

Document for the Raspberry Pi novices

Part 1:

In this part, you will set up the basic Raspberry Pi settings. After finishing it, you can startup in your Raspberry Pi displaying on your own screen, or you can connect to the Raspberry Pi with your laptop/PC with VNC Viewer.

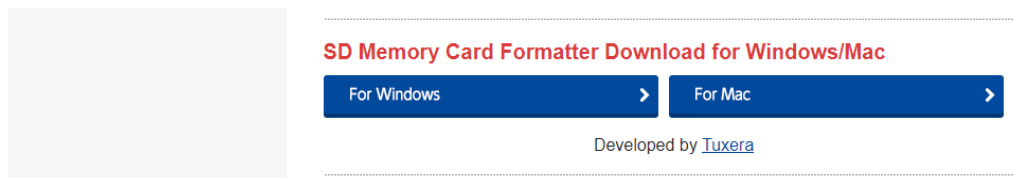
Please follow the steps below:

1. Formatting the SD card:

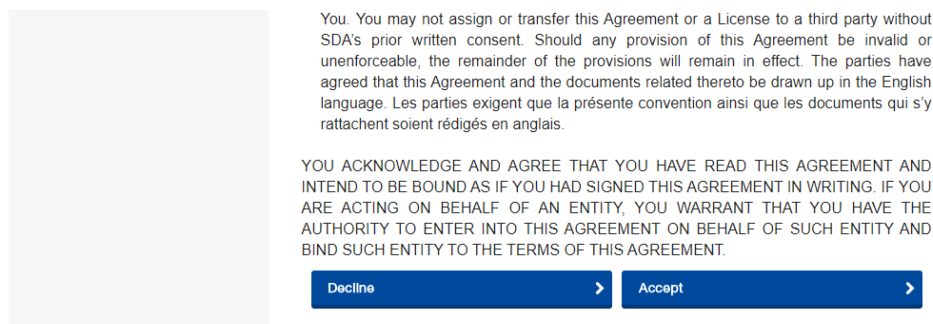
Go to this URL: https://www.sdcard.org/downloads/formatter_4/ to download the SD Memory Card Formatter.



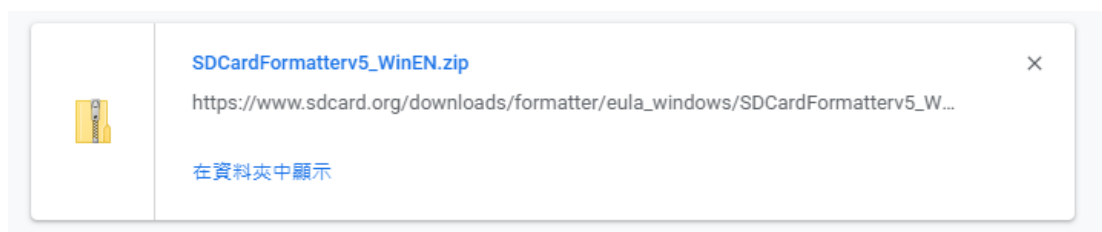
In the bottom of the page, it have the link.



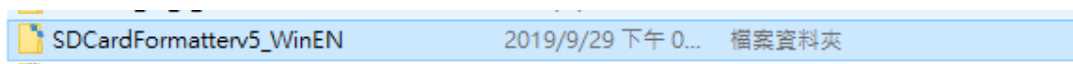
Choose the for your System. For me, it is the Windows.



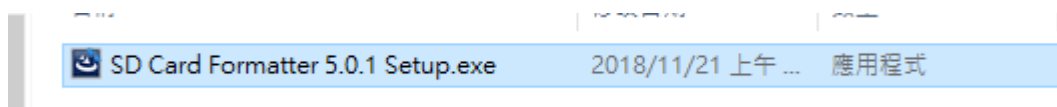
At the next page's bottom have the "Accept" button, click it.



And then, you will get the zip file. Unzip it, you will get the directory.



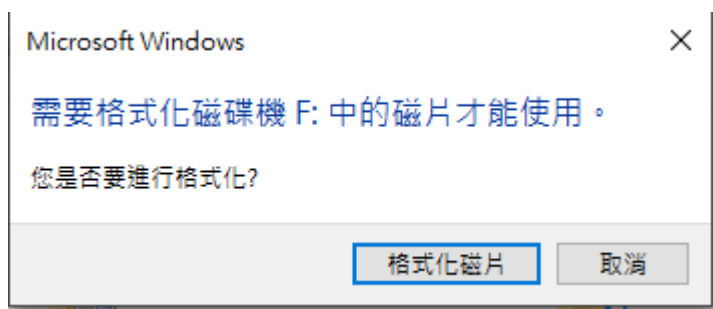
In the directory, it have the file "SD Card Formatter 5.0.1 Setup.exe". We will use this to format our SD Card.



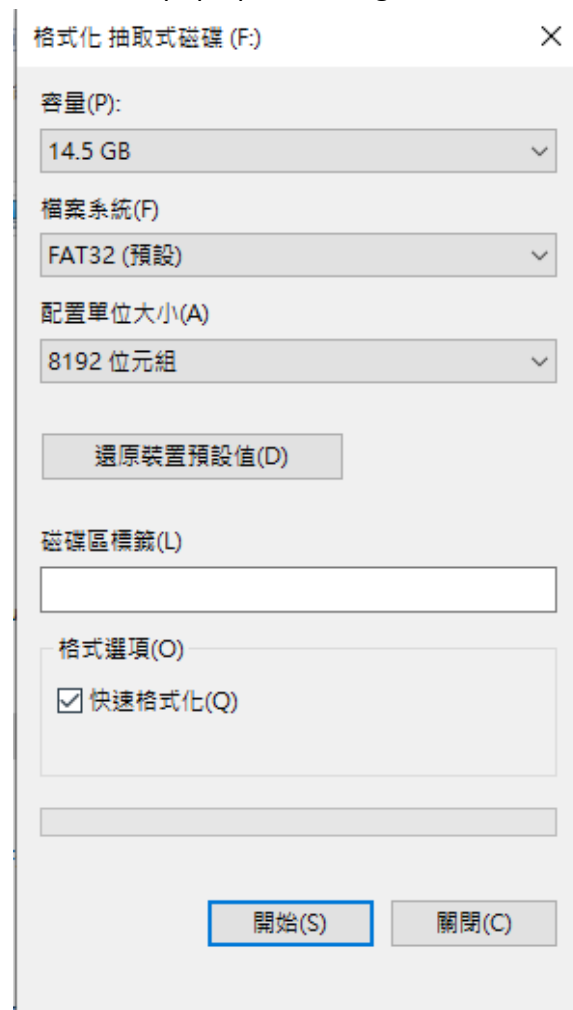
Now, please insert your SD Card and you will see the two disk:



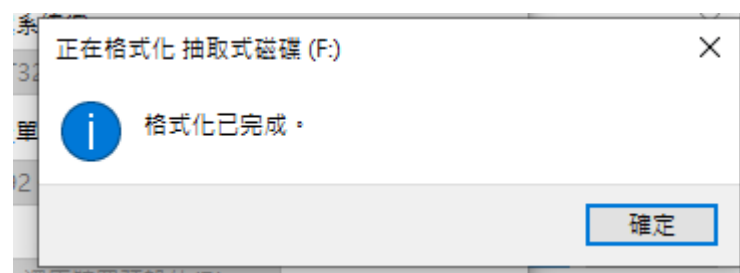
Click the F disk, and click the "Format disk"



Then it will pop up the dialog, and the settings will like this:





Click the start and you will see like this:



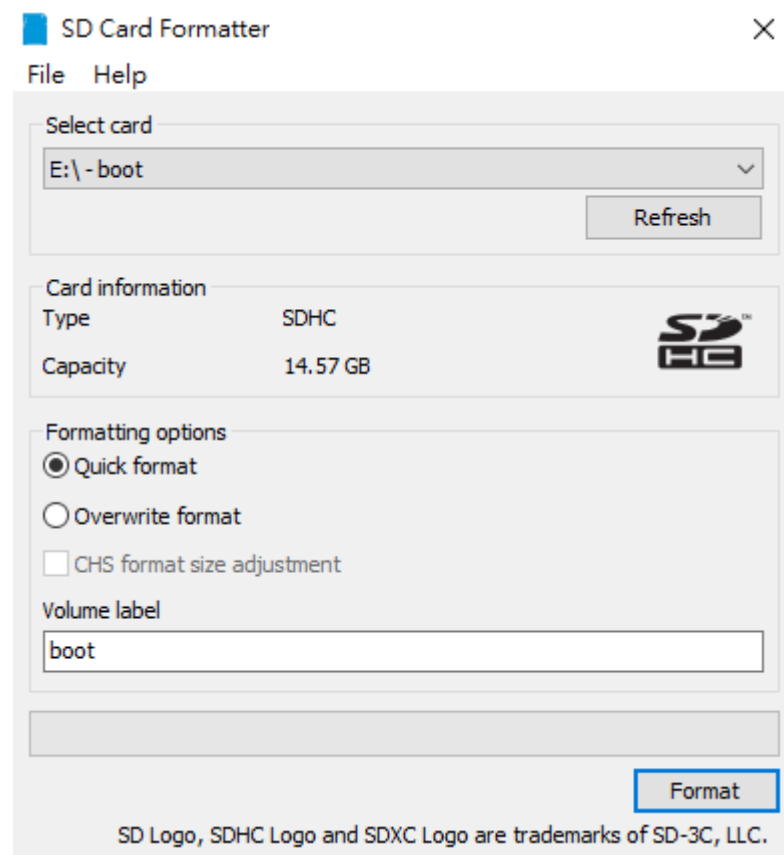
Execute the file you previously download

 SD Card Formatter 5.0.1 Setup.exe	2018/11/21 上午 ...	應用程式	6,877 KB
---	-------------------	------	----------

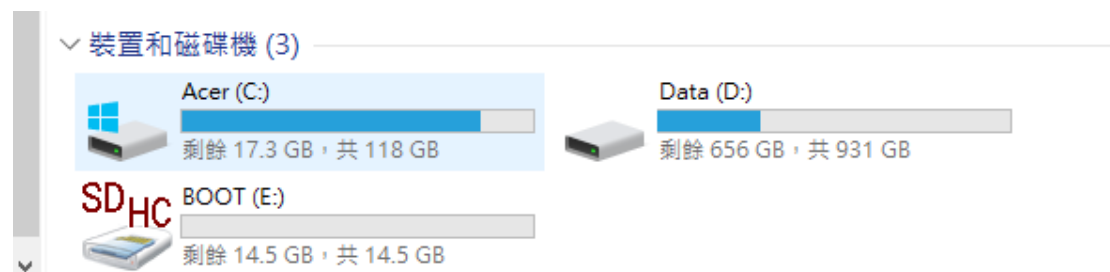
And you will get the file in C:\Program Files (x86)\SDA\SD Card Formatter

C:\Program Files (x86)\SDA\SD Card Formatter				搜尋 SD Card Form
名稱	修改日期	類型	大小	
 format_sd.exe	2018/10/23 下午 ...	應用程式	1,115 KB	
 SD Card Formatter.exe	2018/10/23 下午 ...	應用程式	3,545 KB	

Use the SD Card Formatter.exe with the settings. Select card is which you see with the name “boot”. Then Click the Format.

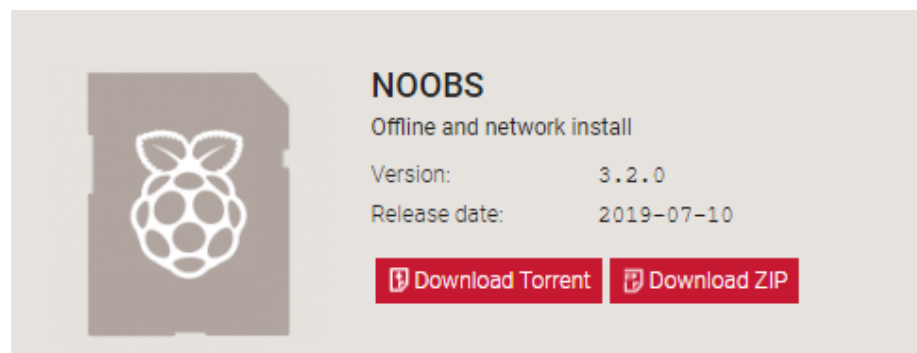


Then you will see the F disk disappears only the E disk left





2. Installing operating system for Raspberry PI:

Go to the <https://www.raspberrypi.org/downloads/noobs/> and choose the Download ZIP.



You will get the zip file, and then unzip it to get the “NOOBS_v3_2_0” directory.

