



# 컴퓨터프로그래밍1

Matplot의 기초

---

천양하

# 목차

- Matplot 개요
- Matplot 설치
- Matplotlib 활용

# Matplot 라이브러리 개요

## ■ 그래프나 그림을 그릴 수 있는 플라팅(plotting) 라이브러리

- 파이썬에서 데이터를 차트나 플롯(Plot)으로 그려주는 라이브러리 패키지로서 가장 많이 사용되는 데이터 시각화(Data Visualization) 패키지로 알려져 있다.
- 파이썬에서 제공하는 함수들을 사용하면 매트랩(Matlab)처럼 다룰 수 있으며 제공되는 기능이나 문법(syntax)도 비슷함
- 파이썬에서 Matplot의 기능을 인터랙티브하게 사용할 수 있음
- Pyplot 모듈을 사용하여 데이터를 시각적으로 표현
- Matplot과 인터페이스 역할을 하는 것으로 상태 머신(state machine)이라고도 함

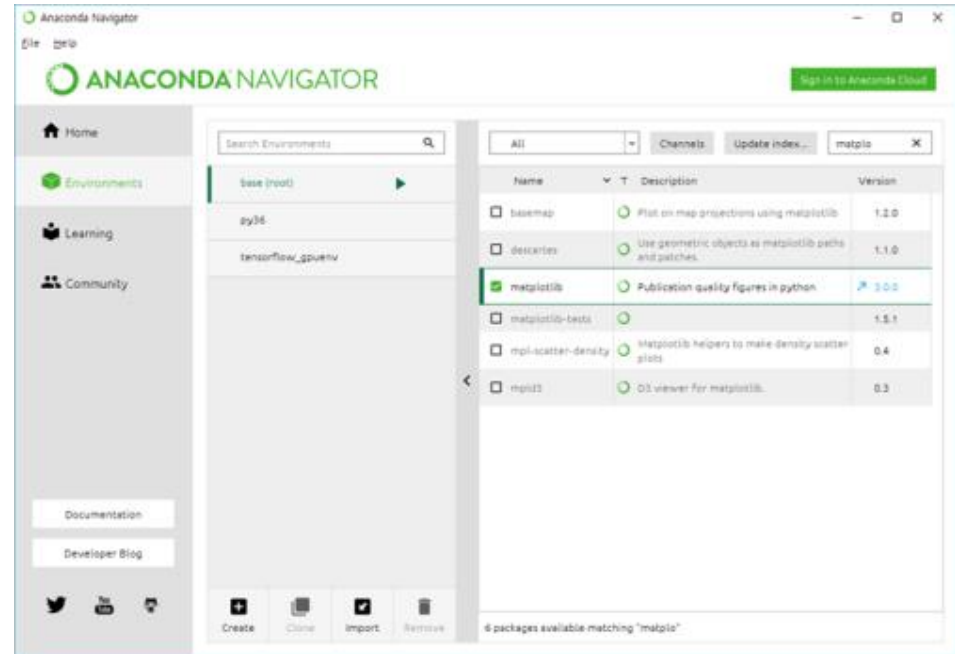
# Matplotlib 설치

## ■ 아나콘다 내비게이터

- Environments 탭 선택
- All 선택하고 matplotlib 검색
- matplotlib 체크하고 설치

## ■ Colab창

- 명령문 실행



명령문	<code>pip list</code> # 설치된 라이브러리들을 보여줌
명령문	<code>pip install matplotlib</code> #라이브러리가 없다면 설치실행 <code>import matplotlib.pyplot as plt</code> #판다스 라이브러리 불러오기

# Matplotlib : 직선 그래프

- 이름이 matplotlib.pyplot으로 너무 길어 plt로 대체
- 그래프를 그리기 전에 다음 명령을 삽입하면 한 번의 실행으로 그 그래프의 출력을 볼 수 있음

