

Matplotlib



Data Visualization

2

คือ รูปแบบหนึ่งของการนำเสนอข้อมูลในรูปแบบต่างๆ เช่นกราฟ แผนภูมิ เพื่อสื่อสารให้ผู้อ่านเข้าใจข้อมูลได้ง่ายขึ้น



pandas , matplotlib , seaborn

Matplotlib

3

เป็น lib ที่ใช้ทำ Data Visualization

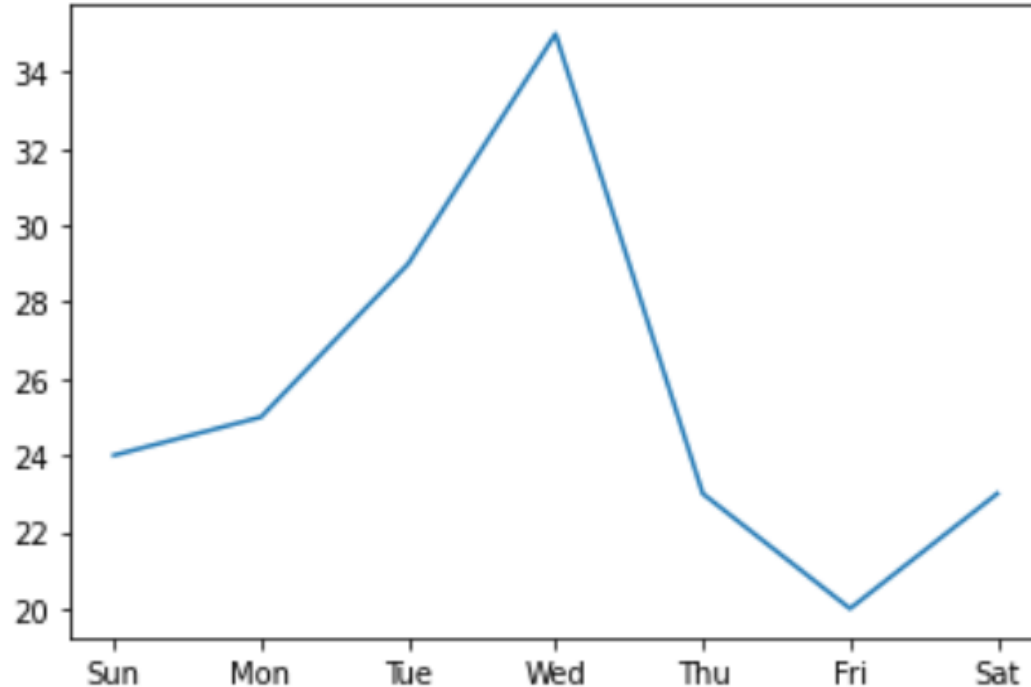
```
import matplotlib.pyplot as plt
```

Matplotlib

4

ตัวอย่างการใช้งาน

```
dic = {"day": ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'] ,  
       "temp": [24, 25, 29, 35, 23, 20, 23] }  
df = pd.DataFrame(dic)  
x = df['day']  
y = df['temp']  
plt.plot(x,y)  
plt.show()
```



Matplotlib – plot()

5

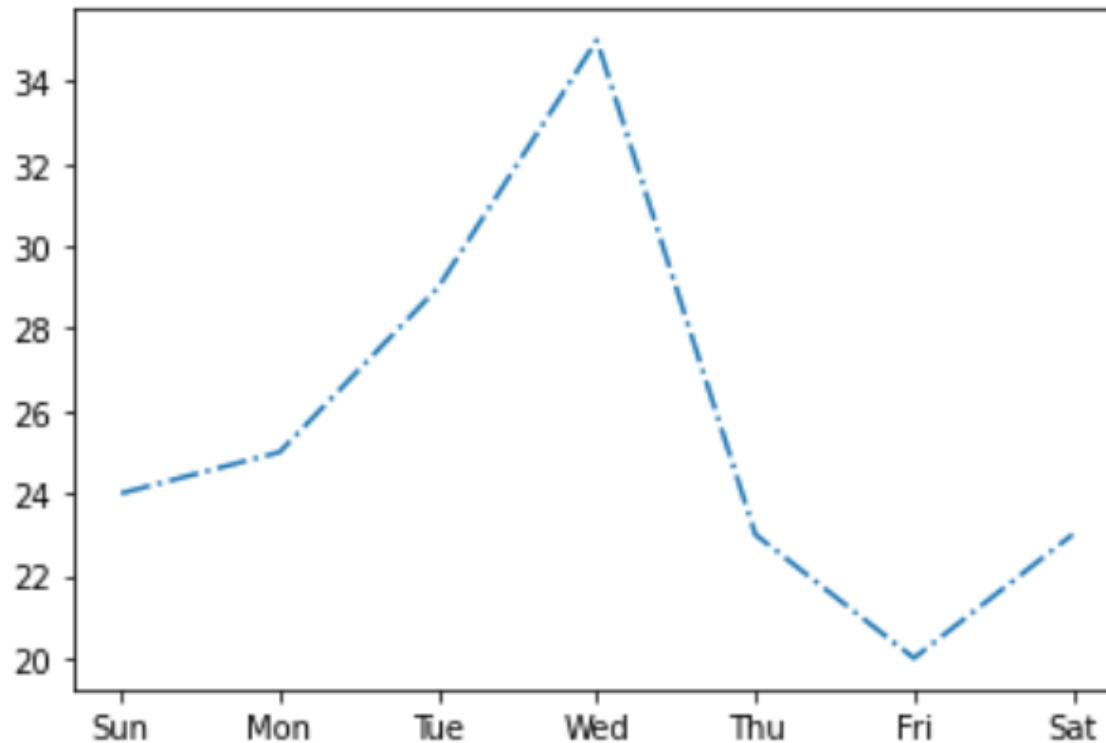
linestyle หรือ ls คือการกำหนดรูปแบบเส้นของกราฟ

