

Matplotlib

- ไลบรารีที่นิยมใช้มากที่สุดในการพลอตกราฟสองมิติจาก array
- มีจุดเริ่มต้นมากจากการจำลองคำสั่งพวงกราฟพีคของ MATLAB
- สามารถแสดงผลข้อมูลได้อย่างรวดเร็วและสามารถบันทึกผลที่ได้ออกมาเป็นรูปภาพได้หลายรูปแบบ

สิ่งที่ต้องการก่อนใช้งาน

- ก่อนที่จะใช้งาน Matplotlib ได้นั้นเราต้องติดตั้งก่อน โดย
- เข้าไปที่โฟลเดอร์ c:\python34\scripts
- พิมพ์คำสั่ง pip install matplotlib

```
C:\Python34\Scripts>pip install matplotlib
You are using pip version 6.0.8, however version 7.1.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Collecting matplotlib
  Downloading matplotlib-1.4.3-cp34-none-win32.whl (3.8MB)
    100% |#####| 3.8MB 93kB/s
Collecting pyparsing>=1.5.6 (from matplotlib)
  Downloading pyparsing-2.0.3-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.6 in c:\pytho
n34\lib\site-packages (from matplotlib)
Collecting six>=1.4 (from matplotlib)
  Downloading six-1.9.0-py2.py3-none-any.whl
Collecting pytz (from matplotlib)
  Downloading pytz-2015.4-py2.py3-none-any.whl (475kB)
    100% |#####| 475kB 415kB/s
Collecting python-dateutil (from matplotlib)
  Downloading python_dateutil-2.4.2-py2.py3-none-any.whl (188kB)
    100% |#####| 192kB 710kB/s
Installing collected packages: python-dateutil, pytz, six, pyparsing, matplotlib

Successfully installed matplotlib-1.4.3 pyparsing-2.0.3 python-dateutil-2.4.2 py
tz-2015.4 six-1.9.0
C:\Python34\Scripts>
```

ใน python มีไลบรารี pylab ที่รวมฟังก์ชันที่สำคัญในการคำนวณด้านคณิตศาสตร์และพลอตกราฟมาไว้ในโมดูลเดียวกัน โดยเราจะ import ในรูปแบบด้านล่าง เพื่อดึงเอาฟังก์ชันเหล่านั้นออกมาไว้ในเนมสเปซโกลบอล ทำให้เรียกใช้งานได้ทันที

```
from pylab import *
```

เทียบเท่ากับเรียก

```
from numpy import *
```

```
from scipy import *
```

```
from matplotlib import *
```

หลักการทำงาน

ในการวาดกราฟเราจะใช้คำสั่ง `plot(a,b)` โดยที่ `a` และ `b` เป็นชุดของข้อมูล เช่น `array` หรือ `list`

ข้อมูล `a` จะอยู่ตามแกน `x` และข้อมูล `b` จะอยู่ตามแกน `y`

กราฟที่โปรแกรมวาดจะคือการจับคู่กันของสมาชิกแต่ละตัวของ `a` และ `b`

เมื่อเรียกฟังก์ชัน `plot()` แล้วจะยังไม่มีการแสดงผลจนกว่าจะมีการเรียกฟังก์ชัน `show()`

ตัวอย่าง

ตัวอย่างความสัมพันธ์กำลังสองและแผนภูมิพาราโบลา

```
from pylab import *
```

```
x = arange(-5,5)
```

```
[-5 -4 -3 -2 -1  0  1  2  3  4]  
[25 16  9  4  1  0  1  4  9 16]
```

