

데이터 처리를 위한 Python 프로그래밍 입문

9-2강. 수학 연산 2

ERICA 2018-2

- ▶ 이차방정식
- ▶ 그래프

이차방정식(1/3)

- ▶ 이차방정식 근의 공식 프로그램 적용

$$ax^2 + bx + c = 0 (a \neq 0)$$

- ▶ 판별식 $D = b^2 - 4ac$

- ▶ $D > 0$: 근이 두 개일 때 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

- ▶ $D = 0$: 근이 한 개일 때 $x = -\frac{b}{2a}$

- ▶ $D < 0$: 근이 없을 때

이차방정식(2/3)

▶ 이차방정식 프로그램 : 코드

File Edit Format Run Options Window Help

```
import math
import sys

print('ax2 + bx + c = 0')

# 계수 a,b,c를 입력받고, 입력단은 문자열을 소수로 바꿉니다.
a = float(input('a : '))
b = float(input('b : '))
c = float(input('c : '))

if a == 0:
    print('a=0 : 이차장방식이 아닙니다.')
    sys.exit() # 이차방정식이 아니면 프로그램 실행을 멈춥니다.

D = b*b-4*a*c # 판별식
```

```
if D > 0:
    x1 = (-b+math.sqrt(D))/(2*a)
    x2 = (-b-math.sqrt(D))/(2*a)
    print('2개의 해 : ', x1, x2)
elif D == 0:
    x = -b/(2*a)
    print("1개의 해 : ", x)
else :
    print("해가 없습니다.")
```

이차방정식(3/3)

▶ 이차방정식 프로그램 : 실행화면

```
ax2 + bx + c = 0
a : 1
b : -3
c : 2
2개의 해 : 2.0 1.0
>>>
ax2 + bx + c = 0
a : 1
b : -2
c : 1
1개의 해 : 1.0
>>>
ax2 + bx + c = 0
a : 1
b : 1
c : 1
해가 없습니다.
>>>
```

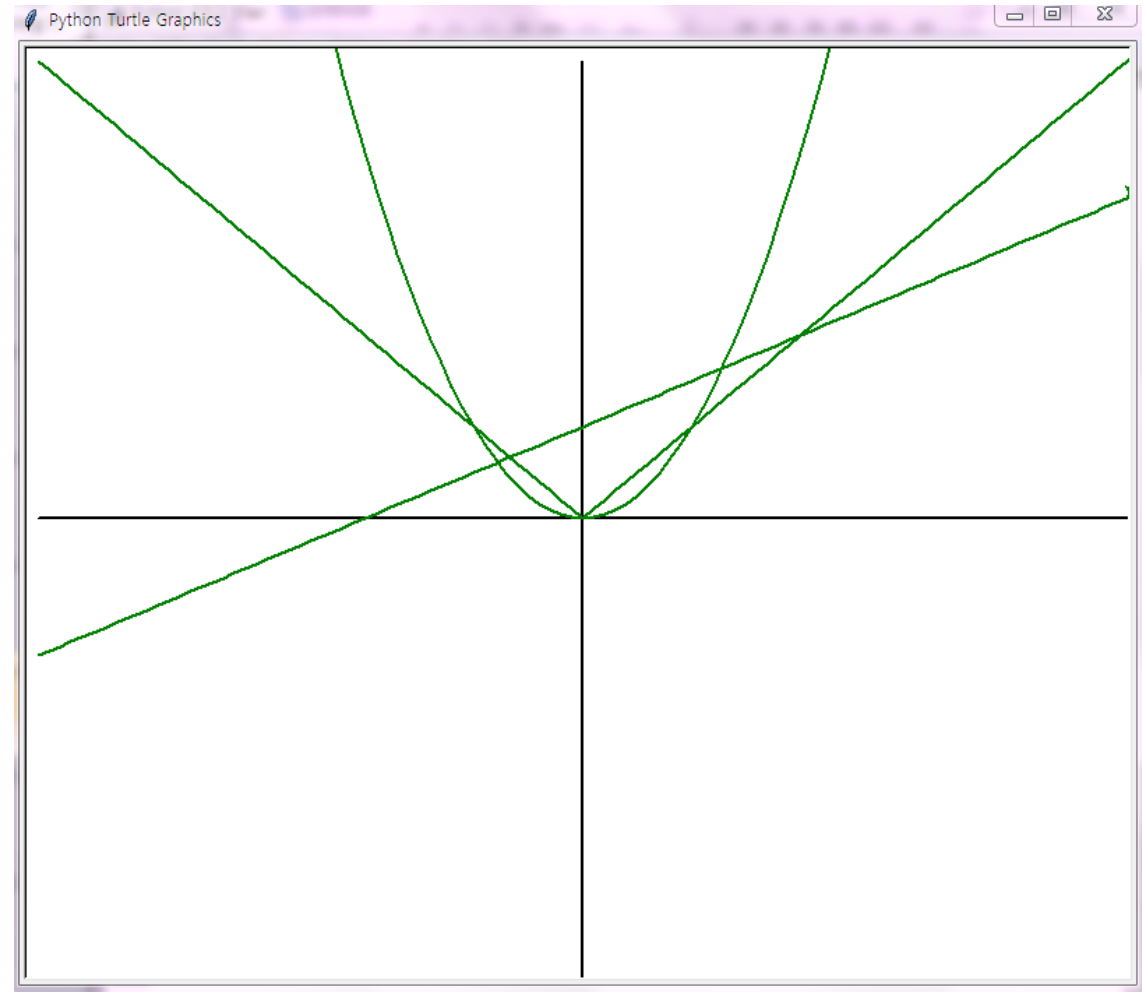
그래프(1/2)

▶ 함수 그래프 : 실행화면

▶ 함수1 : $y = x * x$

▶ 함수2 : $y = abs(x)$

▶ 함수3 : $y = 0.5 * x + 1$



그래프(2/2)

▶ 함수 그래프 : 코드

File Edit Format Run Options Window Help

```
import turtle as t

# 그래프를 그릴 x 좌표 범위
x_min = -5
x_max = +5

# 그래프를 그릴 y 좌표 범위
y_min = -5
y_max = +5

# 그래프를 그릴 간격
space = 0.1

# 그릴 함수의 리스트
func_list = ["y=x*x", "y=abs(x)", "y=0.5*x+1"]
#func_list = [y=x*x, y=abs(x), y=0.5*x+1]

# 좌표 설정, 거북이 속도, 선 굵기
t.setworldcoordinates(x_min, y_min, x_max, y_max)
t.speed(0)
t.pensize(2)
```

```
# x축 그리기
t.up()
t.goto(x_min, 0)
t.down()
t.goto(x_max, 0)

# y축 그리기
t.up()
t.goto(0, y_min)
t.down()
t.goto(0, y_max)

# 그래프 그리기
t.color('green')
for func in func_list: # func_list에 있는 함수를 하나씩 그립니다.
    x = x_min          # x_min부터 계산을 시작합니다.
    exec(func)         # 수식을 계산합니다.
    t.up()
    t.goto(x, y)       # 계산된 좌표로 이동합니다.
    t.down()
    while x <= x_max:  # x_max까지 그래프를 그립니다.
        x = x + space # space만큼 x를 증가시킨 후
        exec(func)    # 수식을 계산합니다.
        t.goto(x, y)  # 계산된 좌표로 이동합니다.
```

Thank you