 NOOBS_v3_2_0.zip	2019/9/17 下午 0...	壓縮的 (zipped) ...	2,200,087...
 NOOBS_v3_2_0	2019/9/17 下午 0...	檔案資料夾	

Copy all of the file in the “NOOBS_v3_2_0” directory into your SD Card (E disk)

組合管理					新增	開啟	選取
Course > 三上 > IOT > Week2 > NOOBS_v3_2_0					搜尋 NOOBS_v3_2_0		
名稱	修改日期	類型	大小				
defaults	2019/9/17 下午 0...	檔案資料夾					
os	2019/9/17 下午 0...	檔案資料夾					
overlays	2019/9/17 下午 0...	檔案資料夾					
bcm2708-rpi-b.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-b-plus.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-cm.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-zero.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-zero-w.dtb	2019/9/17 下午 0...	DTB 檔案	24 KB				
bcm2709-rpi-2-b.dtb	2019/9/17 下午 0...	DTB 檔案	24 KB				
bcm2710-rpi-3-b.dtb	2019/9/17 下午 0...	DTB 檔案	26 KB				
bcm2710-rpi-3-b-plus.dtb	2019/9/17 下午 0...	DTB 檔案	26 KB				
bcm2710-rpi-cm3.dtb	2019/9/17 下午 0...	DTB 檔案	25 KB				
bcm2711-rpi-4-b.dtb	2019/9/17 下午 0...	DTB 檔案	39 KB				
bootcode.bin	2019/9/17 下午 0...	BIN 檔案	52 KB				
BUILD-DATA	2019/9/17 下午 0...	檔案	1 KB				
INSTRUCTIONS-README.txt	2019/9/17 下午 0...	文字文件	3 KB				
recover4.elf	2019/9/17 下午 0...	ELF 檔案	743 KB				

組合管理					新增	開啟	選取
> BOOT (E:)					搜尋 BOOT (E:)		
名稱	修改日期	類型	大小				
defaults	2019/9/17 下午 0...	檔案資料夾					
os	2019/9/17 下午 0...	檔案資料夾					
overlays	2019/9/17 下午 0...	檔案資料夾					
bcm2708-rpi-b.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-b-plus.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-cm.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-zero.dtb	2019/9/17 下午 0...	DTB 檔案	23 KB				
bcm2708-rpi-zero-w.dtb	2019/9/17 下午 0...	DTB 檔案	24 KB				
bcm2709-rpi-2-b.dtb	2019/9/17 下午 0...	DTB 檔案	24 KB				
bcm2710-rpi-3-b.dtb	2019/9/17 下午 0...	DTB 檔案	26 KB				
bcm2710-rpi-3-b-plus.dtb	2019/9/17 下午 0...	DTB 檔案	26 KB				
bcm2710-rpi-cm3.dtb	2019/9/17 下午 0...	DTB 檔案	25 KB				
bcm2711-rpi-4-b.dtb	2019/9/17 下午 0...	DTB 檔案	39 KB				
bootcode.bin	2019/9/17 下午 0...	BIN 檔案	52 KB				
BUILD-DATA	2019/9/17 下午 0...	檔案	1 KB				
INSTRUCTIONS-README.txt	2019/9/17 下午 0...	文字文件	3 KB				
recover4.elf	2019/9/17 下午 0...	ELF 檔案	743 KB				

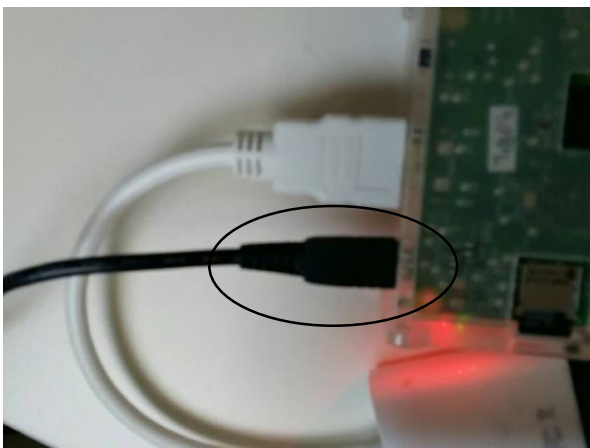
3. Boot up your Raspberry PI:
Insert your SD Card into the Raspberry PI

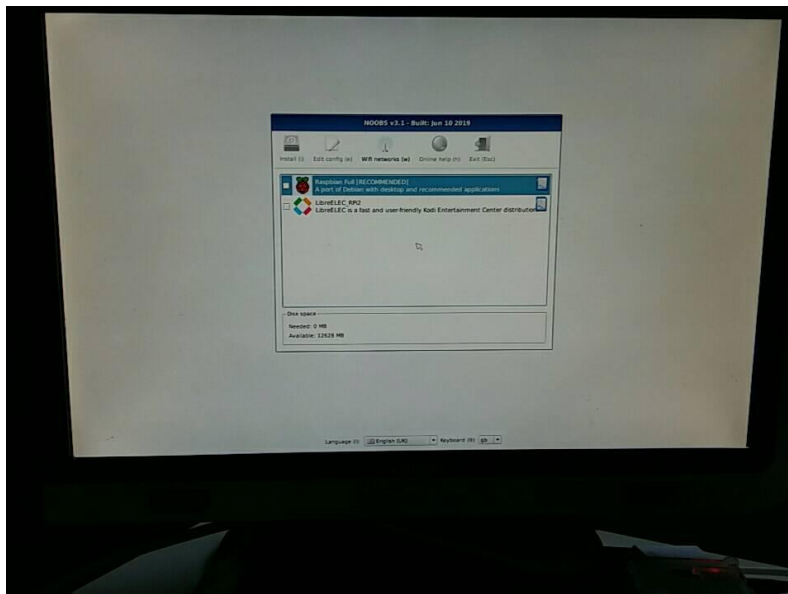


Link the "Keyboard", "Mouse" and "Screen" with the VGA-to-HDMI wire to your Raspberry PI.

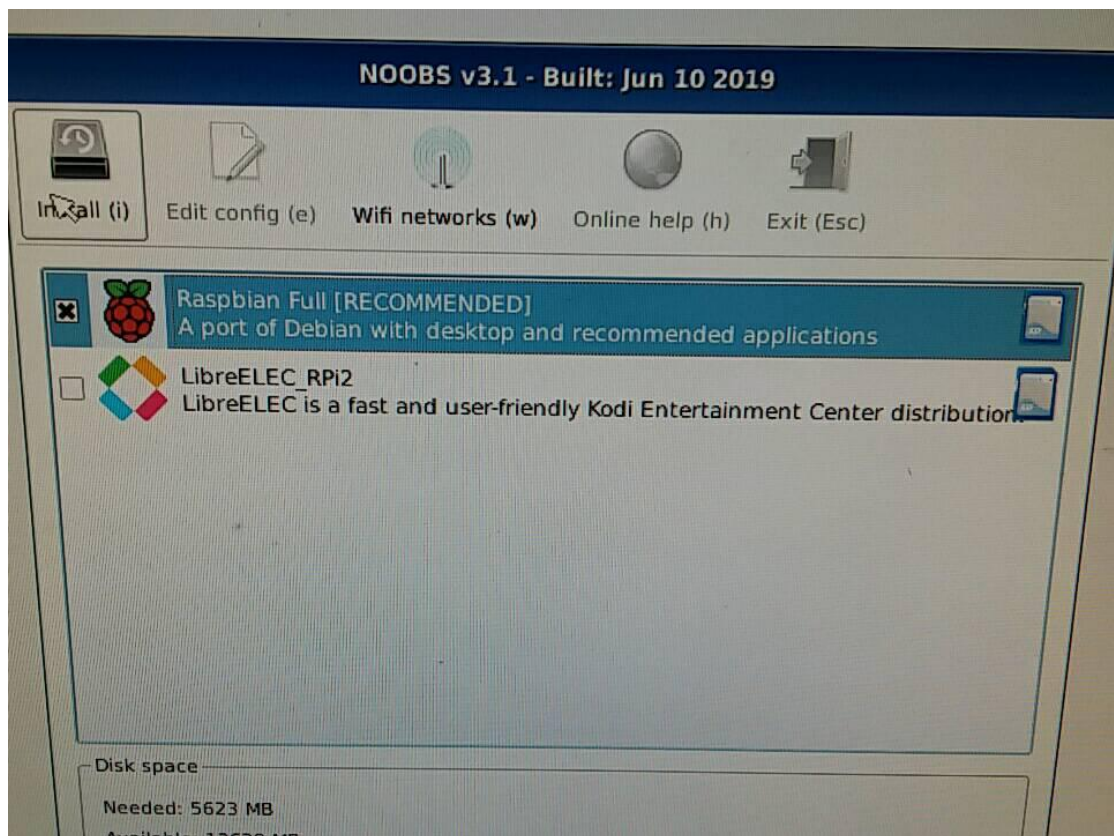


Link the Power wire and the Raspberry PI will boot up.

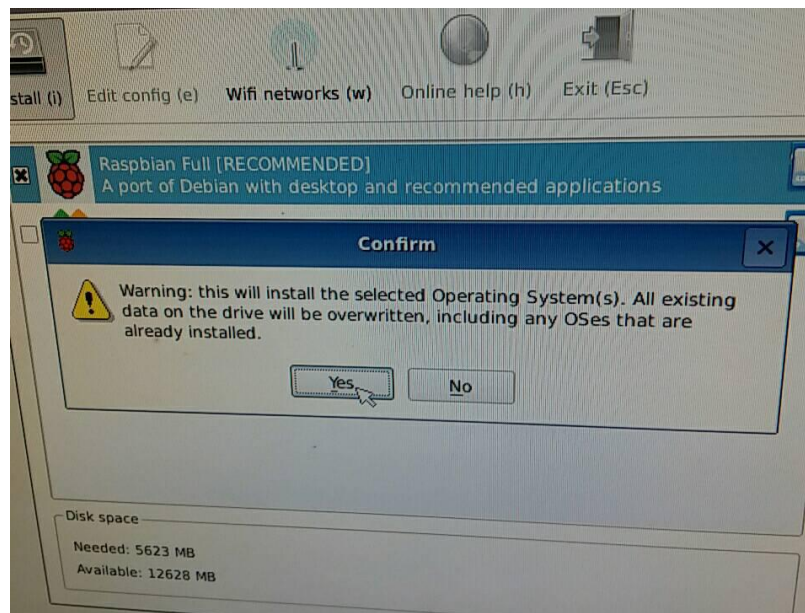




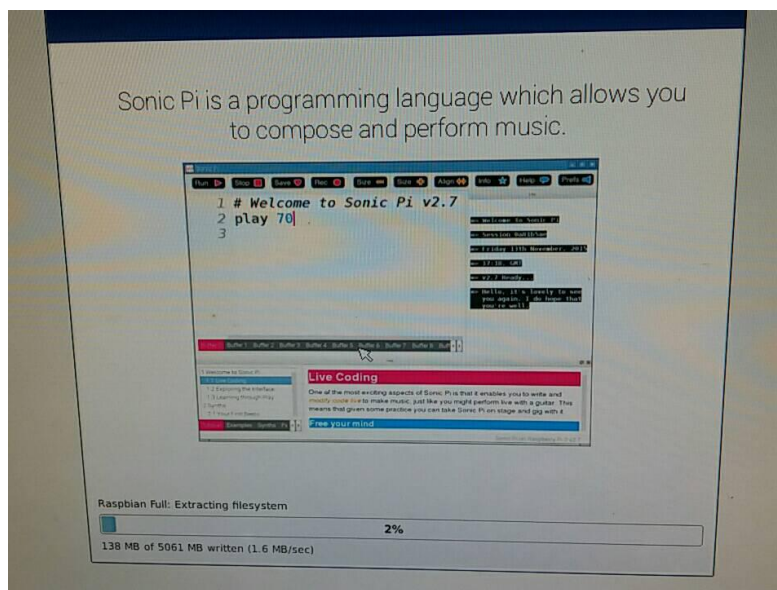
Choose the first and click the install



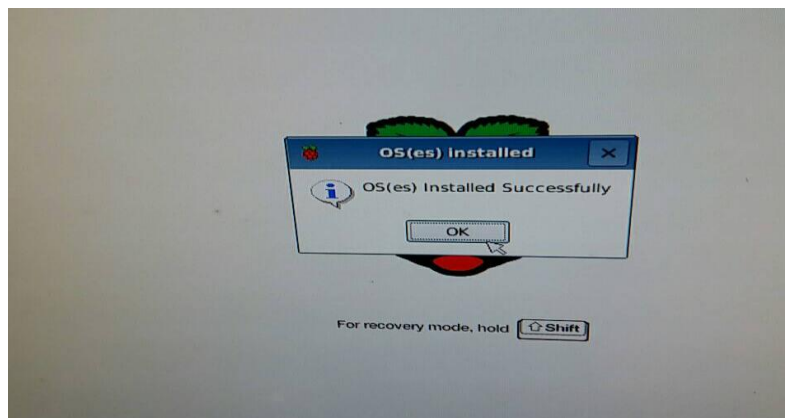
Click "Yes".



