

# Technical Writing on Windows 10/11

via a state-of-the-art open-source toolchain

Russell D. Brunelle

June 7, 2022

## Abstract

This tutorial teaches you to create beautiful PDF documents from simple text files. Using Notepad++ you write Markdown that's backed up via Git, Pandoc typesets your Markdown via LaTeX while adding any flowcharts written in Mermaid, and all of this runs (without needing internet access) via VirtualBox on a virtual Debian Linux machine that's accessed via PuTTY.

## Contents

<b>1</b>	<b>Our goal</b>	<b>3</b>
<b>2</b>	<b>Configuring your basic tools</b>	<b>3</b>
2.1	Notepad++ . . . . .	3
2.1.1	Setup . . . . .	3
2.1.2	Spelling . . . . .	4
2.1.3	Color . . . . .	4
2.2	VirtualBox . . . . .	5
2.3	PuTTY . . . . .	5
2.3.1	Setup . . . . .	5
2.3.2	Use . . . . .	6
<b>3</b>	<b>Creating your virtual machine</b>	<b>6</b>
3.1	Downloading its installation “disc” . . . . .	6
3.2	Defining its virtual hardware . . . . .	7
3.3	Installing its operating system . . . . .	7
<b>4</b>	<b>Configuring your virtual machine</b>	<b>8</b>
4.1	Enabling SSH logins . . . . .	8
4.2	Activating your shared folders . . . . .	8
4.3	Adding your applications . . . . .	9
4.3.1	Pandoc . . . . .	10
4.3.2	Mermaid . . . . .	11

4.4	Refining Pandoc . . . . .	11
<b>5</b>	<b>Starting and stopping your virtual machine</b>	<b>12</b>
<b>6</b>	<b>Using your virtual machine</b>	<b>12</b>
6.1	Pandoc for documents . . . . .	12
6.1.1	Creating a beautiful PDF . . . . .	12
6.1.2	Exporting to Google Docs . . . . .	13
6.2	Mermaid for flowcharts . . . . .	13
<b>7</b>	<b>Free upgrades</b>	<b>15</b>
7.1	Learning keyboard shortcuts . . . . .	15
7.2	Skipping passwords with SSH keys . . . . .	15
7.2.1	Generating your keypair . . . . .	15
7.2.2	Distributing your public key . . . . .	16
7.2.3	Telling PuTTY to use your private key . . . . .	16
7.3	Using Git for backups and version control . . . . .	16
7.3.1	Installing Git itself . . . . .	16
7.3.2	Installing Git's LFS extension . . . . .	17
7.3.3	Personalizing Git . . . . .	17
7.3.4	Selecting a Git host . . . . .	18
7.3.5	Good practices . . . . .	18
7.4	Updating Mermaid . . . . .	18
<b>8</b>	<b>Unicode reference</b>	<b>19</b>
<b>9</b>	<b>Markdown reference</b>	<b>20</b>
9.1	Inline syntax . . . . .	20
9.2	Headings . . . . .	21
9.3	Ordinary lists . . . . .	21
9.4	Definition lists . . . . .	21
9.5	Example lists . . . . .	22
9.6	Blockquotes . . . . .	22
9.7	Figures . . . . .	23
9.8	Tables . . . . .	23
9.9	Code listings . . . . .	23
9.9.1	Plain . . . . .	23
9.9.2	Fancy . . . . .	24
9.10	Equations . . . . .	24
9.11	Metadata blocks . . . . .	25
9.12	Bibliographic blocks . . . . .	26
9.13	Everything else . . . . .	27
	<b>The auto-generated bibliography for our examples</b>	<b>27</b>

# 1 Our goal

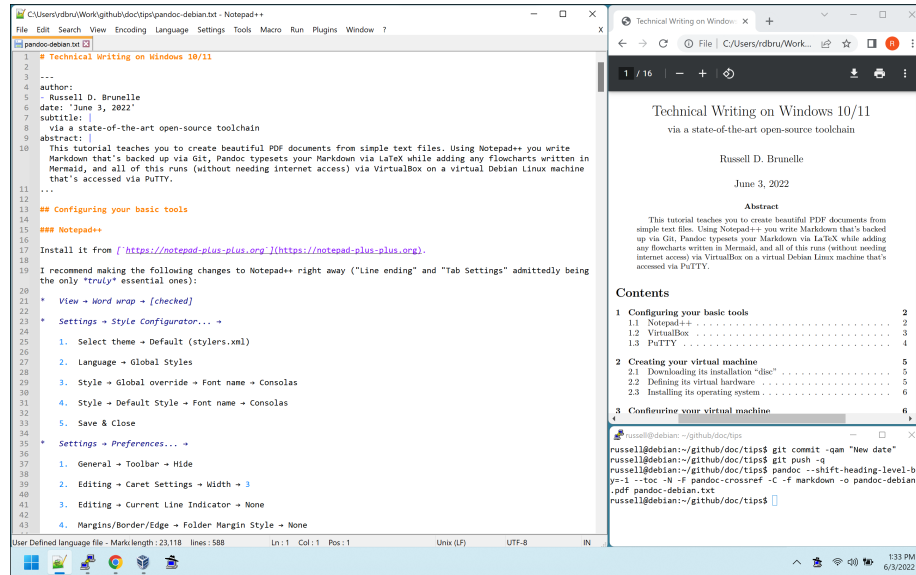


Figure 1: Your Screen

In the screen shown in figure 1, you'll write on the left, issue the typesetting command at bottom right, then proofread at top right.