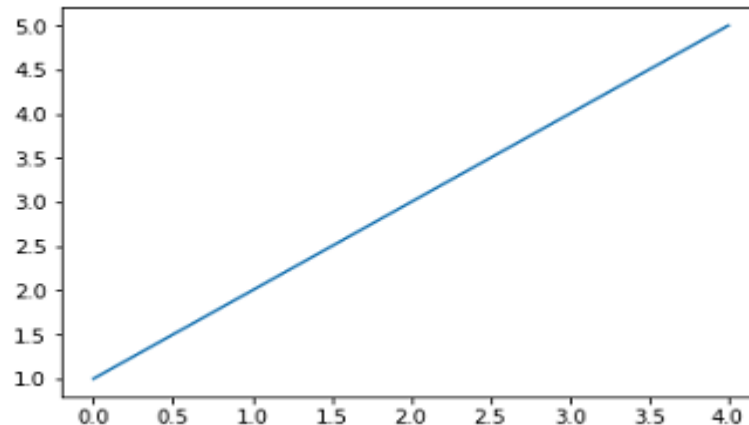
명령문

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

```
y = [13, 16, 15, 18, 16, 17, 16]
```

```
plt.plot(y)
```

```
plt.show( )
```

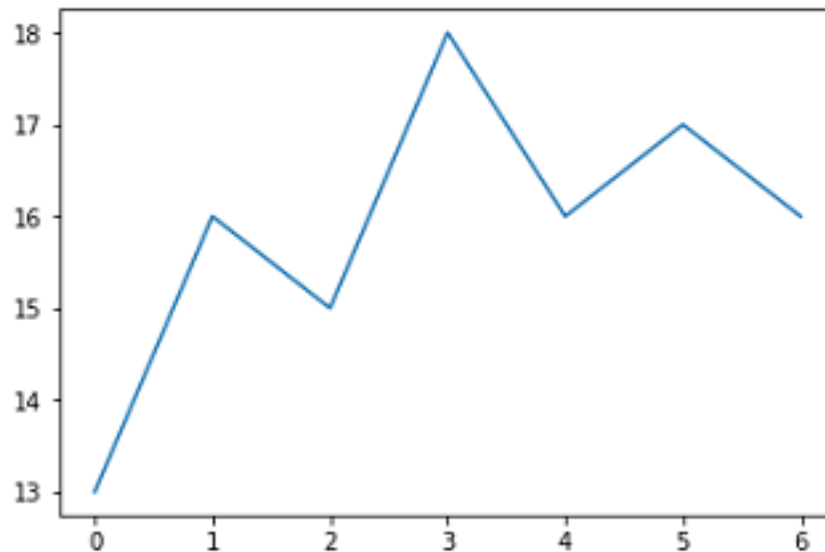


# Matplotlib : 직선 그래프

- **plot() 메소드의 입력 파라미터의 개수가 2개인 경우**
  - 첫 번째 항목은 x축 좌표값 : `range(len(y))`는 `range(7)` 이므로 0에서 6까지의 리스트에 해당
  - 두 번째 항목은 y축 좌표값 : y의 값

명령문

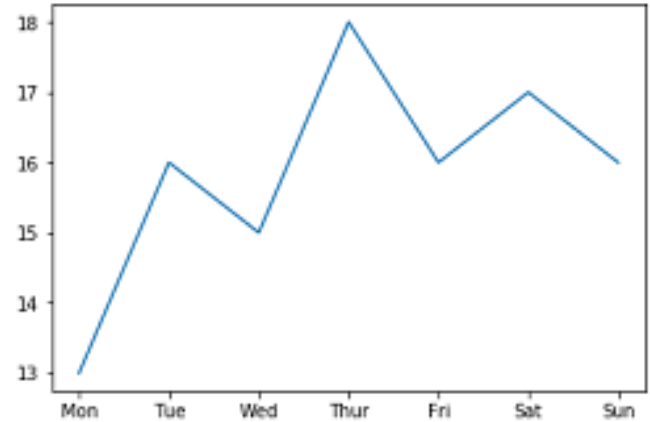
```
y = [13, 16, 15, 18, 16, 17, 16]  
plt.plot(range(len(y)), y)
```