ตัวอย่าง

```
plt.plot(x,y,ls='-.')
```

```
[<matplotlib.lines.Line2D at 0x1d012fcad00>]
```

-
--
-.
:
.



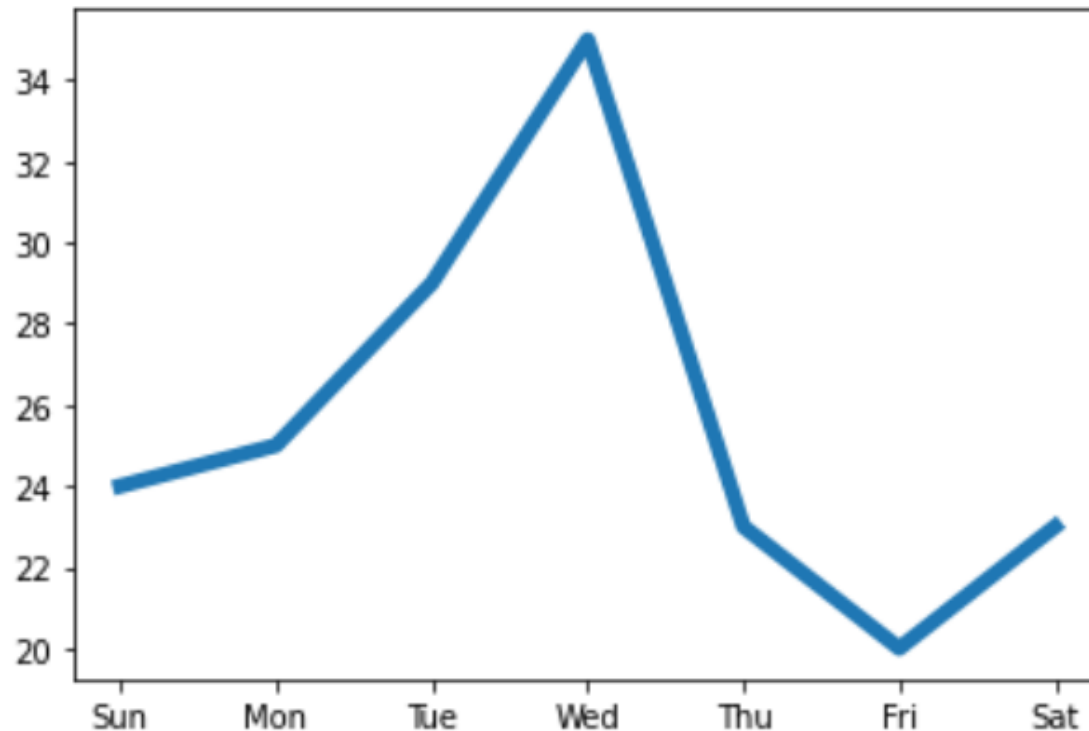
Matplotlib – plot()

6

linewidth หรือ lw คือการกำหนดความกว้างของเส้น
ตัวอย่าง

```
plt.plot(x,y,lw='5')
```

```
[<matplotlib.lines.Line2D at 0x1d013016f40>]
```



Matplotlib – plot()

