

2022년 IoT기반 스마트 솔루션 개발자 양성과정



Programming : Python

8-Matplotlib

담당 교수 : 윤 종 이

010-9577-1696

ojo1696@naver.com

<https://cafe.naver.com/yoons2022>



충북대학교 공동훈련센터


Matplotlib

- 파이썬에서 자료를 차트(chart)나 플롯(plot)으로 시각화(visulaization)하는 패키지
 - 라인 플롯(line plot)
 - 스캐터 플롯(scatter plot)
 - 컨투어 플롯(contour plot)
 - 서피스 플롯(surface plot)
 - 바 차트(bar chart)
 - 히스토그램(histogram)
 - 박스 플롯(box plot)



MatPlotLib.org

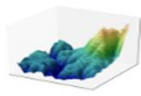
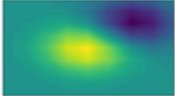
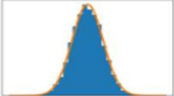

- <https://matplotlib.org>



Version 3.0.0

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](#) shells, the [Jupyter](#) notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [Matplotlib installation instructions](#).

Documentation

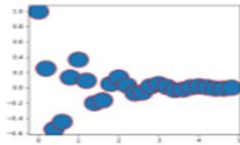
This is the documentation for Matplotlib version 3.0.0.

To get started, read the [User's Guide](#).

