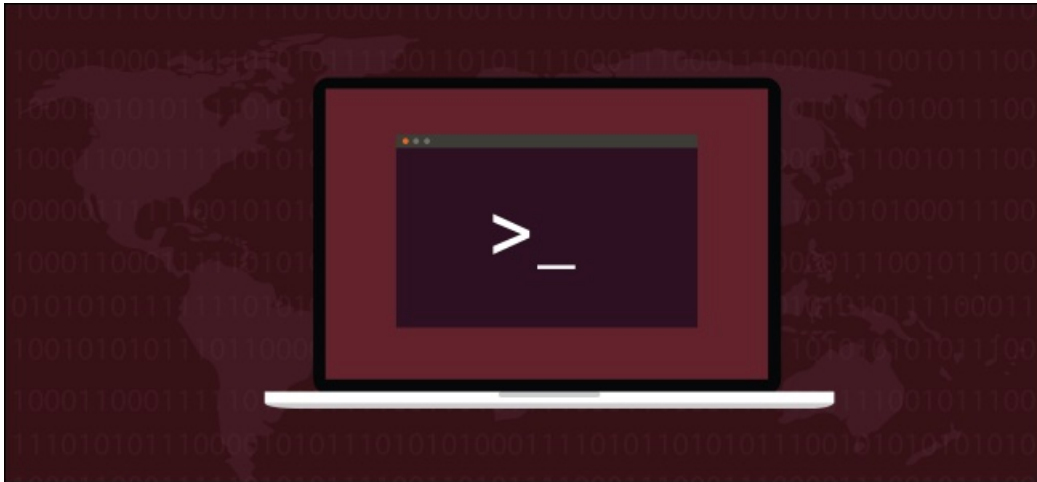


How to Use pandoc to Convert Files on the Linux Command Line



DAVE MCKAY [@thegurkha](#)

JUNE 30, 2020, 8:00AM EDT



Fatmawati Achmad Zaenuri/Shutterstock

You can use `pandoc` on Linux to convert between more than 40 file formats. You can also use it to create a simple docs-as-code system by writing in Markdown, storing in `git`, and publishing in any of its supported formats.

Document Conversion and Docs-as-Code

If you have a document in any of `pandoc`'s [many supported file formats](#), converting it to any of the others is a cinch. That's a handy tool to have!

But the real power of `pandoc` becomes apparent when you use it as the basis of a simple docs-as-code system. The premise of docs-as-code is to adopt some of the techniques and principles of software development and apply them to writing documentation, especially for software development projects. You can apply it to the development of any kind of documentation, though.

Software developers use their favorite editor or [integrated development environment](#) (IDE) to write their programs. The code they type is saved in text files. These contain the [source code](#) for the program.

They use a [version control system](#), or VCS ([Git](#) is the most popular), to capture changes to the source code as it's developed and enhanced. This means the programmer has a complete history of all versions of the source code files. He or she can quickly access any previous version of a file. `Git` stores files in a repository. There's a local repository on each developer's computer and a central, shared, remote repository that's often cloud-hosted.

When they're ready to produce a working version of the program, they use a [compiler](#) to read the source code and generate a binary executable.

By writing your documents in a lightweight, text-based markup language, you can use a VCS

to version control your writing. When you're ready to distribute or publish a document, you can use pandoc to generate as many different versions of your documentation as you need, including web-based ([HTML](#)), word-processed or typeset ([LibreOffice](#), [Microsoft Word](#), [TeX](#)), portable document format ([PDF](#)), e-book ([ePub](#)), and so on.

You can do all of this from one set of version-controlled, lightweight text files.