7

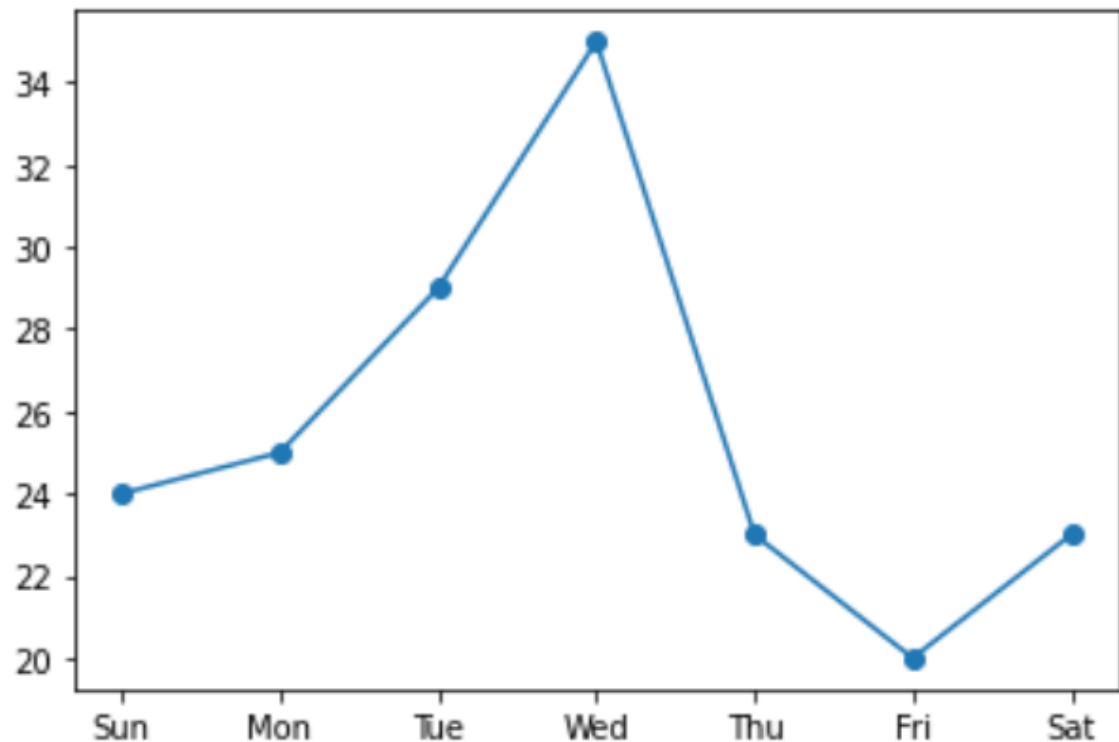
marker คือการกำหนดตัวเน้นจุดพิกัด

ตัวอย่าง

```
plt.plot(x,y,marker='o')
```

```
[<matplotlib.lines.Line2D at 0x1d012f9f790>]
```

* . , X
X + P s
D d p H
h v ^ <
1 2 3 4
| - o



Matplotlib – plot()

8

markersize หรือ ms คือการกำหนดขนาดของตัวเน้นพิกัด

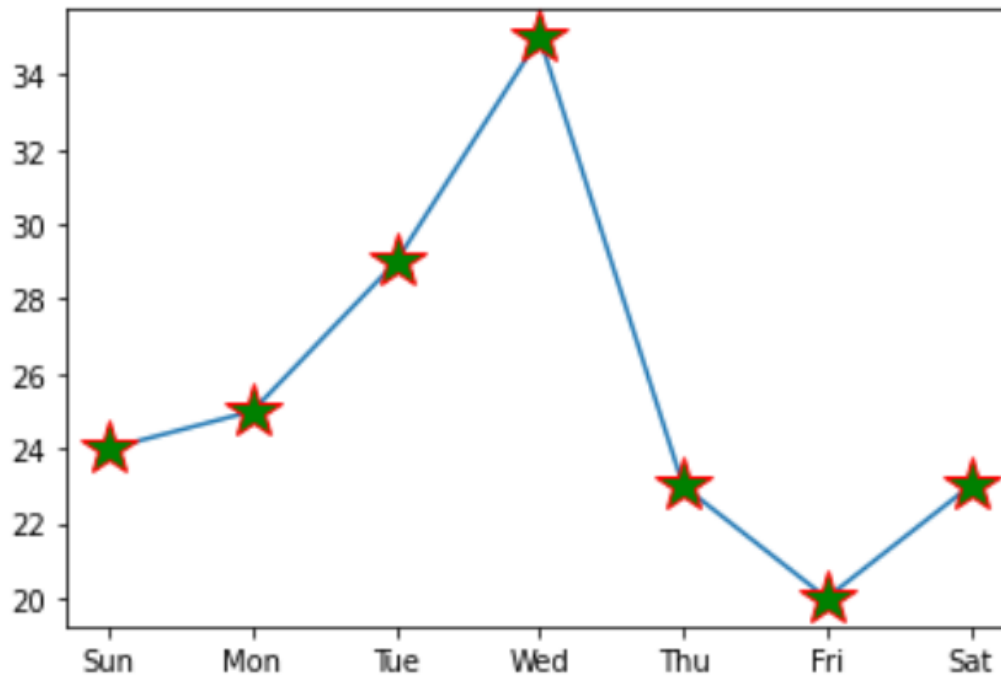
markeredgecolor หรือ mec คือการกำหนดสีขอบของตัวเน้นพิกัด

markerfacecolor หรือ mfc คือการกำหนดสีด้านในของตัวเน้นพิกัด

ตัวอย่าง

```
plt.plot(x,y,marker='*',ms=20,mec='red',mfc='green')
```

[<matplotlib.lines.Line2D at 0x1d0143d0e20>]