## 2 Configuring your basic tools

### 2.1 Notepad++

#### 2.1.1 Setup

Install Notepad++ from <https://notepad-plus-plus.org>.

I recommend making the following changes to Notepad++ right away (“Line ending” and “Tab Settings” admittedly being the only *truly* essential ones):

- View → Word wrap → [checked]
- Settings → Style Configurator... →
  1. Select theme → Default (stylers.xml)
  2. Language → Global Styles
  3. Style → Global override → Font name → Consolas
  4. Style → Default Style → Font name → Consolas
  5. Save & Close

- Settings → Preferences... →
  1. General → Toolbar → Hide
  2. Editing → Caret Settings → Width → 3
  3. Editing → Current Line Indicator → None
  4. Margins/Border/Edge → Folder Margin Style → None
  5. Margins/Border/Edge → Line Number → Constant width
  6. New Document → Format (Line ending) → Unix (LF)
  7. Language → Tab Settings → Replace by space → [checked]
  8. Highlighting → Smart Highlighting → Enable → [unchecked]
  9. Print → Print line number → [unchecked]
  10. Backup → Enable session snapshot and periodic backup → [choose]<sup>1</sup>
  11. Auto-Completion → Enable auto-completion on each input → [unchecked]
  12. Auto-Completion → Function parameters hint on input → [unchecked]
  13. Close

### 2.1.2 Spelling

Spell check is provided via a plugin. Here's how you activate that plugin:

1. Plugins → Plugins Admin... → Available → DSpellCheck → [checked] → Install → Yes
2. Plugins → DSpellCheck → Settings... → Simple → Download... → [check desired language] → Install Selected → OK → Apply → OK
3. Plugins → DSpellCheck → Settings... → Advanced → Underline Style → Straight Box → Apply → OK
4. Plugins → DSpellCheck → Spell Check Document Automatically

### 2.1.3 Color

All of the above only has to be done once. But choosing one of the following two Markdown color modes is something you'll probably want to do every time you open a Markdown file for substantial work.

- If you're in ordinary light conditions:
  1. Language → Markdown (preinstalled)

---

<sup>1</sup>If your computer is stable, you're in the habit of regularly saving, and you'll be keeping your work under version control, then you may prefer to uncheck this.

- If you're in very low light conditions:
  1. Language → Markdown (preinstalled dark mode)
  2. Settings → Preferences... → Dark Mode → Enable dark mode → [checked] → Close

If you chose the low light option above, then you'll probably also want to do most of your work in full screen mode. The F11 key toggles that.

## 2.2 VirtualBox

Install it from <https://www.virtualbox.org>.

Two quick comments:

1. You may notice that a fresh installation of VirtualBox creates a default host-only VirtualBox network. There's no harm in ignoring this virtual network, but for our purposes there's also no harm in removing it via the VirtualBox Host Network Manager; this virtual network is only useful if not only will you be running multiple virtual machines simultaneously, but those virtual machines will need to communicate *with each other*. We're not doing anything remotely like that.
2. VirtualBox can't run on Windows PCs with ARM CPUs (i.e., the radically different type of CPU normally found in smartphones). But "Windows on ARM" PCs are currently only a *very* small part of the total Windows PC market, and it's similarly unlikely this limitation will apply to you.

## 2.3 PuTTY

### 2.3.1 Setup

Install it from <https://www.chiark.greenend.org.uk/~sgtatham/putty/>.

I recommend creating within PuTTY two saved sessions (one for "root" logins and one for ordinary user account logins) as follows:

- Session
  - Host Name: 127.0.0.1
  - Port: 10022
  - Close window on exit: Always
- Terminal
  - Bell
    - \* Action to happen when a bell occurs: None
    - \* Taskbar/caption indication on bell: Steady

- Window
  - Appearance
    - \* Font: Consolas
  - Selection
    - \* Ctrl + Shift + {C,V}: System clipboard
  - Colours
    - \* Use system colours: [checked]
- Connection
  - Data
    - \* Auto-login username: [root for your first saved session, and your preferred ordinary username for your second saved session (all lowercase with no spaces or punctuation)]

### 2.3.2 Use

Please note that you'll use an accompanying application called "Pageant" (which comes with your PuTTY installation) to actually initiate one of the login sessions you just configured. Using Pageant is more elegant, since:

- to initiate an SSH login you can simply right-click Pageant in the Windows system tray and pick the session you want under "Saved Sessions."
- logging out of any particular session doesn't cause Pageant to exit.

Because Pageant will be an everyday application, I strongly recommend making it always visible in the Windows system tray. Here's how you do that:

**Windows 11** Right-click the taskbar, select "Taskbar settings," click "Taskbar corner overflow," and finally switch "PuTTY SSH authentication agent" from Off to On.

**Windows 10** Same as Window 11, except the link you click in the taskbar settings screen is named "Select which icons appear on the taskbar."