Installing pandoc

To install pandoc on Ubuntu, use this command:

```
sudo apt-get install pandoc
```

```
dave@ubuntu20-04:~/pd$ sudo apt-get install pandoc
```

On Fedora, the command you need is the following:

```
sudo dnf install pandoc
```

```
[dave@fedora32 ~]$ sudo dnf install pandoc
```

On Manjaro, you need to type:

```
sudo pacman -Syu pandoc
```

```
[dave@manjaro20-0-1 ~]$ sudo pacman -Syu pandoc
```

You can check which version you have installed by using the `--version` option:

```
pandoc --version
```

```
dave@ubuntu20-04:~/pd$ pandoc --version
pandoc 2.5
Compiled with pandoc-types 1.17.5.4, texmath 0.11.2.2, skylighting 0.7
.7
Default user data directory: /home/dave/.pandoc
Copyright (C) 2006-2018 John MacFarlane
Web: http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.
dave@ubuntu20-04:~/pd$
```

Using pandoc Without Files

If you use pandoc without any command-line options, it also accepts typed input. You just press Ctrl+D to indicate you've finished typing. pandoc expects you to type in Markdown format, and it generates HTML output.

Let's look at an example:

```
pandoc
```

```
dave@ubuntu20-04:~/pd$ pandoc
```

We've typed a few lines of Markdown and are about to hit Ctrl+D.

```
dave@ubuntu20-04:~/pd$ pandoc
# A Level One Heading
## A Level Two Heading
### A Level Three Heading

## Bold and Underline
This is bold and this is italic

### A Level Three Heading
[Link to How to Geek](https://www.howtogeek.com/)

### Include an image
![The Geek](HTG.png)
```

As soon as we do, pandoc generates the equivalent HTML output.

```
## Bold and Underline
This is bold and this is italic

### A Level Three Heading
[Link to How to Geek](https://www.howtogeek.com/)

### Include an image
![The Geek](HTG.png)
<h1 id="a-level-one-heading">A Level One Heading</h1>
<h2 id="a-level-two-heading">A Level Two Heading</h2>
<h3 id="a-level-three-heading">A Level Three Heading</h3>
<h2 id="bold-and-underline">Bold and Underline</h2>
<p>This is bold and this is italic</p>
<h3 id="a-level-three-heading-1">A Level Three Heading</h3>
<p><a href="https://www.howtogeek.com/">Link to How to Geek</a></p>
<h3 id="include-an-image">Include an image</h3>
<figure>
<figcaption>The Geek</figcaption>
</figure>
dave@ubuntu20-04:~/pd$
```

To do anything useful with pandoc, though, we really need to use files.

Markdown Basics

[Markdown](#) is a lightweight markup language, and special meaning is given to certain characters. You can use a plain text editor to create a Markdown file.

Markdown can be read easily, as there are no visually cumbersome tags to distract from the text. Formatting in Markdown documents resembles the formatting it represents. Below are some of the basics:

- To emphasize text with *italics*, wrap it in asterisks. `*This will be emphasized*`
- To **bold** text, use two asterisks. `**This will be in bold**`
- Headings are represented by the number sign/hash mark (#). Text is separated from the hash by a space. Use one hash for a top-level heading, two for a second-level, and so on.
- To create a bulleted list, start each line of the list with an asterisk and insert a space before the text.
- To create a numbered list, start each line with a digit followed by a period, and then insert a space before the text.
- To create a hyperlink, enclose the name of the site in square brackets ([]), and the URL in parentheses (()) like so: [Link to How to Geek](https://www.howtogeek.com/).
- To insert an image, type an exclamation point immediately before brackets (! []). Type any alternative text for the image in the brackets. Then, enclose the path to the image in parentheses (()). Here's an example: ![The Geek](HTG.png).

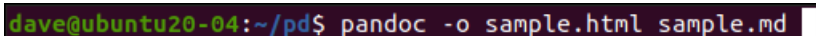
We'll cover more examples of all of these in the next section.

RELATED: [What Is Markdown and How Do You Use It?](#)

Converting Files

File conversions are straightforward. `pandoc` can usually work out which file formats you're working with from their filenames. Here, we're going to generate an HTML file from a Markdown file. The `-o` (output) option tells `pandoc` the name of the file we wish to create:

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



```
dave@ubuntu20-04:~/pd$ pandoc -o sample.html sample.md
```

Our sample Markdown file, `sample.md`, contains the short section of Markdown shown in the image below.

```

1 # A Level One Heading
2 A line of normal text.
3
4 ## A Level Two Heading
5 * This will become
6 * a *bulleted*
7 * list
8
9 ### A Level Three Heading
10 1. This will become
11 2. a **numbered**
12 3. list
13
14 #### A level Four Heading
15 [How to Geek](https://www.howtogeek.com/)
16
17 ![The Geek](HTG.png)

```

A file called sample.html is created. When we double-click the file, our default browser will open it.

