=true, -V linkcolor=magenta -V  
urlcolor=magenta -o syllabus.pdf
```

```
title: "Logic Syllabus"  
author: "Lain Iwakura, PhD, MD"  
date: "6/23/2025"  
geometry: margin=1in  
colorlinks: true  
linkcolor: magenta  
urlcolor: magenta
```

PANDOC YAML

Refer to the YAML file by adding `--metadata-file=syllabus.yaml` in our command:

```
pandoc syllabus.md --metadata-file=syllabus.yaml -s -o syllabus.pdf
```

YAML TOC

Let's add a table of contents, specify the depth of the TOC, number the sections of the syllabus, and specify the language:

```
# TOC
toc: true
toc-depth: 2
numbersections: true
lang: en-US
```

```
pandoc syllabus.md --metadata-file=syllabus.yaml -s -o syllabus.pdf
```

YAML TOC: Unexpected Behavior

NOTE: If you are converting to HTML, your TOC won't appear!

- For HTML conversion, adding `toc: true` to the YAML tells pandoc to add the html nav but not to construct the toc
- In contrast, `--toc` tells pandoc to construct the table.

For more, see here:

1. **Github issue**
2. **Pandoc Writer Options**

YAML TOC: Unexpected Behavior

Similarly, `toc-depth` is not set. So, we need to declare that as a command as well:

```
pandoc syllabus.md --metadata-file=syllabus.yaml --toc --toc-depth=1 -s -o syllabus.pdf
```

YAML Header and page numbers

Let's also add a header and page numbers

```
# Header / footer
header-includes: |
  \usepackage{fancyhdr}
  \pagestyle{fancy}
  \fancyhf{}
  \fancyhead[L]{Logic Syllabus}
  \fancyhead[R]{\thepage}
  \renewcommand{\headrulewidth}{0.4pt}
```

```
pandoc syllabus.md --metadata-file=syllabus.yaml -s -o syllabus.pdf
```

CSS flag

Some options only impact certain writers. If we were to convert to HTML, we could use a stylesheet by using the `--css=style.css` flag.

```
a {  
  color: green;  
}
```

```
pandoc syllabus.md --metadata-file=syllabus.yaml --css=style.css -s -o syllabus.html
```

Templating

- When `pandoc` converts documents with the `standalone` option, it applies a template.
- The template can be reviewed:

```
pandoc -D *FORMAT*
```

So, for example, we could examine the HTML template here:

```
pandoc -D html
```

Adding Text to the Bottom

Let's add some text to the end of every document:

1. We'll add `<p>© 2025 Serial Experiments</p>`
2. Create a new `template.FORMAT` file (ours is `html_template.html`)
3. Paste the template
4. Add `<p>© 2025 Serial Experiments</p>` after `$body$`

Adding Text to the Bottom

```
1 $body$  
2 <p>© 2025 Serial Experiments</p>  
3 $for(include-after)$  
4 $include-after$  
5 $endfor$  
6 </body>
```

```
pandoc syllabus.md --metadata-file=syllabus.yaml --template=html_template.html -s -o  
syllabus.html
```

Adding More Content

- Lets add the date and author to each .html file we create.
- Use conditionals along with the `date` and `author` variables.
- Place them in dollar signs: (`$...$`)

```
$if(author)$  
$if(date)$  
<p>This file was created by $author$ on $date$.</p>  
$else$  
<p>This file was created by $author$.</p>  
$endif$  
$elseif(date)$  
<p>This file was created on $date$.</p>  
$endif$
```

```
pandoc syllabus.md --metadata-file=syllabus.yaml --template=html_template.html -s -o  
syllabus.html
```

Exercise

1. Create a .txt file with some text.
2. Create a YAML file with title, subtitle, and keywords parameters
3. Create a `html` template file that writes each of these parameters on their own lines (if they exist).

Resources

1. **Pandoc installation**
2. **Pandoc try**
3. **Pandoc manual**
4. **Pandoc demos**

Speaker notes