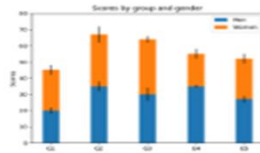


충북대학교 공동훈련센터

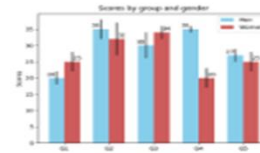
Gallery



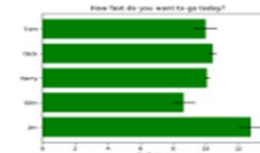
Arctest



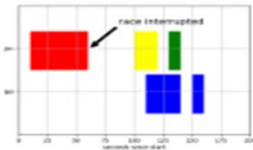
Stacked Bar Graph



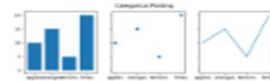
Barchart



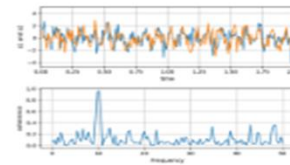
Horizontal bar chart



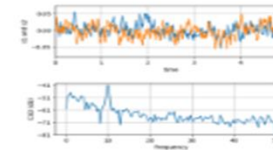
Broken Barh



Plotting categorical variables



Plotting the coherence of two signals

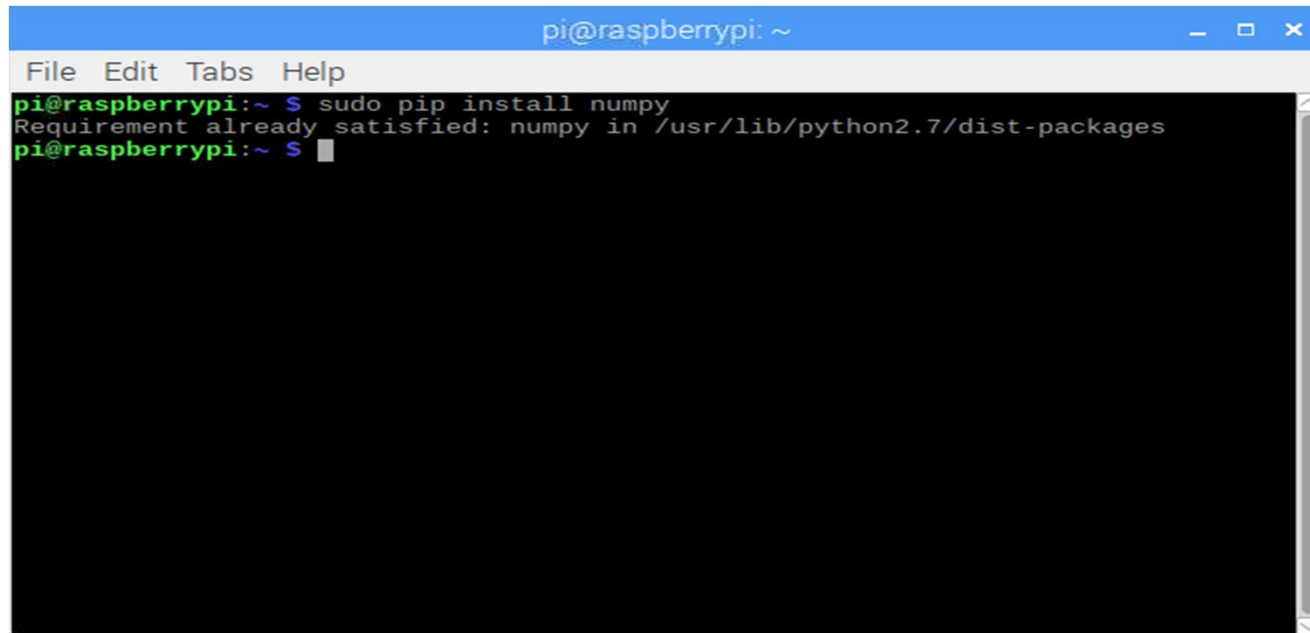


CSD Demo



Install numpy

- \$ sudo pip install numpy

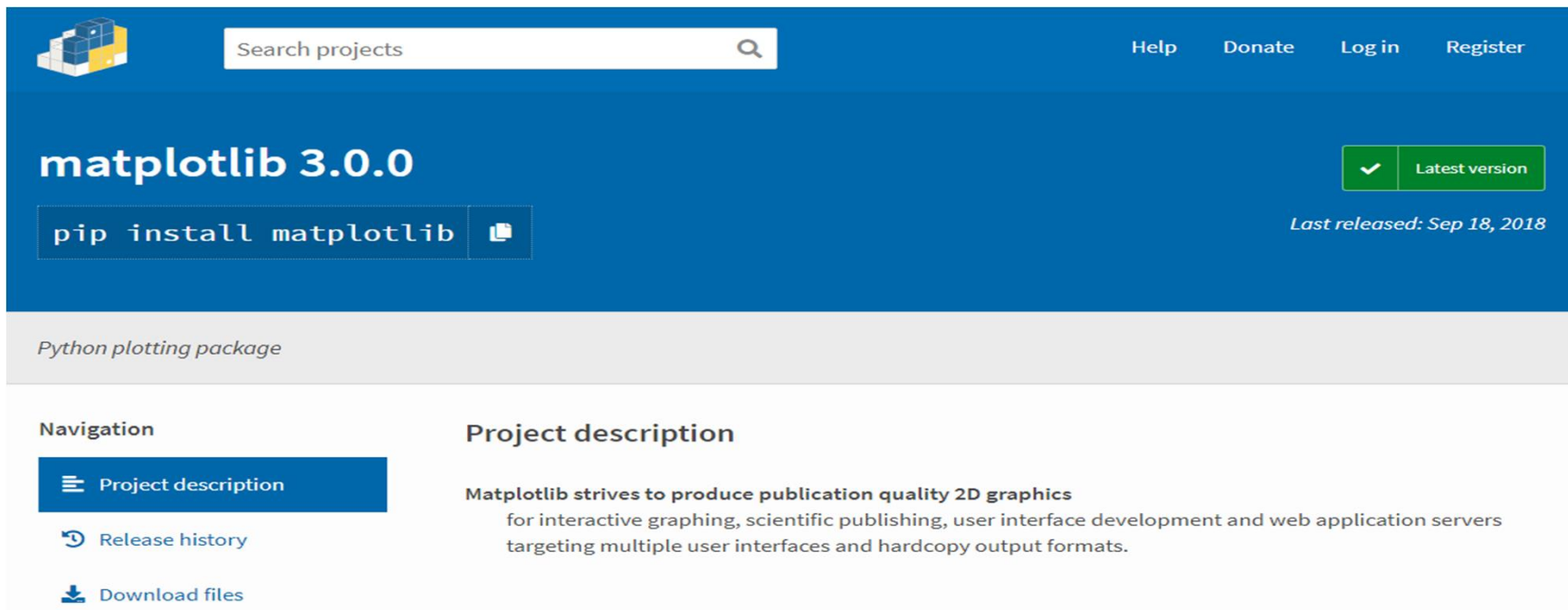


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo pip install numpy  
Requirement already satisfied: numpy in /usr/lib/python2.7/dist-packages  
pi@raspberrypi:~ $
```



PIP install

- <https://pypi.org>




The screenshot shows the PyPI page for the matplotlib 3.0.0 package. The header is blue with the PyPI logo, a search bar, and links for Help, Donate, Log in, and Register. The main section features the package name 'matplotlib 3.0.0' in large white text, a green 'Latest version' button with a checkmark, and a green button with the command 'pip install matplotlib' and a copy icon. Below this, it says 'Python plotting package'. The page is divided into two columns: 'Navigation' on the left with links for 'Project description' (highlighted), 'Release history', and 'Download files'; and 'Project description' on the right, which states that Matplotlib strives to produce publication quality 2D graphics for interactive graphing, scientific publishing, user interface development, and web application servers.

Search projects

Help Donate Log in Register

matplotlib 3.0.0

✓ Latest version

`pip install matplotlib` 

Last released: Sep 18, 2018

Python plotting package

Navigation

- Project description
- Release history
- Download files

Project description

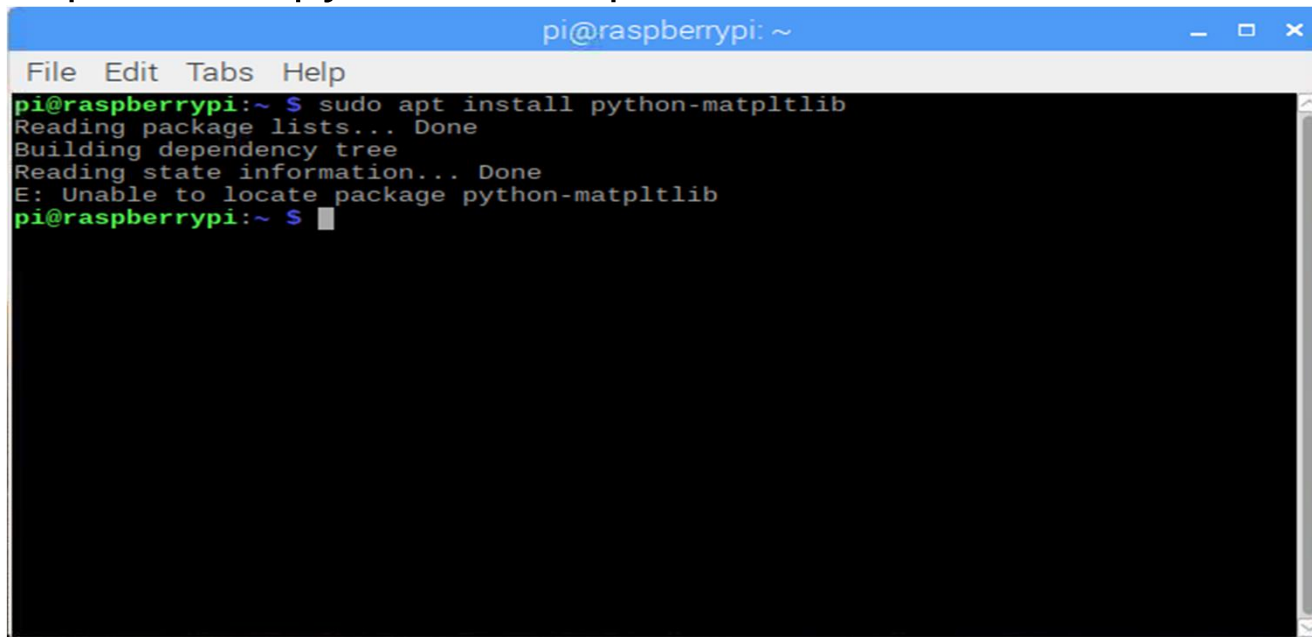
Matplotlib strives to produce publication quality 2D graphics
for interactive graphing, scientific publishing, user interface development and web application servers targeting multiple user interfaces and hardcopy output formats.



충북대학교 공동훈련센터

Install Matplotlib

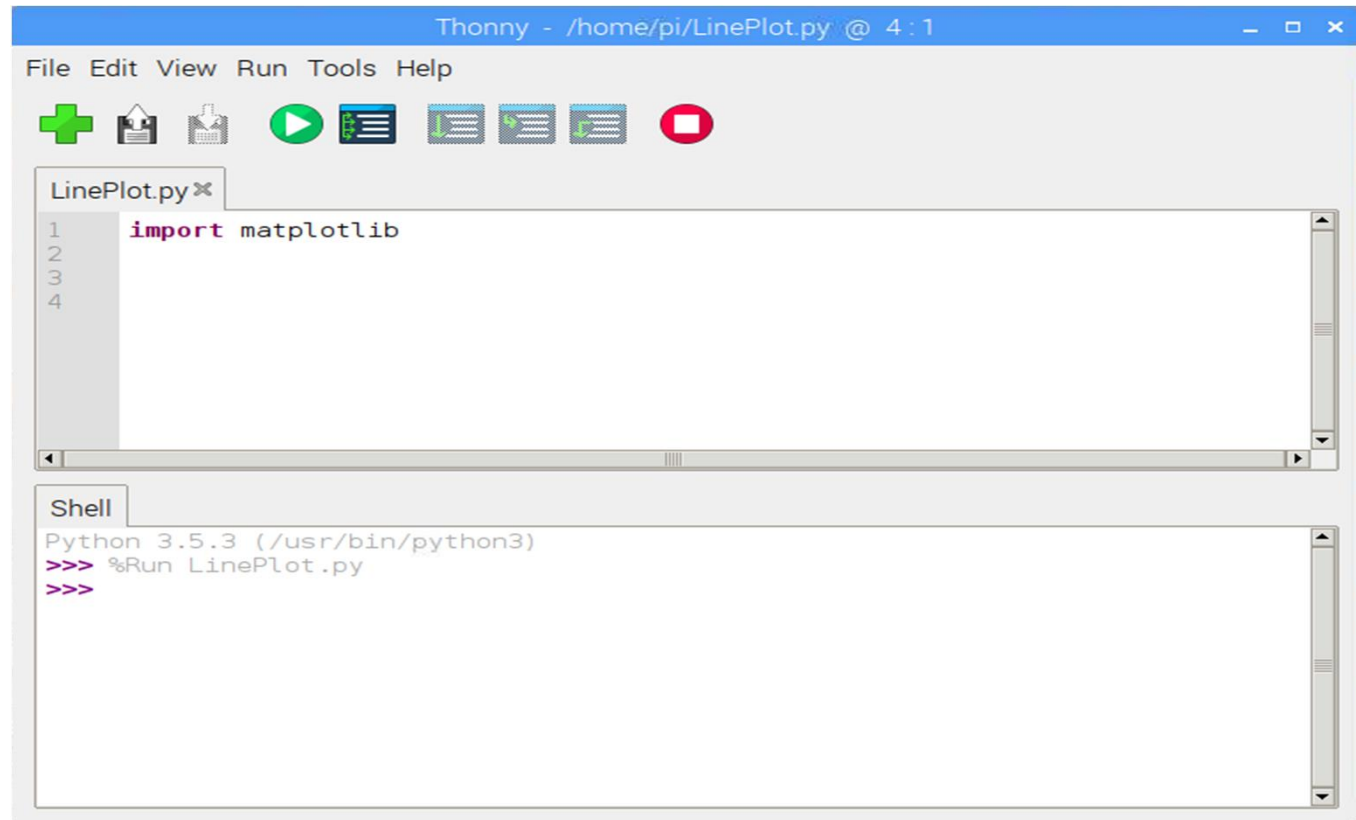
- `$ sudo apt install python3-matplotlib`



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt install python-matplotlib  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
E: Unable to locate package python-matplotlib  
pi@raspberrypi:~ $
```



설치 확인



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - /home/pi/LinePlot.py @ 4 : 1". The menu bar includes "File", "Edit", "View", "Run", "Tools", and "Help". The toolbar contains icons for creating a new file, opening a file, saving a file, running the code (a green play button), and other development tools. The editor window, titled "LinePlot.py", contains the following code:

```
1 import matplotlib
2
3
4
```

Below the editor is a "Shell" window. It shows the output of running the script:

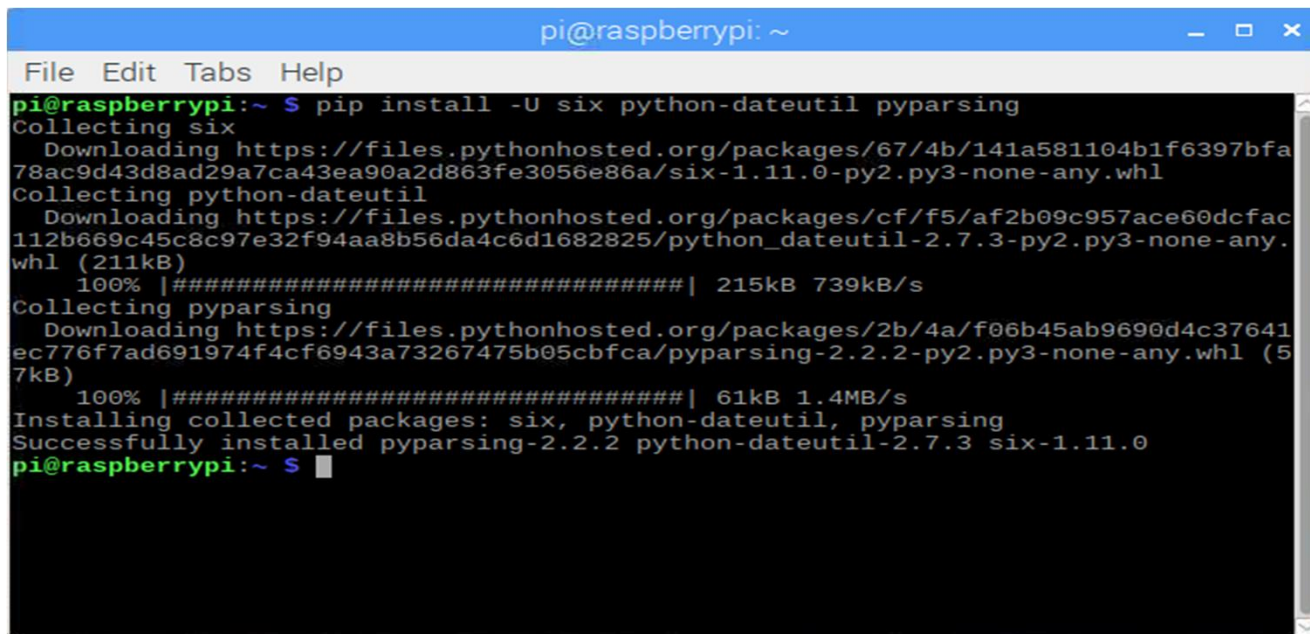
```
Python 3.5.3 (/usr/bin/python3)
>>> %Run LinePlot.py
>>>
```

The shell window indicates that the script was executed successfully using Python 3.5.3.



필수 util 설치

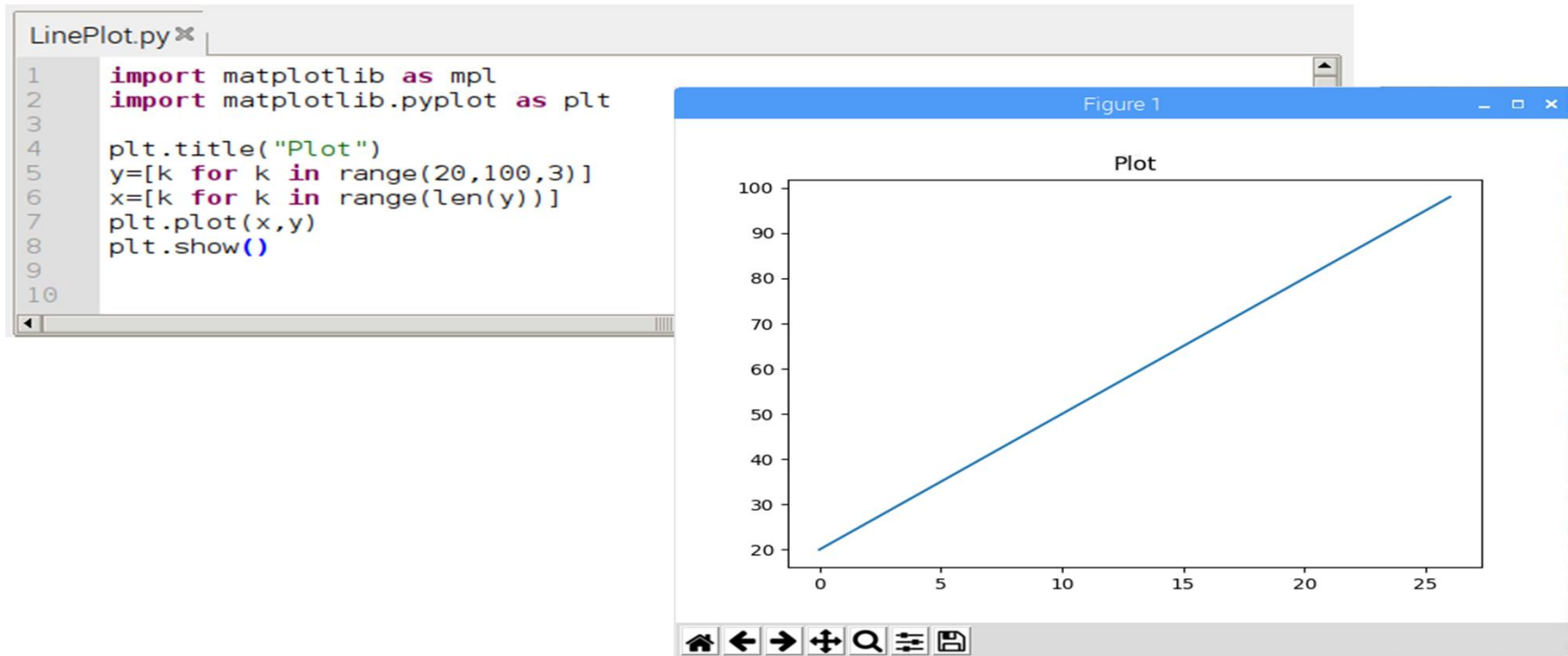
- \$ sudo pip install -U six python-dateutil cyclcr pyparsing



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pip install -U six python-dateutil pyparsing  
Collecting six  
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py2.py3-none-any.whl  
Collecting python-dateutil  
  Downloading https://files.pythonhosted.org/packages/cf/f5/af2b09c957ace60dcfac112b669c45c8c97e32f94aa8b56da4c6d1682825/python_dateutil-2.7.3-py2.py3-none-any.whl (211kB)  
    100% |#####| 215kB 739kB/s  
Collecting pyparsing  
  Downloading https://files.pythonhosted.org/packages/2b/4a/f06b45ab9690d4c37641ec776f7ad691974f4cf6943a73267475b05cbfca/pyparsing-2.2.2-py2.py3-none-any.whl (57kB)  
    100% |#####| 61kB 1.4MB/s  
Installing collected packages: six, python-dateutil, pyparsing  
Successfully installed pyparsing-2.2.2 python-dateutil-2.7.3 six-1.11.0  
pi@raspberrypi:~ $
```



Ex1 : Line Plot

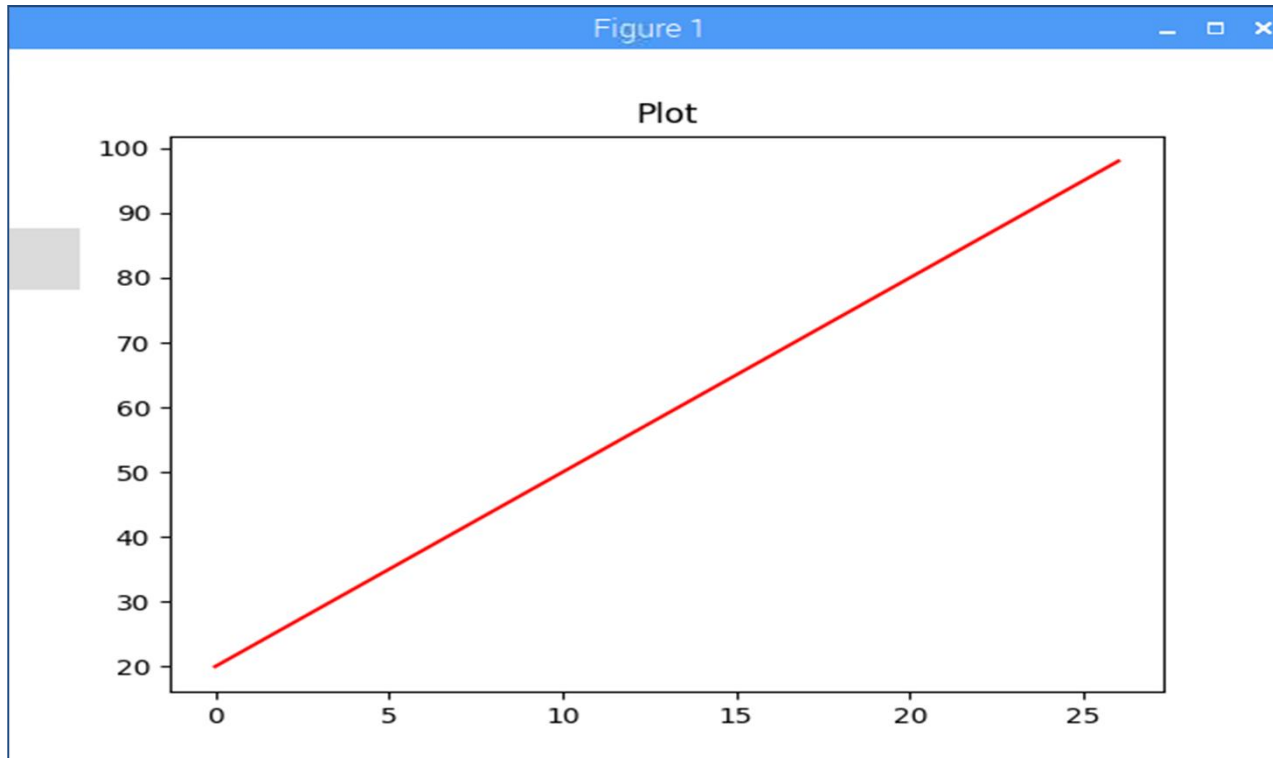


Ex2 : Line Color