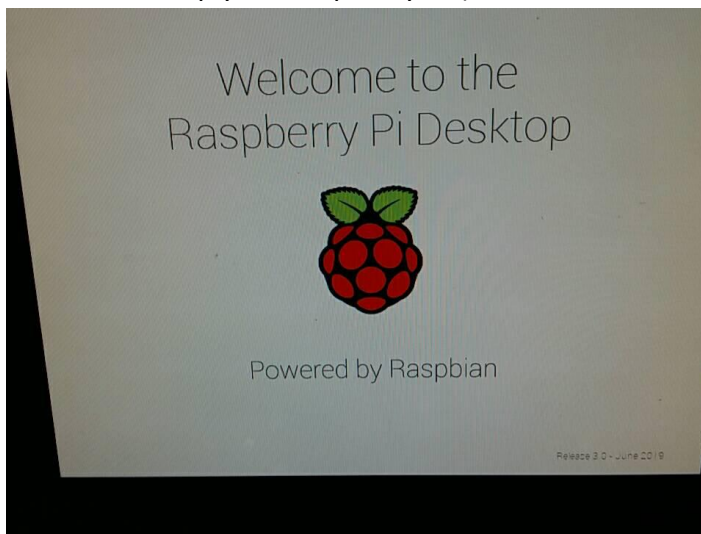
And then, wait a long long time.(Go to buy your dinner...)



After success, click the OK

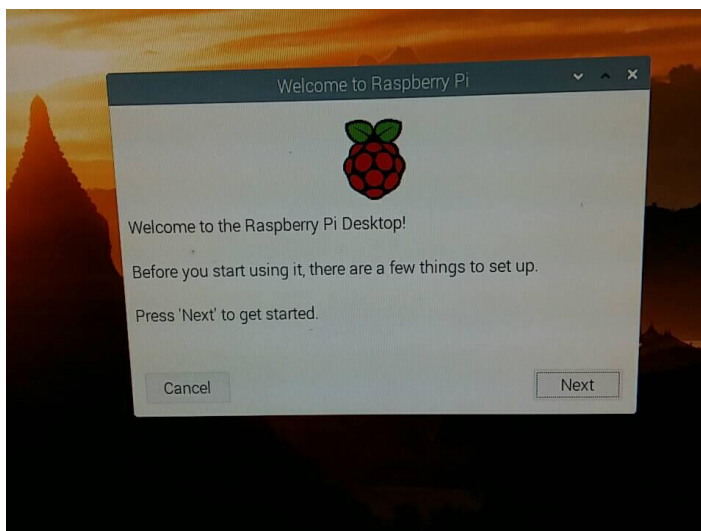


You will boot up your Raspberry PI (And then, let me have my dinner...)



4. Set your Raspberry PI:

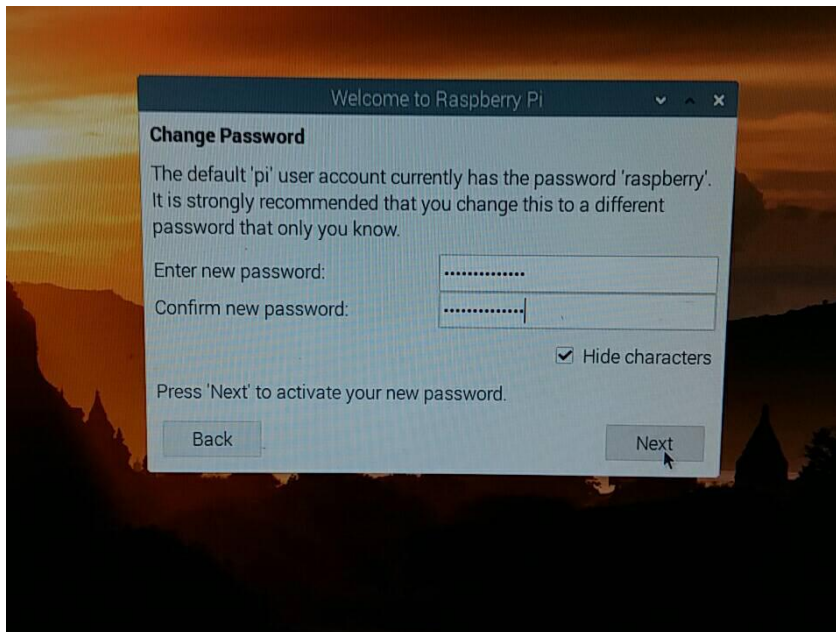
Click the next



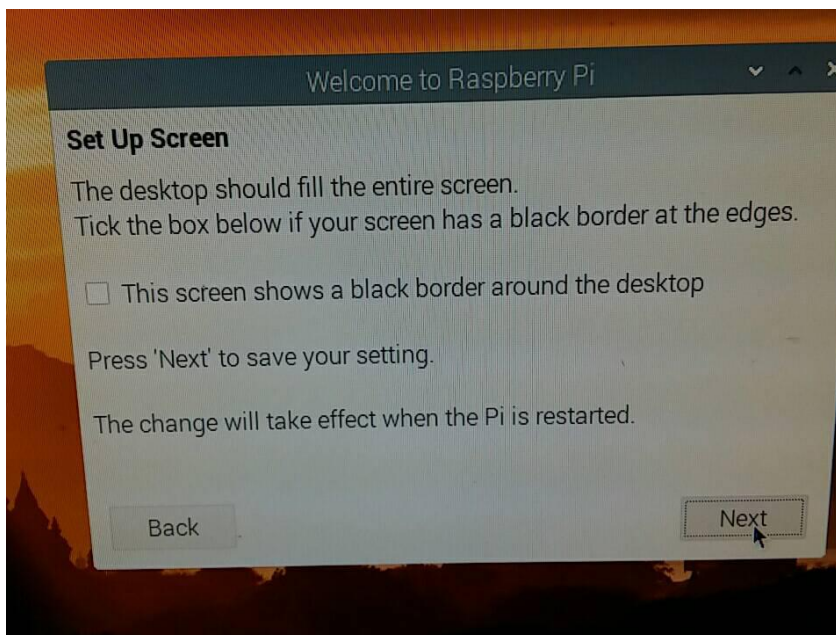
The settings like this. Click Next.



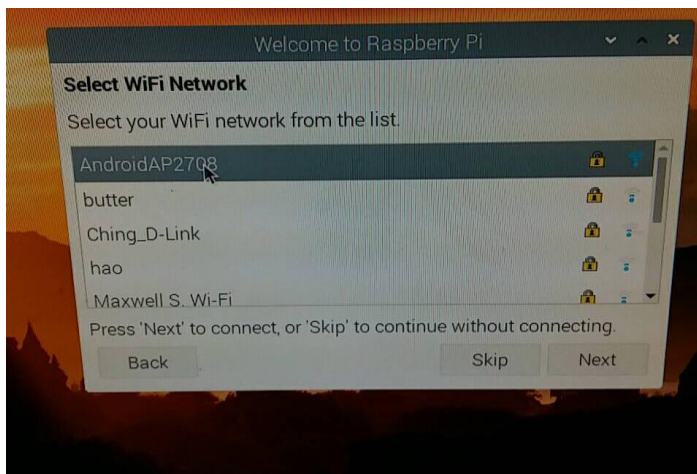
Change the default password to your password



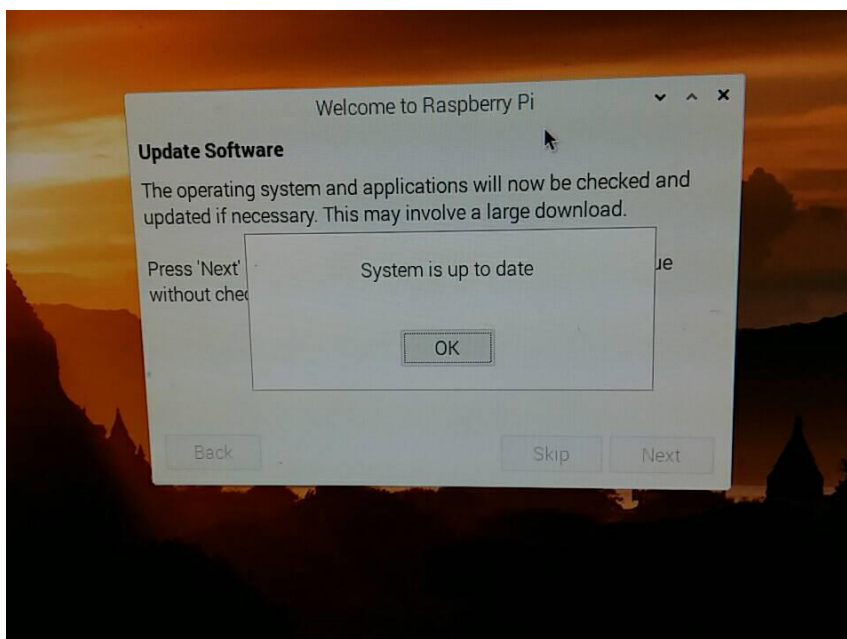
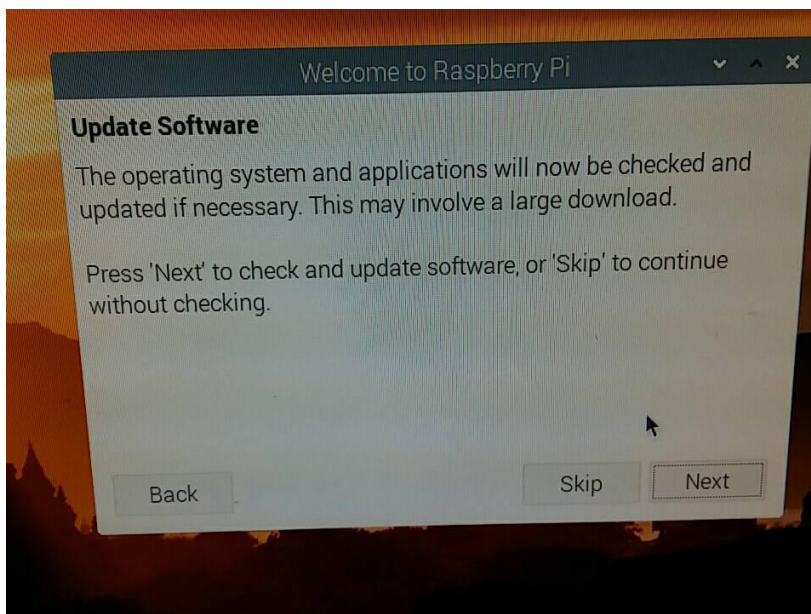
Click next to save your settings



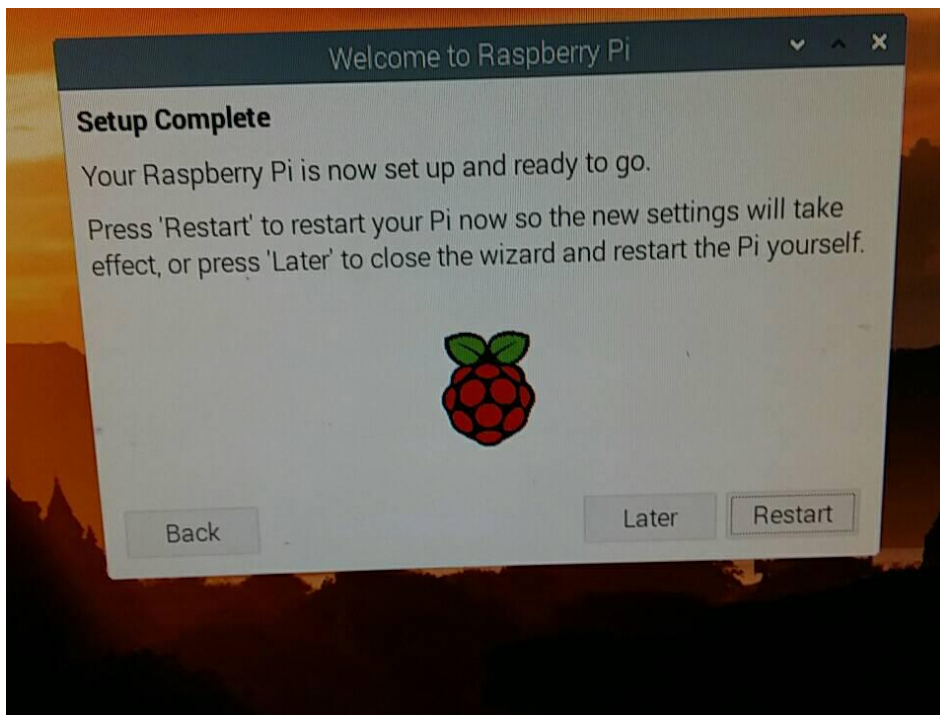
If you already have the WI-FI, you can set the WI-FI first (Or use Ethernet).



And then update your software (Another long long time...)



Now, Congratulation! You have setup the Raspberry PI completely! Restart it.