A Level One Heading

A line of normal text.

A Level Two Heading

- This will become
- a *bulleted*
- list

A Level Three Heading

1. This will become
2. a **numbered**
3. list

A level Four Heading

[How to Geek](https://www.howtogeek.com/)



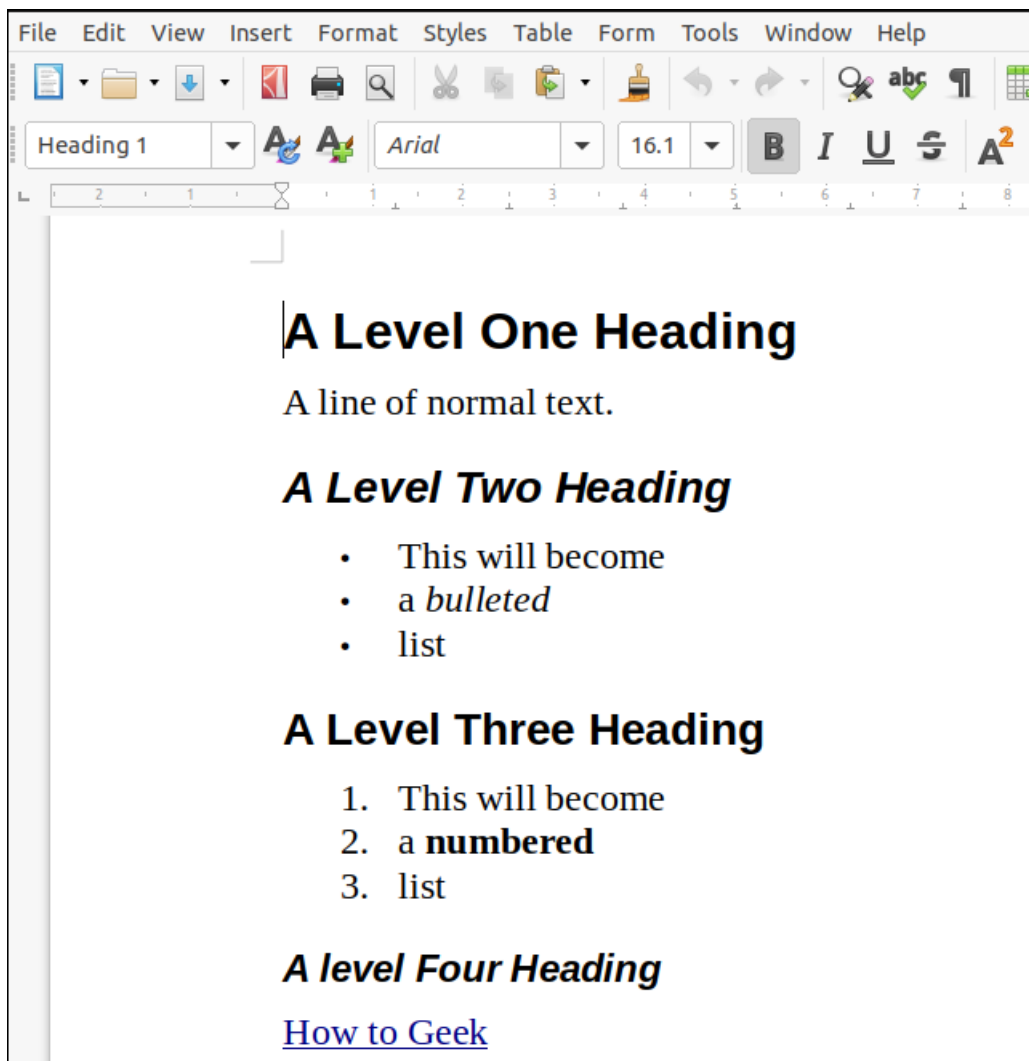
The Geek

Now, let's generate an [Open Document Format](#) text document we can open in [LibreOffice Writer](#):

```
pandoc -o sample.odt sample.md
```

```
dave@ubuntu20-04:~/pd$ pandoc -o sample.odt sample.md
```

The ODT file has the same content as the HTML file.