Matplotlib – Color

9

การกำหนดสีมี 3 รูปแบบ

1. ใส่ชื่อหรือชื่อย่อของสีเช่น 'red' หรือ 'r'
2. ใส่รหัสสีฐาน 16 เช่น '#FF0000'
3. ใส่ในรูปแบบ RGB เช่น (1,0.5,0)

r g b c m y k w

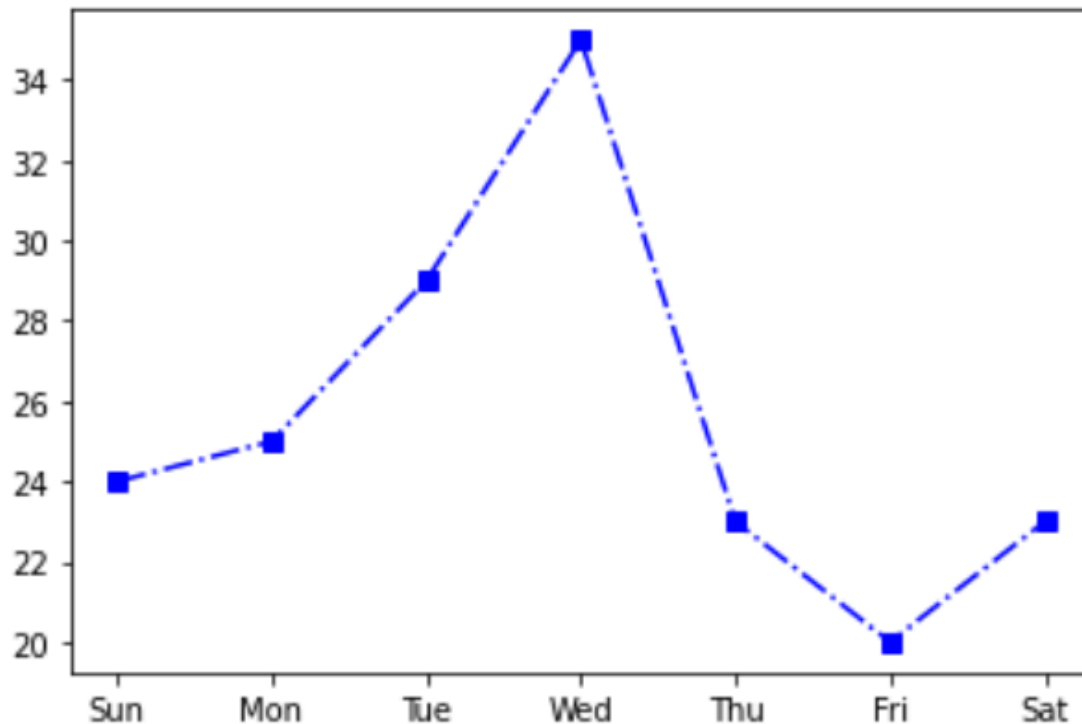
Matplotlib – กำหนดแบบย่อ

10

สามารถย่อการกำหนด marker line color ได้พร้อมกันตามตัวอย่าง

```
plt.plot(x,y, 's-.b' )
```

```
[<matplotlib.lines.Line2D at 0x1d014520760>]
```