## 3 Creating your virtual machine

### 3.1 Downloading its installation "disc"

You'll need to download a boot disc image in order to install your virtual machine's operating system. The correct file has a file name beginning simply with "debian" (not "debian-edu" or "debian-mac") and ending with ".iso", from the following list:

<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/>

## 3.2 Defining its virtual hardware

Start VirtualBox, and create the virtual hardware for your virtual machine as follows:

1. Machine → New...
  - Name: Debian {version}<sup>2</sup>
  - Virtual hard drive type: Fixed size
  - Virtual hard drive size: 15.00 GB (minimum)
2. Machine → Settings...
  - Storage → Empty → Choose a disk file... → [the boot image you just downloaded] → Open
  - Network → Advanced → Port Forwarding → Add → ssh / TCP / 127.0.0.1 / 10022 / [blank] / 22 → OK
  - Shared Folders
    - Add → [folder named “Downloads” (might be the default location for your web browser’s downloads)] → OK
    - Add → [folder named “Work” (might be where you keep all the stuff you’re actively working on)] → OK

## 3.3 Installing its operating system

Its virtual hardware having now been created, and its virtual CD-ROM drive having now been loaded with a virtual installation disc, you can proceed to installing your virtual machine’s operating system. Issue the VirtualBox command “Machine → Start → Normal Start” to begin the process; you’ll be going with either the default or with the only sensible choice at every subsequent installation prompt, except as noted below:

- At the main installer menu: pick “Install” rather than “Graphical install”
- Domain name: localdomain
- Full name: [your choice]
- Software selection: deselect “Debian desktop environment” and “GNOME”
- Device for boot loader: /dev/sda

Note that during this operating system installation process you’re guided through creating two different login accounts: one “root” user account, and one ordinary user account. Once you’ve finished configuring your virtual machine you’ll

---

<sup>2</sup>Including the Debian version number like this isn’t a requirement, but I think it’s helpful. If, for example, your disc image’s filename were `debian-11.3.0-amd64-netinst.iso` then this number would be 11, and hence you’d type: Debian 11

basically never use the former; it's the latter you'll use on a day-to-day basis to issue the `pandoc` commands that will typeset documents from your Markdown-format text files, the `mmdc` commands that will generate flowcharts from your Mermaid-format text files, and the `git` commands that will preserve all the Markdown and Mermaid text files you create.

In any case, once the operating system installation process is complete your virtual machine will automatically reboot, and you'll then be able to log into it using the console window that VirtualBox will present.

## 4 Configuring your virtual machine

### 4.1 Enabling SSH logins

Your first order of business is to enable the ability to log into your virtual machine via SSH using PuTTY, rather than via VirtualBox's console window, so that you can copy and paste commands instead of retyping them.

Log into your virtual machine as root via VirtualBox's console window, and type the following commands (granting permission to upgrade/install whenever asked):

```
apt-get update
```

```
apt-get dist-upgrade
```

```
apt-get install openssh-server
```

```
echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config.d/99-local.conf
```

```
systemctl restart ssh
```

```
exit
```

Congratulations! You can now log into your virtual machine via SSH using PuTTY<sup>3</sup> rather than via its console, which means you can now simply copy (Ctrl-c) and paste (Ctrl-Shift-v) commands from this tutorial into your virtual machine's command line.

### 4.2 Activating your shared folders

Your second order of business is to activate the folders that you shared with your virtual machine from Windows; by doing all of your creative work in these shared folders, Pandoc and Mermaid (running on your virtual machine) will be

---

<sup>3</sup>You'll need to click "Accept" on an alert regarding the "host key" being unknown the first time you do this, but that's normal.



able to access everything you edit in Notepad++ (running on Windows), and you'll be able to view all of their output in Windows.

Start by executing the following VirtualBox menu command, which loads a virtual CD-ROM that can teach your virtual machine how to access its shared folders:

1. Devices →
2. Insert Guest Additions CD image...

Continue by issuing the following commands as root on your virtual machine (as before, granting permission to install whenever asked):

```
apt-get install build-essential dkms linux-headers-$(uname -r)
```

```
mount /media/cdrom0
```

```
cd /media/cdrom0/
```

```
sh ./VBoxLinuxAdditions.run
```

```
mkdir -p /mnt/Downloads
```

```
echo "Downloads /mnt/Downloads vboxsf defaults,\nuid=1000,gid=1000,dmode=755,fmode=755,rw 0 0" >> /etc/fstab
```

```
mkdir -p /mnt/Work
```

```
echo "Work /mnt/Work vboxsf defaults,\nuid=1000,gid=1000,dmode=755,fmode=755,rw 0 0" >> /etc/fstab
```

```
shutdown -h now
```

At this point your virtual machine will be shut down. Back in the VirtualBox menu, issue the following two commands:

1. Machine → Settings... → Storage → VBoxGuestAdditions.iso → Remove Disk from Virtual Drive → OK
2. Machine → Start → Normal Start

### 4.3 Adding your applications

Your third order of business is to add to your virtual machine all the applications you'll actually use on a daily basis (e.g. Pandoc and Mermaid), plus some extra applications that might prove handy in specific circumstances.