# Matplotlib : 직선 그래프

## ■ x축에 의미 있는 문자(열) 넣기

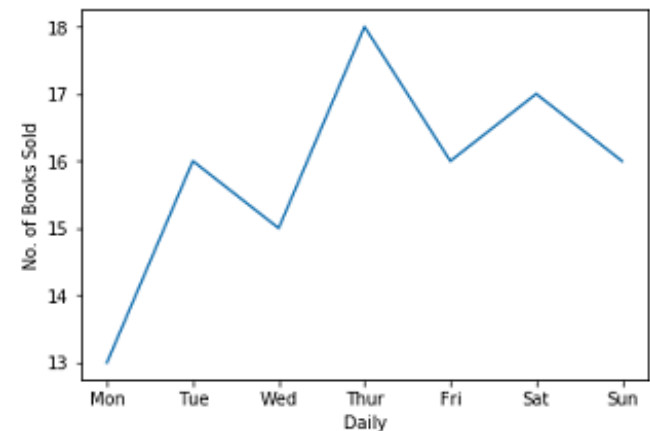
명령문	<pre>x = ['Mon','Tue','Wed','Thur','Fri','Sat','Sun'] y = [13, 16, 15, 18, 16, 17, 16] plt.plot(x, y) plt.show( )</pre>
-----	---



## ■ 레이블(label) 입력

- xlabel(): x축의 레이블 입력
- ylabel(): y축의 레이블 입력

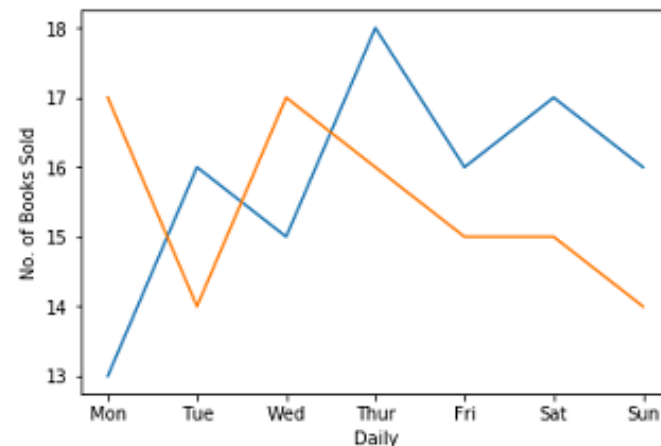
명령문	<pre>x = ['Mon','Tue','Wed','Thur','Fri','Sat','Sun'] y = [13, 16, 15, 18, 16, 17, 16] plt.plot(x, y) plt.xlabel('Daily') plt.ylabel('No. of Books Sold') plt.show( )</pre>
-----	---



# Matplotlib : 직선 그래프

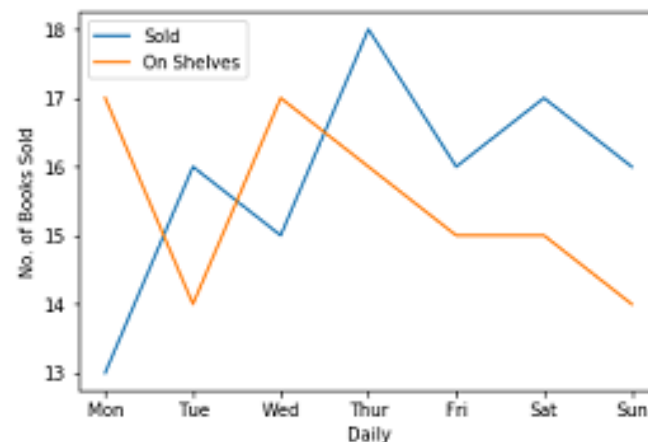
## ■ 그래프 1개 더 추가

명령문	<pre>x = ['Mon','Tue','Wed','Thur','Fri','Sat','Sun'] y1 = [13, 16, 15, 18, 16, 17, 16] y2 = [17, 14, 17, 16, 15, 15, 14] plt.plot(x, y1) plt.plot(x, y2) plt.xlabel('Daily') plt.ylabel('No. of Books Sold') plt.show()</pre>
-----	--



## ■ 레전드(legend) 붙이기

명령문	<pre>x=['mon', 'Tue','Web','Thur','Fri','Sat','Sun'] y1=[13,16,15,18,16,17,16] y2=[17,14,17,16,15,15,14] plt.plot(x, y1) plt.plot(x, y2) plt.xlabel('Daily') plt.ylabel('No. of Books Sold') plt.legend(['Sold','On Shalves'], loc="upper left")</pre>
-----	--



