# 파이썬 프로그래밍

## 18. 그래픽스 지원 API

00

# Contents

## ❖ 수업 목표

- 터틀 그래픽스 개요를 이해할 수 있다.
- 터틀 그래픽스 함수를 사용할 수 있다.
- 터틀 그래픽스에 반복문을 활용할 수 있다.
- 다각형 및 도형을 그릴 수 있다.

## ❖ 세부 목표

00

# 1. 터틀 그래픽스

## ❖ 터틀 그래픽스(거북이 그래픽, Turtle Graphics)

- 소개
  - 컴퓨터에 그래픽을 표시하는 하나의 방식
  - 꼬리에 잉크를 묻인 거북이를 조작하여 그림을 그리기

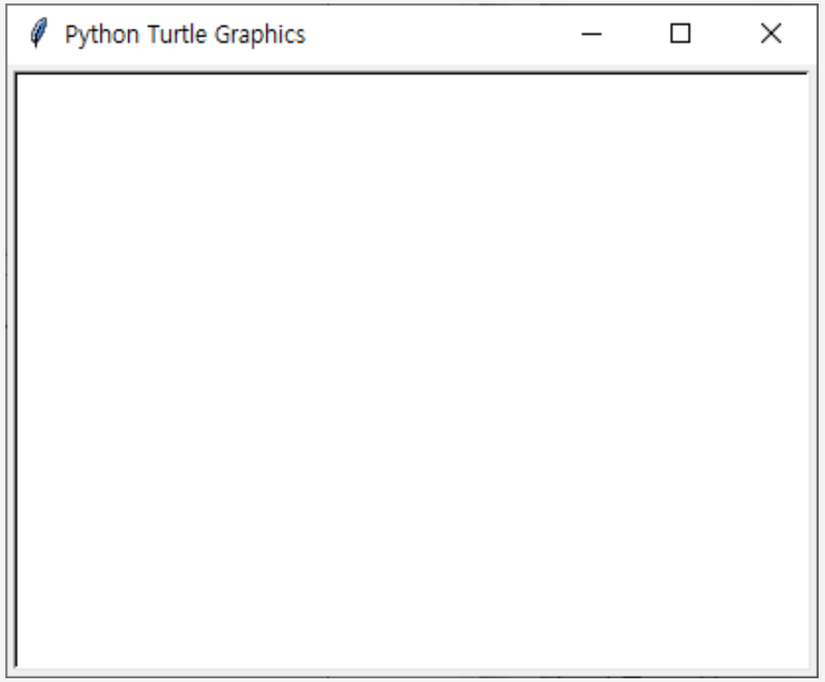
- 사용 방법
  - 파이썬 기본 내장 모듈
  - 임포트 후 사용
    - import turtle

# 1. 터틀 그래픽스

## ❖ 터틀 그래픽스 개요

- 윈도우 생성
  - 라이브러리 임포트
    - import turtle

- 메인루프 실행
  - turtle.mainloop()
  - turtle.done()



# 1. 터틀 그래픽스

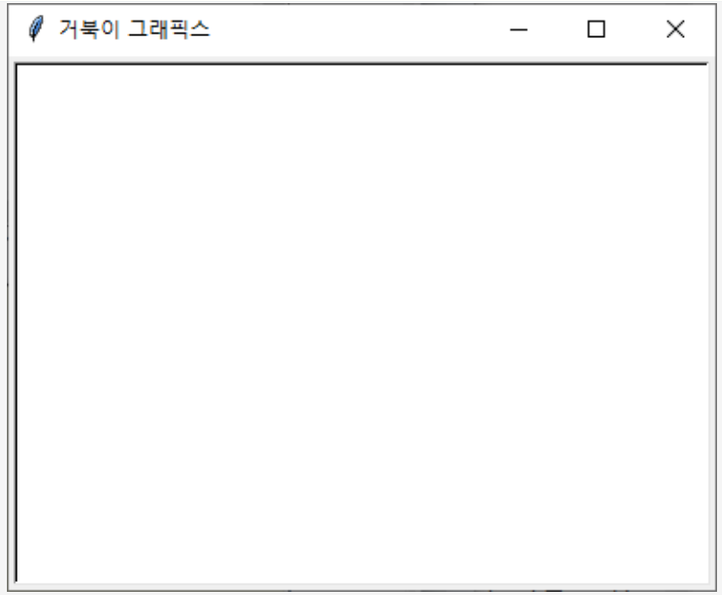
## ❖ 터틀 그래픽스 개요

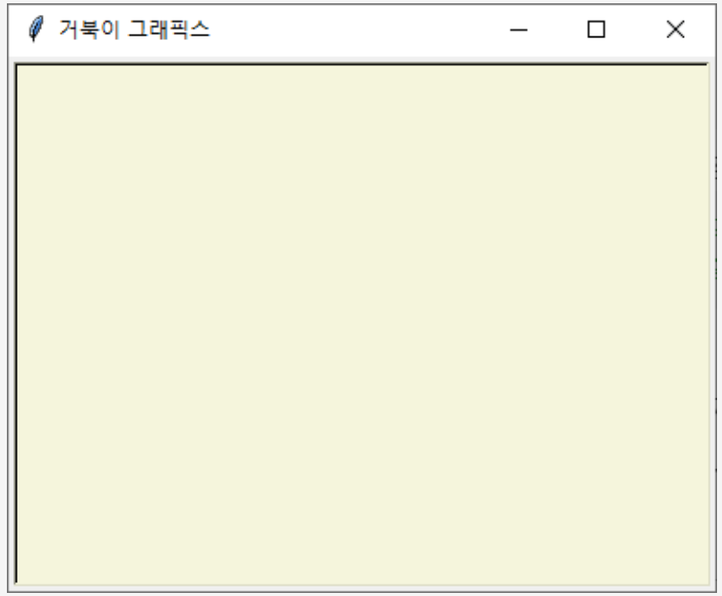
- 윈도우 클릭 시 종료
  - turtle.exitonclick()