### 4.3.1 Pandoc

Issue the following as root:

```
apt-get install zip

apt-get install vim

select-editor # pick /bin/nano

update-alternatives --set editor /bin/nano

apt-get install texlive-full

apt-get install librsvg2-bin

apt-get install curl

curl -fsSL https://deb.nodesource.com/setup_current.x | bash -

apt-get install nodejs

apt-get clean

wget https://github.com/jgm/pandoc/releases/\
download/2.18/pandoc-2.18-linux-amd64.tar.gz

tar xvzf pandoc-2.18-linux-amd64.tar.gz \
--strip-components 1 -C /usr/local/

rm pandoc-2.18-linux-amd64.tar.gz

wget https://github.com/lierdakil/pandoc-crossref/releases/\
download/v0.3.13.0/pandoc-crossref-Linux.tar.xz

xz -dc pandoc-crossref-Linux.tar.xz | tar x

mv pandoc-crossref /usr/local/bin/

gzip pandoc-crossref.1

mv pandoc-crossref.1.gz /usr/local/share/man/man1/

rm pandoc-crossref-Linux.tar.xz

apt-get install dos2unix
```

```
exit
```

### 4.3.2 Mermaid

Issue the following in your ordinary user account (*not* as root):

```
select-editor # pick /bin/nano

npm install @mermaid-js/mermaid-cli

mkdir -p .local

cd .local/

ln -s ../node_modules/.bin bin

exit
```

## 4.4 Refining Pandoc

Your final order of business is to refine Pandoc a bit by equipping it with another good citation style<sup>4</sup>, telling it how you want cross-references labeled, and giving it a directory for templates.

Issue the following in your ordinary user account:

```
mkdir -p .local/share/pandoc/csl

wget -P .local/share/pandoc/csl \
https://raw.githubusercontent.com/citation-style-language/styles/\
80aded68adbacd9873157295541a6d4f87515f46/ieee.csl

mkdir -p .pandoc-crossref

cat <<'EOF' >.pandoc-crossref/config.yaml
secPrefix:
- section
- sections
figPrefix:
- figure
- figures
tblPrefix:
- table
```

---

<sup>4</sup>Specifically the “ieee.csl” style, which of the common citation styles that don’t commandeer footnotes is the least intrusive to the reader; to select this style just add the line `csl: ieee.csl` to your Markdown file’s Pandoc metadata block.

```

- tables
lstPrefix:
- listing
- listings
eqnPrefix:
- equation
- equations
EOF

mkdir -p .local/share/pandoc/templates

exit

```

## 5 Starting and stopping your virtual machine

Assuming you're still comfortable logging in via SSH rather than the console, table 1 summarizes how to start and stop your virtual machine.

Table 1: VirtualBox start/stop commands

Action	Command
Start	Machine → Start → Headless Start
Stop	Machine → Close → ACPI Shutdown

## 6 Using your virtual machine

All of the following must be done in your ordinary user account, *not* as root.

### 6.1 Pandoc for documents

#### 6.1.1 Creating a beautiful PDF

Here's the simplest possible command to typeset a beautiful new PDF file (to be named "out.pdf") from a Markdown text file (in this case named "doc.txt"):

```
pandoc -f markdown -o out.pdf doc.txt
```

I strongly advise *always* adding `--shift-heading-level-by=-1` to the above command, and then adopting the habit of only having one level-one heading in your document, on its first line; this way, that first and only level-one heading is treated as your document's actual title, all following level-two headings are typeset as level-one headings, etc.

There are a few other Pandoc features you could enable:

1. Add `-C` if you'd like to use Pandoc's citation and bibliography features.

2. Before any `-C`, you can add `-N -F pandoc-crossref` if you'd like your document's headings automatically numbered, along with gaining the ability to easily reference those headings *with* their current numbers.<sup>5</sup>
3. Add `--toc` to automatically generate and include a table of contents.

Pulling all this together, if you wanted to enable every optional feature I just mentioned, plus the one setting I advised always making, then your Pandoc command would be this:

```
pandoc --shift-heading-level-by=-1 --toc \  
-N -F pandoc-crossref -C -f markdown -o out.pdf doc.txt
```

In fact, the above command is exactly what was used to typeset this tutorial.

### 6.1.2 Exporting to Google Docs

To create a Google Doc from your Markdown text file, simply change the output file's extension from `.pdf` to `.docx` in your Pandoc command; that will cause the output file to have Microsoft Word format, which Google Docs will accept as an upload.

If your Pandoc command requested either a table of contents or citation processing, then you might improve the Google Docs import by, just prior to uploading, opening your new `.docx` file in Word, answering "Yes" if asked to update fields, and then saving.

Here are a few tips for writing Markdown that will eventually become a Google Doc:

- Avoid nested blockquotes (i.e., indented quoted paragraphs within indented quoted paragraphs).
- If you're using `pandoc-crossref` to cross-reference figures and tables, then give *all* your figures and tables labels — not just the specific ones you actually cross-reference.

## 6.2 Mermaid for flowcharts

Mermaid lets you create attractive diagrams from simple and human-readable text files; basically, it does for a document's flowcharts what Markdown does for a document's words.

The Mermaid input file shown in listing 1, for which the command in listing 2 generated the output shown below, demonstrates all of the flowchart node types along with the most useful link types.

---

<sup>5</sup>Note that I said "before any `-C`"; Pandoc's cross-reference processing and citation processing each function as filters that are applied in the order they're given on the command line, and if the citation processor sees unprocessed cross-references it will mistakenly believe those unprocessed cross-references are just broken citations.