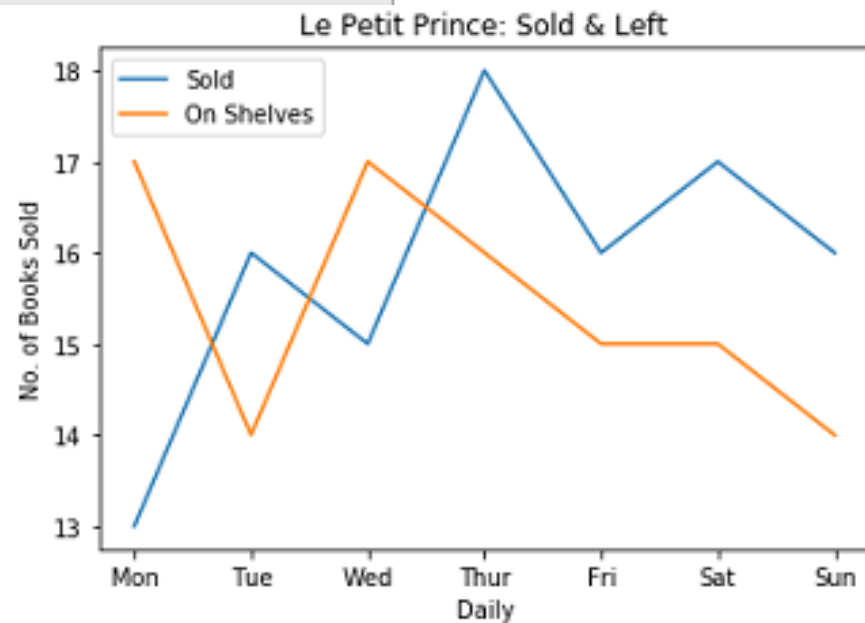


# Matplotlib : 직선 그래프

- 제목 넣기 : title

명령문

```
x = ['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']  
y1 = [13, 16, 15, 18, 16, 17, 16]  
y2 = [17, 14, 17, 16, 15, 15, 14]  
plt.plot(x, y1, label='Sold')  
plt.plot(x, y2, label='On Shelves')  
plt.xlabel('Daily')  
plt.ylabel('No. of Books Sold')  
plt.legend(loc="upper left")  
plt.title('Le Petit Prince: Sold & Left')
```



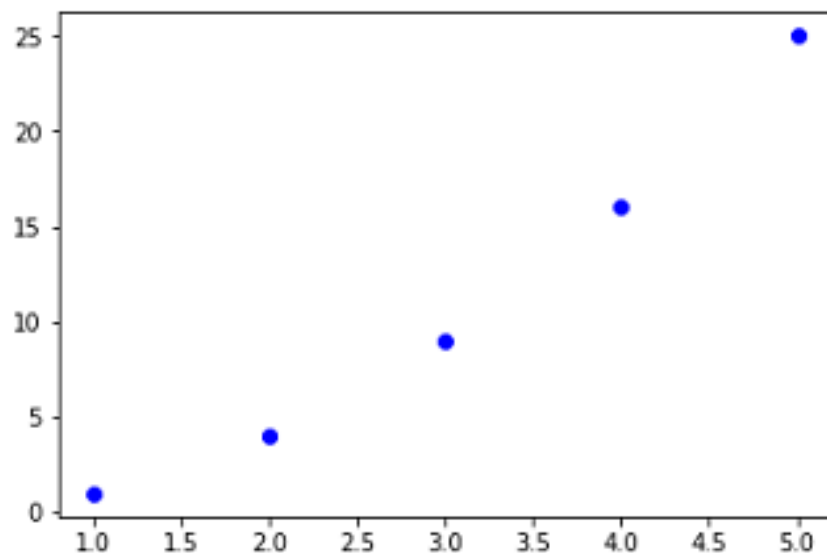
# Matplotlib : 점선 그래프

- **'bo' 옵션**

- b(Blue) 색상의 'o' 모양 점 표시

명령문

```
x = [1, 2, 3, 4, 5]  
y = [1, 4, 9, 16, 25]  
plt.plot(x, y, 'bo')  
plt.show()
```