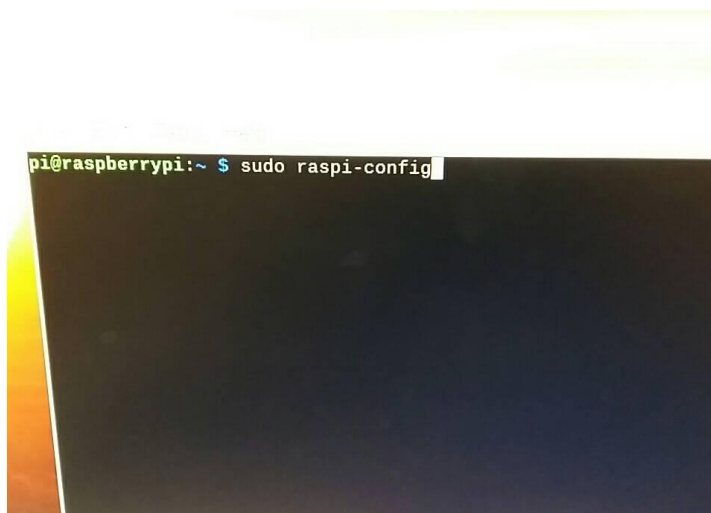
Bonus:

1. Setting the hostname

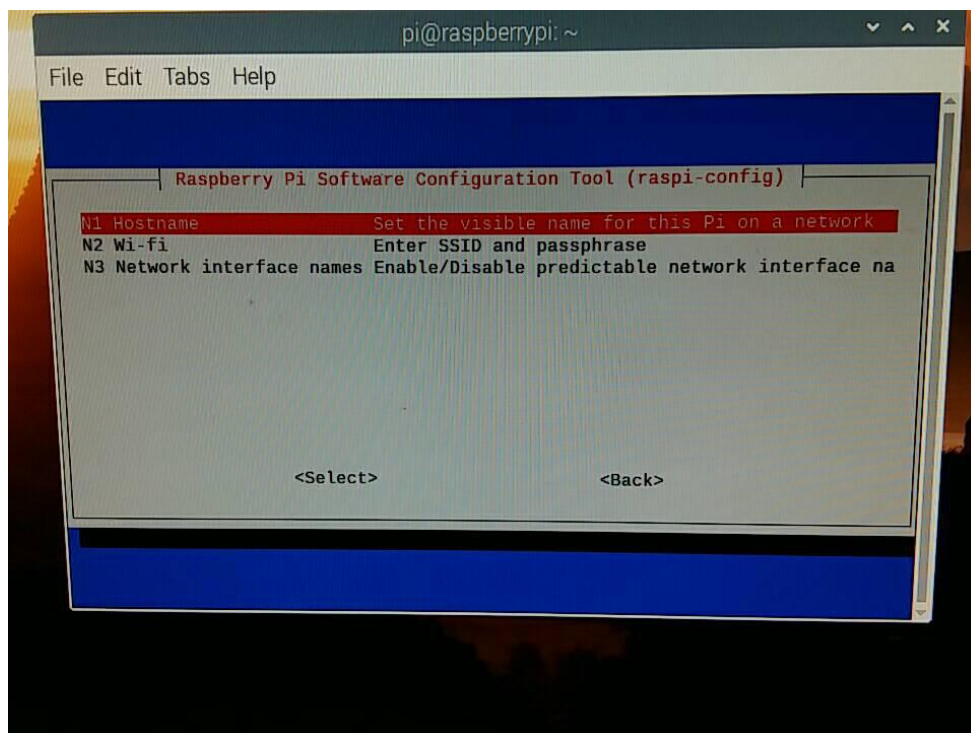
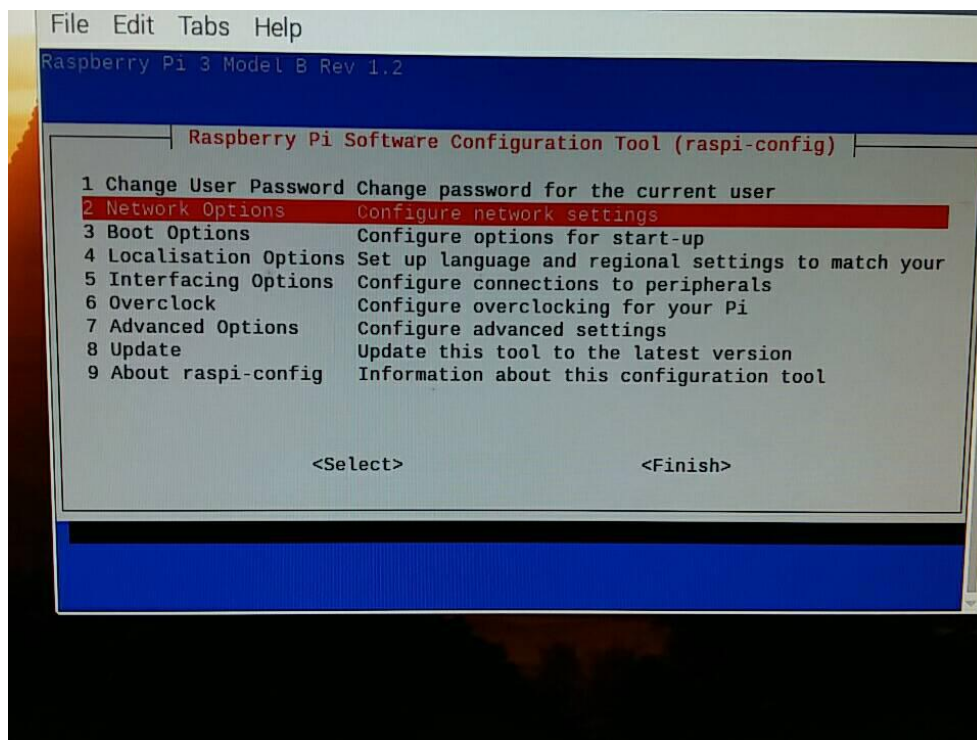
You can see the document at

<https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

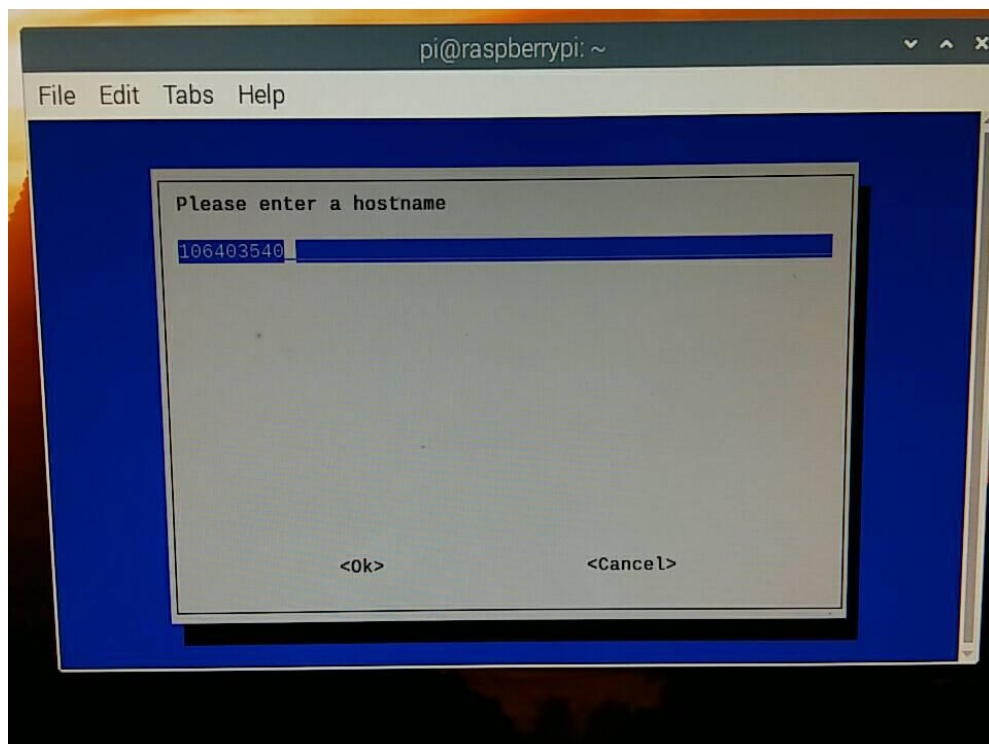
Open your terminal and type like this:



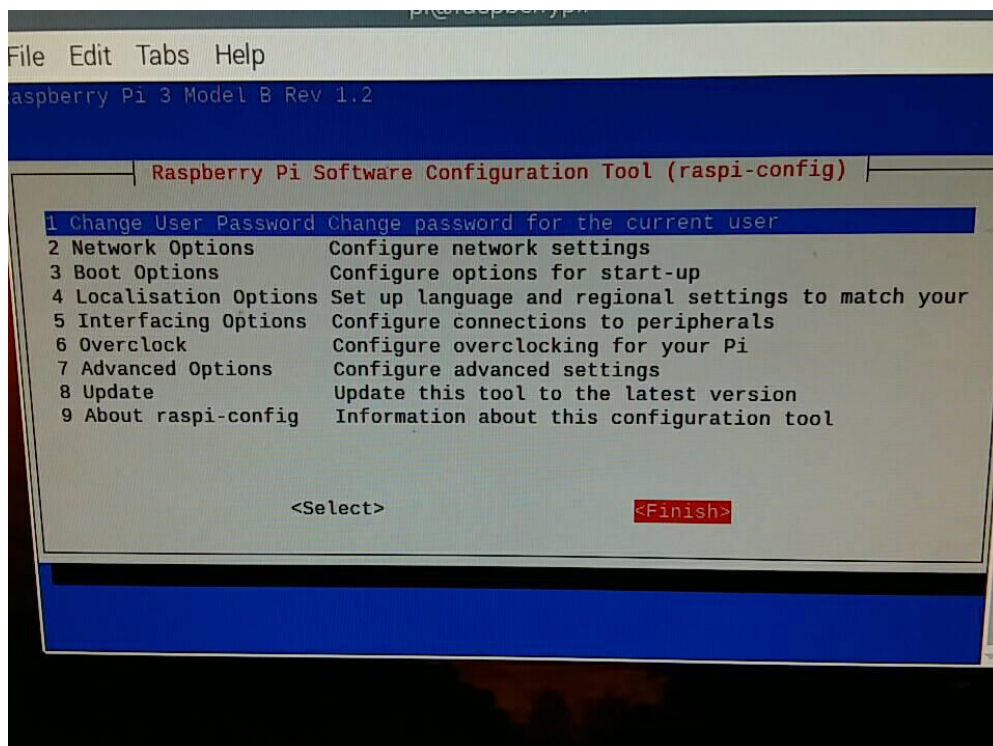
Choose the "2. Network Options"