A neat touch is the alternative text for the image is also used to automatically generate a caption for the figure.



Specifying File Formats

The `-f` (from) and `-t` (to) options are used to tell pandoc which file formats you want to convert from and to. This can be useful if you're working with a file format that shares a file extension with other related formats. For example, [TeX](#), and [LaTeX](#) both use the ".tex" extension.

We're also using the `-s` (standalone) option so pandoc will generate all the LaTeX preamble required for a document to be a complete, self-contained, and well-formed LaTeX document.

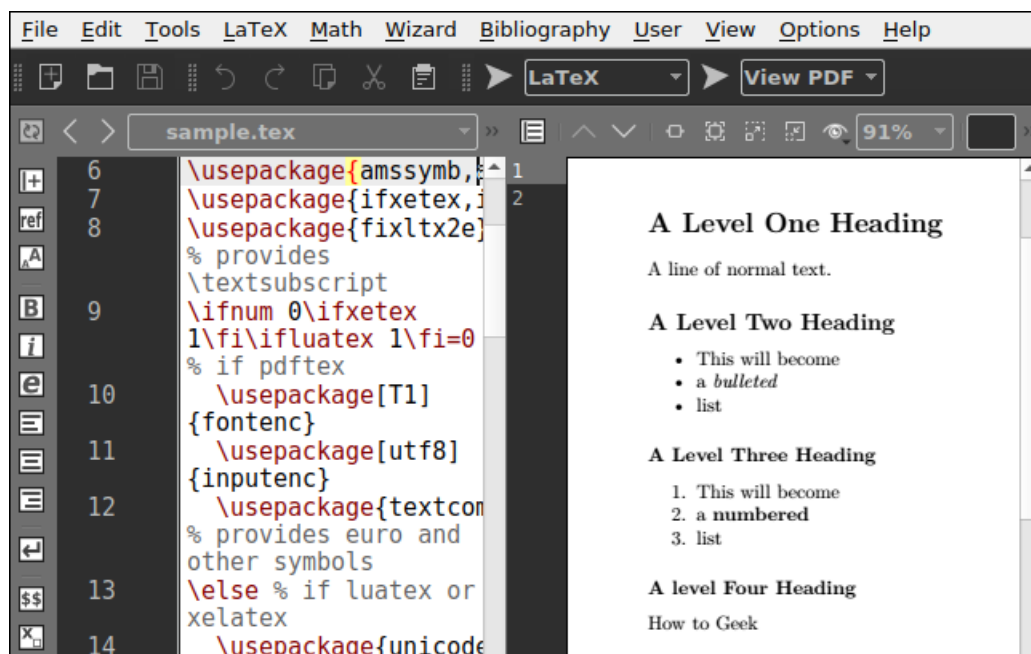
Without the `-s` (standalone) option, the output would still be well-formed LaTeX that could be slotted into another LaTeX document, it wouldn't parse properly as a standalone LaTeX document.

We type the following:

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

```
dave@ubuntu20-04:~/pd$ pandoc -f markdown -t latex -s -o sample.tex sample.md
```

If you open the “sample.tex” file in a text editor, you'll see the generated LaTeX. If you have a LaTeX editor, you can open the TEX file to see a preview of how the LaTeX typesetting commands are interpreted. Shrinking the window to fit the image below made the display look cramped, but, in reality, it was fine.



We used a LaTeX editor called [Texmaker](#). If you want to install it in Ubuntu, type the following:

```
sudo apt-get install texmaker
```

In Fedora, the command is:

```
sudo dnf install texmaker
```

In Manjaro, use:

```
sudo pacman -Syu texmaker
```

Converting Files with Templates

You're probably starting to understand the flexibility that pandoc provides. You can write once and publish in almost any format. That's a great feat, but the documents do look a little vanilla.