- 윈도우 제목 변경
  - turtle.title('거북이 그래픽스')

- 윈도우 크기 변경
  - turtle.setup(410, 310)

- 배경색 변경
  - t.bgcolor("색상")
    - t.bgcolor("beige")

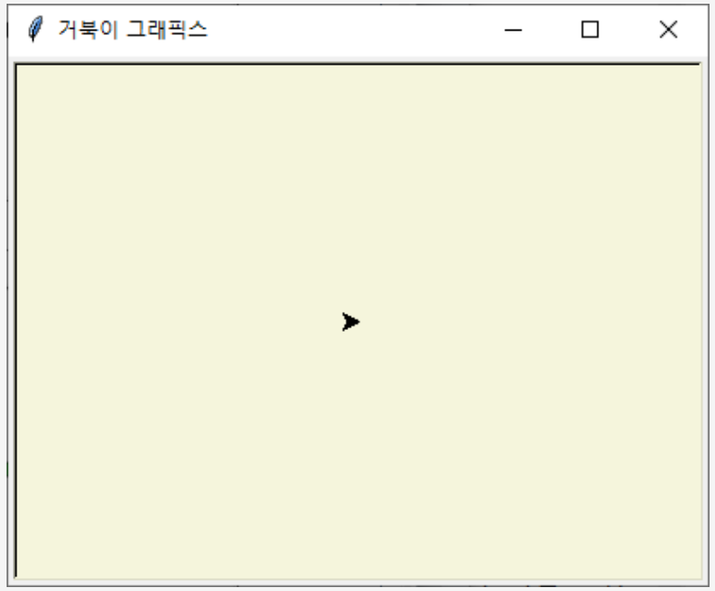


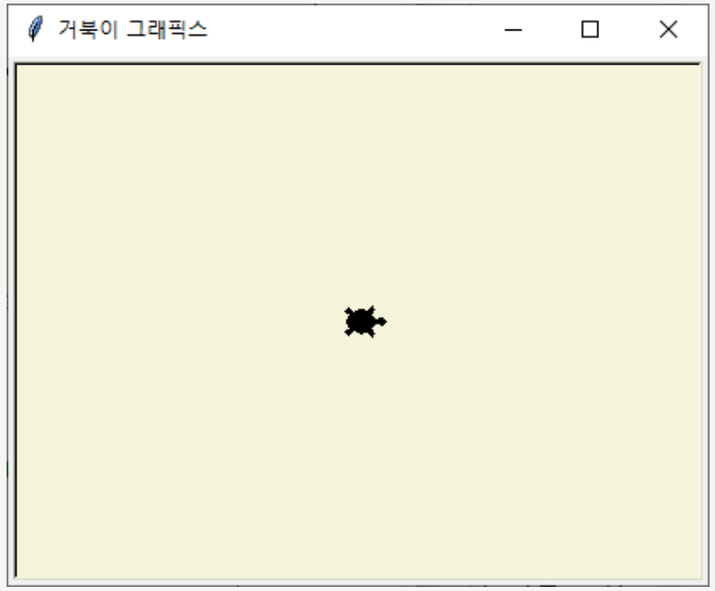
# 1. 터틀 그래픽스

## ❖ 터틀 그래픽스 개요

- Turtle 클래스 객체 생성
  - t = turtle.Turtle()




- 거북이 표시하기
  - t.shape('모양')
    - t.shape('turtle')



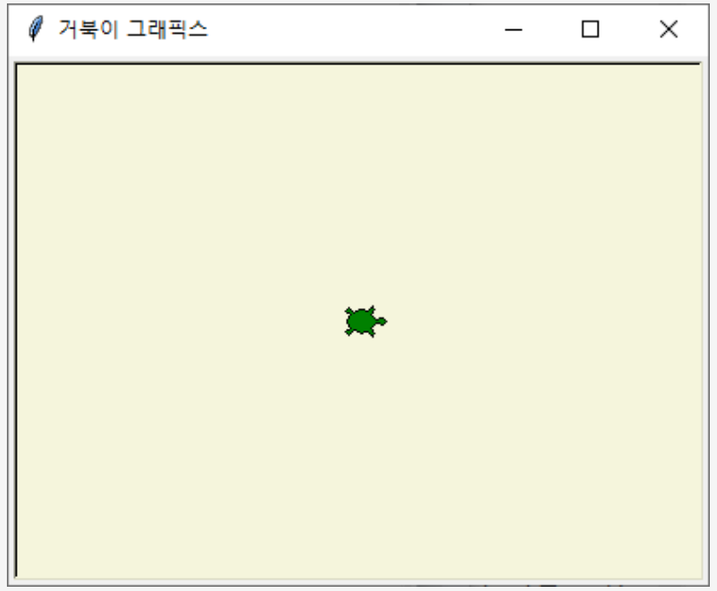
# 1. 터틀 그래픽스

## ❖ 터틀 그래픽스 개요

- 거북이 색상 변경
  - t.color('경계선색', '채움색')
    - t.color('black','green')