---

**Listing 1** Mermaid flowchart demonstration input

---

```
flowchart TB
  A --> B[rectangle]
  B ---|std open| C(rounded)
  C -.-|dot open| D([stadium])
  A --> E[[subroutine]]
  E -->|std arrow| F[(cylinder)]
  F -.->|dot arrow| G((circle))
  A --> H>asymmetric]
  H <-->|std arrow 2| I{rhombus}
  I <-.->|dot arrow 2| J{{hexagon}}
  A --> K[/pgram/]
  K --x|std cross| L[\pgram alt\]
  L -.-x|dot cross| M[/tzoid\]
  A --> N[\tzoid alt/]
  N x--x|std cross 2| O(((circle 2)))
  O x-.-x|dot cross 2| P[rectangle]
```

---

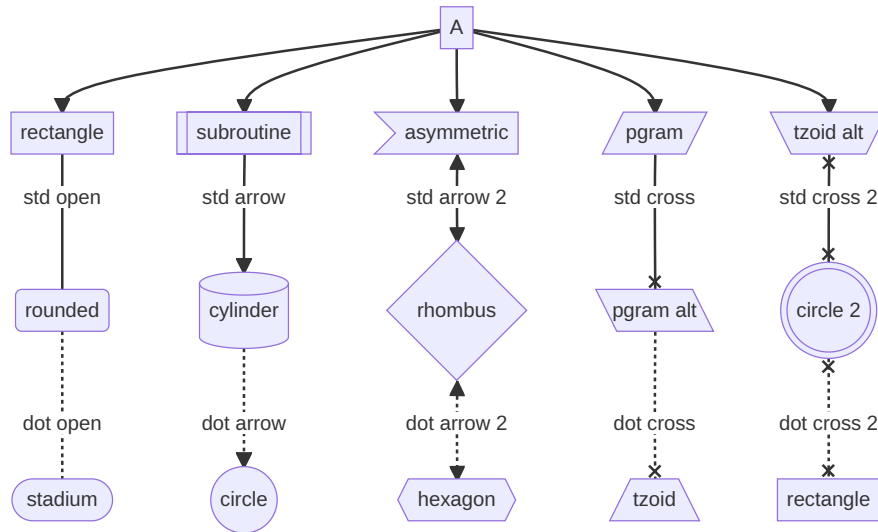
---

**Listing 2** Command to generate Mermaid flowchart demonstration output

---

```
mmdc -f -i pandoc-debian-fc.mmd -o pandoc-debian-fc.mmd.pdf
```

---



See <https://mermaid-js.github.io> to learn additional Mermaid flowchart syntax, as well as the syntax for all of the other diagram types Mermaid can generate (including sequence, class, state, entity relationship, user journey, Gantt, pie, requirement, and Gitgraph).

## 7 Free upgrades

Now that you have the basics, here are a few extra steps you could take.

### 7.1 Learning keyboard shortcuts

Since this workflow involves cycling through three specific Windows applications, it's perfectly suited to Win-# shortcutting. So pin Notepad++ to your taskbar, right-click your running PuTTY terminal on the taskbar to pin it to your taskbar as well, and finally pin your PDF reader to your taskbar. Then, arrange those three pinned applications on your taskbar in that same order, as the very first pins to the right of the Windows start button (as shown in figure 1).

Once that's done you'll always be able to instantly switch to your text editor with Win-1, your terminal with Win-2, and your PDF reader with Win-3, which in turn (assuming you're using Google Chrome as your PDF reader) means you can accelerate your entire workflow per table 2.

Table 2: Workflow shortcuts

Step	Keystroke	Purpose
1	Ctrl-s	Save your writing
2	Win-2	Switch to your terminal
3	Up Arrow	Recall last Pandoc command
4	Enter	Execute that Pandoc command
5	Win-3	Switch to your PDF reader
6	Ctrl-r	Reload your PDF file
7	Ctrl-0/+/-	Adjust the zoom level
8	Mouse Wheel	Scroll through your PDF file
9	Win-1	Switch back to your text editor

### 7.2 Skipping passwords with SSH keys

Typing a password every time you log into your virtual machine can become a drag. Switching from passwords to SSH keys lets you skip that.

#### 7.2.1 Generating your keypair

Your first step is to generate your cryptographic keypair. Here's how you do that using the "PuTTYgen" application that comes with PuTTY:

1. Start PuTTYgen.
2. Select "EdDSA" as the type of key to generate (at present leave the associated "curve" at its default of "Ed25519" for compatibility).

3. Click “Generate” and follow the subsequent instructions regarding randomly moving the pointer until your keypair has finally been generated.
4. This is optional, but I think it makes sense to change the “Key comment” from “eddsa-key-[date]” to “eddsa-key-[date]-virtualbox” just to make its purpose unmistakable later on; either way, assuming you’ll only be using this keypair to log into the virtual machine you just created, don’t put anything in either the “Key passphrase” field or the following “Confirm” field.
5. Click “Save private key,” making your key’s filename the same as its “Key comment” to avoid confusion later on.<sup>6</sup>

## 7.2.2 Distributing your public key

Next, you need to copy the public key from this keypair to your virtual machine, so that its associated private key can validate you instead of your typed-in password validating you.

To accomplish this, issue the following commands in your virtual machine twice: once while logged in as root, and once while logged into your ordinary user account. Where you see “[pasted public key]” in the following commands you should paste everything contained in the “Public key for pasting into OpenSSH authorized\_keys” field for your keypair from your PuTTYgen window.<sup>7</sup>

```
mkdir -m 700 ~/.ssh
```

```
echo '[pasted public key]' >> ~/.ssh/authorized_keys
```

```
exit
```

## 7.2.3 Telling PuTTY to use your private key

Finally, back in PuTTY itself, for each of your saved sessions load and then re-save that session after making the following simple change:

1. Connection → SSH → Auth
2. Choose your “.ppk” file as the “Private key file for authentication”

## 7.3 Using Git for backups and version control

### 7.3.1 Installing Git itself

Simply issue as root: `apt-get install git`

<sup>6</sup>PuTTYgen will automatically give this filename PuTTY’s preferred .ppk filename extension, so don’t actually type that extension when saving your key.