```
LinePlot.py x
1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3
4  plt.title("Plot")
5  y=[k for k in range(20,100,3)]
6  x=[k for k in range(len(y))]
7  plt.plot(x,y,'r')
8  plt.show()
9
10
```



Ex2 : Run



Char	Color
b	Blue
g	Green
r	Red
c	Cyan
m	Magenta
Y	Yellow
k	Black
w	White

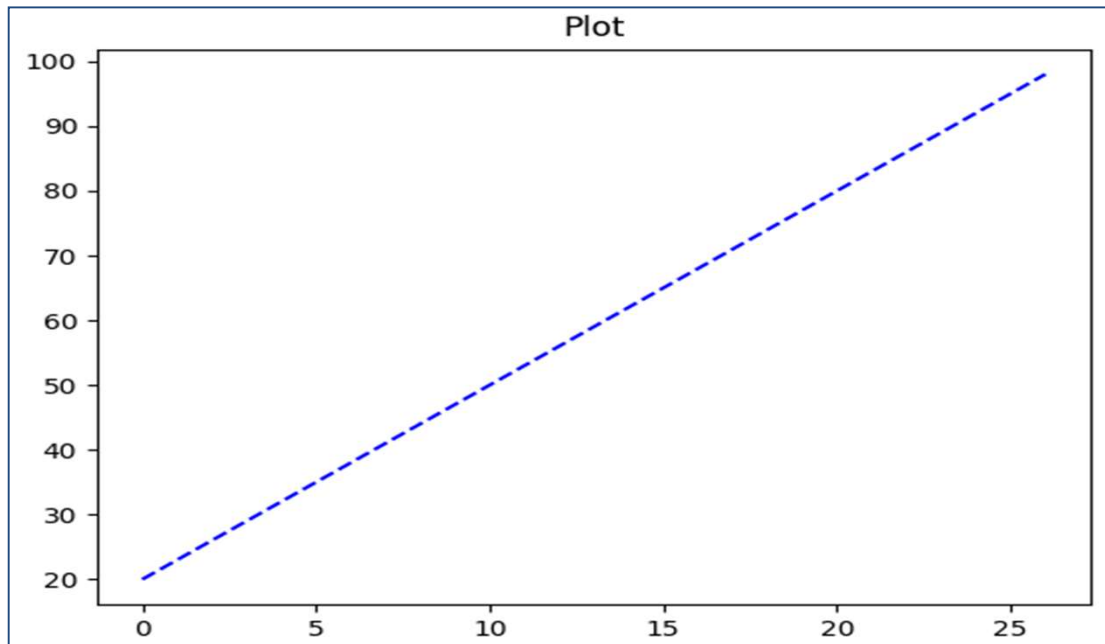


Ex3 : Line Style

```
LinePlot.py *  
1  import matplotlib as mpl  
2  import matplotlib.pyplot as plt  
3  
4  plt.title("Plot")  
5  y=[k for k in range(20,100,3)]  
6  x=[k for k in range(len(y))]  
7  plt.plot(x,y,'b--')  
8  plt.show()  
9  
10 |  
11  
12
```



Ex3 : Run



Char	Style
-	Solid
--	Dashed
-.	Dash-dot
:	Dotted
.	Point



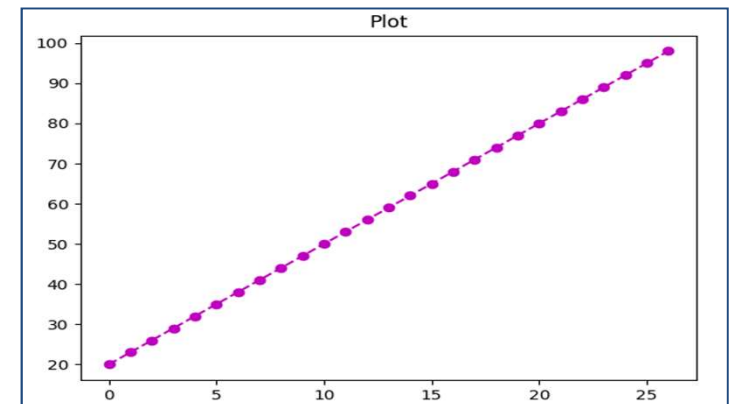
Ex4 : Marker

```
LinePlot.py ✕
1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3
4  plt.title("Plot")
5  y=[k for k in range(20,100,3)]
6  x=[k for k in range(len(y))]
7  plt.plot(x,y,'m--o')
8  plt.show()
9
10
11
12
```



Ex4 : Run

Char	Style	Char	Style
.	Point	s	Square
,	Pixel	p	Pentagon
o	Circle	*	Star
v	Triangle-down	h	Hexagon1
^	Triangle-up	H	Hexagon2
<	Triangle-left	+	Plus
>	Triangle-right	x	X
1	Tri-down	D	Diamond
2	Tri-up	d	Thin-diamond
3	Tri-left		Vline
4	Tri-right	_	Hline

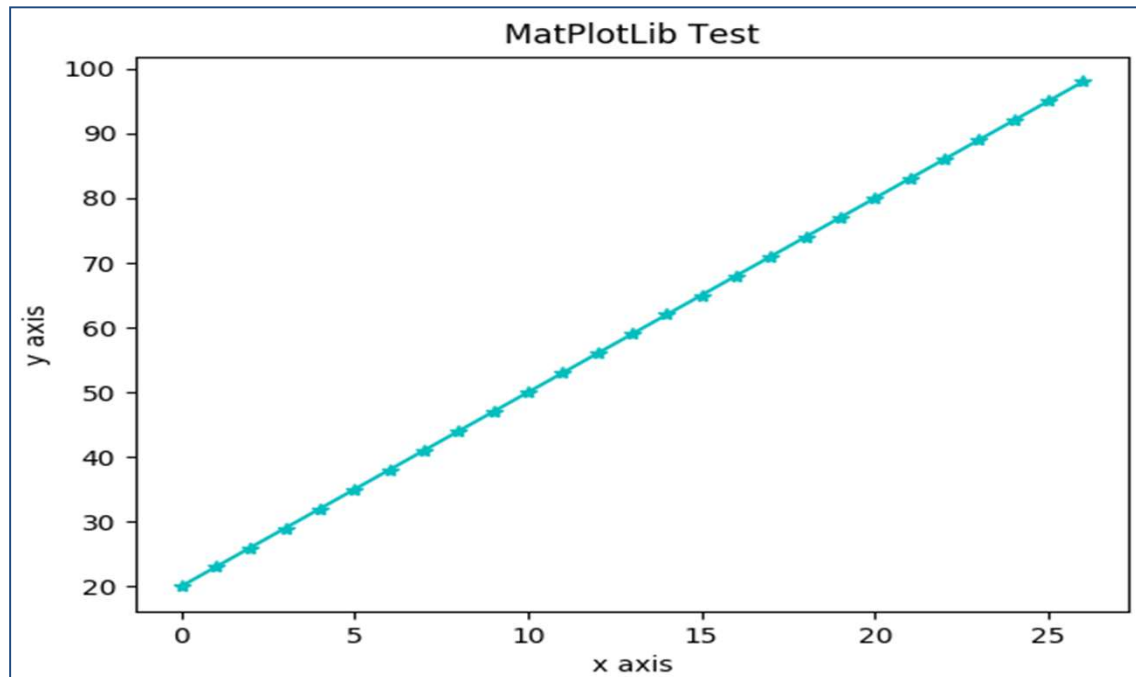


Ex5 : label & Title