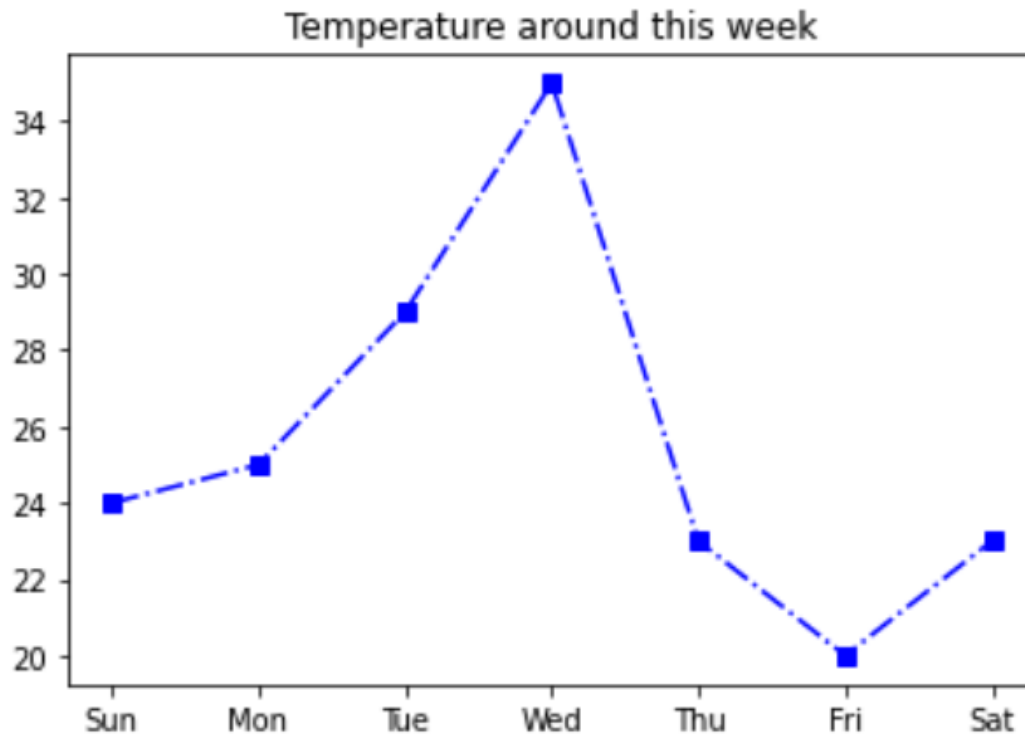
Matplotlib – Title

11

Title คือการกำหนดหัวเรื่องของกราฟ ดังตัวอย่าง

```
plt.plot(x,y,'s-.b' )  
plt.title('Temperature around this week')
```

```
Text(0.5, 1.0, 'Temperature around this week')
```

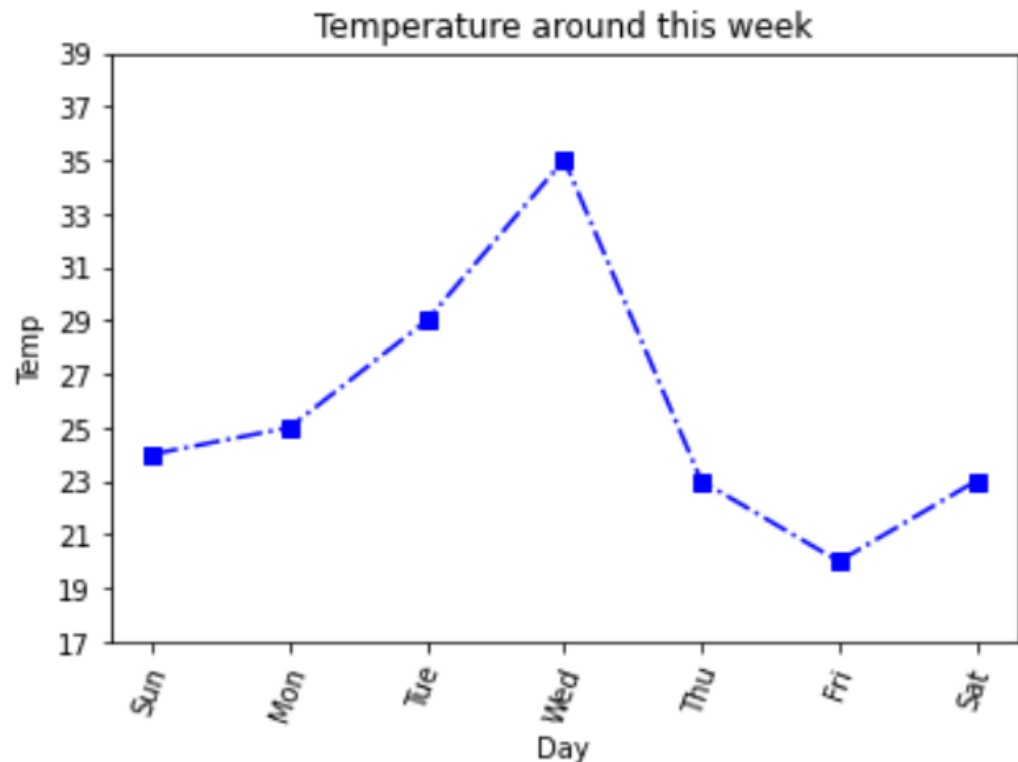


Matplotlib – label และ tick

12

xlabel หรือ ylabel เป็นการกำหนดชื่อของแกน
xticks หรือ yticks เป็นการกำหนดลำดับของข้อมูล

```
plt.plot(x,y,'s-.b' )  
plt.title('Temperature around this week')  
plt.xlabel('Day')  
plt.ylabel('Temp')  
plt.xticks(rotation=70)  
plt.yticks(np.arange(17,41,2))  
plt.show()
```



