

1. Graph라는 클래스를 만들어 cos, sin, tan, log 함수를 각각 화면에 출력하는 메서드를 구현하시오. (class와 math, matplotlib 활용)

- 코드

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

# Graph 클래스
class Graph:
    def __init__(self):
        plt.figure(figsize=(8, 5))
        # y값을 저장할 리스트
        self.y = []

    def sin(self):
        x = np.arange(0, 6, 0.1)
        # 함수에 x를 입력하여 결과값 y 계산 후 리스트에 추가
        for i in x:
            y = math.sin(i)
            self.y.append(y)
        # matplotlib을 이용하여 함수 출력
        plt.xlabel('x'), plt.ylabel('y')
        plt.title("sin"), plt.plot(x, self.y)

    def cos(self):
        x = np.arange(0, 6, 0.1)
        for i in x:
            y = math.cos(i)
            self.y.append(y)
        plt.xlabel('x'), plt.ylabel('y')
        plt.title("cos"), plt.plot(x, self.y)

    def tan(self):
        x = np.linspace(-1.3, 1.3)
        for i in x:
            y = math.tan(i)
            self.y.append(y)
        plt.xlabel('x'), plt.ylabel('y')
        plt.title("tan"), plt.plot(x, self.y)

    def log(self):
        x = np.arange(0.001, 3, 0.001)
        for i in x:
            y = math.log(i)
            self.y.append(y)
        plt.xlabel('x'), plt.ylabel('y')
        plt.title("log"), plt.plot(x, self.y)

# 사용자로부터 함수 입력받기
Function = input("Enter sin cos tan log : ")

# 각 입력에 맞게 클래스의 함수 실행
graph = Graph()
if Function == "sin":
    graph.sin()
elif Function == "cos":
    graph.cos()
elif Function == "tan":
    graph.tan()
elif Function == "log":
    graph.log()
else:
    print("Enter sin cos tan log")

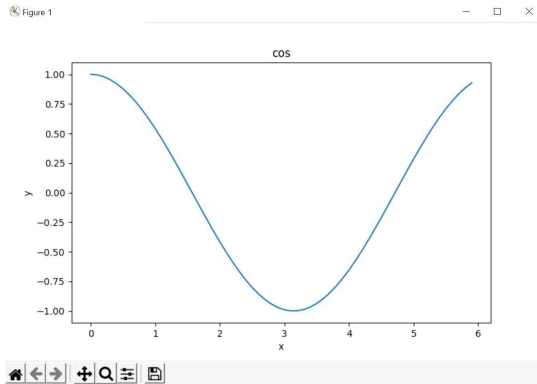
plt.show()
```

math 라이브러리를 이용하여 함수 구현 시
입력값으로 list가 아닌 숫자 하나가 필요하지만,
matplotlib 라이브러리를 이용하여 그래프 출력 시
x, y에 해당하는 list 필요

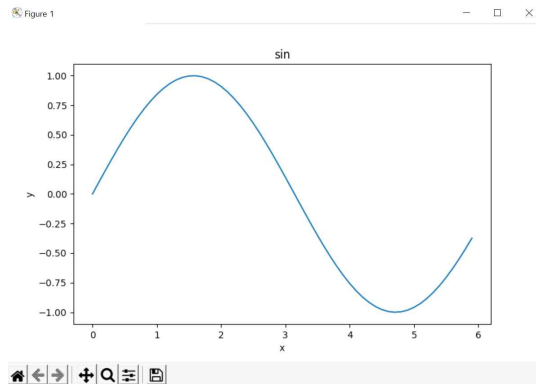
x는 np.arrange를 이용하여 list 형태로 생성
y는 for문으로 x list의 원소에 접근하여 계산 후
self.y list에 추가
x, self.y list로 그래프 출력

- 실행 결과

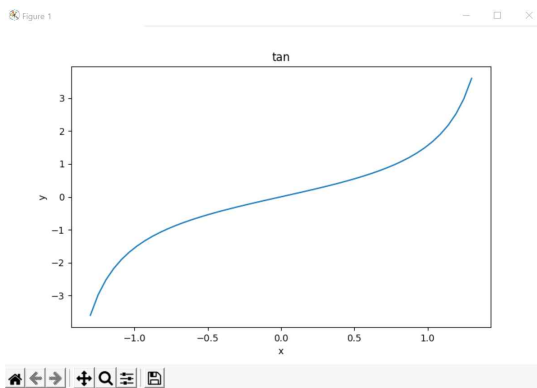
1) Cos



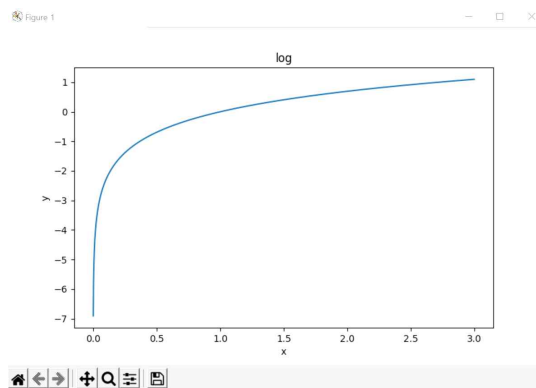
2) Sin



3) Tan



4) Log



2. 본인의 사진을 찍어 해당 파일을 읽어 이미지를 표시하시오. (imshow, imread 활용)

- 코드

```
import cv2
import argparse

# Parser 라이브러리
parser = argparse.ArgumentParser()
parser.add_argument('--path', default="C:/Users/user/PycharmProjects/test1/Eunjung.jpg")
params = parser.parse_args()

# imread - color 이미지 로드
img = cv2.imread(params.path)

# imshow - color 이미지 출력
cv2.imshow("Color Image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 실행 결과



3. 본인의 사진을 Gray Scale 이미지로 변환하여 표시하시오. (imshow, imread 활용)

- 코드

```
import cv2
import argparse

# Parser 라이브러리
parser = argparse.ArgumentParser()
parser.add_argument('--path', default="C:/Users/user/PycharmProjects/test1/Eunjung.jpg")
params = parser.parse_args()

# imread - gray 이미지 로드
gray = cv2.imread(params.path, 0)

# imshow - gray 이미지 출력
cv2.imshow("Gray Image", gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 실행 결과