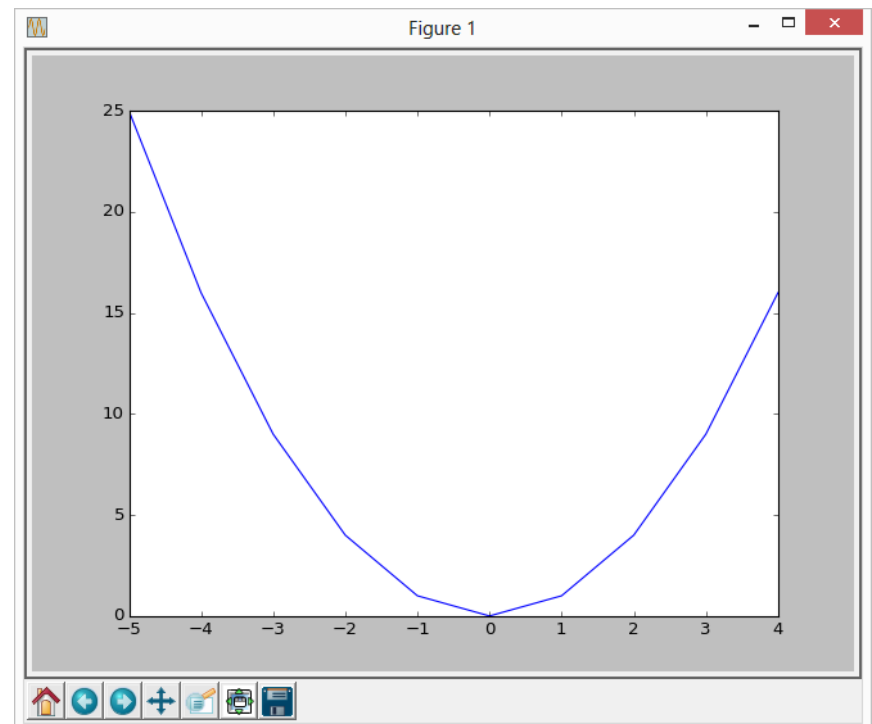
```
print(x)
```

```
y = x**2
```

```
print(y)
```

```
plot(x,y)
```

```
show()
```



```
from pylab import *
```

เป็นการเรียกฟังก์ชันจากโมดูล pylab

```
x = arange(-5,5)
```

ฟังก์ชัน arange() เป็นฟังก์ชันใน pylab ที่คืนค่าเป็น array

```
plot(x,y)
```

เป็นการนำเอา x และ y มาสร้างแผนภูมิเส้นจากความสัมพันธ์ $y = x^2$

```
show()
```

เป็นคำสั่งในการแสดงกราฟที่วาดจากคำสั่ง plot()

ตัวอย่างการเพิ่มเส้นกริดและข้อความบนแกน x y

```
from pylab import *
```

```
x = arange(-5,5)
```

```
y = x**2
```

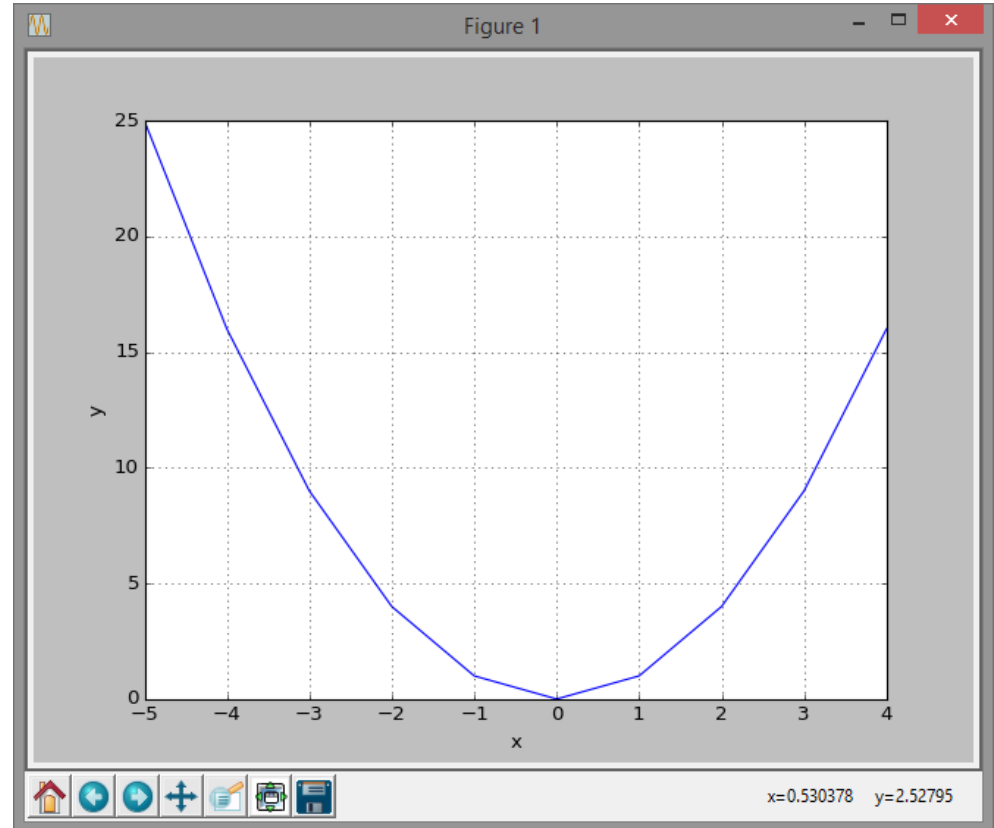
```
plot(x,y)
```

```
grid(True)
```

```
xlabel('x')
```

```
ylabel('y')
```

```
show()
```



