Installation of Miniconda3,Berryconda3 and Jupyter for RPi

Conda

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages.The conda package and environment manager is included in all versions of Anaconda and Miniconda.

Anaconda

Anaconda is a full distribution of the central software in the PyData ecosystem, and includes Python itself along with the binaries for several hundred third-party open-source projects. Miniconda is essentially an installer for an empty conda environment, containing only Conda, its dependencies, and Python

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others.

Installation Procedure:

$wget http://repo.continuum.io/miniconda/Miniconda3-latest-Linux-armv7l.sh
$sudo md5sum Miniconda3-latest-Linux-armv7l.sh
$sudo /bin/bash Miniconda3-latest-Linux-armv7l.sh

Accept the license agreement with yes

When asked, change the install location: /home/pi/miniconda3

Do you wish the installer to prepend the Miniconda3 install location to PATH in your /root/.bashrc ? No

Now add the install path to the PATH variable:

sudo nano /home/pi/.bashrc

Go to the end of the file .bashrc and add the following line:

export PATH="/home/pi/miniconda3/bin:$PATH"

Save the file and exit.

To test if the installation was successful, open a new terminal and enter

$conda

If you see a list with commands you are ready to go

Berryconda3

Berryconda is a conda based Python distribution for the Raspberry Pi. With it, you can install and manage a scientific or Pydata stack on your Raspberry Pi using conda, a package and environment management system.

Installation Procedure:

Then download berryconda

https://github.com/jjhelmus/berryconda

The download file look like

Berryconda3-2.0.0-Linux-armv7l.sh

$sudo chmod -R 777 Berryconda3-2.0.0-Linux-armv7l.sh

$sudo ./Berryconda3-2.0.0-Linux-armv7l.sh

$conda config --add channels rpi

Only now I was able to install Python 3.5 or 3.6 without the need for compiling it myself:

$conda install python=3.5

$conda install python=3.6

This commands are opational

(Afterwards I was able to create environments with the added Python version, e.g. with Python 3.5:

$conda create --name py35 python=3.5

The new environment "py35" can now be activated:

$source activate py35)

we need to install Python Libraries

NumPy

Top-most:array,arange,reshape,shape

Pandas

Top-most:DataFrame,groupby,apply,loc,reset_index

Matplotlib

Top-most:plot,show,figure,subplots

Seaborn

Top-most:show,DataFrame,plots,subplots

Scikit-Learn

Top-most:fit,fit_transform,predict,array

Libaries installation:

conda install pandas
conda install numpy
conda install matplotlib
conda install scikit-learn
conda install seaborn

Jupyter-notebook:

What is Jupyter in Python?

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more

Installation Procedure:

$conda install jupyter

To start a web application ,open the terminal type jupyter-notebook

$jupyter-notebook

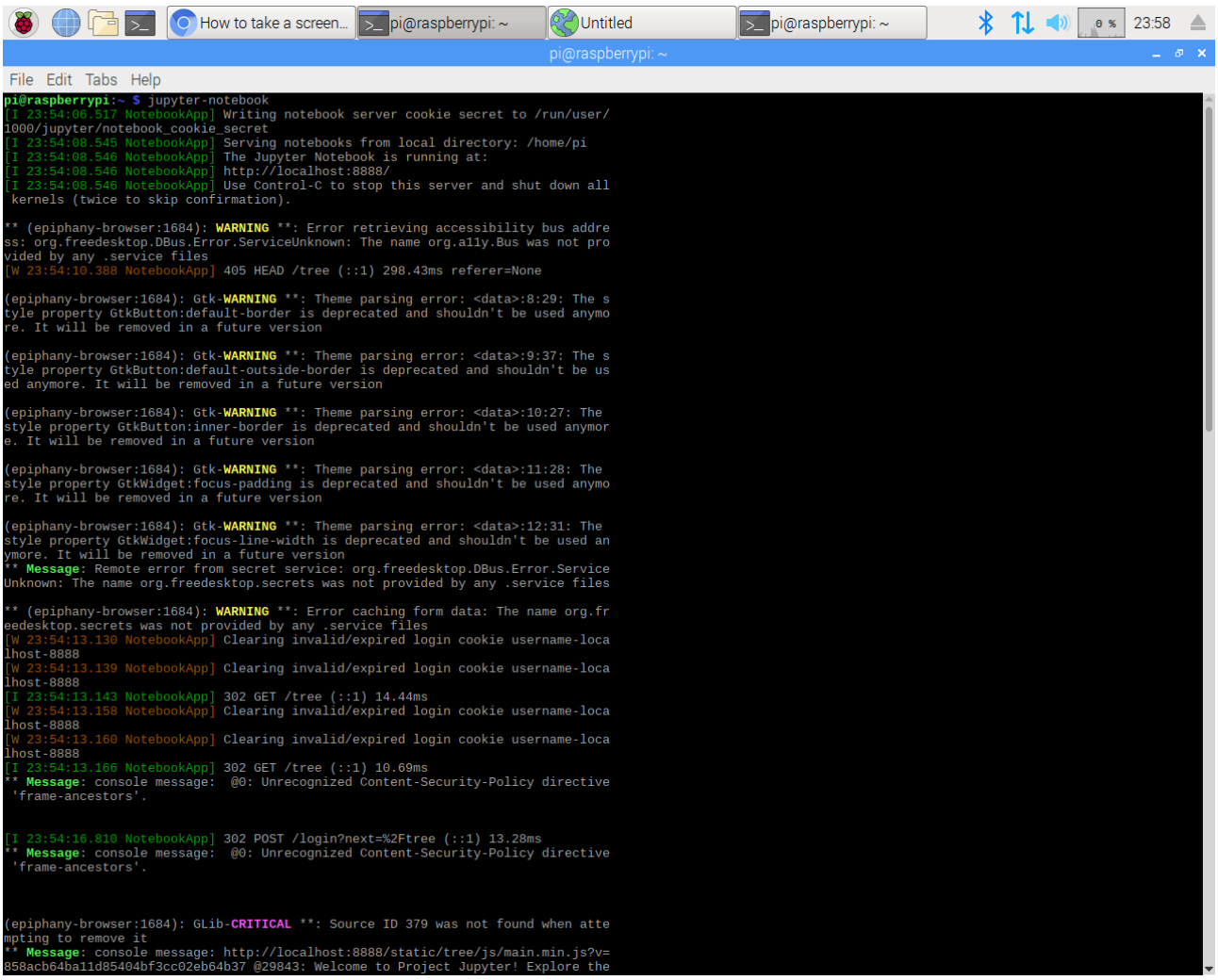
if ask for the password or token (copy the first token generated in server in terminal)

