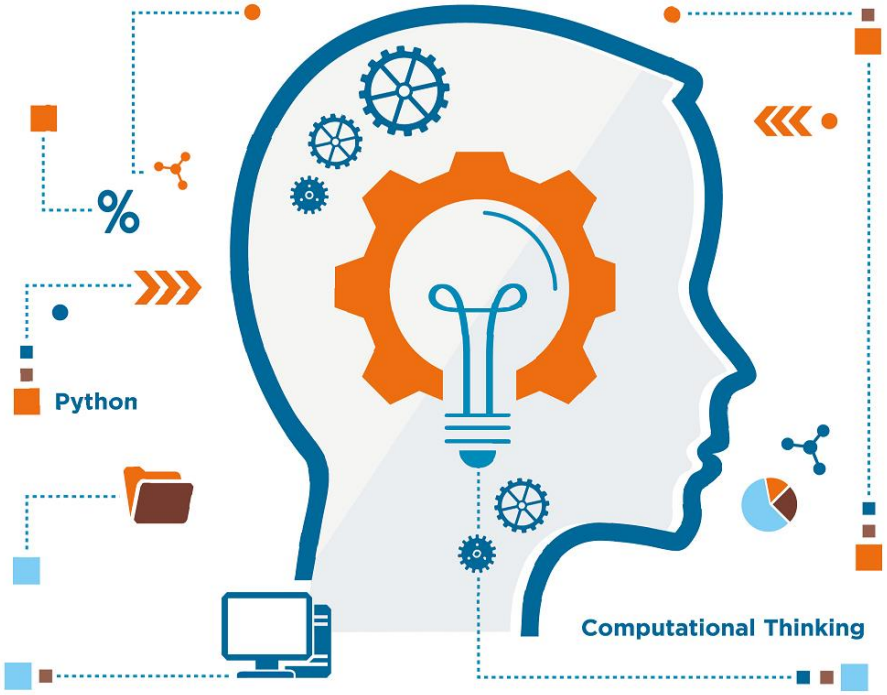


[Python]

Python으로 배우는

소프트웨어 원리

Appendix 05. Turtle 그래픽스 모듈 활용

- xy좌표 그리기 -

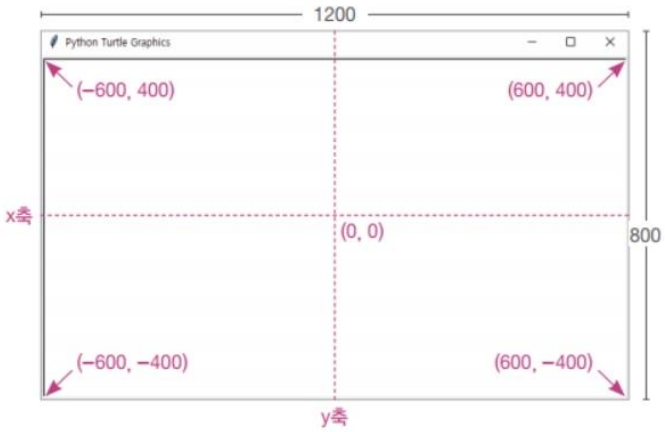
01. 제어 구조

[Python실습] Turtle모듈 활용: x-y좌표 그리기

Ch05-Turtlexy좌표.py

◆ **(0, 0)을 기준으로 x-y 좌표축 그리기 (함수호출 사용)**

- Turtle Speed : 3
- Window Size : 500 * 500 pixel




```
import turtle as t

t.speed(3)

##[함수] (x1, y1) - (x2, y2) 직선 그리기
def line(x1, y1, x2, y2) :
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)

## x, y축 좌표 그리기
line(-500, 0, 500, 0)     # x축 라인
line(0, -500, 0, 500)     # y축 라인


t.exitonclick()  # 실행 창을 닫지 않도록
```