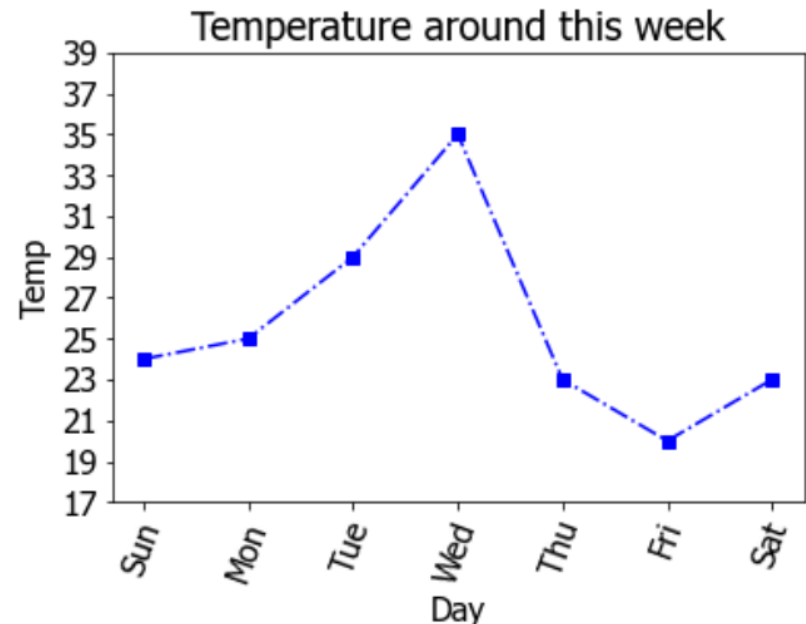
Matplotlib – font

13

สามารถกำหนดขนาดข้อความและ font ได้ แต่ว่าจะกระทบกับทุกกราฟเพราะเป็นการกำหนดตัวแปรรวมของ plt

```
plt.rcParams['font.family'] = 'sans-serif'  
plt.rcParams['font.sans-serif'] = 'Tahoma'  
plt.rcParams['font.size'] = 16
```

```
plt.rcParamsdefaults()
```



Matplotlib – อื่นๆที่ควรรู้

14

กำหนดขนาดของภาพ

การเซฟภาพ

```
plt.figure(figsize=[10,7])  
plt.plot(x,y, 's-.b' )  
plt.title('Temperature around this week')  
plt.savefig('temp.png')  
plt.show()
```

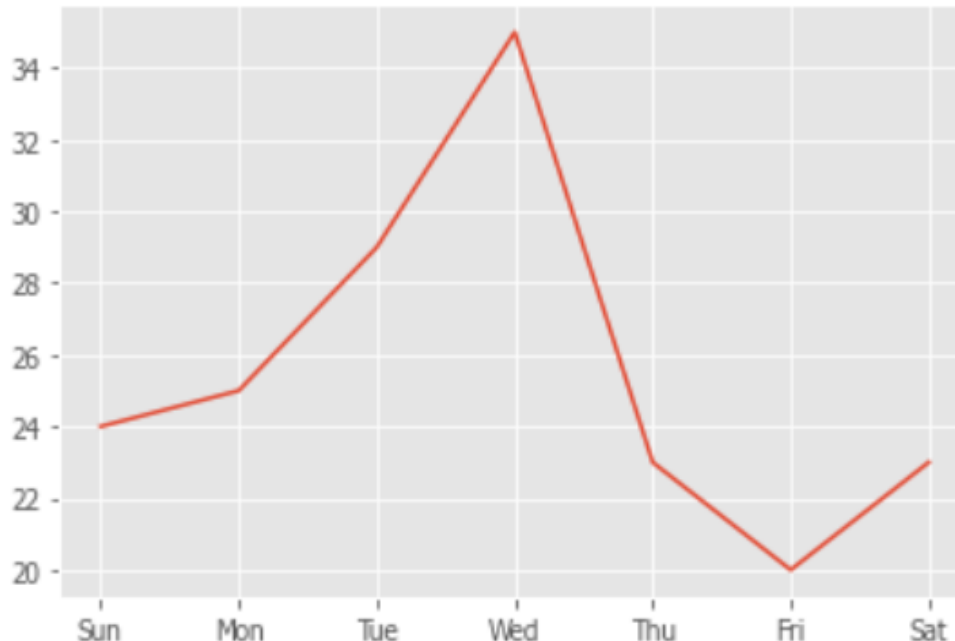
Matplotlib – อื่นๆที่ควรรู้

15

กำหนดรูปแบบกราฟ
ที่มีให้อยู่แล้ว

```
print(plt.style.available)
plt.style.use('ggplot')
plt.plot(x,y)
plt.show()
```

```
['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic',  
e', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-  
p', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-  
'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```