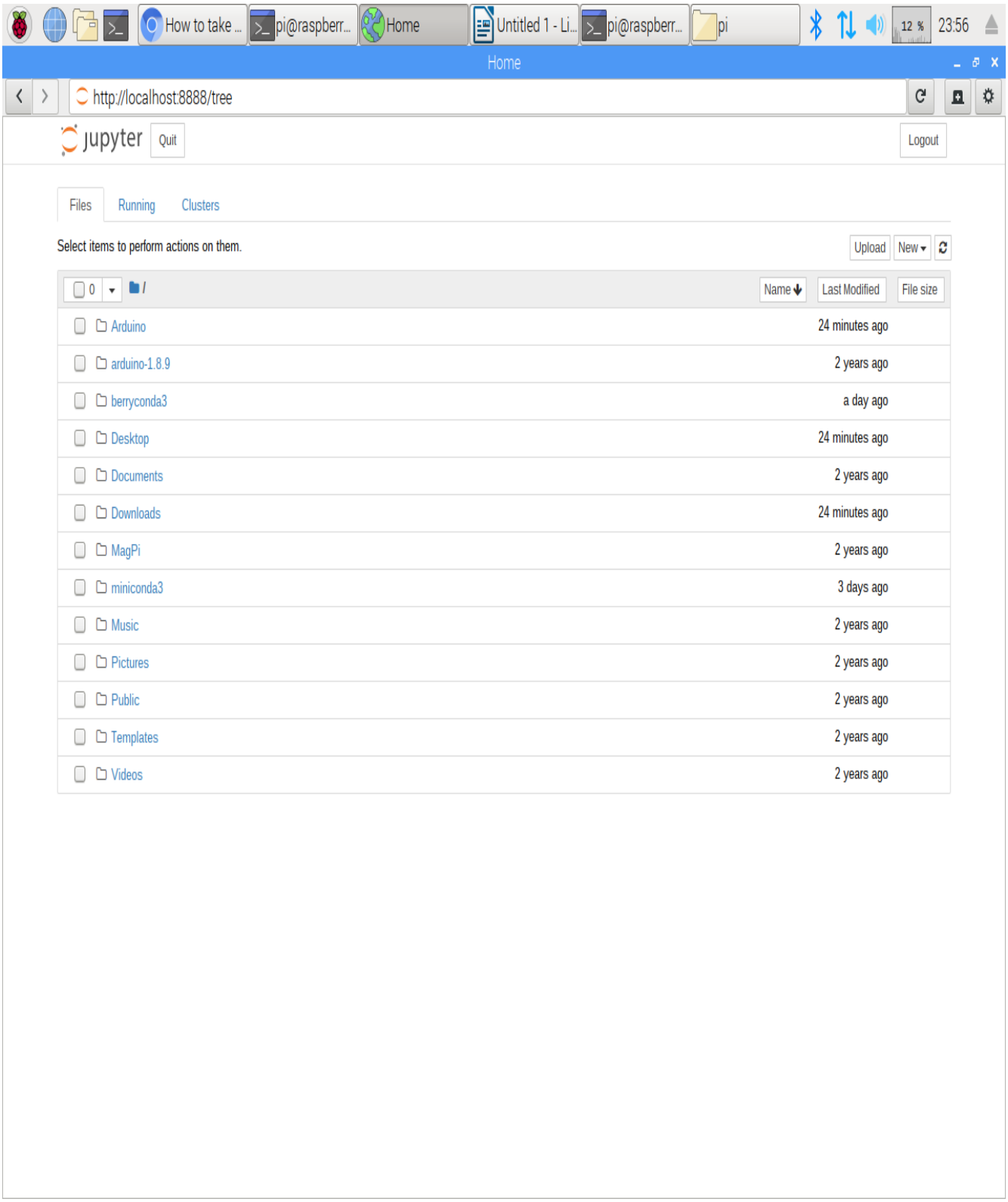
or

follow the comands to reset jupyter password

$jupyter notebook --generate-config

$jupyter notebook password




How to take ...
pi@raspberr...
Home
Untitled 1 - Li...
pi@raspberr...
pi
12 %
23:56
Home
http://localhost:8888/tree
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
Arduino 24 minutes ago
arduino-1.8.9 2 years ago
berryconda3 a day ago
Desktop 24 minutes ago
Documents 2 years ago
Downloads 24 minutes ago
MagPi 2 years ago
miniconda3 3 days ago
Music 2 years ago
Pictures 2 years ago
Public 2 years ago
Templates 2 years ago
Videos 2 years ago

172.16.34.16/iot/exa... | pi@raspberrypi: ~ | pi | Untitled1 | 17 % | 00:02

Untitled1

http://localhost:8888/notebooks/Untitled1.ipynb?kernel_name=python3

Home | Untitled1

jupyter **Untitled1** Last Checkpoint: Last Thursday at 13:00 (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Code

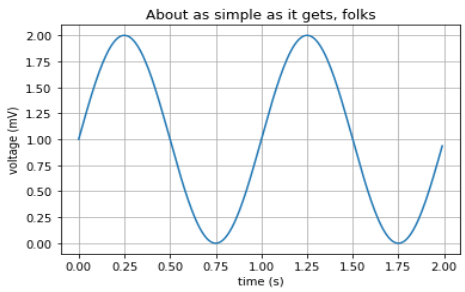
In [1]:
```
import numpy as np
a=np.array([1,2,3])
print(a)
```

```
[1 2 3]
```

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2*np.pi*t)
plt.plot(t, s)

plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.title('About as simple as it gets, folks')
plt.grid(True)
plt.savefig("test.png")
plt.show()
```



In [4]:
```
import pandas as pd
df=pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot([0, 1, 2, 3, 4, 5])

plt.show()
```