

Convert Markdown to Word with Pandoc

Let's look at how to convert Markdown (.md) files to Word (.docx) using Pandoc.

Pandoc is a command-line tool and does not have a graphical interface.

At first check if Pandoc is installed on your computer:

- Open a **terminal** and run the command: `brew info pandoc`

If the response says `not installed`, then Pandoc is not installed.

Install Pandoc

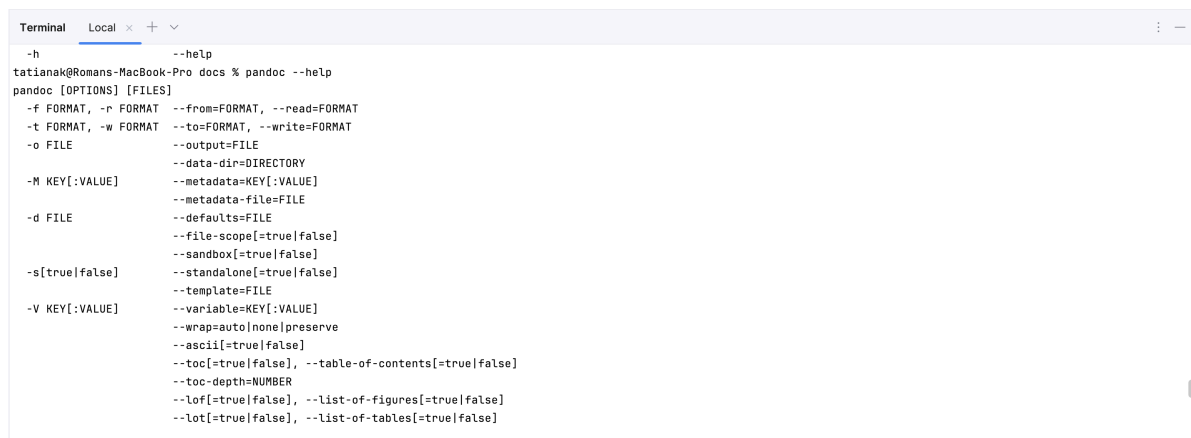
To install Pandoc, run the following command in the **terminal**: `brew install pandoc`

Pandoc is written in **Haskell** and is installed via **brew**, not **pip**.

There is a **Python** wrapper called **pypandoc**, but it wraps **Pandoc**.

Basics of working with Pandoc

After installation, check Pandoc help to get acquainted with its capabilities: `pandoc --help`



```
Terminal Local x + v
--help
tatianak@Romans-MacBook-Pro docs % pandoc --help
pandoc [OPTIONS] [FILES]
-f FORMAT, -r FORMAT --from=FORMAT, --read=FORMAT
-t FORMAT, -w FORMAT --to=FORMAT, --write=FORMAT
-o FILE --output=FILE
--data-dir=DIRECTORY
-M KEY[:VALUE] --metadata=KEY[:VALUE]
--metadata-file=FILE
-d FILE --defaults=FILE
--file-scope[=true|false]
--sandbox[=true|false]
-s[true|false] --standalone[=true|false]
--template=FILE
-V KEY[:VALUE] --variable=KEY[:VALUE]
--wrap=auto|none|preserve
--ascii[=true|false]
--toc[=true|false], --table-of-contents[=true|false]
--toc-depth=NUMBER
--lof[=true|false], --list-of-figures[=true|false]
--lot[=true|false], --list-of-tables[=true|false]
```

Command structure:

- Enter the command `pandoc`.
- Specify the required options (the list can be found in the help).

- Upload files for conversion.

Convert Markdown files to Word

For example, we will convert Markdown files to Word.

To perform the conversion, use the following command:

```
pandoc -f markdown -t docx -o <path_to_result.docx> <path_to_source.md>
```

Explanation of command parameters:

`-f markdown` – source file format (Markdown). `-t docx` – result format (Word). `-o` – path and name of the resulting file.

Example command: `pandoc -f markdown -t docx -o 'docs/Project Setup.docx' docs/Project\ Setup.md`

Note that in the shell we need to specify a backslash for the md file name so that the terminal will read the command as one argument. You can also wrap the entire document in quotes.

If the file name contains spaces, note:

- Use a backslash \ to indicate a space (e.g. `Project\ Setup.md`).
- Or wrap the file name in quotes (e.g. `Project Setup.md`).