```
plt.style.use('default')
```

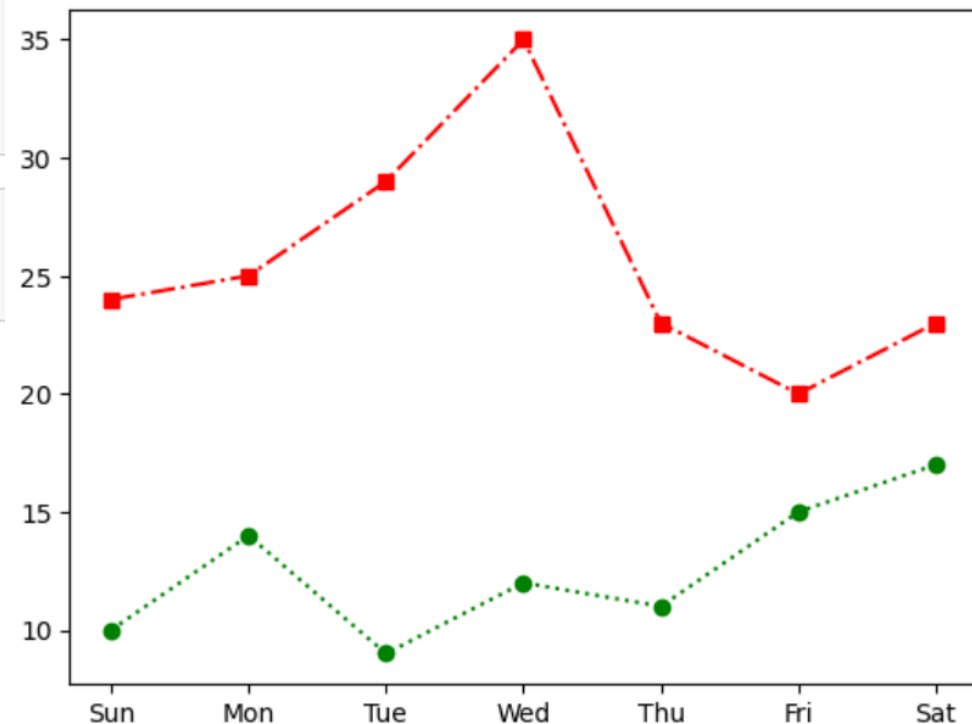
Matplotlib – Multiline

16

สามารถเขียนเส้นกราฟพร้อมกันได้หลายเส้นในกราฟเดียว

```
dic = {"day": ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'] ,  
       "temp1": [24, 25, 29, 35, 23, 20, 23] ,  
       "temp2": [10, 14, 9, 12, 11, 15, 17]}  
df = pd.DataFrame(dic)  
x = df['day']  
y = df['temp1']  
z = df['temp2']
```

```
plt.plot(x,y,'s-.r')  
plt.plot(x,z,'o:g')
```



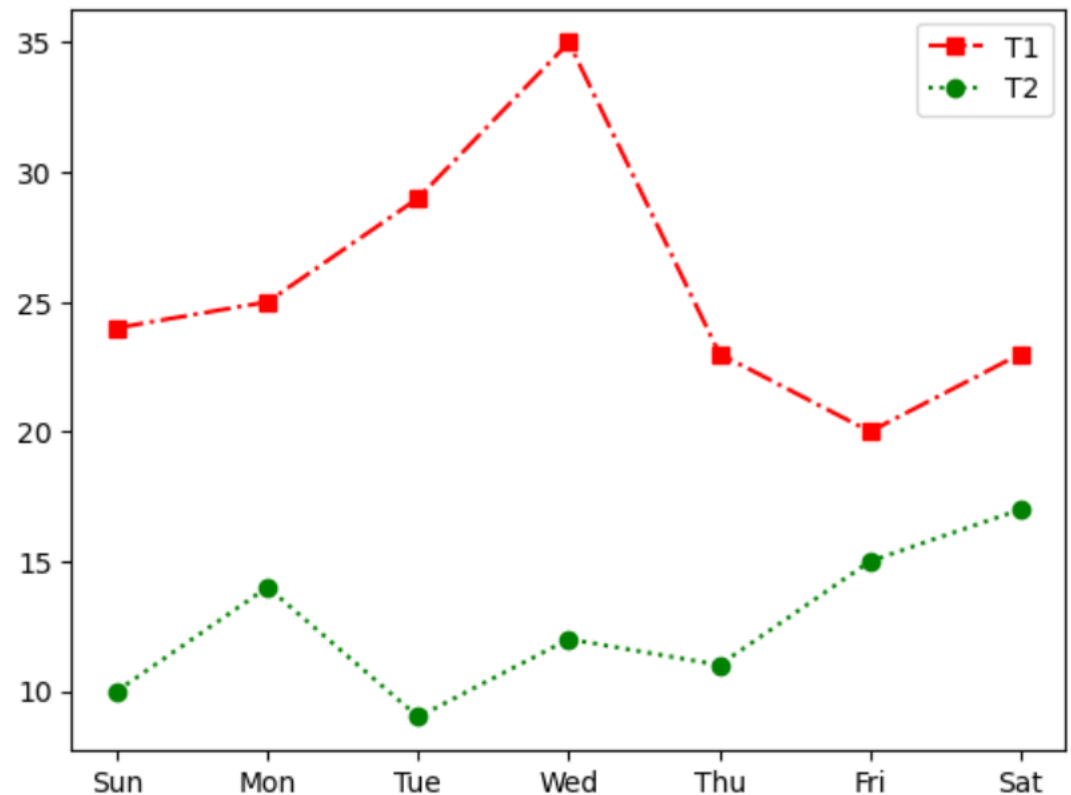
Matplotlib – legend()

17

กรณีกราฟมีหลายเส้นให้กำหนดชื่อด้วย label
และกำหนดตำแหน่งกรอบสรุป legend