```
LinePlot.py *  
1  import matplotlib as mpl  
2  import matplotlib.pyplot as plt  
3  
4  plt.title("Plot")  
5  y=[k for k in range(20,100,3)]  
6  x=[k for k in range(len(y))]  
7  plt.plot(x,y,'c-*')  
8  plt.xlabel('x axis')  
9  plt.ylabel('y axis')  
10 plt.title('MatPlotLib Test')  
11 plt.show()  
12 |
```



Ex5 : Run

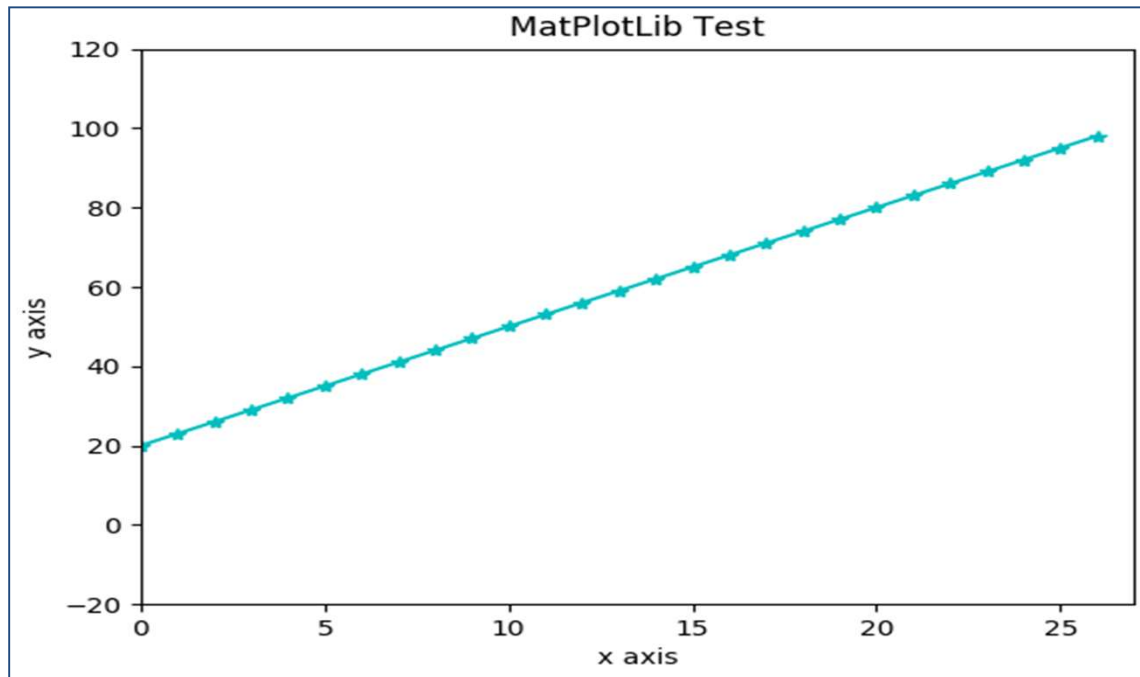


Ex6 : Plot Limit

```
LinePlot.py
1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3
4  plt.title("Plot")
5  y=[k for k in range(20,100,3)]
6  x=[k for k in range(len(y))]
7  plt.xlim(0,len(y))
8  plt.ylim(-20,120)
9  plt.plot(x,y,'c-*')
10 plt.xlabel('x axis')
11 plt.ylabel('y axis')
12 plt.title('MatPlotLib Test')
13 plt.show()
14
```



Ex6 : Run

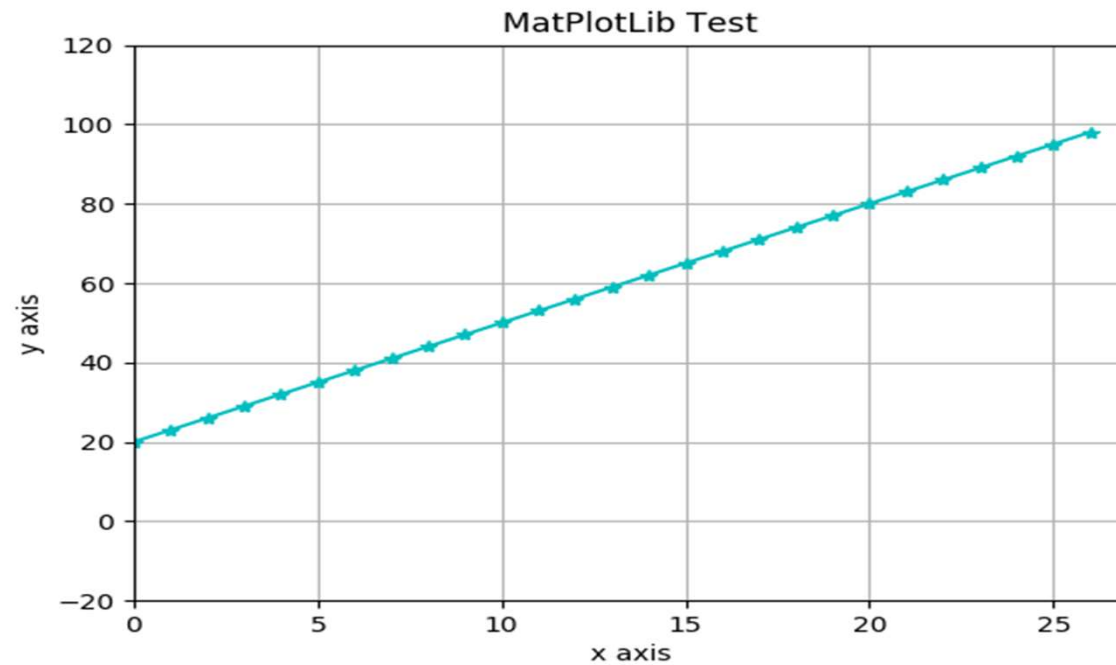


Ex7 : Grid

```
LinePlot.py x
1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3
4  plt.title("Plot")
5  y=[k for k in range(20,100,3)]
6  x=[k for k in range(len(y))]
7  plt.xlim(0,len(y))
8  plt.ylim(-20,120)
9  plt.plot(x,y,'c-*')
10 plt.xlabel('x axis')
11 plt.ylabel('y axis')
12 plt.title('MatPlotLib Test')
13 plt.grid(True)
14 plt.show()
15
~(
```



Ex7 : Run

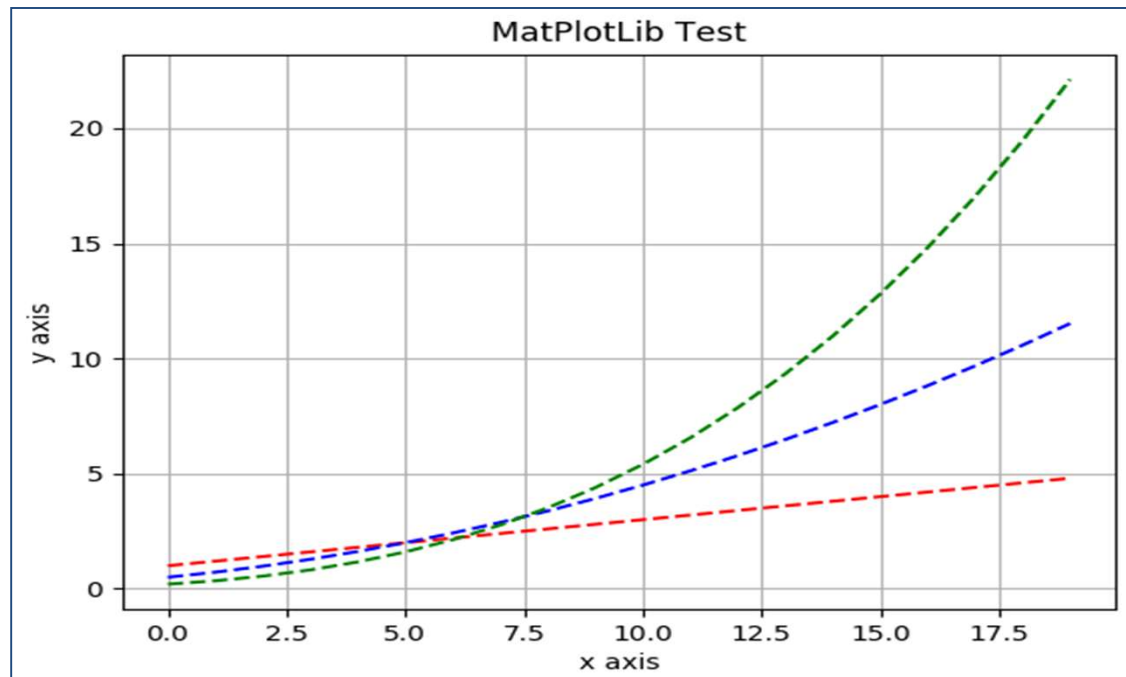


Ex8 : Multi Line