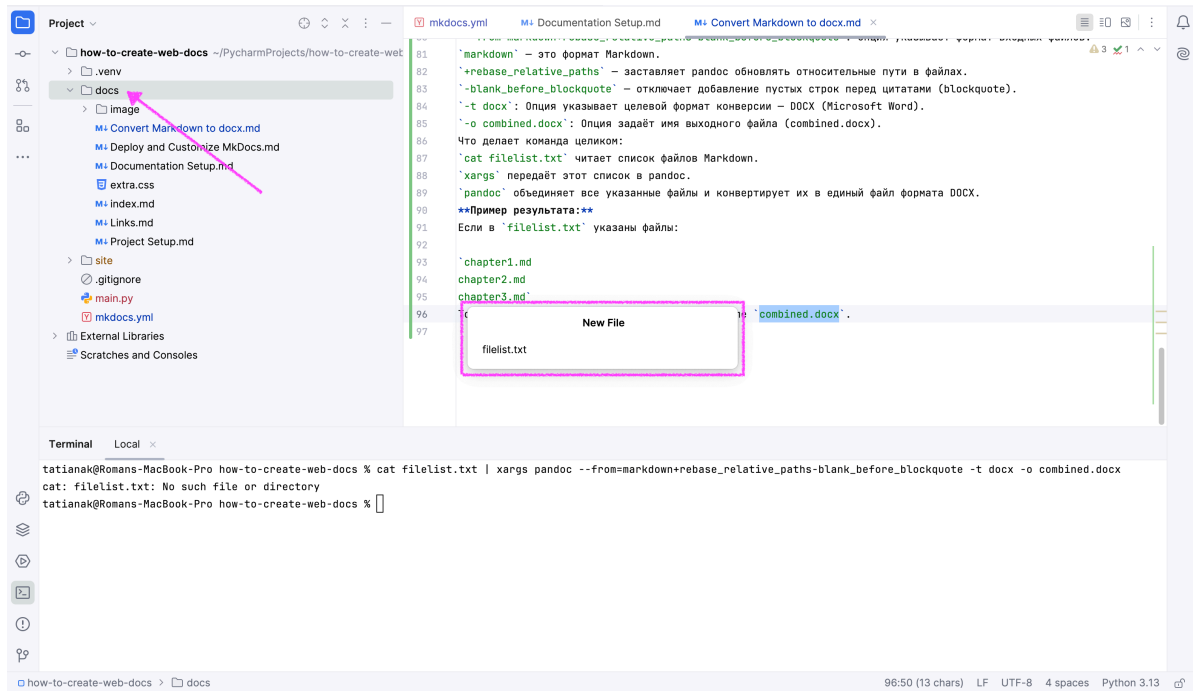
Use Tab to autocomplete file names in the terminal - this will also automatically add the necessary slashes.

Converting multiple Markdown-files into a single DOCX file

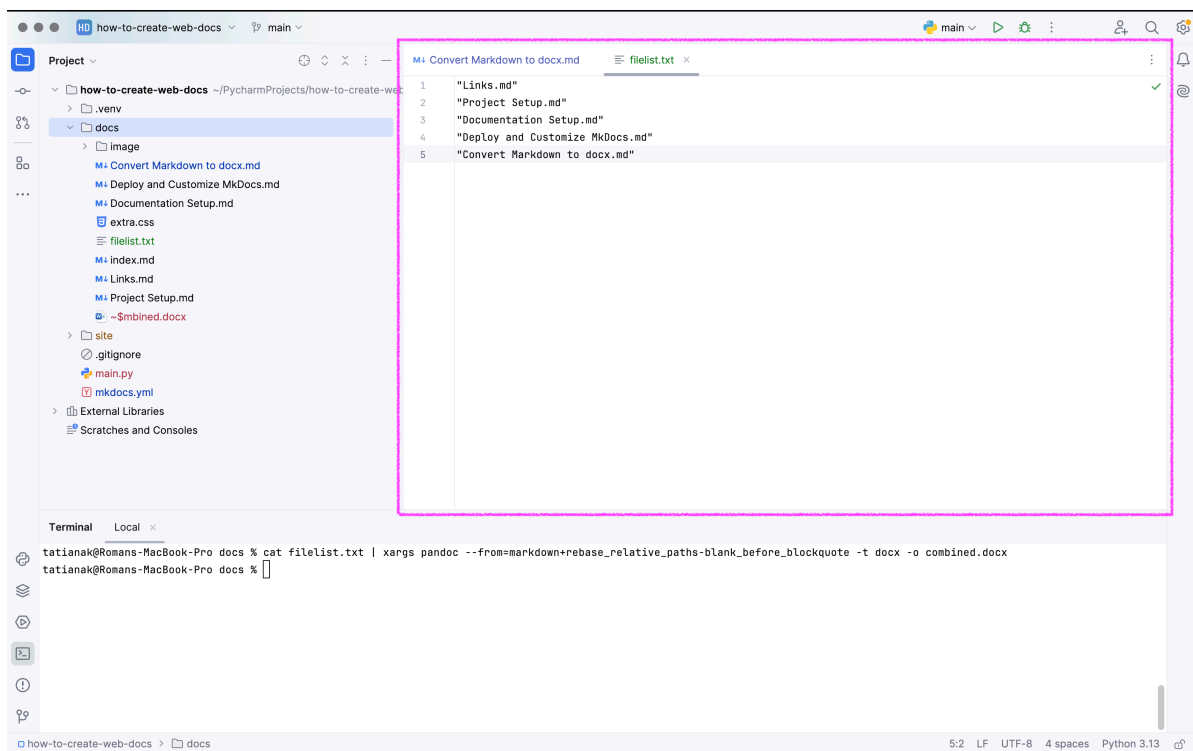
In the previous section, we learned how to convert Markdown files to docx, but we encountered the fact that each file needs to be converted separately. This is inconvenient, since you will have to manually assemble the document later. In order to convert multiple Markdown files to a DOCX file in the specified order, you need to create an additional file and run just one command.