ตัวอย่างนี้เพิ่มเส้นกริดแสดงช่วงจำนวนและคำอธิบายแกน x และ y

ตัวอย่างการมีกราฟหลายเส้น

```
from pylab import *
```

```
x = arange(-5,5)
```

```
y1 = x**2
```

```
y2 = 2*x
```

```
plot(x,y1)
```

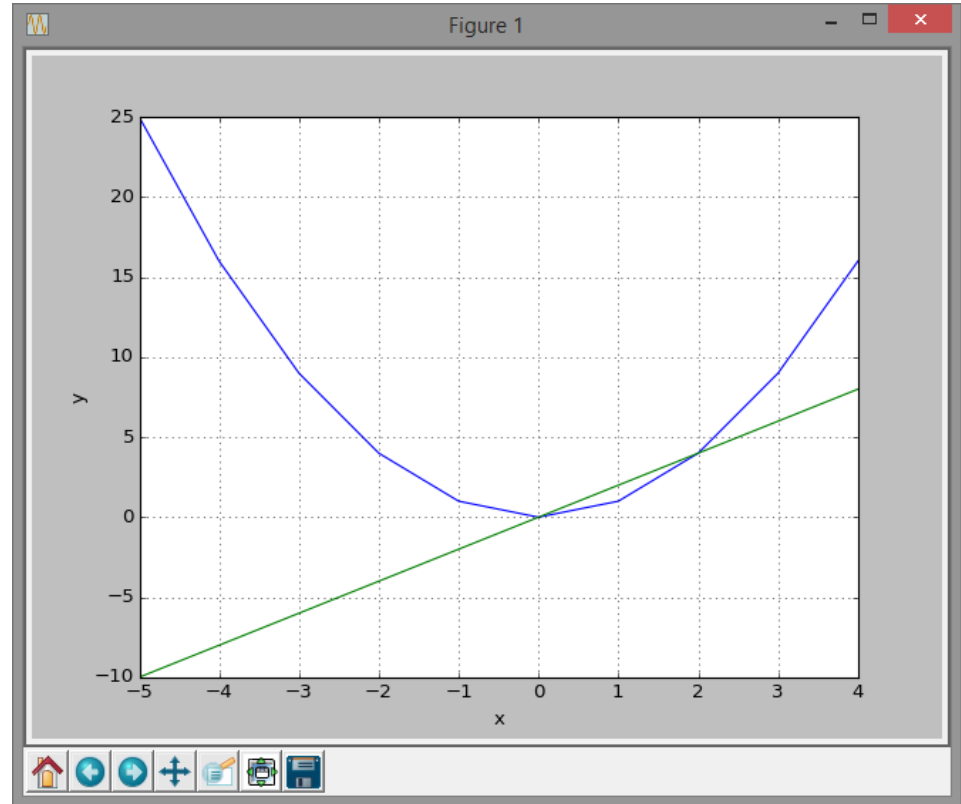
```
plot(x,y2)
```

```
grid(True)
```

```
xlabel('x')
```

```
ylabel('y')
```

```
show()
```



คำสั่ง `plot(x,y2)` เป็นการสั่งให้ python วาดเส้นเพิ่มลงไปอีกเส้น

ตัวอย่าง

```
from pylab import *
```

```
x = randn(10000)
```

```
hist(x, 100)
```

```
show()
```

เป็นการสุ่มตัวเลข 10000 ตัว ให้กับ array x จากนั้นทำการพลอต histogram 100 ช่วงจากข้อมูล x ที่ได้รับ แล้วแสดงผล