- 회전
  - t.right(angle): 오른쪽
    - t.right(180)
  - t.left(angle): 왼쪽
    - t.left(90)

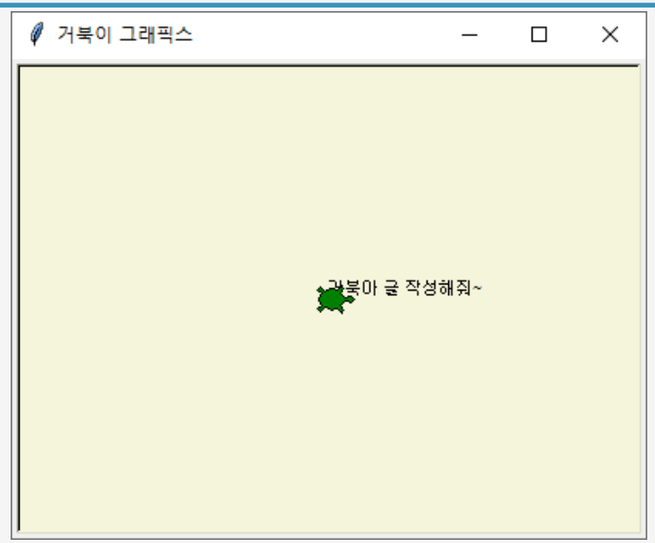
- 홈(원) 위치로 이동
  - t.home()



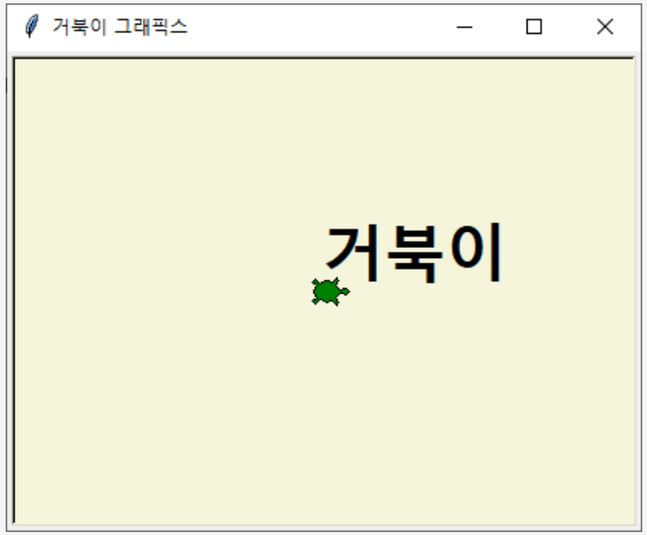

# 1. 터틀 그래픽스

## ❖ 터틀 그래픽스 개요

- 문자열 출력
  - t.write("문자열")
    - t.write("거북아 글 작성해줘~")




- 문자열 폰트 변경
  - t.write("거북이", font=(fontname, fontsize, fonttype))
    - t.write("거북이", font=("arial", 30, "bold"))