CREATING A FILELIST.TXT FILE

Setting the order of Markdown documents is necessary, since **PyCharm** automatically sorts documents alphabetically. To do this, I create a document `filelist.txt`.



Then I fill the document with paths in the order in which they should be written in DOCX.



I use **relative paths** and escape the paths in quotes.

RUNNING THE COMMAND

Once the `filelist.txt` document is created, I use the following command: `cat filelist.txt | xargs pandoc --from=markdown+rebase_relative_paths-blank_before_blockquote -t docx -o combined.docx`

This will generate the `combined.docx` file.

If you get `No such file or directory` as a result of running the command, check which directory you are running the command from. Run `'pwd'` and make sure you are running the document conversion from the same directory where the Markdown files and images are located. You are probably running the convert command from the top directory, so change to the correct directory (use `cd` command)

My **project structure** is shown below:

```
how-to-create-web-docs/
├── .env/ # Directory for storing the Python virtual environment
├── docs/ # Folder with the main project documentation
│   ├── image/ # Document images
│   ├── Convert Markdown to docx.md # Markdown file
│   ├── Deploy and Customize MkDocs.md # Markdown file
│   ├── Documentation Setup.md # Markdown file
│   ├── Links.md # Markdown file
│   ├── Project Setup.md # Markdown file
│   ├── index.md # Main documentation page
│   ├── extra.css # Additional styles for formatting the documentation
│   └── filelist.txt # File containing a list of Markdown documents
├── site/ # Directory where MkDocs generates a static site
├── .gitignore # Git configuration file that specifies which files or folders to
exclude from the repository
└── mkdocs.yml # Main MkDocs configuration file
```

COMMAND EXPLANATION

This command converts multiple Markdown files specified in `filelist.txt` into a single DOCX document using the `pandoc` utility. Here's a step-by-step explanation:

1. `cat filelist.txt` The `cat` command outputs the contents of `filelist.txt`. This file is assumed to list the paths to the Markdown files to be processed. Each path is on a new line.

Example contents of `filelist.txt`:

```
chapter1.md
chapter2.md
```

chapter3.md

2. | (pipe) The | character routes the output of the cat command as input to the command xargs.
3. xargs The xargs command takes the strings passed through the pipe and substitutes them as arguments to the pandoc command. This allows each of the specified files to be processed.
4. pandoc --from=markdown+rebase_relative_paths-blank_before_blockquote -t docx -o combined.docx This command uses pandoc to convert Markdown files into a single DOCX document. Let's look at pandoc's arguments:
 - --from=markdown+rebase_relative_paths-blank_before_blockquote : The option specifies the format of the input files.
 - markdown is the Markdown format.
 - +rebase_relative_paths makes pandoc update relative paths in files.
 - -blank_before_blockquote disables adding blank lines before blockquotes.
 - -t docx : The option specifies the target conversion format - DOCX (Microsoft Word).
 - -o combined.docx : The option specifies the name of the output file (combined.docx).

What the whole command does:

cat filelist.txt reads a list of Markdown files.

xargs passes this list to pandoc.

pandoc combines all the specified files and converts them into a single DOCX file.

Example output:

If filelist.txt specifies the files:

chapter1.md

chapter2.md

chapter3.md Then pandoc will combine their contents into a single file combined.docx .