<sup>7</sup>There’s no harm done if you accidentally close the PuTTYgen window for your newly-generated keypair before copying and pasting its public key field; since the public key is recoverable from the material saved in the file for its associated private key, all you need to do is load back into PuTTYgen your .ppk private key file and you’ll see its public key again.



### 7.3.2 Installing Git's LFS extension

The following steps install a Git extension called “LFS,” which gives you the option of telling repository hosting services that certain types of files within your Git repository (particularly large binary files) should be handled in a more efficient way behind the scenes.

In the case of Markdown authors, directing right from a repository's beginning that any PNG, JPG, and PDF files be handled via LFS should help it maintain great performance even after a lifetime of heavy use.

Issue as root:

```
wget https://github.com/git-lfs/git-lfs/releases/download/v3.2.0/git-lfs-linux-amd64-v3.2.0.tar.gz

tar xzvf git-lfs-linux-amd64-v3.2.0.tar.gz

rm git-lfs-linux-amd64-v3.2.0.tar.gz

cd git-lfs-3.2.0/

./install.sh

chown -R root:root man

gzip man/man1/*.1

gzip man/man5/*.5

mkdir -p /usr/local/man/man1

cp man/man1/*.1.gz /usr/local/man/man1/

mkdir -p /usr/local/man/man5

cp man/man5/*.5.gz /usr/local/man/man5/

cd ..

rm -rf git-lfs-3.2.0/

exit
```

### 7.3.3 Personalizing Git

Issue the following in your ordinary user account, replacing [name] and [address] with however you'd like your commits logged.

```
git config --global user.name "[name]"  
  
git config --global user.email "[address]"  
  
git config --global pull.rebase false  
  
git lfs install  
  
git config --global core.filemode false
```

In case you were curious, the above “core.filemode” setting is helpful because you’ll presumably be cloning your repository to a working directory in one of your shared Windows folders; since Windows filesystems don’t preserve UNIX filemodes, Git might otherwise mistakenly believe those filemodes were intentionally changed.

### 7.3.4 Selecting a Git host

Next, sign up for a free account with a Git hosting service such as <https://github.com>. You can then follow that service’s instructions to:

1. Create a private Git repository for your writing.
2. Configure a suitable form of authentication between your virtual machine and your hosted Git account (one popular choice being SSH keypairs).
3. “Clone” your new repository to the computer you’re using, via a command from your virtual machine.
4. Declare any filename extensions you want handled via LFS.

### 7.3.5 Good practices

- Issue your Git commands from the command line of your virtual machine’s ordinary user account.
- Clone your repository to a working directory on one of your virtual machine’s shared folders, so that you can edit its files with Notepad++.
- Don’t commit Pandoc or Mermaid outputs — only their inputs.

## 7.4 Updating Mermaid

Of all the applications now installed on your virtual machine, Mermaid is unique in that you can update it to later versions entirely from the command line of your ordinary user account.

This convenience is fortunate, since given GitHub’s early 2022 enablement of Mermaid syntax across all its discussion forums, Mermaid’s development is likely to accelerate.

So, here's how you can update Mermaid while logged into your ordinary user account:

```
npm outdated
npx npm-check-updates -u
npm install
```

## 8 Unicode reference

Table 3 summarizes the supported printable characters from the following code charts: Latin-1 Supplement (00A-00F), Latin Extended-A (010-017), Latin Extended-B (01C-023), Latin Extended Additional (1E0-1E2), General Punctuation (201-204), Currency Symbols (20A-20B), and Arrows (219).

If you'd like to learn more about a character you don't recognize, then go to <https://www.unicode.org/charts/> and click its code chart name.

Table 3: Select Unicode characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00A		ı	ċ	ƒ	ꝛ	¥	ı	§	¨	©	ª	«	¬		®	¯
00B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
010	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Č	č	Č	č	Ď	ď
011	Đ	đ	Ē	ē	Ĕ	ĕ	Ė	ė	Ě	ě	Ĝ	ĝ	Ğ	ğ	Ģ	ģ
012	Ġ	ġ	Ģ	ģ	Ĥ	ĥ			Ī	ī	Ĭ	ĭ	Ĵ	ĵ	Ĳ	ĳ
013	Ĭ	ı			Ĵ	ĵ	ķ	ķ		Ĺ	ĺ	Ł	ł	Ł	ł	
014		Ł	ł	Ń	ń	Ņ	ņ	Ñ	ñ		Đ	đ	Ō	ō	Ŏ	ö
015	Œ	œ	Ɔ	œ	Ř	ř	Ŕ	ŕ	Ř	ř	Ś	ś	Ŝ	ŝ	Ş	ş
016	Š	š	Ţ	ţ	Ť	ť			Ũ	ũ	Ū	ū	Ŭ	ŭ	Ů	ů
017	Ů	ů	Ū	ū	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ž	ž	Ž	ž	Ž	ž	
01C														Ă	ă	Ĭ
01D	ÿ	ÿ	ÿ	ÿ	ÿ											
01E			Æ	æ			Ğ	ğ	Ķ	ķ	Œ	œ				
01F	ĵ				Ģ	ģ										
021									Ş	ş	Ţ	ţ				
023			Ȳ	ȳ				J								
1E0			Ȳ	ȳ												
1E2	Ĝ	ĝ														
201	-		-	-	-	-			‘	’	,		“	”	„	
202	†	‡	•				...									

