Pandoc a universal document converter

Getting started with pandoc

This document is for people who are unfamiliar with command line tools. Command-line experts can go straight to the User's Guide or the pandoc man page.

Step 1: Install pandoc

First, install pandoc, following the instructions for your platform.

Step 2: 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

```
pandoc --version
```

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

Step 3: Changing directories

First, let's see where we are. Type

```
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

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.

Step 4: 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!

```
<p>Hello <em>pandoc</em>!</p>
<ul>
<li>one</li>
<li>two</li>
</ul>
```

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?

Step 5: 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`.

Note: If you use plain text a lot, you'll want a better editor than `Notepad` or `TextEdit`. You might want to look at Visual Studio Code or Sublime Text or (if you're willing to put in some time learning an unfamiliar interface) Vim or Emacs.

Step 6: 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, but it doesn't hurt to include them.

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
```

Step 7: 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 discussion forum. But be sure to check the FAQs first, and search through the forum to see if your question has been answered before.