```
LinePlotz.py x
1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3  import numpy
4
5  plt.title('MatPlotLib Test')
6  plt.xlabel('x axis')
7  plt.ylabel('y axis')
8
9  y1=numpy.arange(1.0,5.0,0.2)
10 x=[k for k in range(len(y1))]
11 plt.plot(x,y1,'r--',
12          x,0.5*y1**2,'b--',
13          x,0.2*y1**3,'g--')
14 plt.grid(True)
15 plt.show()
```



Ex8 : Run



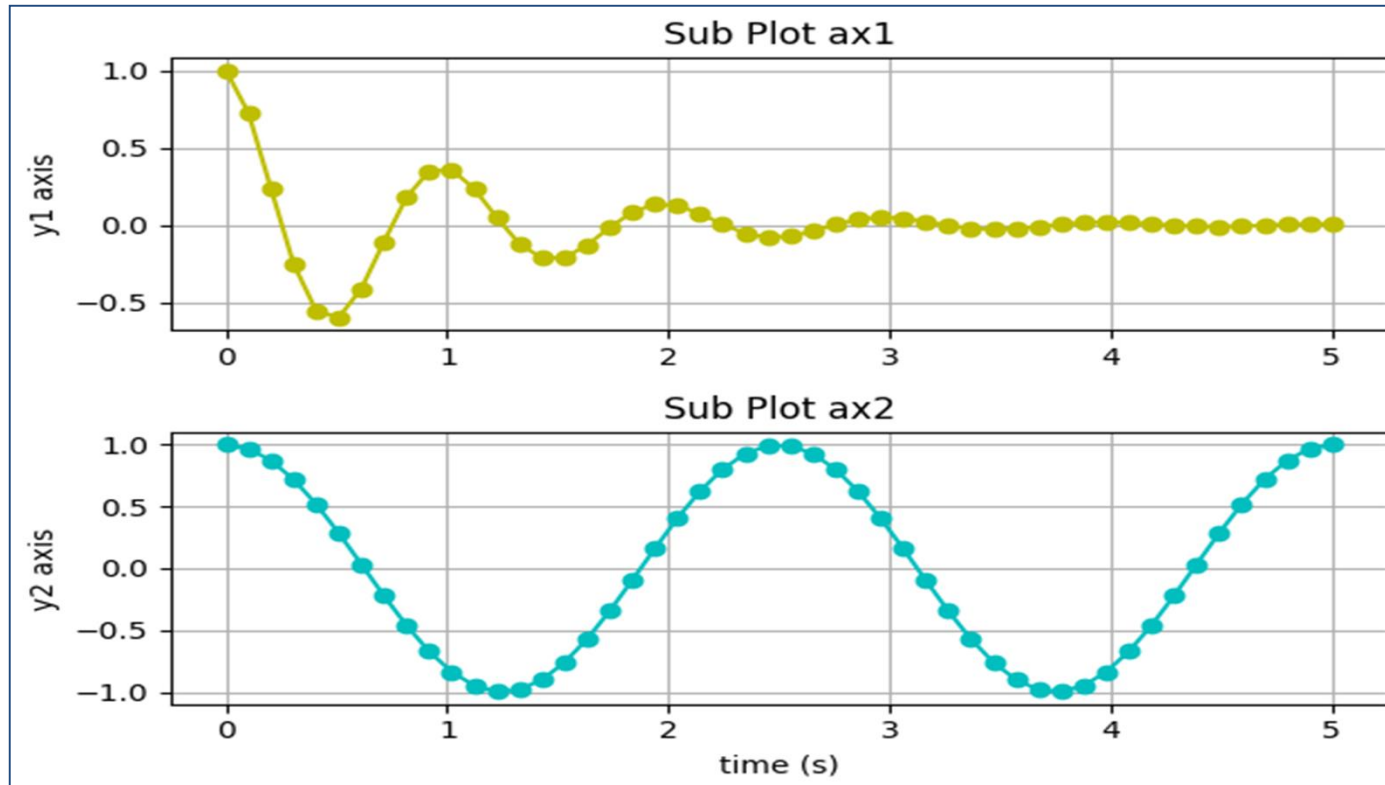
충북대학교 공동훈련센터

Ex9 : Sub Plot