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
203	%									<	>				?	
204					/											
20A		©			£		№			₩		¤	€			
20B		®														
219	←	↑	→	↓												

## 9 Markdown reference

### 9.1 Inline syntax

Table 4: Basic inline formatting

Text	Result
<code>*italic*</code>	<i>italic</i>
<code>**bold**</code>	<b>bold</b>
<code>***italic bold***</code>	<i><b>italic bold</b></i>
<code>`code`</code>	code
<code>~~strikeout~~</code>	<del>strikeout</del>
<code>~sub~script</code>	sub <sub>script</sub>
<code>^super^script</code>	super <sup>script</sup>
<code>'single quotes'</code>	‘single quotes’
<code>"double quotes"</code>	“double quotes”
<code>en--dash</code>	en–dash
<code>em---dash</code>	em—dash
<code>\$c=\sqrt{a^2+b^2}\$</code>	$c = \sqrt{a^2 + b^2}$
<code>foot^[note]</code>	foot <sup>8</sup>

Table 5: Cross-reference formatting

Text	Result
<code>[@Sec:headings]</code>	Section 9.2
<code>[@sec:headings]</code>	section 9.2
<code>[@fig:ui]</code>	figure 1
<code>[@tbl:veg]</code>	table 8
<code>[@lst:codedemo]</code>	listing 3
<code>[@eq:ra;@eq:ma]</code>	equations 1, 2

<sup>8</sup>note

Table 6: Default citation formatting

<b>Text</b>	<b>Result</b>
@PCM2022	Proulx, Conroy, and McNair (2022)
@WatsonCrick1953 [p. 737]	J. D. Watson and Crick (1953, 737)
[@Watson2020, pp. 2-3]	(E. Watson 2020, 2-3)

Table 7: URL formatting

<b>Text</b>	<b>Result</b>
[`http://hp.com`](http://hp.com)	<a href="http://hp.com">http://hp.com</a>

## 9.2 Headings

The heading of the subsection you're reading now was typed as follows, with an optional #sec label for the sake of cross-referencing from table 5:

```
### Headings {#sec:headings}
```

## 9.3 Ordinary lists

If you type this:

1. First step
  - \* One consideration
  - \* Another consideration

2. Second step

You get this:

1. First step
  - One consideration
  - Another consideration
2. Second step

## 9.4 Definition lists

If you type this:

```
Term 1
: This is an explanation of "Term 1."
```

Term 2  
: This is an explanation of "Term 2."

**You get this:**

**Term 1** This is an explanation of "Term 1."  
**Term 2** This is an explanation of "Term 2."

## 9.5 Example lists

**If you type this:**

(©) A first example

Any amount of material can be placed here.

Sequential numbering of the examples won't be disturbed.

Ever.

(©) A second example

(@good) A third example

As (@good) illustrates, labels facilitate cross-referencing.

**You get this:**

(1) A first example

Any amount of material can be placed here.

Sequential numbering of the examples won't be disturbed.

Ever.

(2) A second example

(3) A third example

As (3) illustrates, labels facilitate cross-referencing.

## 9.6 Blockquotes

**If you type this:**

```
> Creating blockquotes is easy, and manual\  
line breaks can\  
be very useful if you're quoting poetry.  
>  
>> You can also nest blockquotes.
```

### You get this:

Creating blockquotes is easy, and manual line breaks can be very useful if you're quoting poetry.

You can also nest blockquotes.

## 9.7 Figures

Here's the Markdown line near the beginning of this tutorial which added figure 1, gave it its caption, gave it a label for later cross-referencing, and specified how much of the available page width it should be scaled to use:

```
![Your Screen](pandoc-debian-ui.png){#fig:ui width=100%}
```

In general, use PDF for vector graphics (including Mermaid charts), JPG for photographs, and PNG for everything else (including user interface screenshots).

## 9.8 Tables

Pandoc offers several methods for defining tables. Here's a simple example of one of the simpler methods, which includes both an optional caption and an optional label for later cross-referencing.

If you type this, you get table 8.

```
**veg** | **price**  
-----:|:-----  
garlic   | 1.12  
kale     | 9.63  
carrot   | 4.27
```

: A simple table example {#tbl:veg}

Table 8: A simple table example

veg	price
garlic	1.12
kale	9.63
carrot	4.27

## 9.9 Code listings

### 9.9.1 Plain

An easy way to present computer code (or anything else you want typeset verbatim in a monospaced font) is to simply bracket it with lines of backticks.

For example:

```
****
Code line 1
Code line 2
Code line 3
Code line 4
****
```

### 9.9.2 Fancy

The fancy way to present computer code, which also provides captioning and cross-referencing, is instead something like the following. When typeset, it yields listing 3.

```
```{#lst:codedemo caption="Demo of code block crossref/caption"}
main :: IO ()
main = putStrLn "Code block crossref/caption demo"
```
```

---

**Listing 3** Demo of code block crossref/caption

---

```
main :: IO ()
main = putStrLn "Code block crossref/caption demo"
```

---

### 9.10 Equations

If you bracket an equation with two dollar signs on each side, instead of just one dollar sign on each side, then your equation will be typeset in display format instead of inline format. In the following two examples you'll also see an optional #eq label immediately after the typed-in equation, for the sake of our cross-referencing examples in table 5.