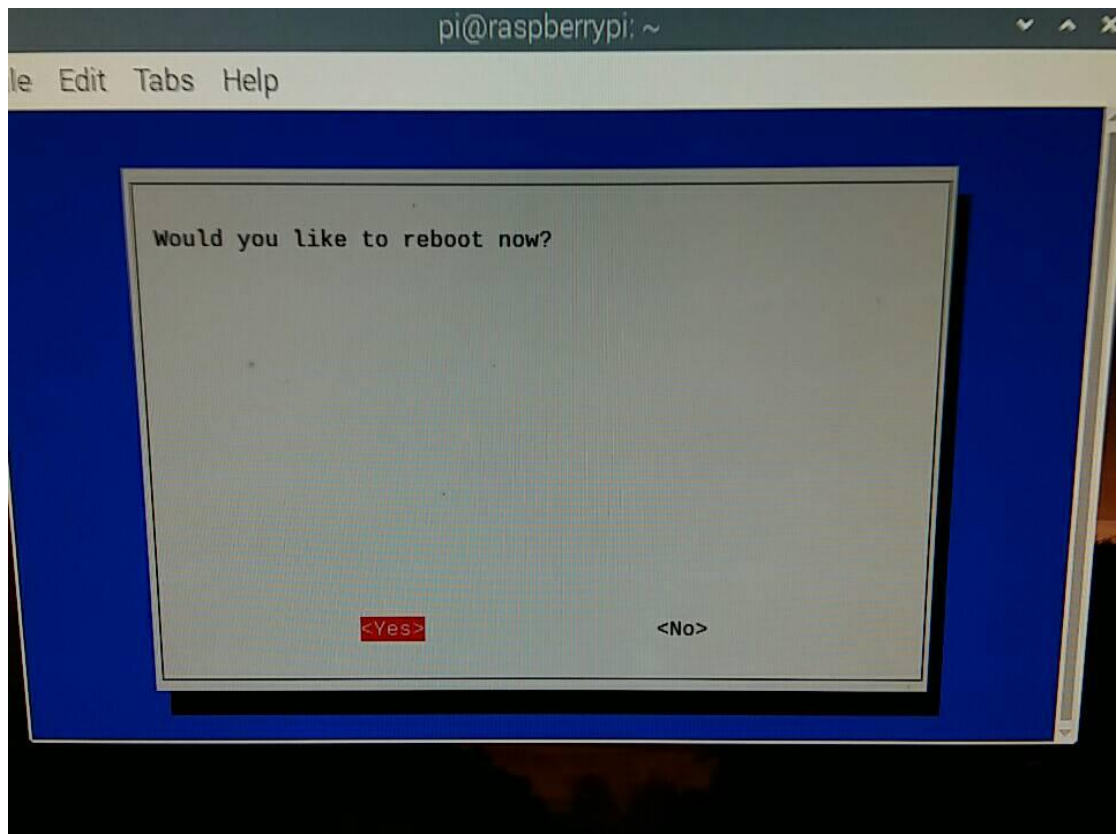
Type your hostname



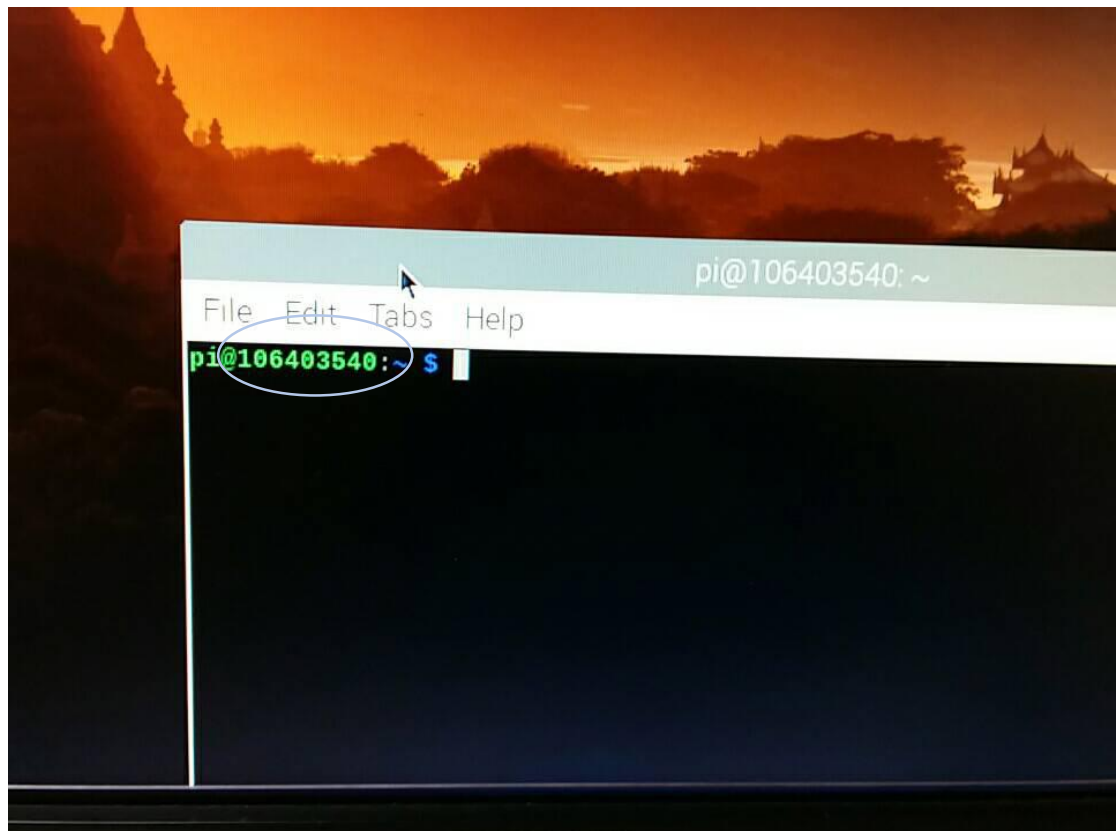
Finish it to change the configuration



Reboot it



Success!

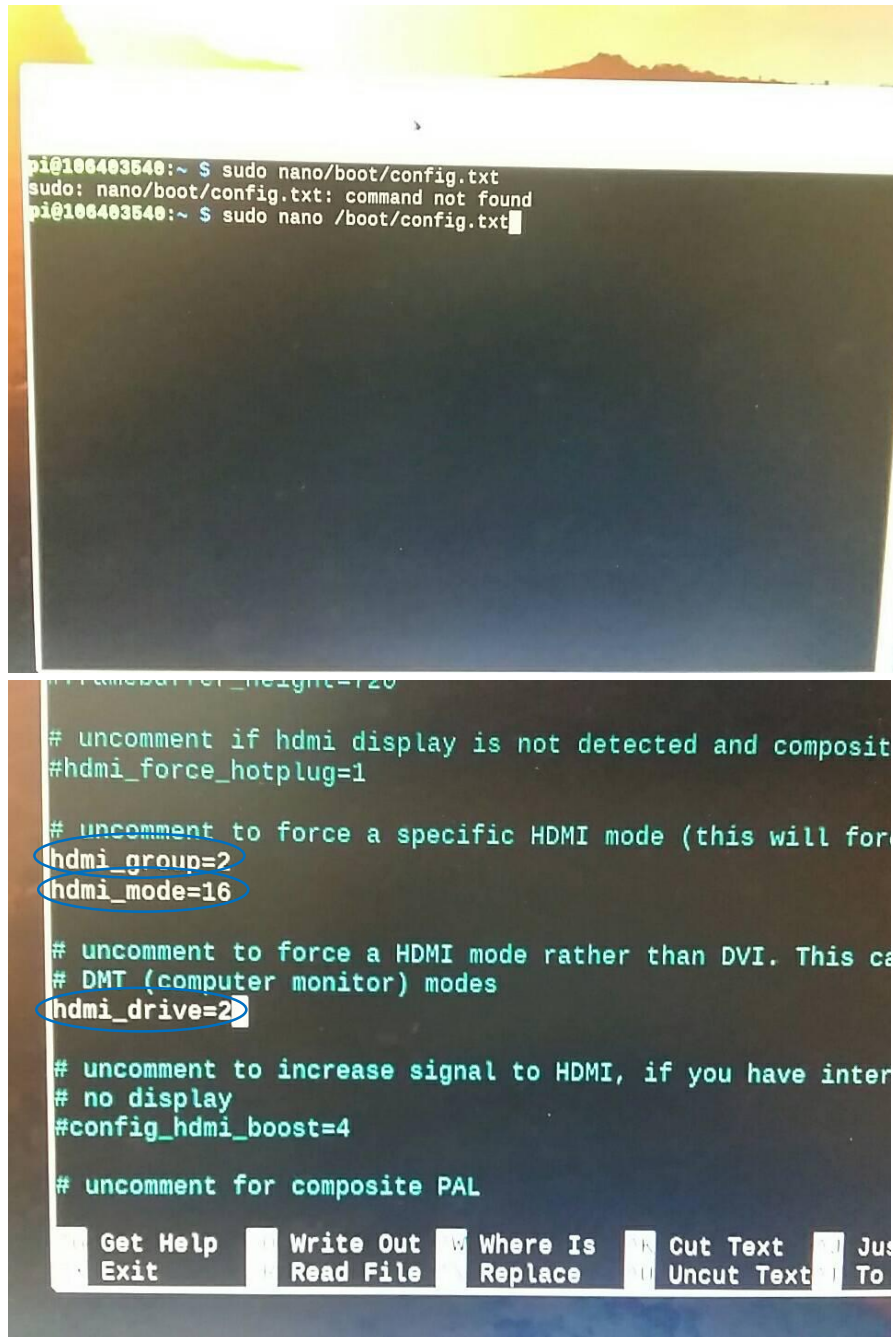


2. Editing the config.txt for video output

You can see the document

<https://www.raspberrypi.org/documentation/configuration/config-txt/>

Type the sudo nano /boot/config.txt



```
pi@106403540:~ $ sudo nano /boot/config.txt
sudo: nano /boot/config.txt: command not found
pi@106403540:~ $ sudo nano /boot/config.txt

# uncomment if hdmi display is not detected and composite
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force HDMI)
hdmi_group=2
hdmi_mode=16

# uncomment to force a HDMI mode rather than DVI. This can be useful to
# DMT (computer monitor) modes
hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference
# no display
#config_hdmi_boost=4

# uncomment for composite PAL

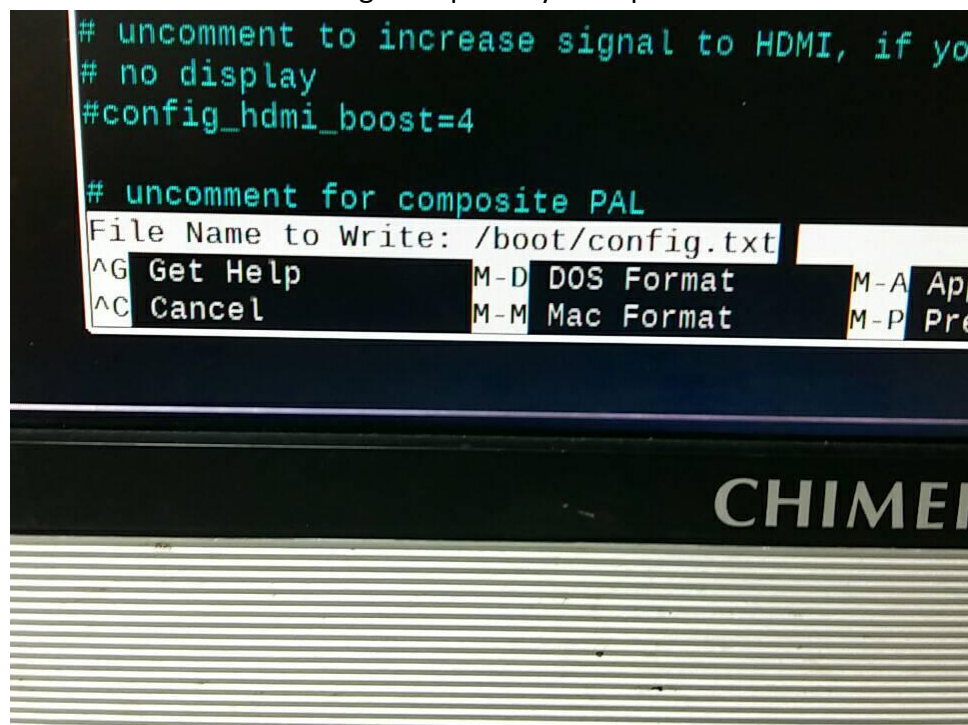
Get Help  Write Out  Where Is  Cut Text  Jus
Exit      Read File  Replace   Uncut Text To
```

Uncomment the hdmi_group, hdmi_mode and hdmi_drive. Then set the parameter to what you need or you can follow my settings.

The else settings you can see in the document

<https://www.raspberrypi.org/documentation/configuration/config-txt/video.md>

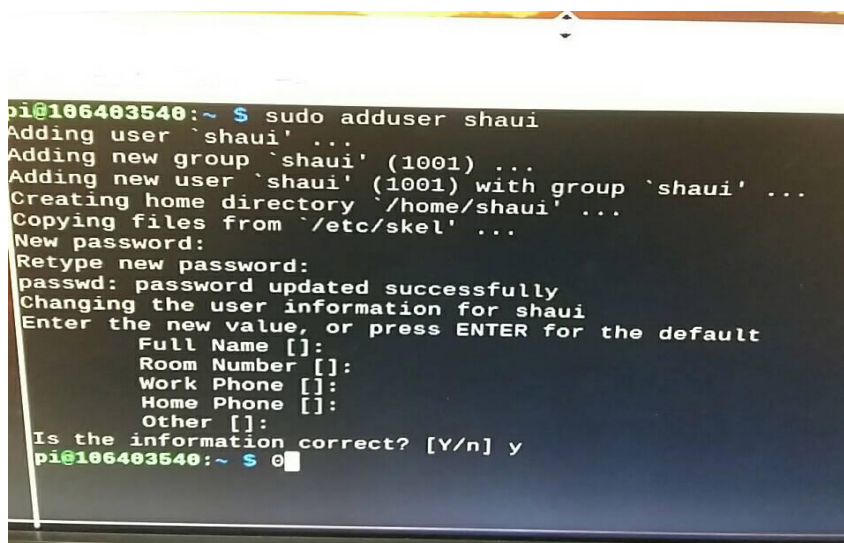
“Ctrl + x” to save the settings and press “y” and press “enter”



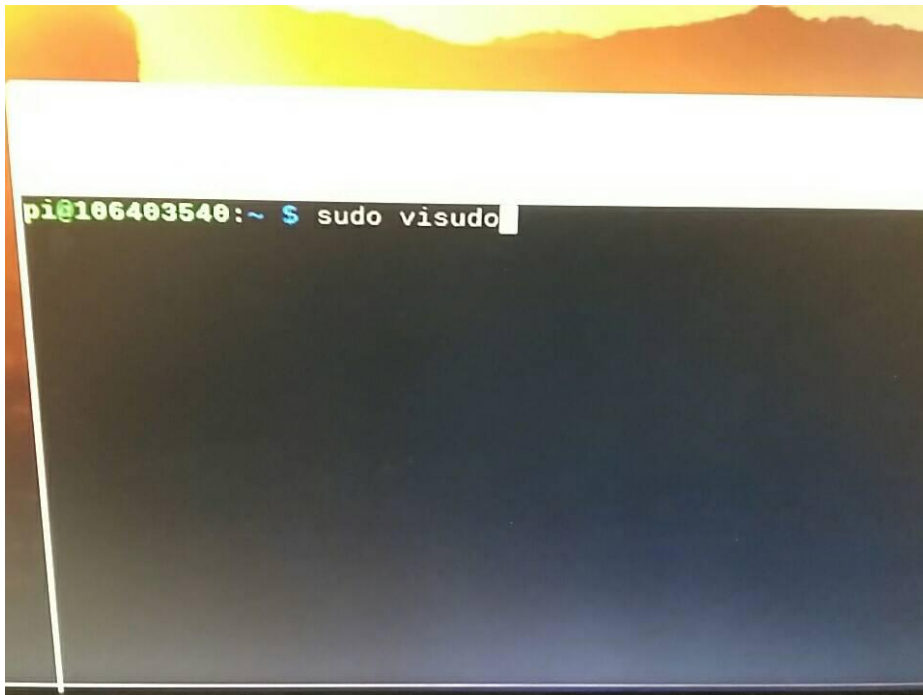
Part 2 :

In this part, you will create a new user in your Raspberry Pi and give it the root privilege. You can see the document: <https://raspi.tv/2012/how-to-create-a-new-user-on-raspberry-pi>

1. Type “sudo adduser yourname” yourname is the name you want to name. Just need to type the password, other configurations can be press “enter” to pass.