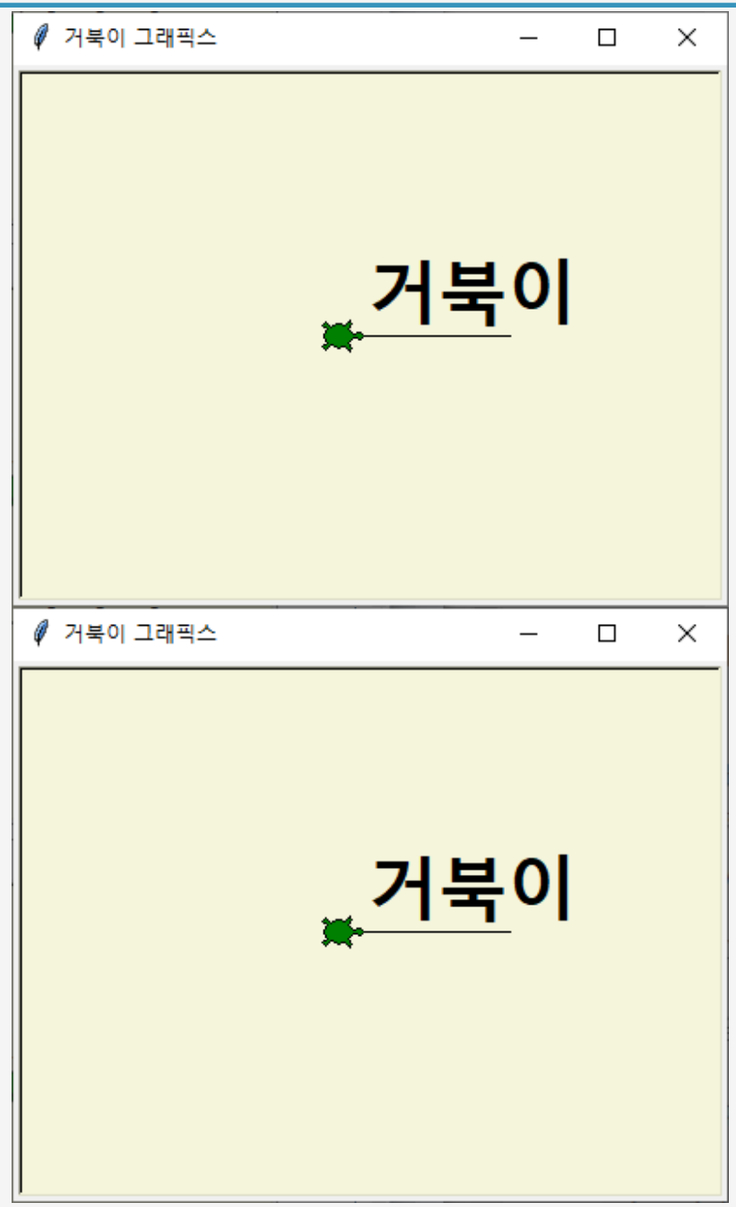
00

# 2. 사각형 그리기

## ❖ 사각형 그리기

- 이동
  - 기본 이동 시 선이 그려짐
  - t.forward(distance): 앞으로
    - t.forward(80)
  - t.backward(distance): 뒤로
    - t.backward(100)

- 그리기
  - 펜을 들고 이동 시 선이 안 그려짐
  - 펜을 내리고 이동 시 선 그려짐
  - t.penup() : 펜 들기
    - t.penup()
  - t.pendown() : 펜 내리기
    - t.pendown()



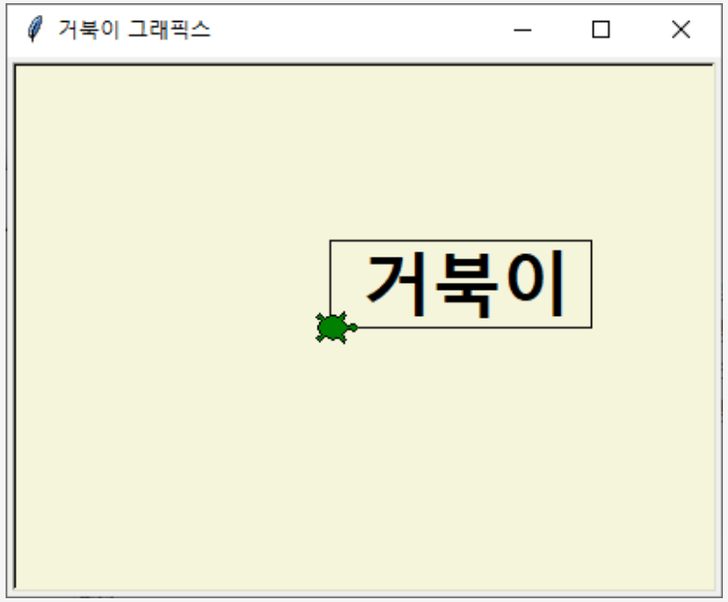
# 2. 사각형 그리기

## ❖ 사각형 그리기

### ▪ 문자열 둘레에 선 그리기

- 기본 이동 시 선이 그려짐
  - t.left(90); t.fd(50)
  - t.right(90); t.fd(150)
  - t.right(90); t.fd(50)
  - t.right(90); t.fd(150)
  - t.left(180)

- forward() = fd()
- backward() = bk()