**If you type this:**

```
$$\frac{1}{\text{Bigl}(\sqrt{\phi \sqrt{5}}-\phi\text{Bigr})}
e^{\frac{25}{\pi}} = 1+\frac{e^{-2\pi}}{1+\frac{e^{-4\pi}}{1+\frac{e^{-6\pi}}{1+\frac{e^{-8\pi}}{1+\dots}}}} } }$$ {#eq:ra}
```

```
$$\begin{aligned} \nabla \times \vec{\mathbf{B}} - \frac{1}{c} \frac{\partial \vec{\mathbf{E}}}{\partial t} &= \frac{4\pi}{c} \vec{\mathbf{j}} \quad \nabla \cdot \vec{\mathbf{E}} &= 4\pi \rho \quad \nabla \times \vec{\mathbf{E}} &= -\frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} \quad \nabla \cdot \vec{\mathbf{B}} &= 0 \end{aligned}$$ {#eq:ma}
```



You get this:

$$\frac{1}{(\sqrt{\phi\sqrt{5}} - \phi)e^{\frac{2}{3}\pi}} = 1 + \frac{e^{-2\pi}}{1 + \frac{e^{-4\pi}}{1 + \frac{e^{-6\pi}}{1 + \frac{e^{-8\pi}}{1 + \dots}}}} \quad (1)$$

$$\begin{aligned} \nabla \times \vec{\mathbf{B}} - \frac{1}{c} \frac{\partial \vec{\mathbf{E}}}{\partial t} &= \frac{4\pi}{c} \vec{\mathbf{j}} \\ \nabla \cdot \vec{\mathbf{E}} &= 4\pi\rho \\ \nabla \times \vec{\mathbf{E}} + \frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} &= \vec{\mathbf{0}} \\ \nabla \cdot \vec{\mathbf{B}} &= 0 \end{aligned} \quad (2)$$

## 9.11 Metadata blocks

Please consider the first 18 lines of this tutorial's Markdown file:

```
# Technical Writing on Windows 10/11

---
author:
- Russell D. Brunelle
date: 'June 7, 2022'
subtitle: |
  via a state-of-the-art open-source toolchain
abstract: |
  This tutorial teaches you to create beautiful PDF documents
  from simple text files. Using Notepad++ you write Markdown
  that's backed up via Git, Pandoc typesets your Markdown via
  LaTeX while adding any flowcharts written in Mermaid, and all
  of this runs (without needing internet access) via VirtualBox
  on a virtual Debian Linux machine that's accessed via PuTTY.
...
```

```
## Our goal
```

Since the Pandoc command that converted this tutorial's Markdown file to a PDF file included a `--shift-heading-level-by=-1` option, the `# Technical Writing on Windows 10/11` line above becomes this document's title rather than a top-level section heading, and the `## Our goal` line becomes a top-level section heading rather than a second-level section heading.

In the metadata block between those two lines, one could:

- Remove any definitions (author, date, subtitle, or abstract) that aren't needed, or even remove the entire block if none of them is needed.

- Add co-authors by adding more - [name] lines to the author definition.
- Add, on a line of its own, the definition `cs1: ieee.csl` if you'd like to switch from Pandoc's default citation style (*Chicago Manual of Style* "author-date") to the IEEE citation style.

## 9.12 Bibliographic blocks

Here is the bibliographic block that was typed into this tutorial's Markdown file, for the sake of the citation demonstrations in table 6.

---

references:

```
- id: PCM2022
  author:
    - family: Proulx
      given: Mike
    - family: Conroy
      given: Emma
    - family: McNair
      given: Corey
  title: "Cancel culture is loud, but for most brands it's noise"
  publisher: Forrester Research, Inc.
  issued:
    - year: 2022
      month: 1
      day: 10
  type: report

- id: WatsonCrick1953
  author:
    - family: Watson
      given: James Dewey
    - family: Crick
      given: Francis Harry Compton
  title: 'Molecular structure of nucleic acids'
  container-title: Nature
  issued:
    - year: 1953
      month: 4
      day: 25
  volume: 171
  issue: 4356
  page: 737-738
  DOI: 10.1038/171737a0
  type: article-journal
```

```
- id: Watson2020
  author:
  - family: Watson
    given: Erin
  title: '3 things to remember to avoid being cancelled'
  issued:
  - year: 2020
    month: 2
  URL: https://banfield.agency
  accessed:
  - year: 2022
    month: 5
    day: 7
  type: webpage
...
```

### 9.13 Everything else

We haven't covered absolutely everything, but the following will:

- <https://pandoc.org>
  - **For Pandoc's Markdown extensions:**  
Documentation → User's Guide → Pandoc's Markdown
  - **For Pandoc's citation features:**  
Documentation → User's Guide → Citations
- <https://lierdakil.github.io/pandoc-crossref/>

## The auto-generated bibliography for our examples

- Proulx, Mike, Emma Conroy, and Corey McNair. 2022. "Cancel Culture Is Loud, but for Most Brands It's Noise." Forrester Research, Inc.
- Watson, Erin. 2020. "3 Things to Remember to Avoid Being Cancelled." February 2020. <https://banfield.agency>.
- Watson, James Dewey, and Francis Harry Compton Crick. 1953. "Molecular Structure of Nucleic Acids." *Nature* 171 (4356): 737–38. <https://doi.org/10.1038/171737a0>.