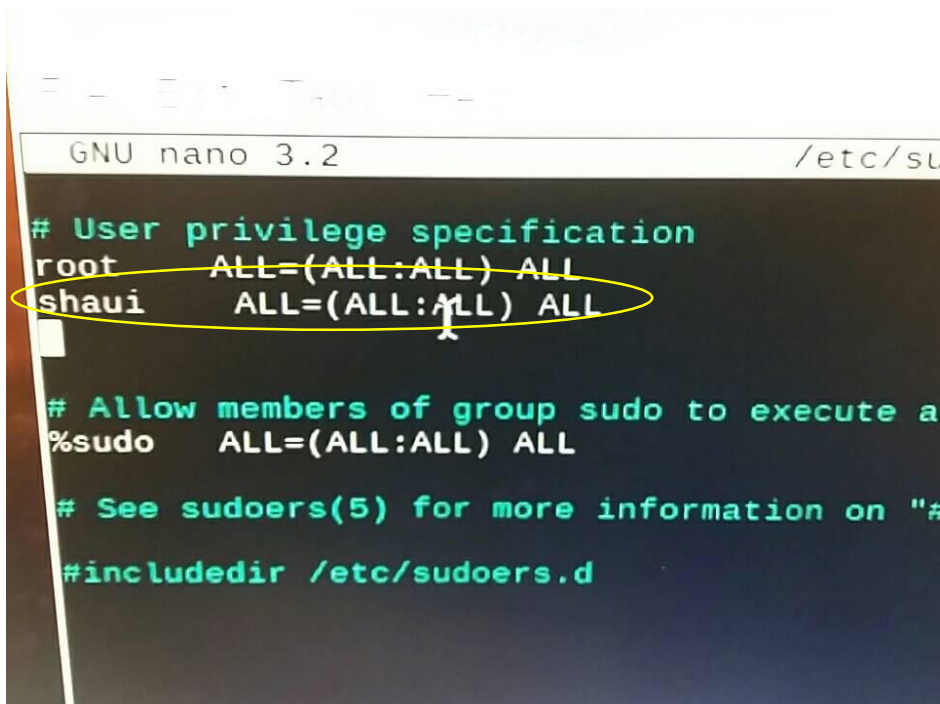


2. Give the root privilege to your account
Type the "sudo visudo" to set the privilege



```
pi@106403540:~ $ sudo visudo
```

Find the User privilege and add your account to have the root privilege



```
GNU nano 3.2 /etc/sudoers

# User privilege specification
root    ALL=(ALL:ALL) ALL
shau1    ALL=(ALL:ALL) ALL

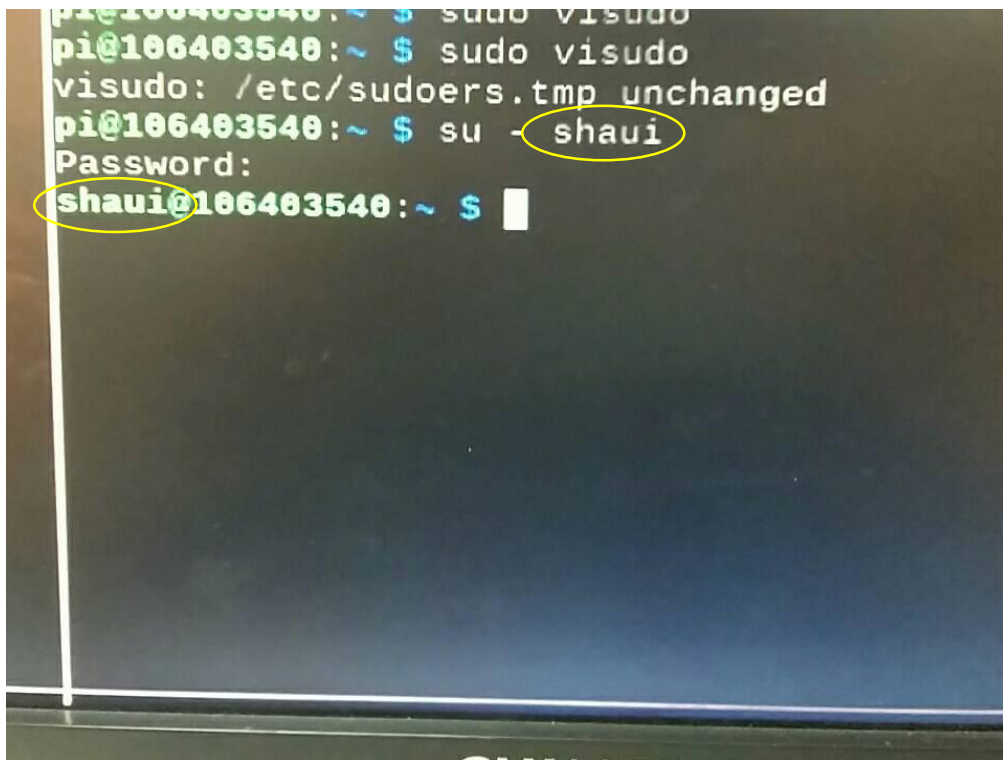
# Allow members of group sudo to execute a
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#
#includedir /etc/sudoers.d
```

"Ctrl + x" to save, then press "y" to check, finally press "enter" to finish.

3. Change the branch to your account

Type “su – yourname” and then type your password to login

A photograph of a terminal window on a Raspberry Pi. The terminal shows the following commands and output:
1. `pi@106403540:~$ sudo visudo`
2. `pi@106403540:~$ sudo visudo`
3. `visudo: /etc/sudoers.tmp unchanged`
4. `pi@106403540:~$ su - shaui` (The username 'shaui' is circled in yellow)
5. `Password:`
6. `shaui@106403540:~$` (The prompt 'shaui' is circled in yellow)

Part 3:

In this part, you will install the “Conda” for managing the python package. And then, you will create a “virtual environment” with conda and install the package you need in the virtual environment. Finally you will make the kernel in virtual environment available in Jupyter notebook, and then run the python code in Jupyter.

1. Installing Conda:

You can choose either Miniconda or Berryconda, although Berryconda is recommended for Raspbian OS. For more details on Berryconda:

<https://github.com/jjhelmus/berryconda>

Important:

Berryconda should be installed in a user account, not by root or using sudo. So you can use something that in Part 2.

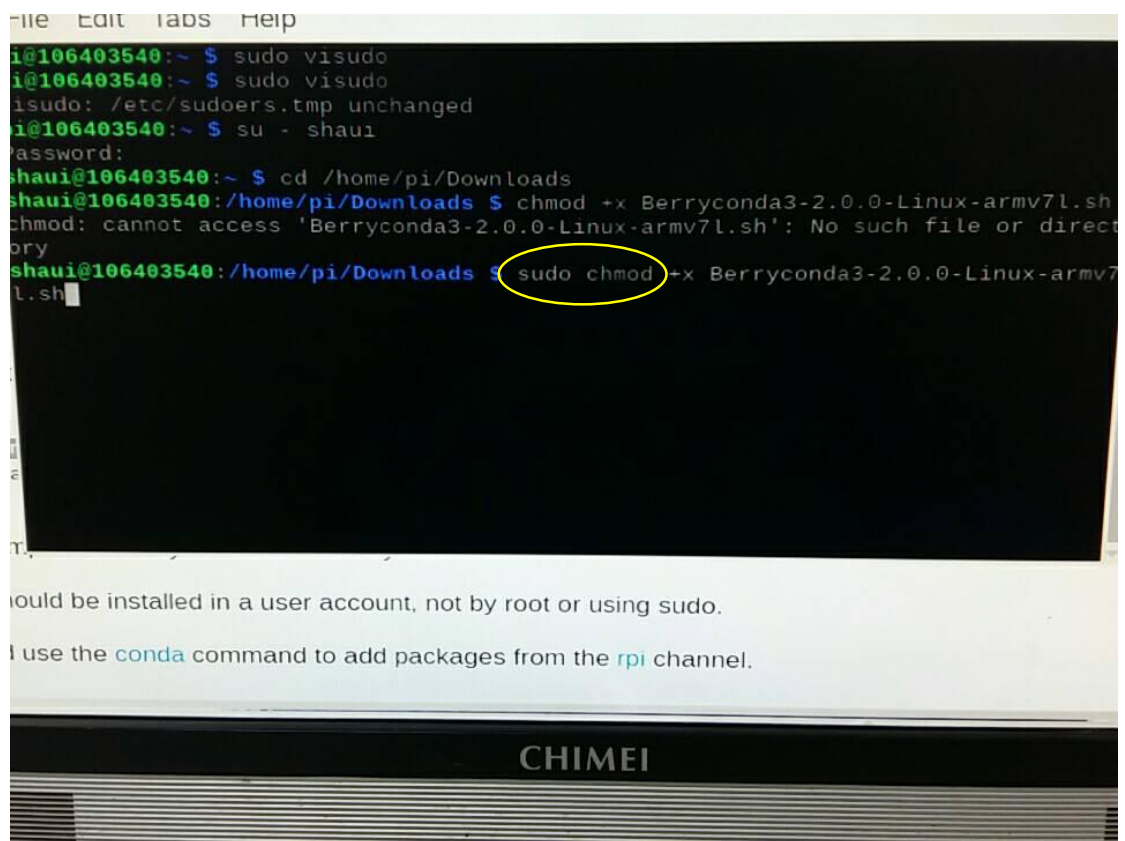
Download this for python3

armv7l installers (Raspberry Pi 2 or 3)

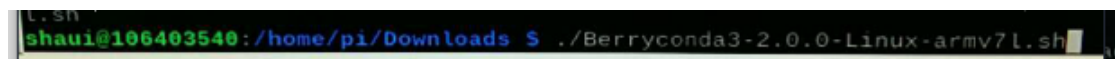
- [Berryconda3-2.0.0-Linux-armv7l.sh](#)
- [Berryconda2-2.0.0-Linux-armv7l.sh](#)

Change the path to your download directory, and then type the command as the picture, because “chmod” command need the root, so you need to add “sudo”.

