

Matplotlib

Matplotlib 기본 사용

Matplotlib 라이브러리를 이용해서 그래프를 그리는 일반적인 방법

Pyplot 소개

matplotlib.pyplot은

Matplotlib을 MATLAB과 비슷하게 동작하도록 하는 명령어 스타일의 함수의 모음

각각의 pyplot 함수를 사용해서 그림(figure)에 변화를 줄 수 있다

예를 들어,
그림을 만들어서 플롯 영역을 만들고
몇개의 라인을 플롯하고
라벨(label)들로 꾸미는 등의 일을 할 수 있다.

기본그래프

pyplot으로 어떤 값들을 시각화하는 것은 매우 간단.

pyplot.plot()에 하나의 리스트를 입력함으로써 그래프가 그려진다.

matplotlib은

리스트의 값들이 y 값들이라고 가정하고,
x 값들([0, 1, 2, 3])을 자동으로 만들어낸다.

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4])
plt.ylabel('y-label')
plt.show()
```

plot()은 다재다능한 (versatile)한 명령어여서,
임의의 개수의 인자를 받을 수 있다.

예를 들어, 아래와 같이 입력하면, x와 y값을 그래프로 나타낼 수 있다.

```
plt.plot([1,2,3,4], [1,4,9,16], 'ro')

# axis()를 이용해서 축의 [xmin, xmax, ymin, ymax] 범위를 지정.
plt.axis([0, 6, 0, 20])
plt.show()
```

여러 개의 그래프 그리기

matplotlib에서 리스트만 가지로 작업하는 것은 제한적이기 때문에,
일반적으로 Numpy 어레이를 이용.

사실, 모든 시퀀스는 내부적으로 Numpy 어레이로 변환된다.

다양한 포맷 스타일의 여러개의 라인을 하나의 그래프로 그리기.

```
# 다양한 포맷 스타일의 여러개의 라인을 하나의 그래프로 그리기.
import matplotlib.pyplot as plt
import numpy as np

# 200ms 간격으로 균일한 샘플된 시간
t = np.arange(0., 5., 0.2)

# 빨간 대쉬, 파란 사각형, 녹색 삼각형
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```

Matplotlib 라벨 설정 하기

xlabel(), ylabel() 함수를 사용해서
그래프의 x, y축에 대한 라벨을 설정할 수 있다.

```
plt.plot([1, 2, 3, 4], [1, 4, 5, 16])

# xlabel() 과 ylabel() 에 텍스트를 입력해주면, 각각의 축에 라벨이 나타난다.
plt.xlabel('X-Label')
```

```
plt.ylabel('Y-Label')
plt.axis([0, 5, 0, 20])
plt.show()
```

axis() 에 [xmin, xmax, ymin, ymax] 의 형태로
x, y 축의 범위를 지정

입력 리스트는
꼭 네개의 값(xmin, xmax, ymin, ymax)이 있어야 한다.

입력값이 없으면
데이터에 맞게 자동(Auto)으로 범위를 지정

Matplotlib 색깔 지정하기

자주 사용하는 색깔 외에도 다양한 생상을 지정할 수 있다.

plot() 에 color = 'springgreen' 과 같이 입력해주면,
springgreen에 해당하는 색깔이 지정된다.

```
plt.plot([1, 2, 3, 4], [1, 4, 5, 16], color='#e35f62',
         marker='o', linestyle='--')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.axis([0, 5, 0, 20])
plt.show()
```

Matplotlib 여러 곡선 그리기

세개의 곡선을 하나의 그래프에 그리기

Numpy를 사용해서 array를 생성.

numpy.arange()

주어진 간격에 따라 균일한 array를 생성한다.

```
import matplotlib.pyplot as plt
import numpy as np

a = np.arange(5)
b = np.arange(1, 5)
```

```
c = np.arange(2, 10, 2)

print(a)
print(b)
print(c)
```

plot()에 x값, y값, 스타일을 순서대로 세번씩 입력하면,
세개의 곡선 ($y=x$, $y=x^2$, $y=x^3$)이 동시에 그려진다.

```
plt.plot(a, a, 'r--',
         a, a**2, 'bo',
         a, a**3, 'g-.')
```

세세한 스타일 설정

```
plt.plot(a, a, 'r--',
         a, a**2, 'bo',
         a, a**3, 'g-.')
```

세개의 곡선에 세세한 스타일을 설정할 수 있다.

```
a = np.arange(0, 2, 0.2)
```

첫번째 곡선의 스타일은 'bo'로

```
plt.plot(a, a, 'bo')
```

두번째 곡선은 color='#e35f62', marker='*', linewidth=2로

```
plt.plot(a, a**2, color='#e35f62', marker='*', linewidth=2)
```

세 번째 곡선은 color='springgreen', marker='^', markersize=9로

```
plt.plot(a, a**3, color='springgreen', marker='^', markersize=9)
```

Matplotlib 그리드와 틱 설정하기

grid() 와 tick_params() 를 이용해서

그래프의 그리드와 틱의 그타일을 설정 할 수 있다.

```
a = np.arange(0, 2, 0.2)

plt.plot(a, a, 'bo')
plt.plot(a, a**2, color='#e35f62', marker='*', linewidth=2)
plt.plot(a, a**3, color='springgreen', marker='^', markersize=9)
```

그리드가 표시되도록 하려면,
grid() 의 첫번째 파라미터를 True로 설정.

axis='y' 로 설정하면 y축의 그리드만 표시.

alpha 는 투명도를 설정합니다.
0으로 설정하면 투명하게,
1은 불투명하게 표시.

linestyle을 대쉬(Dashed)로 설정.

```
plt.grid(True, axis='y', color='gray', alpha=0.5, linestyle='--')
```

tick_params() 를 이용해서 그래프의 틱(Tick)에 관련된 설정을 할 수 있다.

axis='both' 로 설정하면 x, y축의 틱에 모두 적용.

direction='in' 으로 틱의 방향을 그래프 안쪽으로 설정.

틱의 길이(length)를 3만큼으로 하고,
틱과 라벨의 거리(pad)를 6만큼.
틱 라벨의 크기(labelsize)를 14로 설정.

```
plt.tick_params(axis='both', direction='in', length=3, pad=6, labelsz=14)

plt.show()
```

Matplotlib 타이틀 설정하기

title() 을 이용해서 그래프의 제목(타이틀)을 설정.

plt.title()을 이용해서 그래프의 타이틀을 'Sample graph'로 설정.

```
plt.title('Sample graph')
plt.show()
```

2 - 위치와 오프셋

```
plt.title('Sample graph', loc = 'right', pad=20)
```

loc = 'right'로 설정하면,
타이틀이 그래프의 오른쪽 위에 나타나게 된다.

'left', 'center', 'right'로 설정할 수 있으며
디폴트는 'center'

pad = 20 은
타이틀과 그래프와의 간격(오프셋)을 포인트 단위로 설정.

3 - 폰트 설정

fontdict에 딕셔너리 형태로 폰트에 대한 설정을 입력할 수 있다.
'fontsize'를 16으로, 'fontweight'를 'bold'로 설정.

'fontsize'는 포인트 단위의 숫자를 입력하거나,
'smaller', 'x-large' 등의 상대적인 설정을 할 수 있다.

'fontweight'에는 'normal', 'bold', 'heavy', 'light', 'ultrabold', 'ultralight'의
설정을 할 수 있다.

```
title_font = {
    'fontsize':16,
    'fontweight':'bold'
}
plt.title('Sample graph',fontdict=title_font, loc = 'left', pad=20)
```