```
SubPlot.py *  
1  import matplotlib.pyplot as plt  
2  import numpy as np  
3  
4  x1=np.linspace(0.0, 5.0)  
5  x2=np.linspace(0.0, 2.0)  
6  y1=np.cos(2 * np.pi * x1) * np.exp(-x1)  
7  y2=np.cos(2 * np.pi * x2)  
8  
9  ax1=plt.subplot(2,1,1)  
10 plt.plot(x1,y1,'yo-')  
11 plt.title('Sub Plot ax1')  
12 plt.ylabel('y1 axis')  
13 plt.grid(True)  
14 print(ax1)  
15  
16 ax2=plt.subplot(2,1,2)  
17 plt.plot(x1,y2,'co-')  
18 plt.title('Sub Plot ax2')  
19 plt.ylabel('y2 axis')  
20 plt.xlabel('time (s)')  
21 plt.grid(True)  
22 print(ax2)  
23  
24 plt.tight_layout()  
25 plt.show()  
26
```



Ex9 : Run

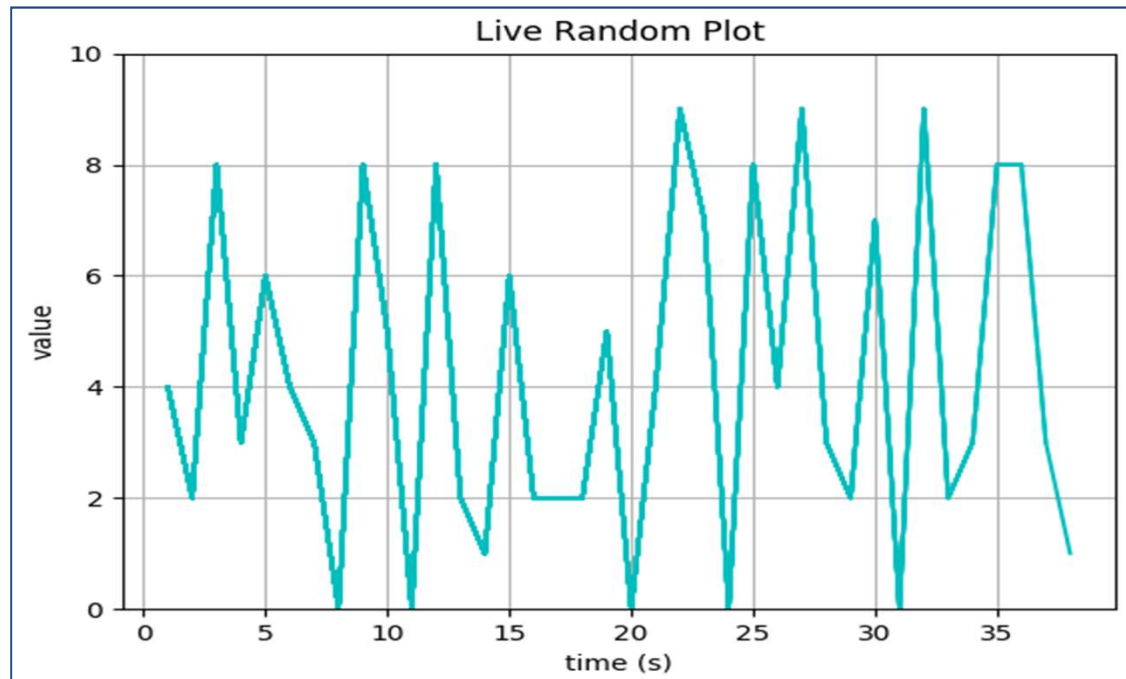


Ex10 : Live Random Plot

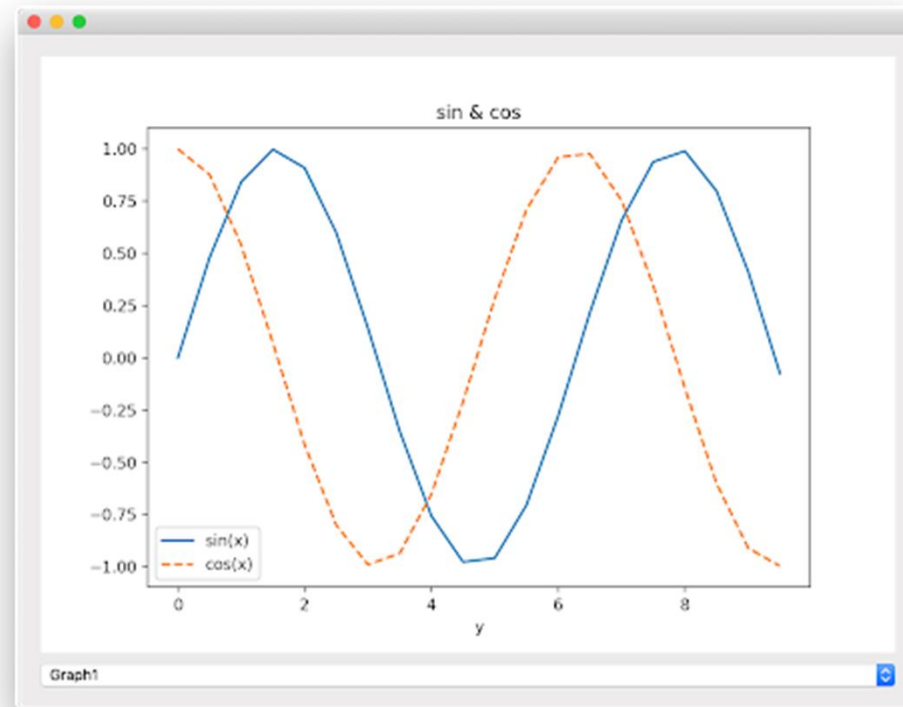
```
LiveRandomPlot.py *  
1  import matplotlib.pyplot as plt  
2  import numpy as np  
3  import random  
4  
5  plt.title('Live Random Plot')  
6  plt.ylabel('value')  
7  plt.xlabel('time (s)')  
8  plt.ylim(0,10)  
9  plt.grid(True)  
10 plt.ion()  
11  
12 x=[]  
13 y=[]  
14 k=0  
15  
16 while True:  
17     k=k+1  
18     x.append(k)  
19     y.append(random.randrange(0,10))  
20  
21     plt.plot(x,y,'c-')  
22     plt.show()  
23     plt.pause(0.00001)  
24
```



Ex10 : Run

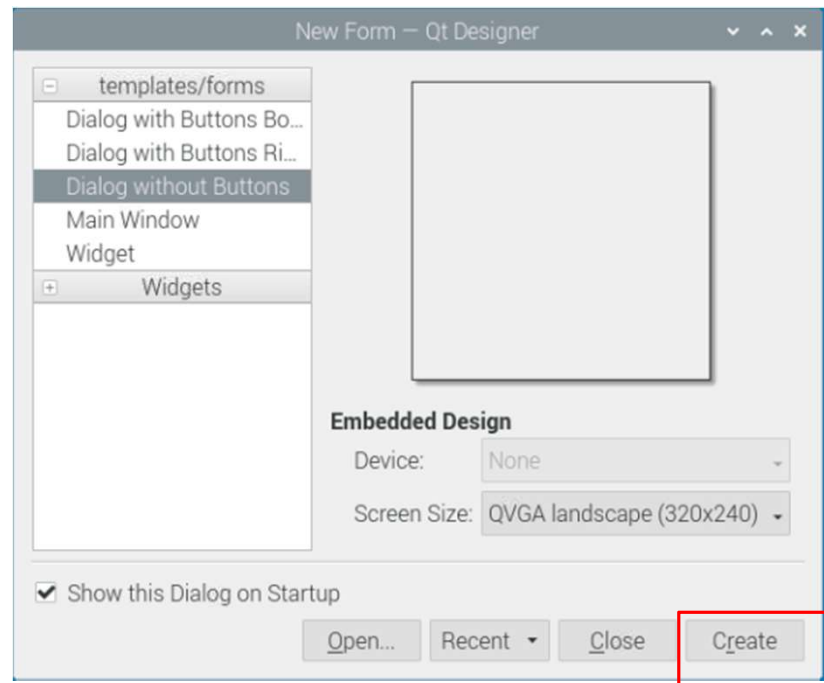


Matplotlib in PyQt5

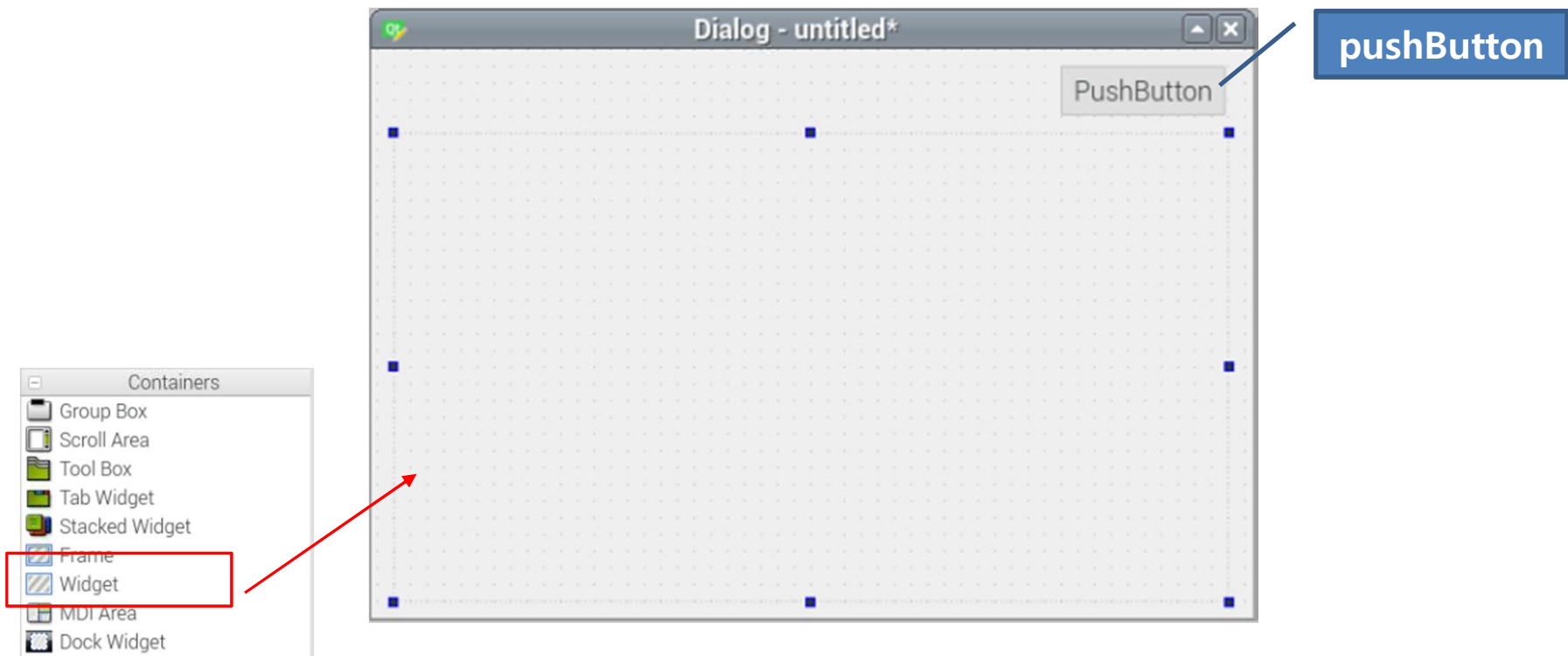


충북대학교 공동훈련센터

New Form – Qt Designer

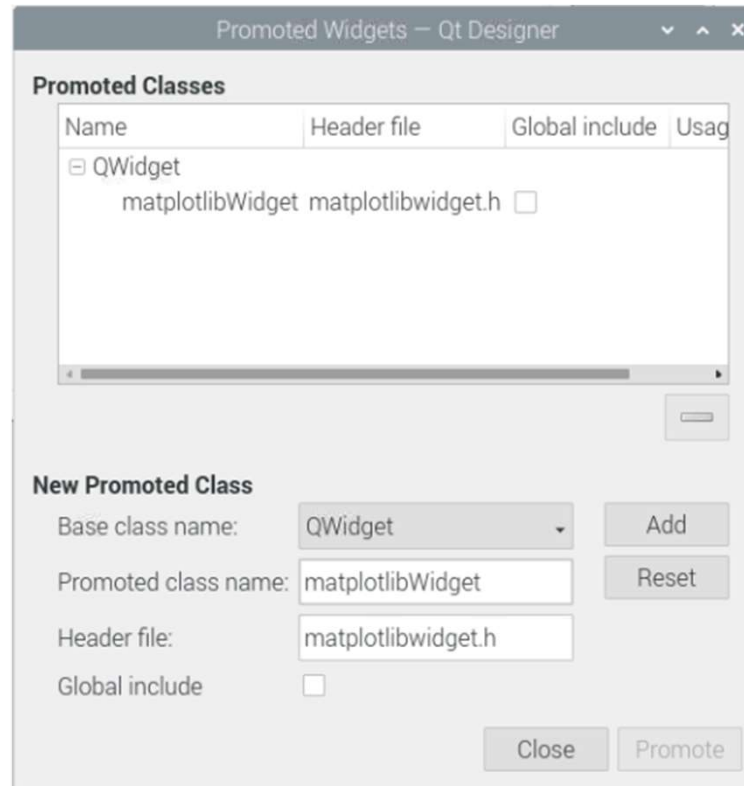
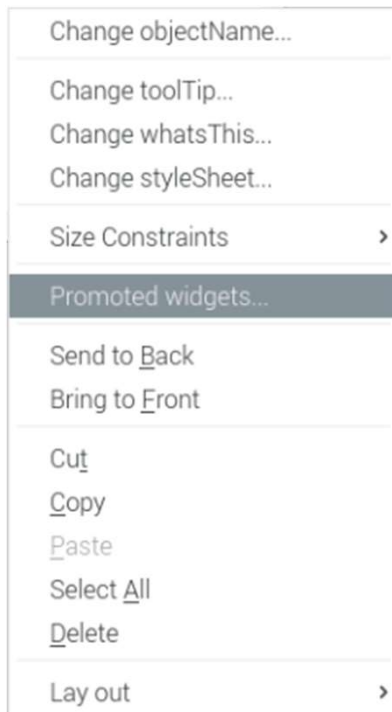


Dialog



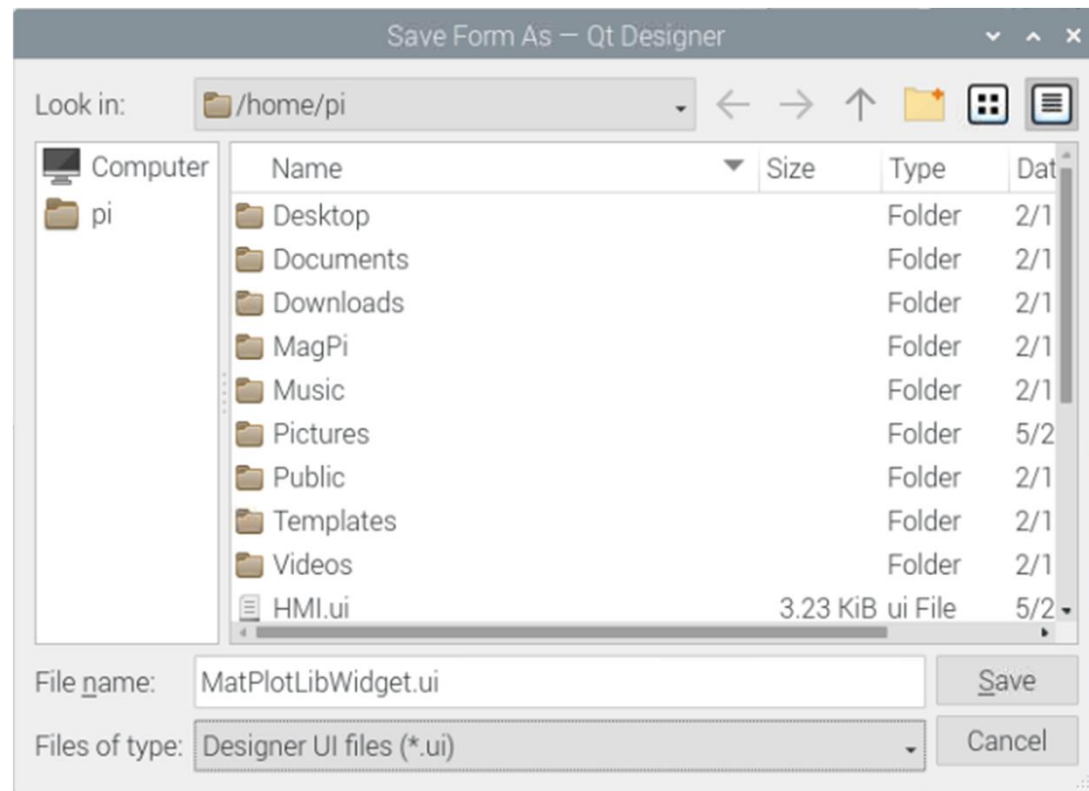
Promoted widget

@ Widget {Mouse-Right}



충북대학교 공동훈련센터

Save as – MatPlotLibWidget.ui



충북대학교 공동훈련센터

Thonny – matplotlibwidget.py