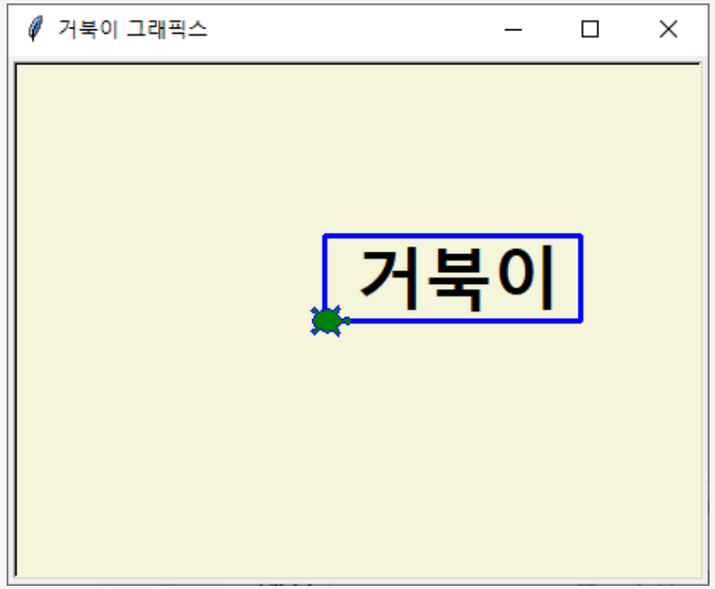


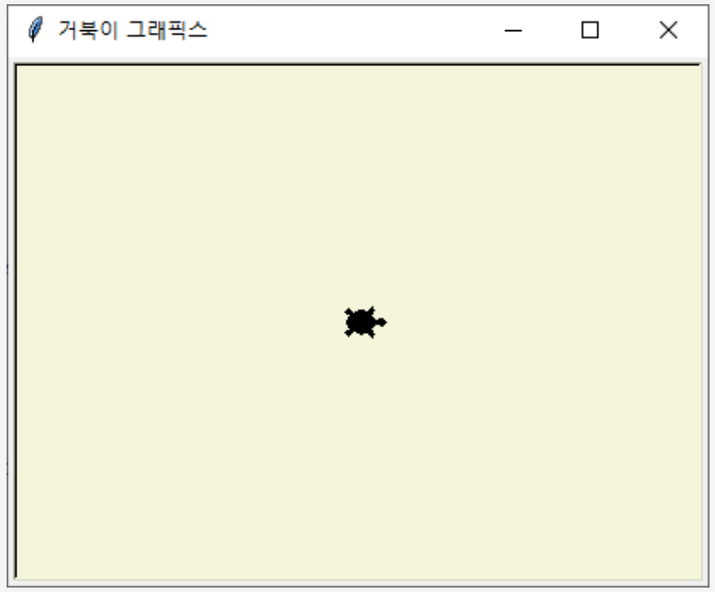
# 2. 사각형 그리기

## ❖ 사각형 그리기

- 펜 색상과 두께 변경
  - t.pencolor(color): 펜 색상
    - t.pencolor("blue")
  - t.pensize(size): 펜 두께
    - t.pensize(3)

- 객체 초기화
  - t.reset() : 초기화



# 2. 사각형 그리기

## ❖ 사각형 그리기

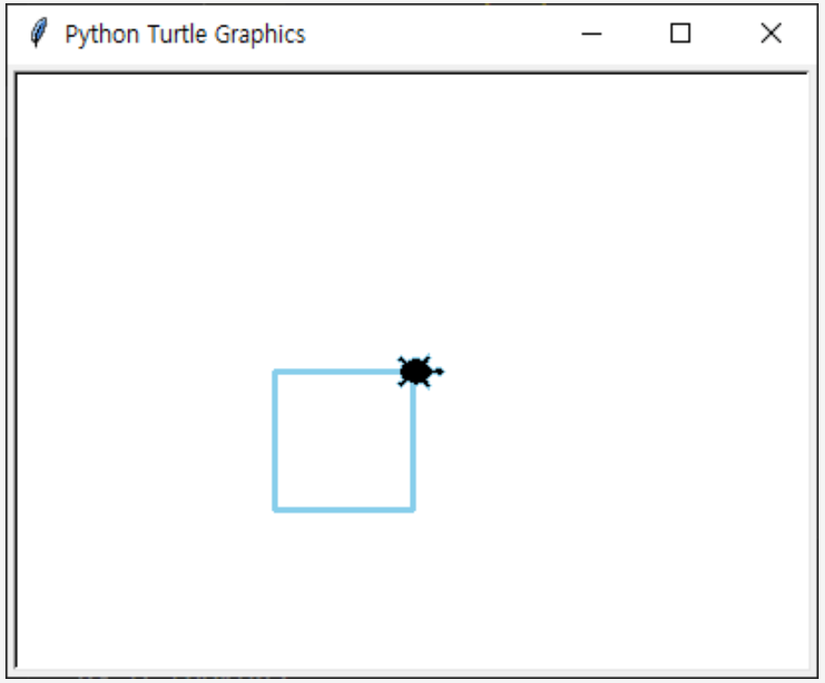
- 터틀 객체를 활용하여 정사각형을 그려보기
  - 가로 70, 세로 70
  - 펜 색상: skyblue
  - 펜 사이즈: 5

- 터틀 객체를 활용하여 직사각형을 그려보기
  - 배경색 yellogreen
  - 가로 120, 세로 80
  - 펜 색상: green
  - 펜 사이즈: 3

# 3. 다각형 그리기

## ❖ 다각형 그리기

- 반복문(for) 활용
  - 정사각형 그리기
    - 다음 내용을 4번 반복
      - » 오른쪽 90도 회전
      - » 앞으로 이동