01. 제어 구조

 # for 반복문

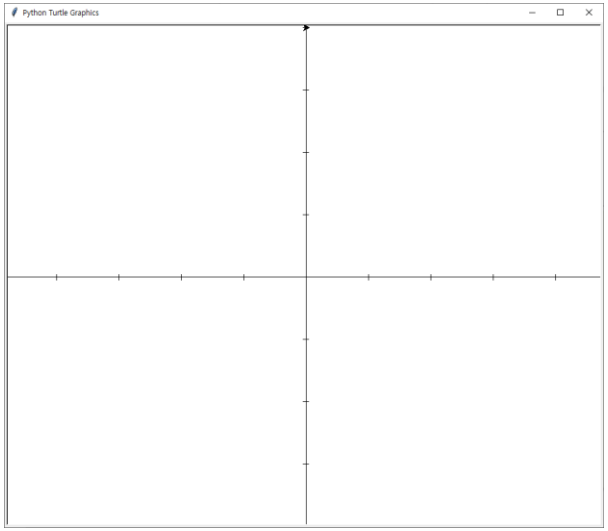
for *반복값_변수* | _ **in range(***반복_초기값, 반복_종료값* [, *반복_증감값*]**)**

[Python실습] Turtle모듈 활용: x-y좌표 그리기

Ch05-Turtlexy좌표.py

- **100 pixel마다 눈금 그리기 (함수호출 사용)**
  - for i in range(-500, 500, 100) :

    line(i, -5, i, 5)

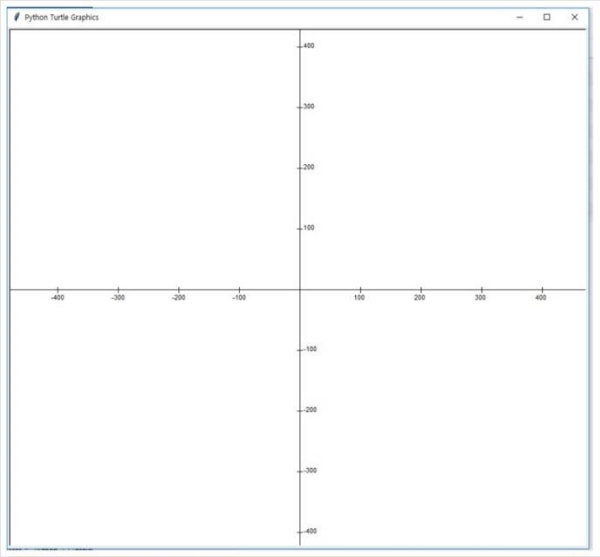



```
##[함수] (x1, y1) - (x2, y2) 직선 그리기
def line(x1, y1, x2, y2) :
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)

## x, y축 좌표 그리기
line(-500, 0, 500, 0)     # x축 라인
line(0, -500, 0, 500)     # y축 라인

for i in range(-500, 500, 100) :
    line(i, -5, i, 5)         # x축 눈금
for i in range(-500, 500, 100) :
    line(-5, i, 5, i)         # y축 눈금

t.exitonclick()  # 실행 창을 닫지 않도록
```

[Python실습] Turtle모듈 활용: x-y좌표 눈금 값 쓰기

◆ **눈금에 값 쓰기 (함수호출 사용)**

- ##[함수] (x,y)에 text 쓰기
  def txtwrite(x, y, text):
- Turtle 감추기
  turtle.hideturtle()



➢ 좌표축과 눈금 값 위치를 잘 맞춰보세요.

```
##[함수] (x,y)에 텍스트 쓰기
def txtwrite(x, y, text) :
    t.up()
    t.goto(x, y)
    t.down()
    t.write(text)

## x, y축 좌표 그리기
line(-500, 0, 500, 0)      # x축 라인
line(0, -500, 0, 500)      # y축 라인

for i in range(-500, 500, 100) :
    line(i, -5, i, 5)          # x축 눈금
    if i != 0 :
        txtwrite(i-10, -20, i)
for i in range(-500, 500, 100) :
    line(-5, i, 5, i)          # y축 눈금
    if i != 0 :
        txtwrite(20, i-10, i)
```