```
matplotlibwidget.py ✕
1 import PyQt5
2 from PyQt5.QtGui import *
3 from PyQt5.QtWidgets import *
4
5 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAff as FigureCanvas
6 from matplotlib.figure import Figure
7
8 class MplCanvas(FigureCanvas):
9
10     def __init__(self):
11         self.fig = Figure()
12         self.ax = self.fig.add_subplot(111)
13
14         FigureCanvas.__init__(self, self.fig)
15         FigureCanvas.setSizePolicy(self, QSizePolicy.Expanding, QSizePolicy.Expanding)
16         FigureCanvas.updateGeometry(self)
17
18 class matplotlibWidget(QWidget):
19
20     def __init__(self, parent = None):
21         QWidget.__init__(self, parent)
22         self.canvas = MplCanvas()
23         self.vbl = QVBoxLayout()
24         self.vbl.addWidget(self.canvas)
25         self.setLayout(self.vbl)
26 --
```

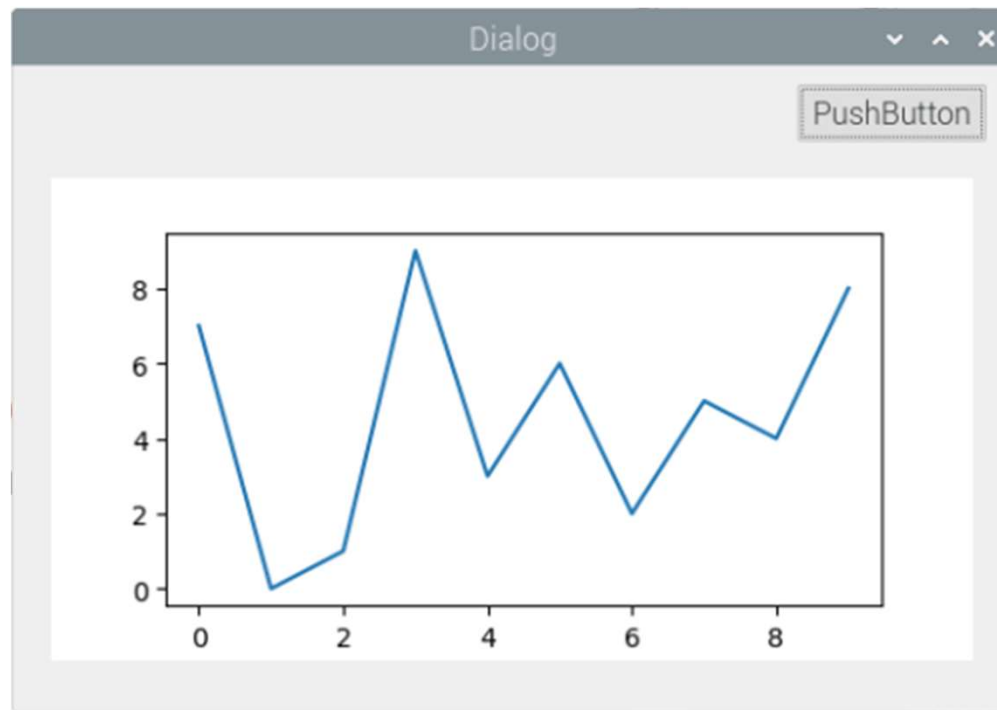


Thonny – PyQt5_MatplotlibWidget.py

```
PyQt5_MatplotlibWidget.py ✕
1 import sys
2 from PyQt5.QtWidgets import *
3 from PyQt5 import uic
4 import random
5
6 uiDialog='MatplotlibWidget.ui'
7
8 class MyDialog(QDialog):
9     def __init__(self):
10         QDialog.__init__(self, None)
11         uic.loadUi(uiDialog, self)
12         self.pushButton.clicked.connect(self.buttonClick)
13
14     def buttonClick(self):
15         randomNumbers = random.sample(range(0, 10), 10)
16         self.widget.canvas.ax.clear()
17         self.widget.canvas.ax.plot(randomNumbers)
18         self.widget.canvas.draw()
19
20 if __name__ == '__main__':
21     app=QApplication(sys.argv)
22     form=MyDialog()
23     form.show()
24     app.exec()
```



Run



충북대학교 공동훈련센터