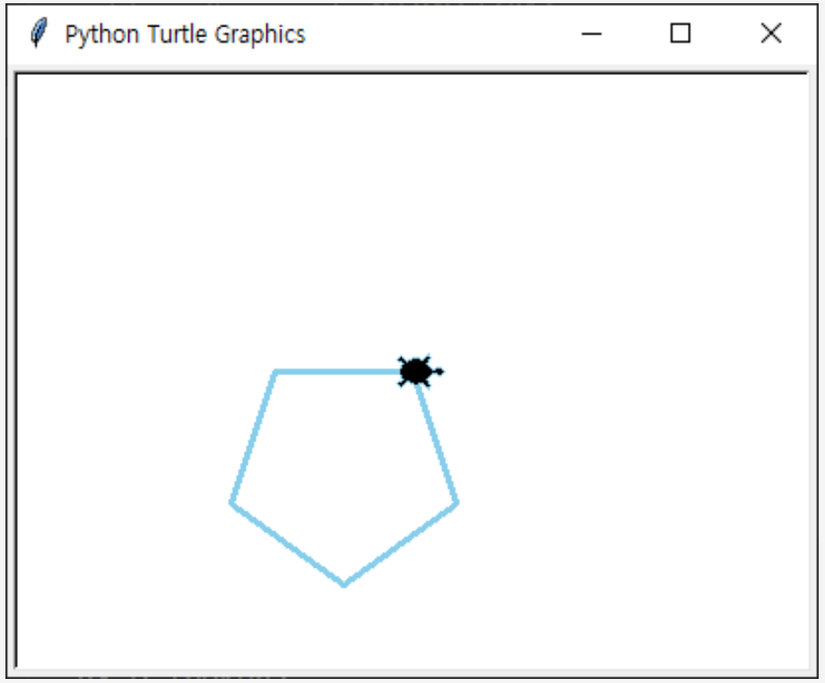



```python
import turtle

t = turtle.Turtle()
t.shape('turtle')
t.pencolor("skyblue")
t.pensize(3)

for i in range(4):
    t.right(90)
    t.forward(70)

turtle.exitonclick()
```

# 3. 다각형 그리기

## ❖ 다각형 그리기

- 반복문(for) 활용
  - 오각형 그리기
    - 다음 내용을 5번 반복
      - » 오른쪽 (360/5)도 회전
      - » 앞으로 이동



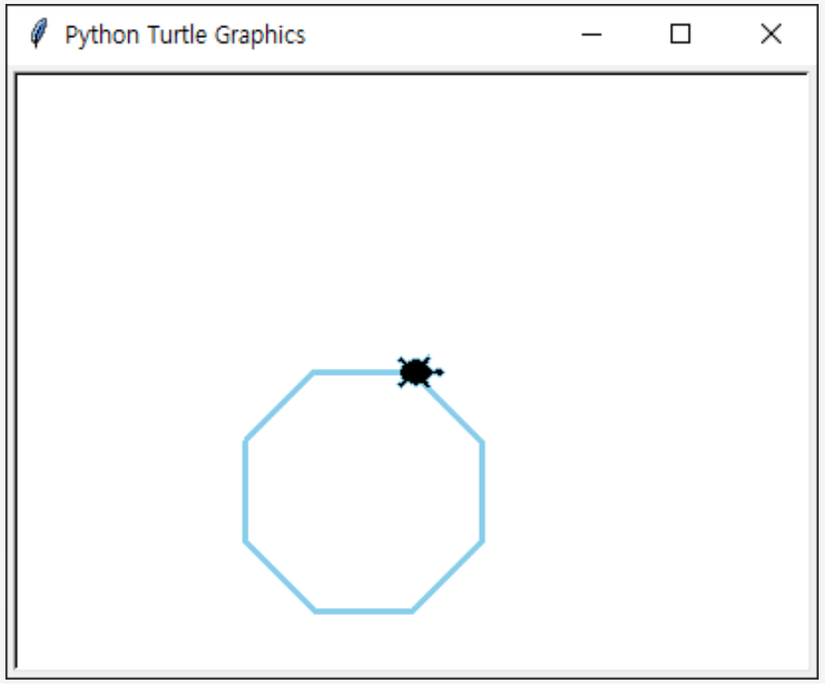

```python
import turtle

t = turtle.Turtle()
t.shape('turtle')
t.pencolor("skyblue")
t.pensize(3)

# 오각형
for i in range(5):
    t.right(360/5)
    t.forward(70)

turtle.exitonclick()
```

# 3. 다각형 그리기

## ❖ 다각형 그리기

### ▪ 반복문(for) 활용

- 다각형 그리기
  - 다음 내용을 n번 반복
    - » 오른쪽 (360/n)도 회전
    - » 앞으로 이동
  - 8각형 예시




```python
import turtle

t = turtle.Turtle()
t.shape('turtle')
t.pencolor("skyblue")
t.pensize(3)

# 다각형
n = int(input("입력: "))
for i in range(n):
    t.right(360/n)
    t.forward(70)

turtle.exitonclick()
```