01. 제어 구조

[Python실습] Turtle모듈 활용: x-y좌표 그리기

```
import turtle as t

##[함수] (x1, y1) - (x2, y2) 직선 그리기
def line(x1, y1, x2, y2):
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)

##[함수] (x,y)에 텍스트 쓰기
def txtwrite(x, y, text):
    t.up()
    t.goto(x, y)
    t.down()
    t.write(text)
```

```
## 메인 시작
t.speed(3)
t.hideturtle()

## x, y축 좌표 그리기
line(-500, 0, 500, 0)       # x축 라인
line(0, -500, 0, 500)       # y축 라인
for i in range(-500, 500, 100) :
    line(i, -5, i, 5)       # x축 눈금
    if i != 0:
        txtwrite(i-10, -20, i) #x축 눈금 값
for i in range(-500, 500, 100) :
    line(-5, i, 5, i)       # y축 눈금
    if i != 0:
        txtwrite(20, i-10, i)   #y축 눈금 값

t.exitonclick()  # 실행 창을 닫지 않도록
```

01. 제어 구조

[과제-2] **Turtle모듈 활용:** x-y좌표 그리기 Ch05-Turtlexy좌표.py

◆ **x, y 좌표를 그리는 부분을 함수 호출로 해결하시오.**

```
##[함수] x, y 좌표 그리기
def draw_xy(wsize, step):
```

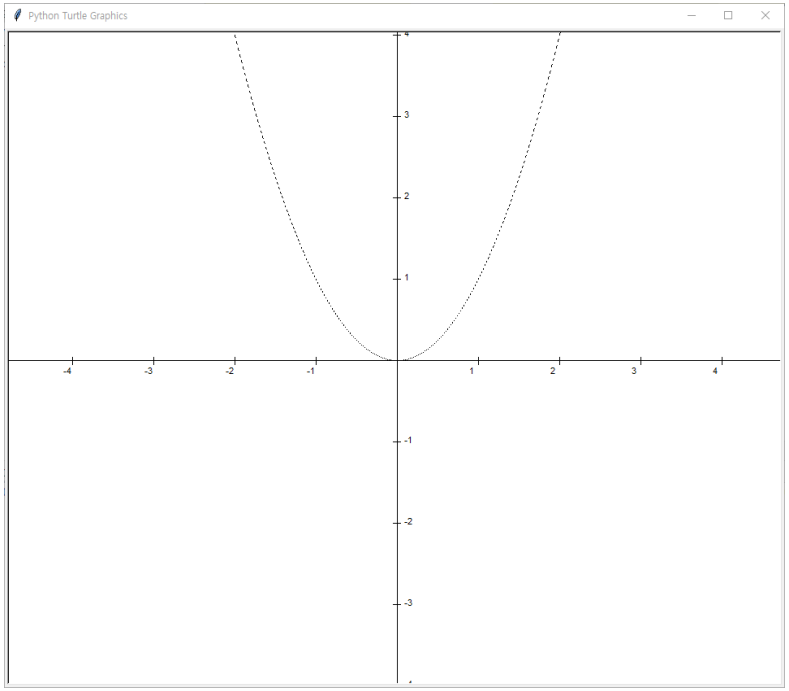
```
## 메인 시작
t.speed(3)
t.hideturtle()

## x, y축 좌표 그리기
wsize = 500
step = 100
draw_xy(wsize, step)


t.exitonclick()  # 실행 창을 닫지 않도록
```

[과제-2도전] $y = x^2$ 함수 그래프를 그리는 프로그램을 완성하시오.

◆ **$y = x^2$ 함수 그래프를 그리는 프로그램 완성**

- Turtle Speed : 0
- Window Size : 500 * 500 pixel
- 좌표축 눈금 간격 크기 : 100pixel 당 1 또는 50pixel 당 1로 축소하여 사용

➢ $y = ax^2 + bx + c$ 형식의 그래프를 그릴 수 있도록 a, b, c 변수를 함수로 넘기는 방식을 적용해보세요.



Thank You !