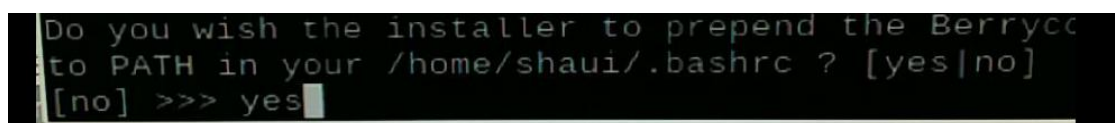
```
chmod +x Berryconda3-2.0.0-Linux-armv7l.sh
./Berryconda3-2.0.0-Linux-armv7l.sh
```



Then type “./Berryconda3-2.0.0-Linux-armv7l.sh”



All settings in the command just press “enter” and “yes”. Like this:



Please reopen your terminal and change to your account. Type “conda -V” to see the version, “conda list” to see the package.

```
shau1@106403540:~$ su - shau1
password:
shau1@106403540:~$ conda -V
conda 4.3.22
shau1@106403540:~$ conda list
# packages in environment at /home/shau1/berryconda3:
#
asn1crypto                0.22.0                py36
ca-certificates            2017.4.17             py36
certifi                    2017.4.17             py36
cffi                       1.10.0                py36
chardet                    3.0.4                 py36
conda                      4.3.22                py36
conda-env                  2.6.0
cryptography               1.9                   py36
idna                       2.5                   py36
libffi                     3.2.1
ncurses                    5.9
openssl                    1.0.2l
packaging                  16.8                  py36
pip                        9.0.1                 py36
pycosat                    0.6.1                 py36
pycparser                   2.17                  py36
```

2. Installing a package “Jupyter” Notebook:

Type “conda install -c rpi jupyter”, “-c” is to select the channel for download “rpi” is the channel name “jupyter” is the package name. All of the suggestion in the install time just press “yes”

```
shau1@106403540:~$ conda install -c rpi jupyter
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment /home/shau1/berryconda3:

The following NEW packages will be INSTALLED:

backcall: 0.1.0-py_0
bleach: 2.1.4-py_1
decorator: 4.3.0-py_0
entrypoints: 0.2.3-py36_2
html5lib: 1.0.1-py36h4e0ed57_0
```

```
conda: 4.
conda-env: 2.
pycosat: 0.

Proceed ([y]/n)? y

conda-env-2.6. 100% |#####
libsodium-1.0. 100% |#####
zeromq-4.2.5-h 100% |#####
backcall-0.1.0 100% |#####
```

3. Create a virtual environment using Conda:

You can see the document:

<https://uoa-ereseach.github.io/ereseach-cookbook/recipe/2014/11/20/conda/>
<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

1. To create an environment:

```
conda create --name myenv
```

```
- Validating: OK

shau1@106403540:~ $ conda create --name pyenv_1
Solving environment: done

## Package Plan ##

  environment location: /home/shau1/berryconda3/envs/pyenv_1

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use:
# > source activate pyenv_1
#
# To deactivate an active environment, use:
# > source deactivate
#

shau1@106403540:~ $
```

4. Install package in your virtual environment

First, activate your virtual environment by “source activate yourenvname”

```
shau1@106403540:~ $ source activate pyenv_1
(pyenv_1) shau1@106403540:~ $
```

And then, like the step-2(install the package)

```
shau1@106403540:~ $ source activate pyenv_1
(pyenv_1) shau1@106403540:~ $ conda install -c rpi numpy
Solving environment: done

## Package Plan ##

environment location: /home/shau1/berryconda3/envs/pyenv_1

added / updated specs:
- numpy

The following packages will be downloaded:
```

package	build	
openssl-1.0.2r	hdff2a78_0	2.2
setuptools-40.2.0	py36_0	554
pip-18.0	py36_1	1.8
libgfortran-3.0.0	0	206
ca-certificates-2018.8.24	0	135
blas-1.1	openblas	2
ncurses-6.1	h4f752ac_1	1.2
readline-7.0	hcb560eb_1	376

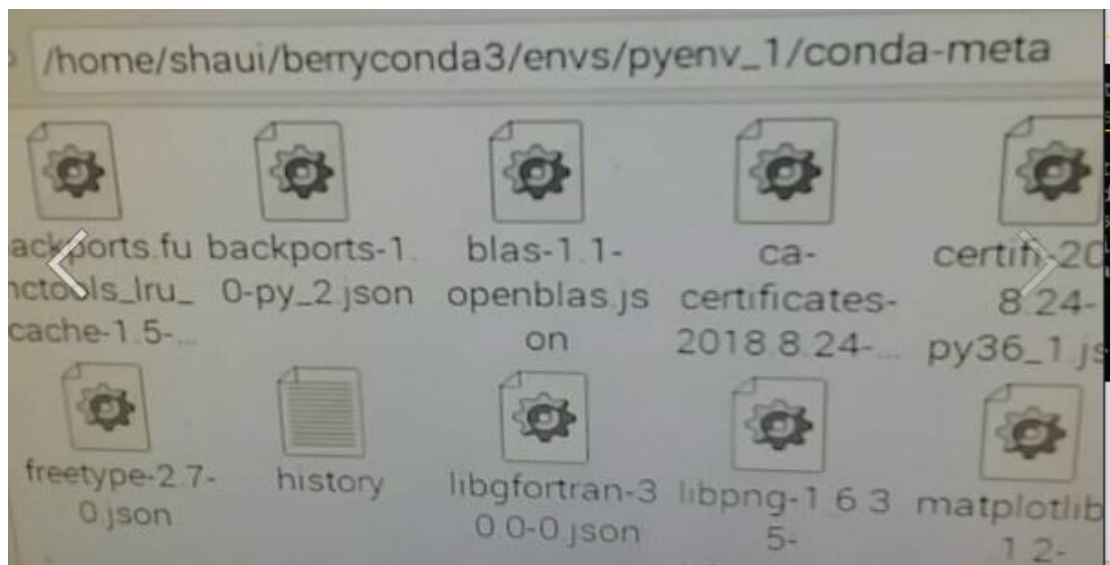
Repeat this step to install these package:

- numpy
- scipy
- matplotlib
- pandas
- scikit-learn

The “rpi.gpio” package cannot use the conda to install, so you need to use the “pip” in your virtual environment to install.

```
1) shau1@106403540:~ $ pip install rpi.gpio
Looking in indexes: https://pypi.org/simple, https://w
Installing rpi.gpio
loading https://files.pythonhosted.org/packages/
fe303266ble3b85e2952238f0da43fed4e/RPi.GPIO-0.7.
Installing wheels for collected packages: rpi.gpio
Installing setup.py bdist_wheel for rpi.gpio ... done
Stored in directory: /home/shau1/.cache/pip/wheels/e
97418affe12e05d42213f3e0ca
Successfully built rpi.gpio
Installing collected packages: rpi.gpio
Successfully installed rpi.gpio-0.7.0
```

You can see all the package you install in the path



5. Make the kernel in virtual environment available in Jupyter notebook. You can see the document:

https://ipython.readthedocs.io/en/stable/install/kernel_install.html

Install the "ipykernel" package in your virtual environment

```
shaui@106403540: ~  
(pyenv_1) shaui@106403540:~ $ conda install -c rpi ipykernel  
Solving environment: done  
  
## Package Plan ##  
  
environment location: /home/shaui/berryconda3/envs/pyenv_1  
  
added / updated specs:  
- ipykernel  
  
The following NEW packages will be INSTALLED:  
  
backcall: 0.1.0-py_0  
decorator: 4.3.0-py_0  
ipykernel: 4.9.0-py36_0  
ipython: 6.5.0-py36_0  
ipython_genutils: 0.2.0-py_1  
jedi: 0.12.1-py36_0  
jupyter_client: 5.2.3-py_1  
jupyter_core: 4.4.0-pyh2c3be9_0  
libsodium: 1.0.16-0  
parso: 0.3.1-py_0  
pexpect: 4.6.0-py36_0
```


Type like the picture. “- name” is your virtual environment’s name, “- display-name” is your kernel’s name

```
Executing transaction: done
(pyenv_1) shau1@106403540:~ $ python -m ipykernel install --user --name pyenv_1
--display-name "Python(MyPyEnv_1)"
Installed kernelspec pyenv_1 in /home/shau1/.local/share/jupyter/kernels/pyenv_1
(pyenv_1) shau1@106403540:~ $
```

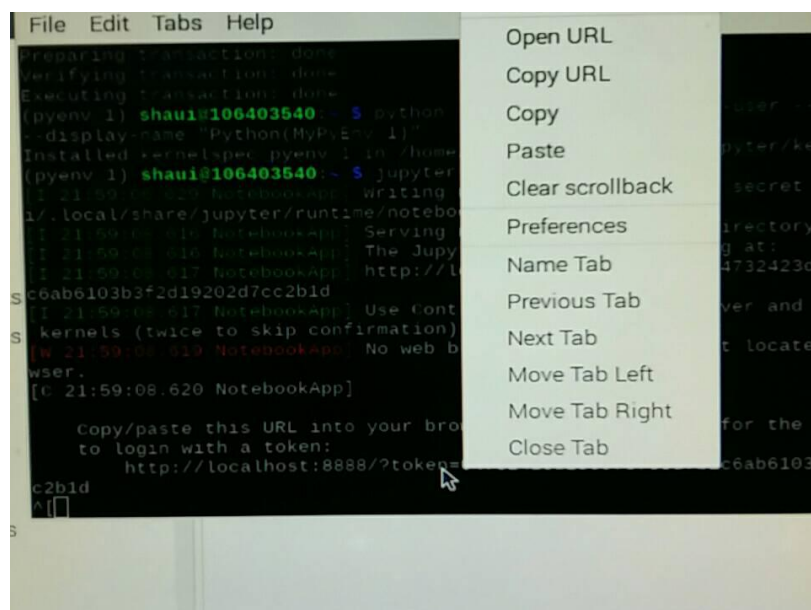
6. Run a Jupyter Notebook with the above installed packages, and check the kernel if it is install successfully.

Create a Jupyter notebook with the command “jupyter notebook”

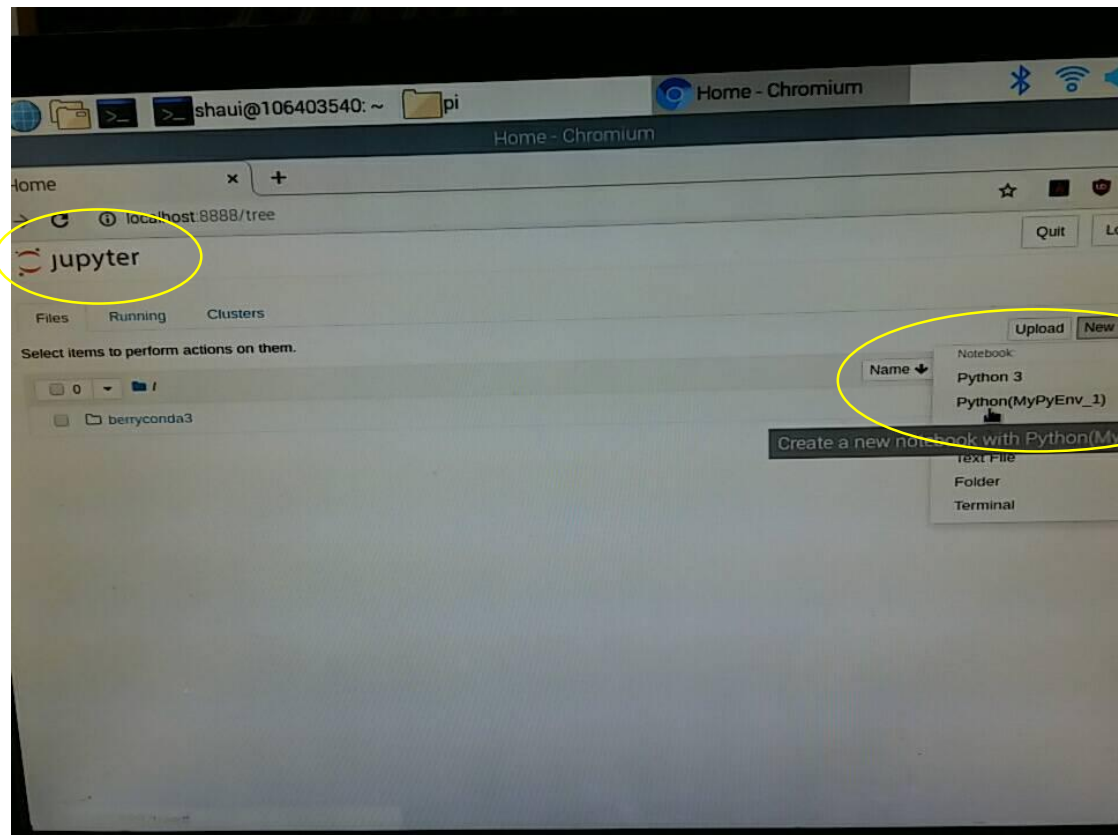
```
nv_1) shau1@106403540:~ $ python -m ipykernel in
splay-name "Python(MyPyEnv_1)"
alled kernelspec pyenv_1 in /home/shau1/.local/s
nv_1) shau1@106403540:~ $ jupyter notebook
1:59:06.629 NotebookApp] Writing notebook server
ocal/share/jupyter/runtime/notebook_cookie_secre
1:59:08.616 NotebookApp] Serving notebooks from
1:59:08.616 NotebookApp] The Jupyter Notebook is
1:59:08.617 NotebookApp] http://localhost:8888/?
6103b3f2d19202d7cc2b1d
1:59:08.617 NotebookApp] Use Control-C to stop th
nals (twice to skip confirmation).
1:59:08.619 NotebookApp] No web browser found: co
.
1:59:08.620 NotebookApp]

Copy/paste this URL into your browser when you co
to login with a token:
http://localhost:8888/?token=04732423d15e97cf
ld
```