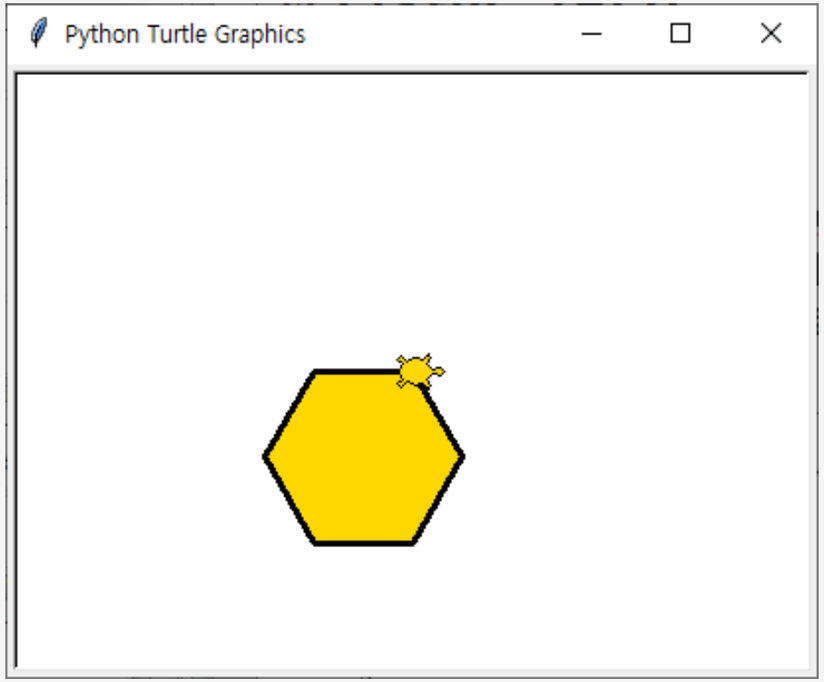
# 3. 다각형 그리기

## ❖ 다각형 그리기

### ▪ 반복문(for) 활용

#### • 다각형 색칠하기

- t.color("black",'gold'): 색상
- t.begin_fill() : 채우기 시작
- t.end_fill : 채우기 종료




```python
import turtle

t = turtle.Turtle()
t.shape('turtle')
t.pensize(3)
t.color("black",'gold')

# 다각형
n = int(input("입력: "))
t.begin_fill()
for i in range(n):
    t.right(360/n)
    t.forward(50)
t.end_fill()

turtle.exitonclick()
```

# 3. 다각형 그리기

## ❖ 다각형 그리기

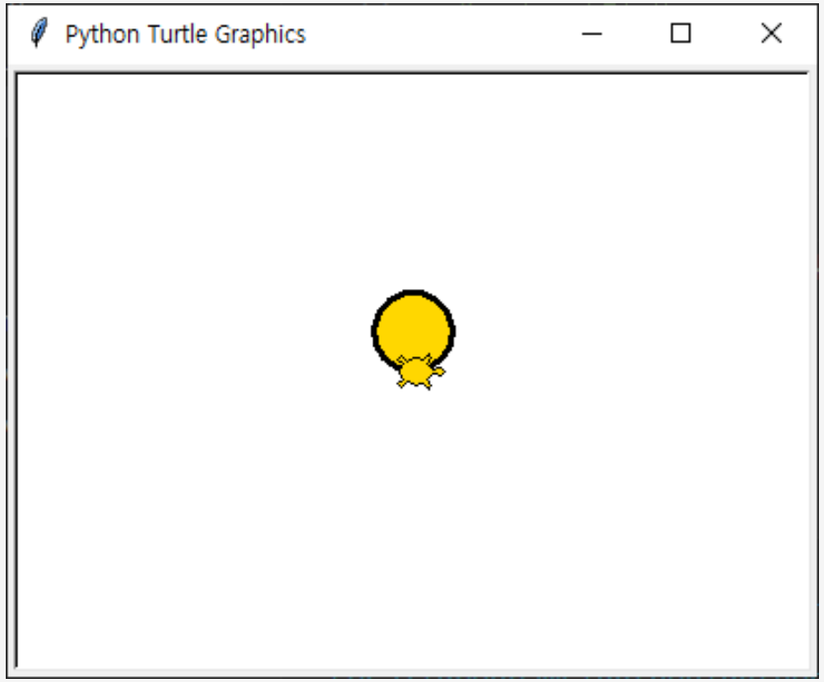
- 터틀 객체를 활용하여 십이각형을 그려보기
  - 한변 길이 : 80
  - 펜 사이즈: 7
  - 색상(테두리/내부): black/olive

- 터틀 객체를 활용하여 다각형을 만드는 함수를 정의하고 함수를 호출하여 삼각형을 그리시오.
  - 펜 사이즈: 7
  - 색상(테두리/내부): black/olive
  - 다각형 함수
    - 입력: 각개수, 변길이
    - 인수: 3, 80

# 4. 복잡한 도형 그리기

## ❖ 원 그리기

- t.circle(radius)
  - t.circle(20)

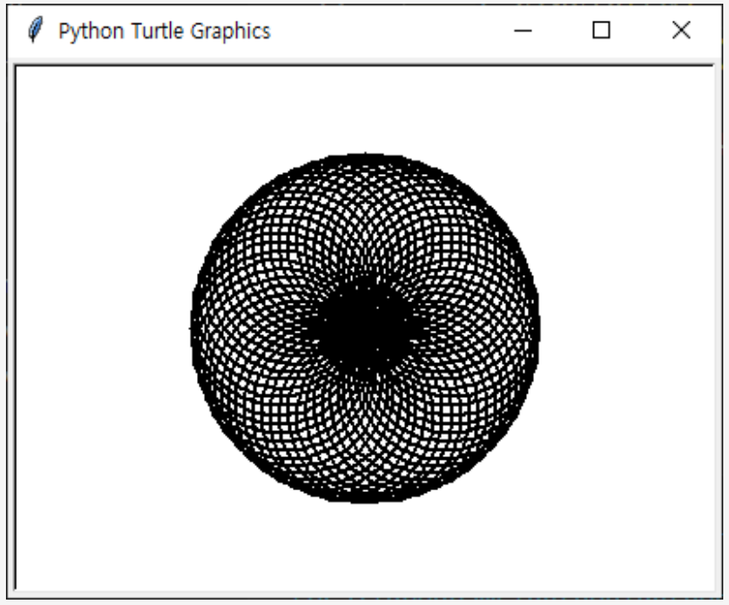



```python
import turtle
turtle.setup(410, 310)

t = turtle.Turtle()
t.shape('turtle')
t.pensize(3)
t.color("black",'gold')

t.begin_fill()

# 원 그리기
t.circle(20)

t.end_fill()

turtle.exitonclick()
```