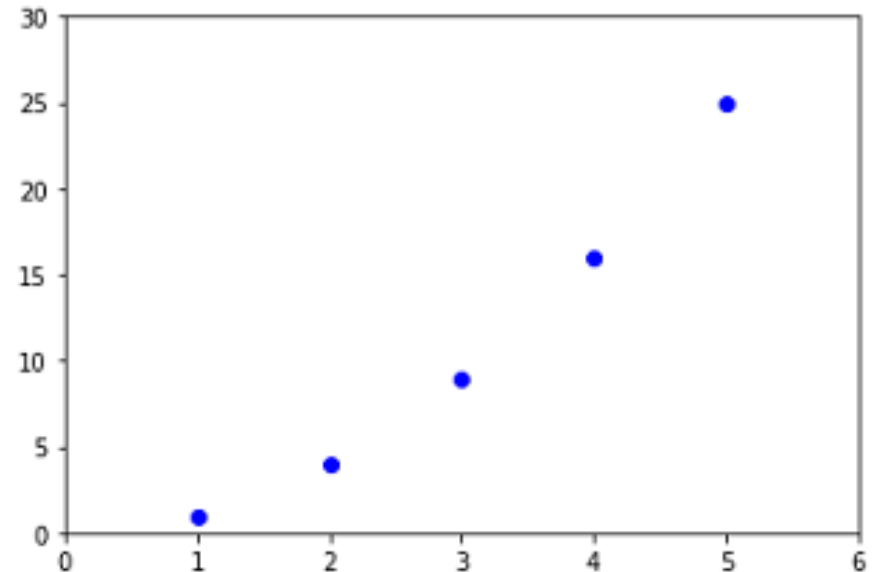


# Matplotlib : 점선 그래프

- **axis() 메소드 : 그래프의 여백을 할당**
  - **x축과 y축 범위 지정**  
**[xmin, xmax, ymin, ymax]**

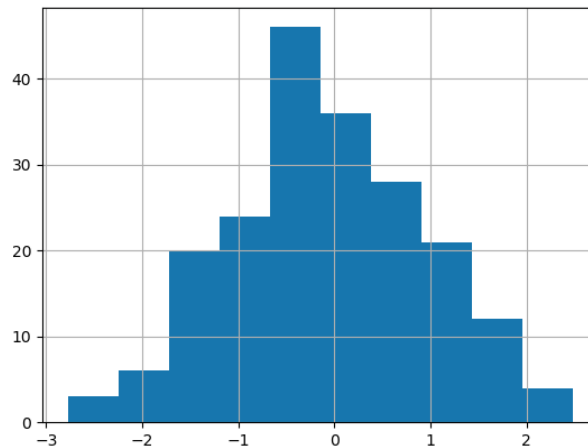
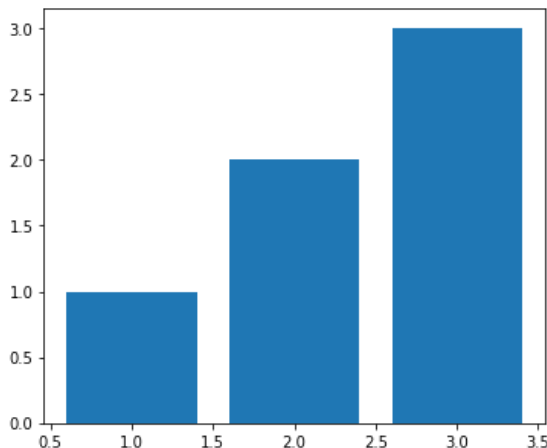
명령문

```
x = [1, 2, 3, 4, 5]  
y = [1, 4, 9, 16, 25]  
plt.plot(x, y, 'bo')  
plt.axis([0, 6, 0, 30])  
plt.show()
```



