

Pandoc document converter

Introduction

[Pandoc](#) converts files from one markup format into another. In the pandoc web page it is defined as the swiss-army knife of document converting.

What Pandoc is capable of?

Pandoc understands a lot of markdown syntax extensions, including:

- document metadata (title, author, date)
- footnotes
- tables
- definition lists
- superscript and subscript
- strikeouts
- enhanced ordered lists
- running example lists
- delimited code blocks with syntax highlighting
- smart quotes, dashes, and ellipses
- markdown inside HTML blocks; and inline LaTeX.

[LaTeX](#) math (and even macros) can be used in markdown documents. Several different methods of rendering math in HTML are provided, including [MathJax](#) and translation to [MathML](#). LaTeX math is converted (as needed by the output format) to [unicode](#), native Word equation objects, MathML, or roff eqn.

Pandoc includes a powerful system for automatic citations and bibliographies. This means that you can write a citation like [see @doe99, pp. 33-35; also @smith04, ch. 1] and pandoc will convert it into a properly formatted citation using any of hundreds of CSL styles (including footnote styles, numerical styles, and author-date styles), and add a properly formatted bibliography at the end of the document. The bibliographic data may be in [BibTeX](#), [BibLaTeX](#), [CSL JSON](#), or [CSL YAML](#) format. Citations work in every output format.

There are many ways to customize pandoc to fit your needs, including a template system and a powerful system for writing filters.

Pandoc includes a [Haskell](#) library and a standalone command-line program. The library includes separate modules for each input and output format, so adding a new input or output format just requires adding a new module.

Pandoc is free software, released under the GPL. Copyright 2006–2020 [John MacFarlane](#)

Installing pandoc

The simplest way to get the latest pandoc release is to use the installer. For alternative ways to install pandoc, see below under the heading for your operating system.

Installing on Windows

There is a package installer at pandoc's download page. This will install pandoc, replacing older versions, and update your path to include the directory where pandoc's binaries are installed. It is also obtain a zip file that contains pandoc's binaries and documentation. Simply unzip this file and move the binaries to a directory of your choice.

Alternatively, you can install pandoc using Chocolatey:

```
choco install pandoc
```

[Chocolatey](#) can also install other software that integrates with Pandoc, e.g. rsvg-convert (from librsvg, covering formats without SVG support), Python (to use Pandoc filters), and MiKTeX (to typeset PDFs with LaTeX):

Installing on Mac OSX

There is a package installer at pandoc's download page. If you later want to uninstall the package, you can do so by downloading this script and running it with `perl uninstall-pandoc.pl`.

Alternatively, you can install pandoc using [Homebrew](#):

```
brew install pandoc
```

Homebrew, as in Chocolatey can also install other software that integrates with Pandoc ¹.

It also exists a zip file containing the binaries and man pages, for those who prefer not to use the installer. Simply unzip the file and move the binaries and man pages to whatever directory you like.

By default, Pandoc creates PDFs using LaTeX which means that should be installed on your system. If you are a Latex user (and probably have installed [MacTeX](#) it works immediately. But that Mactex distribution uses four gigabytes of disk space, so if you need Latex only for conversion purposes, [BasicTeX](#) or [TinyTeX](#) are good choices

Installing on Linux distributions

Check whether the pandoc version in your package manager is not outdated. Pandoc is in the [Debian](#), [Ubuntu](#), [Slackware](#), [Arch](#), [Fedora](#), [NiXOS](#), [openSUSE](#), [Gentoo](#) and [Void](#) repositories.

To get the latest release, there is a binary package for amd64 architecture on the download page.

The executable is statically linked and has no dynamic dependencies or dependencies on external data files ².

Both a tarball and a deb installer are provided.

To install the deb:

```
sudo dpkg -i $DEB
```

where \$DEB is the path to the downloaded deb. This will install the pandoc executable and man page.

To install in a RPM-based distro, you may be able to install the deb from our download page using [alien](#).

On any distro, you may install from the tarball into *DEST* (say, */usr/local* or *HOME/.local*) by doing

```
tar xvzf $TGZ --strip-components 1 -C $DEST
```

where \$TGZ is the path to the downloaded zipped tarball.

For PDF output, you'll need LaTeX. We recommend installing TeX Live via your package manager. (On Debian/Ubuntu, apt-get install texlive.)

Installing on Chrome OS

On [Chrome OS](#), pandoc can be installed using the chromebrew package manager with the command:

```
crew install pandoc
```

This will automatically build and configure pandoc for the specific device you are using.