With templates, you can dictate which styles pandoc uses when it generates documents. For example, you can tell pandoc to use the styles defined in a [Cascading Style Sheets](#) (CSS) file with the `--css` option.

We've created a small CSS file containing the text below. It changes the spacing above and below the level header one style. It also changes the text color to white, and the background color to a shade of blue:

```
h1 {
  color: #FFFFFF;
  background-color: #3C33FF;
  margin-top: 0px;
  margin-bottom: 1px;
}
```

The full command is below—note that we also used the standalone option (`-s`):

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

pandoc uses the single style from our minimalist CSS file and applies it to the level one header.

A Level One Heading

A line of normal text.

A Level Two Heading

- This will become
- a *bulleted*
- list

A Level Three Heading

1. This will become
2. a **numbered**
3. list

Another fine-tuning option you have available when working with HTML files is to include HTML markup in your Markdown file. This will be passed through to the generated HTML file as standard HTML markup.

This technique should be reserved for when you're only generating HTML output, though. If you're working with multiple file formats, pandoc will ignore the HTML markup for non-HTML files, and it will be passed to those as text.

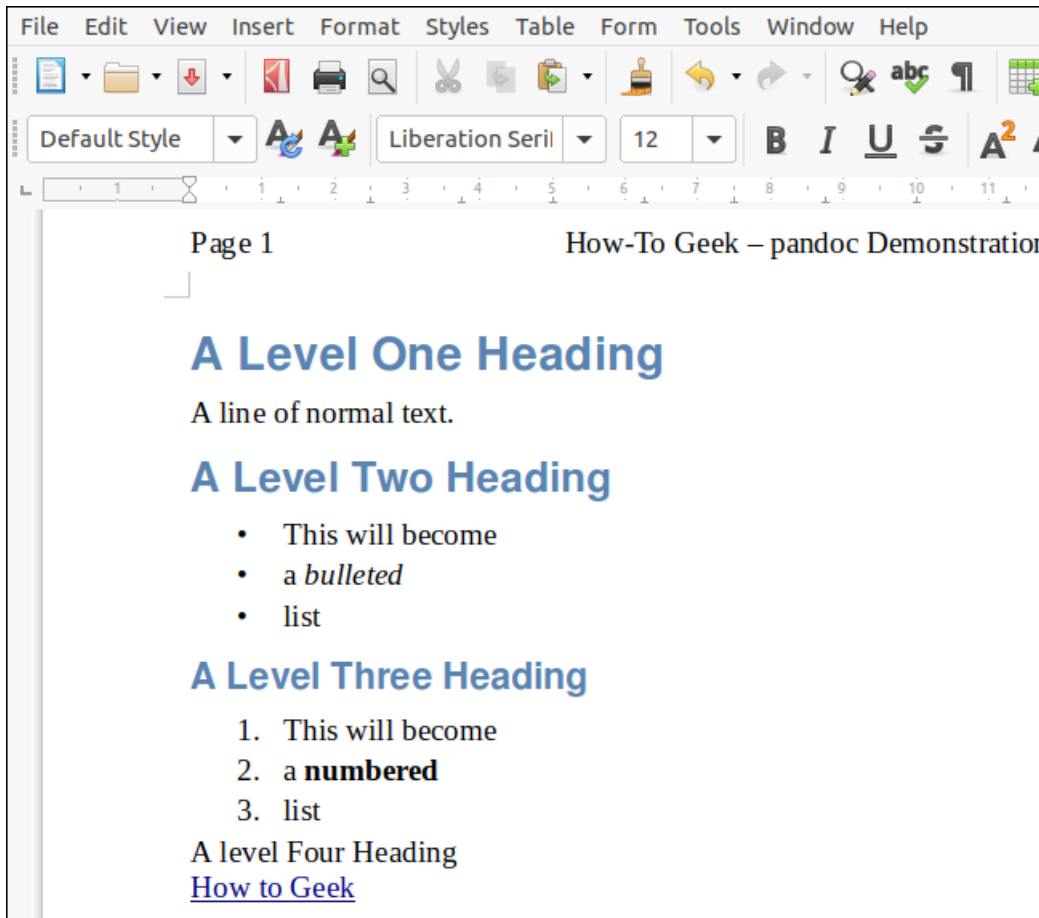
We can specify which styles are used when ODT files are generated, too. Open a blank LibreOffice Writer document and adjust the heading and font styles to suit your needs. In our example, we also added a header and footer. Save your document as "odt-template.odt."

