How not to install Debian Gnu/Linux in a HP chromebook 11 G4.

I am loosely following this guide (but with some extra explanations)
http://linux-exynos.org/wiki/HP_Chromebook_11/Installing_Linux

Disclaimer.
I'm sure you have heard that this will void the warranty, that linux is super weird, difficult and dangerous, that you will have to pay for the computer or that you will become autistic if you use linux. Not that I don't care about that, but I have written this procedure WITHOUT actually having done it myself.
Oh, how I would like to get my hands on one of those things! But seriously, this procedure is really unreliable, by far the worst procedure I have ever written.
The upside is that if you actually do make it, then you can improve this document for the good of all chromebook-enduring students.

For this procedure you will need your fully charged chromebook (if possible with the charger itself), a usb memory stick with a bootable image of Debian, an accessible wireless network that is connected to the internet, a full backup of all the files that are remotely important and patience.

The first step is to gain root access to your chromebook.

Root is a user that has permissions to make any changes in the operating system.
Root is not a specific person, but a role: you can “act as root” for some time, when you want to make the changes, and then you go back to being a regular user, with regular permissions.
In order to gain root access you have to enable “developer mode” in your chromebook.
While the chromebook is on, press and hold the esc button and refresh keys, and while those two are pressed, press the power button and release all three.

The screen will go black and the computer will reboot, but this time a white screen will appear saying something like “chrome os is missing or damaged. Please insert a recovery usb stick.”
As usual, it means almost the opposite to what it says; chrome os is fine (but not for long).

To go further you have to push the Ctrl key and while it is pushed, push d and release both.
Another message will appear saying that you have to push enter to turn OS verification off (which is the idea). This is the step when all the files are deleted and the system will reboot. Press enter (it might take a few seconds to react).
While it works and before it reboots, another message reminding you that the OS verification is off will appear. Don't re-enable it, just wait until it reboots.
A message saying that the transition to developer mode is taking place will appear. Again, just wait. Depending on the computer this may take a few seconds or one minute (it will say that it is working and you can't turn off the computer now).
Even though all your files are deleted, the operating system ChromeOS is still working (which raises a number of questions, in my opinion), so it will boot into chromeOS just fine.

Now that we are in developer mode, we can open a shell.
Shell is (for those of you who are not familiar with the awesomeness of UNIX-like systems) something like a window where you can write commands for the computer to execute (my friends in Dlabs Hackerspace would flog me for such a loose definition).
When we open a shell we have a huge repertoir of commands available, but not all of them. Only the root user can give any command.

When ChromeOS boots, log in normally and when you are logged in, open a shell by pushing and holding Ctrl and Alt keys and while these two are pressed, press t and release all three.

A black window with white letters will appear. That is your shell.

It is a good shell, but not good enough, we want a “bash shell”, and in order to get that we type:

```
shell
```

and press enter.

Now we have technically a linux bash shell, which is great, but we need to log in as root. To do that we type:

```
sudo su
```

and press enter.

Now we are root and we have a shell. There is no end to the things we can do with this.

But we are only going to do two:

We need to tell the computer that the next time it boots, try to boot from the usb stick (where we will have the Debian bootable image). We do so by typing

```
crossystem dev_boot_usb=1 dev_boot_signed_only=0
```

and press enter.

That command also tells the computer that, when booting, it should not demand “signed” operating systems (signed means that you can only install “certain” operating systems).

Now reboot the system (you can give the command reboot and press enter) so that the changes take effect. As you reboot, insert the usb stick with the debian operating system and, if everything is correct, the debian installation program will take over.

From here on, the procedure is just the installation of a Debian operating system, and you can check how to do it in this file a weirdo wrote

tachikoma.org:6550/share/complete.pdf

If you want to get familiar with linux, you can join the Linux Support Group in Telegram
https://t.me/joinchat/ENtDFQ1yzoOS_QcwNefiBw