Installing on BSD

Pandoc is in the [NetBSD](#) and [FreeBSD](#) ports repositories.

Docker

The official Docker images for pandoc can be found at <https://github.com/pandoc/dockerfiles> and at dockerhub.

The pandoc/core image contains pandoc.

The pandoc/latex image also contains the minimal LaTeX installation needed to produce PDFs using pandoc.

To run pandoc using Docker, converting README.md to README.pdf:

```
docker run --rm --volume "`pwd`::/data" --user `id -u`:`id -g` pandoc/latex README.md -o README.pdf
```

GitHub Actions

Pandoc can be run through GitHub Actions. For some examples, see <https://github.com/pandoc/pandoc-action-example>.

Compiling from source

If for some reason a binary package is not available for your platform, or if you want to hack on pandoc or use a non-released version, you can install from source.

You can also install from source, using the instructions to Compiling from source³.

Using Pandoc

Once installed following the instructions for your platform you can use it in a terminal or command window depending on your OS.

Open a terminal

Pandoc is a command-line tool. There is no graphic user interface. So, to use it, you'll need to open a terminal window:

- * On OS X, the Terminal application can be found in /Applications/Utilities. Open a Finder window and go to Applications, then Utilities. Then double click on Terminal. (Or, click the spotlight icon in the upper right hand corner of your screen and type Terminal - you should see Terminal under Applications.)
- * On Windows, you can use either the classic command prompt or the more modern PowerShell terminal. If you use Windows in desktop mode, run the cmd or powershell command from the Start menu. If you use the Windows 8 start screen instead, simply type cmd or powershell, and then run either the "Command Prompt" or "Windows Powershell" application. If you are using cmd, type chcp 65001 before using pandoc, to set the encoding to UTF-8.
- * On Linux, there are many possible configurations, depending on what desktop environment you're using:
 - * In Unity, use the search function on the Dash, and search for Terminal. Or, use the keyboard shortcut Ctrl-Alt-T.
 - In Gnome, go to Applications, then Accessories, and select Terminal, or use Ctrl-Alt-T.
 - In XFCE, go to Applications, then System, then Terminal, or use Super-T.
 - In KDE, go to KMenu, then System, then Terminal Program (Konsole).

You should now see a rectangle with a "prompt" (possibly just a symbol like %, but probably including more information, such as your username and directory), and a blinking cursor.

Let's verify that pandoc is installed. Type a command, e.g.:

```
pandoc --version
```

and hit enter. You should see a message telling you which version of pandoc is installed, and giving you some additional information.

Changing directories

```
pwd
```

on Linux or OSX, or

```
echo %cd%
```

on Windows, and hit enter.

Your terminal should print your current working directory. (Guess what `pwd` stands for?) This should be your home directory.

Let's navigate now to our Documents directory: type

```
cd Documents
```

and hit enter. Now type

```
pwd
```

(or `echo %cd%` on Windows) again. You should be in the Documents subdirectory of your home directory. To go back to your home directory, you could type

```
cd ..
```

The `..` means "one level up."

Go back to your Documents directory if you're not there already. Let's try creating a subdirectory called `pandoc-test`:

```
mkdir pandoc-test
```

Now change to the `pandoc-test` directory:

```
cd pandoc-test
```

If the prompt doesn't tell you what directory you're in, you can confirm that you're there by doing

```
pwd
```

(or `echo %cd%`) again.

OK, that's all you need to know for now about using the terminal. But here's a secret that will save you a lot of typing. You can always type the up-arrow key to go back through your history of commands. So if you want to use a command you typed earlier, you don't need to type it again: just use up-arrow until it comes up. Try this. (You can use down-arrow as well, to go the other direction.) Once you have the command, you can also use the left and right arrows and the backspace/delete key to edit it.

Most terminals also support tab completion of directories and filenames. To try this, let's first go back up to our Documents directory:

```
cd ..
```

Now, type

```
cd pandoc-
```

and hit the tab key instead of enter. Your terminal should fill in the rest (`test`), and then you can hit enter.

To review:

`pwd` (or `echo %cd%` on Windows) to see what the current working directory is.

`cd foo` to change to the `foo` subdirectory of your working directory.

`cd ..` to move up to the parent of the working directory.

`mkdir foo` to create a subdirectory called `foo` in the working directory.

up-arrow to go back through your command history.
tab to complete directories and file names.