```
plt.plot(x,y,'s-.r',label='T1')  
plt.plot(x,z,'o:g',label='T2')  
plt.legend(loc='best')
```

right
left
upper
lower
center



Matplotlib – MultiGraph

18

สามารถเขียนกราฟพร้อมกันได้หลายกราฟ

แบ่งออกเป็น 2 วิธีหลักๆ

1. แบ่งเป็น row , col

2. วาดอิสระ (กำหนดตำแหน่งด้วย x,y และขนาดภาพ)

Matplotlib – MultiGraph

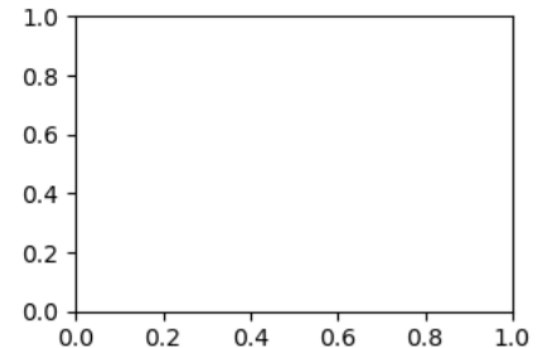
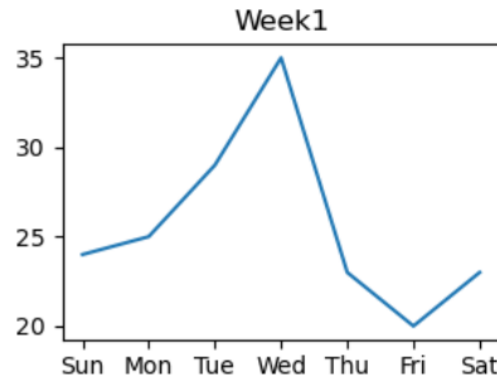
19

1. แบ่งเป็น row , col

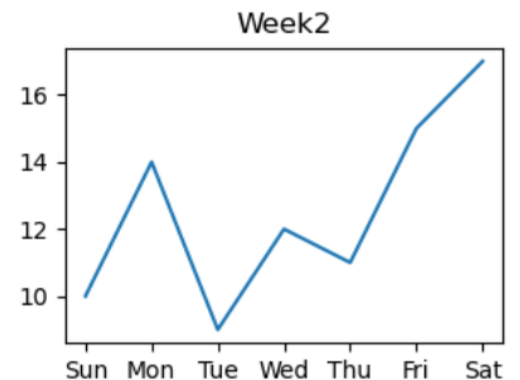
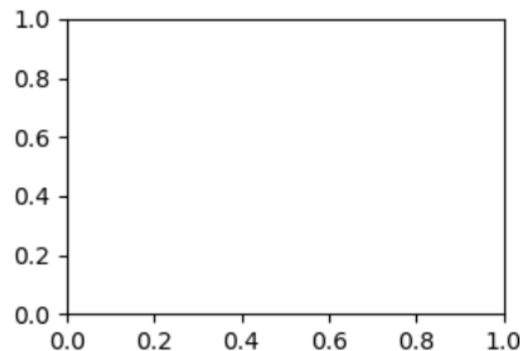
```
fig , axes = plt.subplots(nrows = 2 , ncols = 2)
axes[0,0].plot(x,y)
axes[0,0].set_title('Week1')

axes[1,1].plot(x,z)
axes[1,1].set_title('Week2')

fig.tight_layout()
plt.show()
```



ถ้ามี 1 มิติ ให้เข้าถึงด้วยมิติเดียว
axes[0]



Matplotlib – MultiGraph

20

2. วาดอิสระ

```
fig = plt.figure()
axes1 = fig.add_axes([0,0,1,1])
axes1.plot(x,y,'r--')
axes1.set_title('Axes 1')
axes1.set_xlabel('X Axis')
axes1.set_ylabel('Y Axis')

axes2 = fig.add_axes([1.2,0,1,1])
axes2.plot(x,z)
axes2.set_title('Axes 2')
```