We can now use this as a template with the `--reference-doc` option:

```
pandoc -o sample.odt --reference-doc=odt-template.odt sample.md
```

```
dave@ubuntu20-04:~/pd$ pandoc -o sample.odt --reference-doc=odt-template.odt sample.md
```

Compare this with the ODT example from earlier. This document uses a different font, has colored headings, and includes headers and footers. However, it was generated from the exact same “sample.md” Markdown file.



Reference document templates can be used to indicate different stages of a document’s production. For example, you might have templates that have “Draft” or “For Review” watermarks. A template without a watermark would be used for a finalized document.

Generating PDFs

By default, pandoc uses the LaTeX PDF engine to generate PDF files. The easiest way to make sure you have the appropriate LaTeX dependencies satisfied is to install a LaTeX editor, such as Texmaker.

That’s quite a big install, though—Tex and LaTeX are both pretty hefty. If your hard drive space is limited, or you know you’ll never use TeX or LaTeX, you might prefer to generate an ODT file. Then, you can just open it in LibreOffice Writer and save it as a PDF.

Docs-as-Code

There are several advantages to using Markdown as your writing language, including the following:

- **Working in plain text files is fast:** They load faster than similarly sized word processor files, and tend to move through the document faster, too. Many editors, including `gedit`, `Vim`, and `Emacs`, use syntax highlighting with Markdown text.
- **You'll have a timeline of all versions of your documents:** If you store your documentation in a VCS, such as Git, you can easily see the differences between any two versions of the same file. However, this only really works when the files are plain text, as that's what a VCS expects to work with.
- **A VCS can record who made any changes, and when:** This is especially helpful if you often collaborate with others on large projects. It also provides a central repository for the documents themselves. Many [cloud-hosted Git services, such as GitHub, GitLab, and BitBucket](#), have free tiers in their pricing models.
- **You can generate your documents in multiple formats:** With just a couple of simple shell scripts, you can pull in the styles from CSS and reference documents. If you store your documents in a VCS repository that integrates with [Continuous Integration and Continuous Deployment](#) (CI/CD) platforms, they can be generated automatically whenever the software is built.

RELATED: [What Is GitHub, and What Is It Used For?](#)

Final Thoughts

There are many more options and features within [pandoc](#) than what we've covered here. The conversion processes for most file types can be tweaked and fine-tuned. To learn more, check out the excellent examples on the official (and extremely detailed) [pandoc web page](#).

READ NEXT

- › [How to Restart a Google TV Streaming Device](#)
- › [How to Control the Exposure in the iPhone's Camera App](#)
- › [How to Add Folder Shortcuts to the Start Menu's Left Sidebar on Windows 10](#)
- › [How to Force Users to Change Their Passwords on Linux](#)
- › [What Is RAM? Everything You Need to Know](#)



DAVE MCKAY

Dave McKay first used computers when punched paper tape was in vogue, and he has been programming ever since. After over 30 years in the IT industry, he is now a full-time technology journalist. During his career, he has worked as a freelance programmer, manager of an international software development team, an IT services project manager, and, most recently, as a Data Protection Officer. Dave is a Linux evangelist and open source advocate. [READ FULL BIO »](#)

The above article may contain affiliate links, which help support How-To Geek.

How-To Geek is where you turn when you want experts to explain technology. Since we launched in 2006, our articles have been read more than 1 billion times. [Want to know more?](#)

How-To Geek



Enter Your Email

Sign Up