Using pandoc as a filter

Type

```
pandoc
```

and hit enter. You should see the cursor just sitting there, waiting for you to type something. Type this:

```
Hello *pandoc*!  
  
- one  
- two
```

When you're finished (the cursor should be at the beginning of the line), type Ctrl-D on OS X or Linux, or Ctrl-Z followed by Enter on Windows. You should now see your text converted to HTML!

Hello *pandoc*!

- one
- two

What just happened? When pandoc is invoked without specifying any input files, it operates as a “filter,” taking input from the terminal and sending its output back to the terminal. You can use this feature to play around with pandoc.

By default, input is interpreted as pandoc markdown, and output is HTML. But we can change that. Let's try converting from HTML to markdown:

```
pandoc -f html -t markdown
```

Now type:

```
<p>Hello <em>pandoc</em>!</p>
```

and hit Ctrl-D (or Ctrl-Z followed by Enter on Windows). You should see:

Hello *pandoc*!

Now try converting something from markdown to LaTeX. What command do you think you should use?

Text editor basics

You'll probably want to use pandoc to convert a file, not to read text from the terminal. That's easy, but first we need to create a text file in our pandoc-test subdirectory.

Important: To create a text file, you'll need to use a text editor, not a word processor like Microsoft Word. On Windows, you can use Notepad (in Accessories). On OS X, you can use TextEdit (in Applications). On Linux, different platforms come with different text editors: Gnome has GEdit, and KDE has Kate.

Start up your text editor. Type the following:

title: Test

Test!

This is a test of *pandoc*.

- list one
- list two

Now save your file as `test1.md` in the directory `Documents/pandoc-test`. If you use plain text a lot, you'll want a better editor than Notepad or TextEdit. You might want to look at Sublime Text or (if you're willing to put in some time learning an unfamiliar interface) Vim or Emacs.

Converting a file

Go back to your terminal. We should still be in the `Documents/pandoc-test` directory. Verify that with `pwd`.

Now type

```
ls
```

or

```
dir
```

if you're on Windows). This will list the files in the current directory. You should see the file you created, `test1.md`.

To convert it to HTML, use this command:

```
pandoc test1.md -f markdown -t html -s -o test1.html
```

The filename `test1.md` tells pandoc which file to convert. The `-s` option says to create a “standalone” file, with a header and footer, not just a fragment. And the `-o test1.html` says to put the output in the file `test1.html`. Note that we could have omitted `-f markdown` and `-t html`, since the default is to convert from markdown to HTML.

Check that the file was created by typing `ls` again. You should see `test1.html`. Now open this in a browser. On OS X, you can type

```
open test1.html
```

On Windows, type

```
.\test1.html
```

You should see a browser window with your document.

To create a LaTeX document, you just need to change the command slightly:

```
pandoc test1.md -f markdown -t latex -s -o test1.tex
```

Try opening test1.tex in your text editor.

Pandoc can often figure out the input and output formats from the filename extensions. So, you could have just used:

```
pandoc test1.md -s -o test1.tex
```

Pandoc knows you're trying to create a LaTeX document, because of the .tex extension.

Now try creating a Word document (with extension docx).

If you want to create a PDF, you'll need to have LaTeX installed. (See MacTeX on OS X, MiKTeX on Windows, or install the texlive package on Linux.) Then do:

```
pandoc test1.md -s -o test1.pdf
```

Command-line options

You now know the basics. Pandoc has a lot of options. At this point you can start to learn more about them by reading the User's Guide.

Here's an example. The `--mathml` option causes pandoc to convert TeX math into MathML. Type

```
pandoc --mathml
```

then enter this text, followed by Ctrl-D (Ctrl-Z followed by Enter on Windows):

```
 $x = y^2$ 
```

Now try the same thing without `--mathml`. See the difference in output?

If you forget an option, or forget which formats are supported, you can always do

```
pandoc --help
```

to get a list of all the supported options.

On OS X or Linux systems, you can also do

```
man pandoc
```

to get the pandoc manual page. All of this information is also in the User's Guide.

If you get stuck, you can always ask questions on the pandoc-discuss mailing list. But be sure to check the FAQs first, and search through the mailing list to see if your question has been answered before.

1. On unsupported versions of macOS (more than three releases old), Homebrew installs from source, which takes additional time and disk space for the ghc compiler and dependent Haskell libraries. [↩](#)

2. Because of the static linking, the pandoc binary from this package cannot use lua filters that require external lua modules written in C. [↩](#)

3. Most distros have the Haskell platform in their package repositories. For example, on Debian/Ubuntu, you can install it with `apt-get install haskell-platform`. [↩](#)