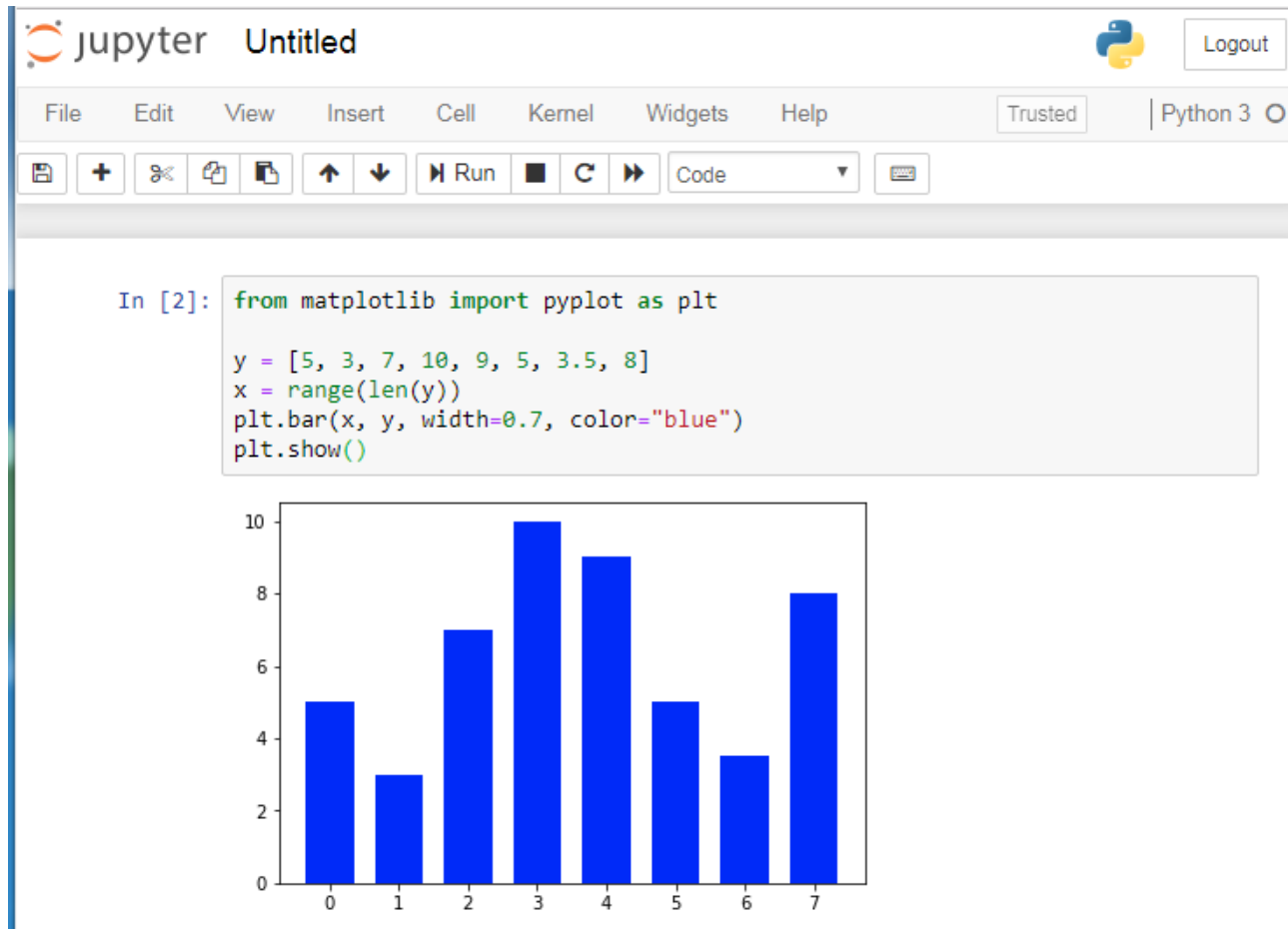
# Matplotlib : 다양한 그래프 함수

- **Matplotlib**는 위에서 예시한 라인 플롯 이외에 여러 다양한 차트/플롯을 그릴 수 있는데,
- 각 차트/플롯마다 다른 함수들을 사용한다.
- 예를 들어, 막대 차트를 그리기 위해서는 **plt.bar()** 함수를 사용하고, **Pie** 차트를 그리기 위해서는 **plt.pie()**를, 히스토그램을 그리기 위해선 **plt.hist()** 함수를 사용한다.



# Matplotlib : 막대 그래프

- Bar 차트 예시 : plt.bar()를 호출하여 막대 차트를 그리고 있다



# Matplotlib : plot의 다양한 서식에 관한 메소드

서식	메소드
legend 추가	legend( )
축교정	axis( )
축의 레이블 설정	xlabel( ), ylabel( ),
그림 저장	savefig( )



**THANK YOU FOR  
YOUR ATTENTION**

---