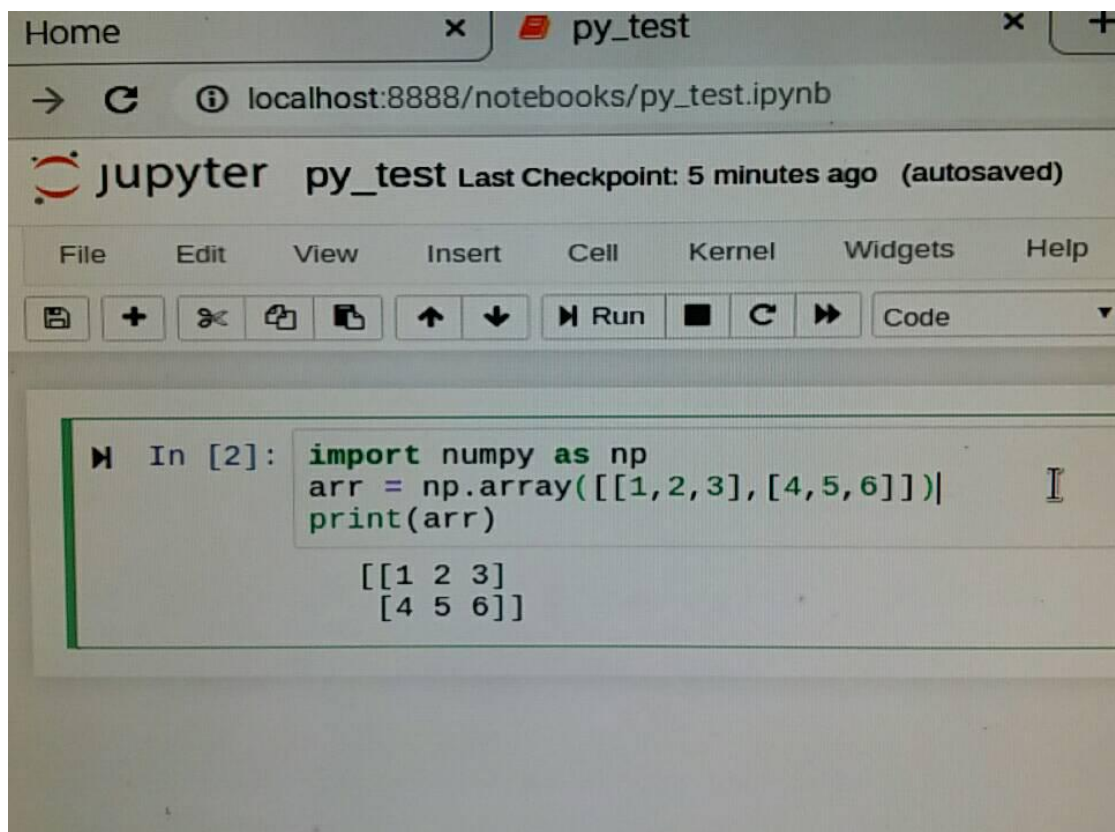
Open the URL at the bottom



And then, you will see the jupyter with browser and you can see the kernel we previously create at the right-side.



Try to import a package we installed to check it was success.



Bonus:

1. Fork a Jupyter Note from Kaggle and run it on your RPi

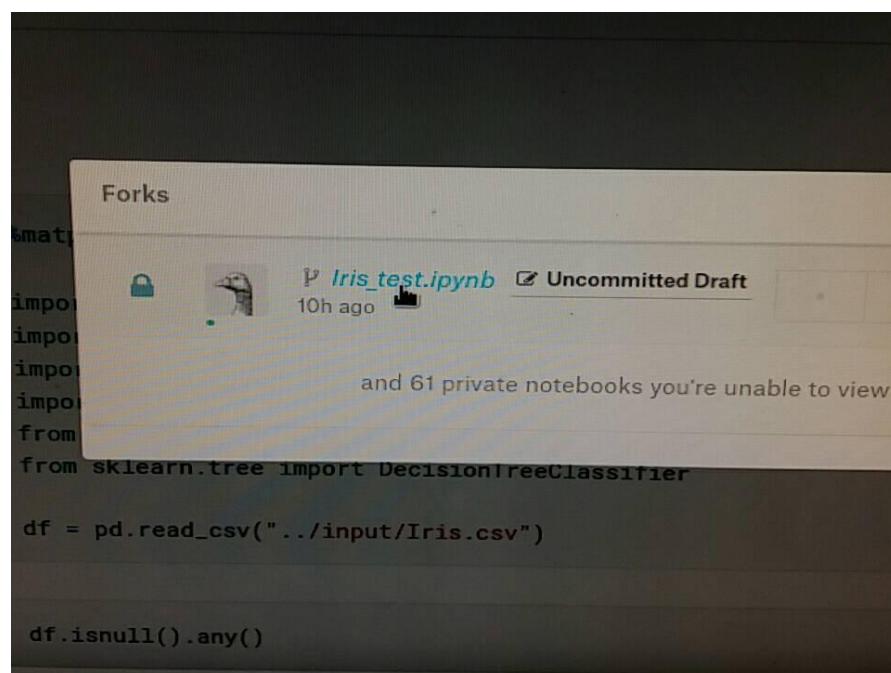
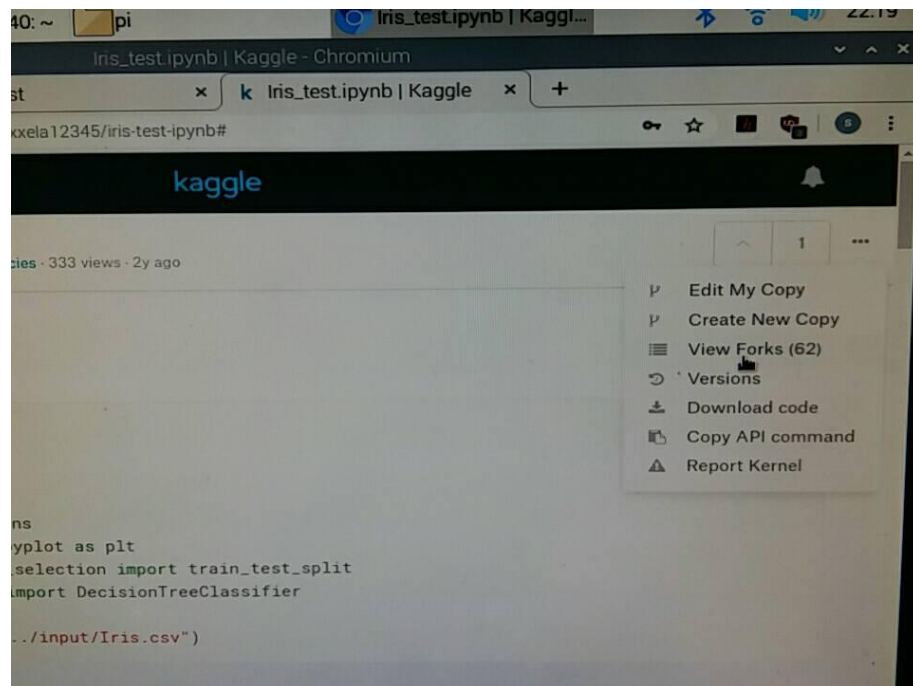
Important:

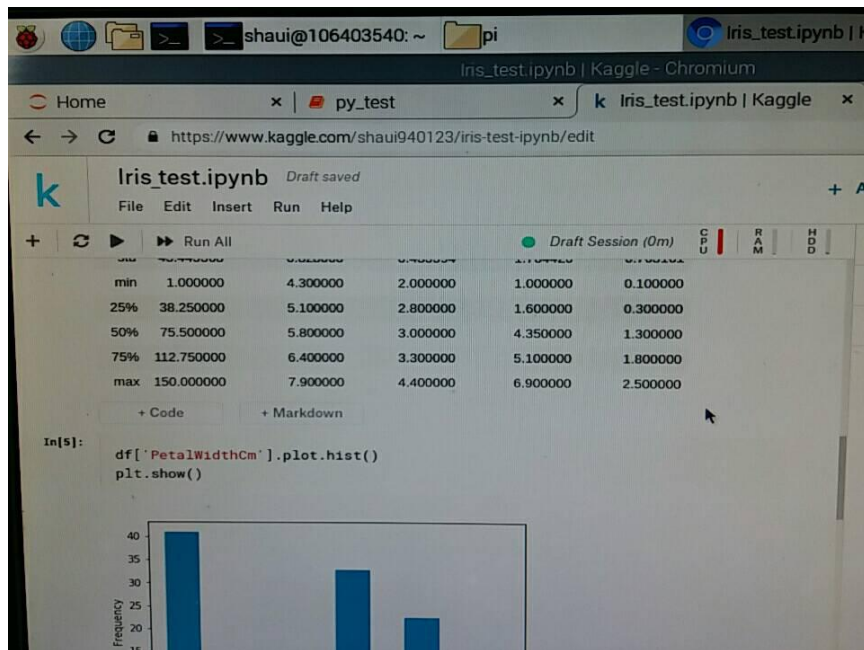
Note that you'd need to sign up for a Kaggle account to do this step

You can try to fork the notebook from here:

<https://www.kaggle.com/alexxela12345/iris-test-ipynb>

Fellow my step in the picture:





Congratulation! Now, you can code your python code with the virtual environment in your Raspberry PI.

2. Backup you Raspberry PI with zip:

You can see the document:

<https://www.raspberrypi.org/magpi/back-up-raspberry-pi/>

```
cd /home/  
sudo tar czf pi_home.tar.gz pi
```

Copy the `pi_home.tar.gz` file to a USB flash drive for safe keeping.

The tar command is a “Packing tool”, not will let the file become the “gzip file”, so We set the parameter “czf”:

“c” let our file become a “tar file” it’s mean that the file was packed.

“z” let out file become a “gzip file” it’s mean the tar file become a “gzip file”

“f” specify a file to output, just type your file name. For me, I know it’s the tar.gz file, so I give me a name “pi_home” + “tar.gz”.

Finally, the “pi” at the end is what file or directory you want to compress, in the example it’s “pi” directory.

For more detail Linux command, you can see the document:

<http://www.vixual.net/blog/archives/127>

Because most of our configuration is at the home, so backup it! And then Copy the file to your USB.

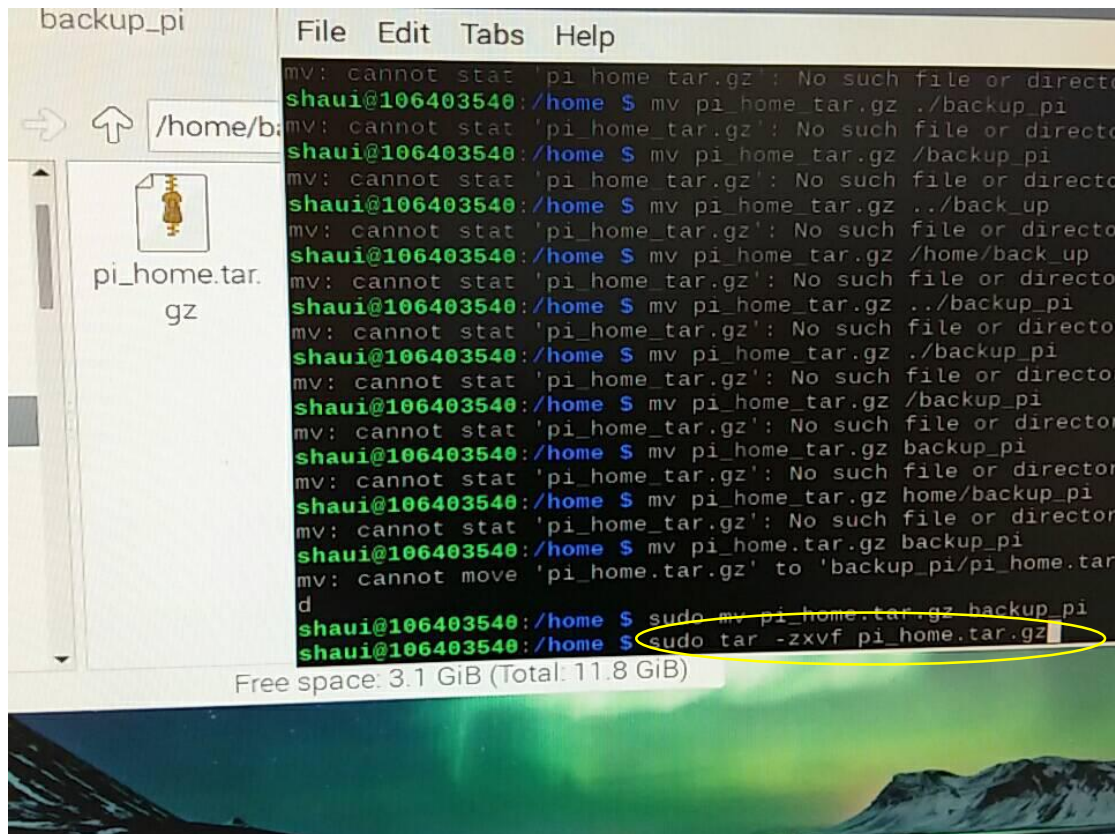


3. Restore your Raspberry PI with your backup zip file.

You can see the document:

<https://blog.gtwang.org/linux/tar-command-examples-in-linux-2/>

Move your gzip file to where you want your file be unzip, and then type the command "sudo tar -zxvf pi_home.tar.gz"



Then, you will see the directory you backup previously!