# 4. 복잡한 도형 그리기

## ❖ 원 여러개 그리기

- t.speed(속도): 속도(0~10) 설정
  - “fastest” : 0
  - “fast” : 10
  - “normal” : 6
  - “slow” : 3
  - “slowest” : 1
- t.speed(0)

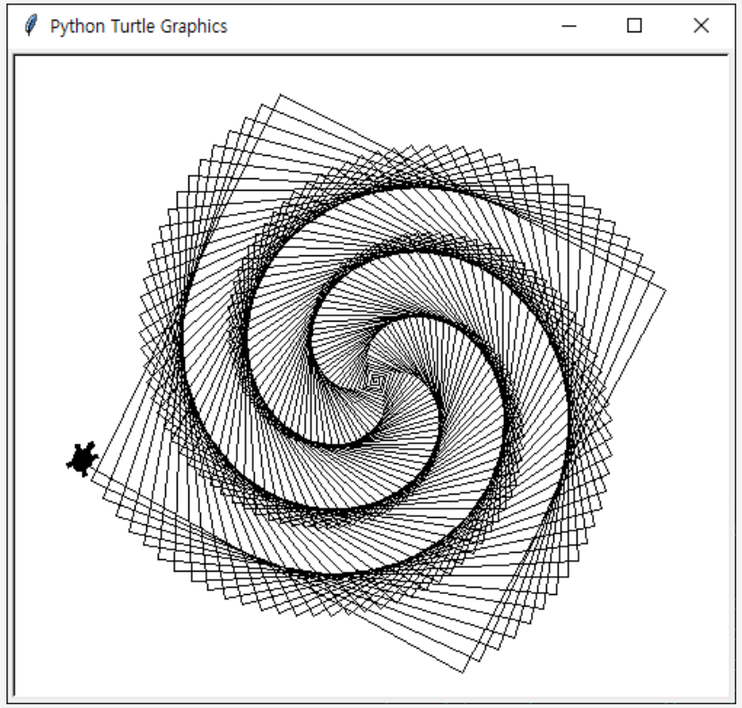



```
1   import turtle
2   turtle.setup(410, 310)
3
4   t = turtle.Turtle()
5   t.shape('turtle')
6   t.pensize(2)
7   t.speed(0)  # 속도(fastest)
8
9   n = 60      # 원 개수
10  for i in range(n):
11      t.circle(50)
12      t.right(360/n)
13
14  turtle.exitonclick()
```

# 4. 복잡한 도형 그리기

## ❖ 복잡한 무늬 그리기

- 속도: “fastest”
- 반복횟수: 300
- 선(이동) 길이 증가
- 오른쪽 91도 회전

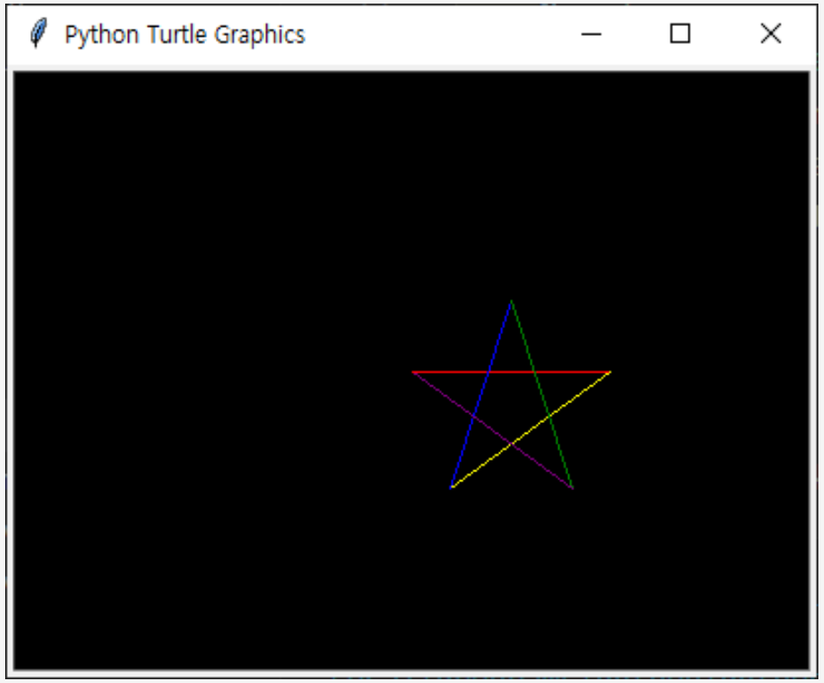



```
1   import turtle
2   turtle.setup(410, 310)
3
4   t = turtle.Turtle()
5   t.shape('turtle')
6   t.pensize(1)
7   t.speed(0)
8
9   n = 300
10  for i in range(n):
11      t.forward(i)
12      t.right(91)
13
14  turtle.exitonclick()
```

# 4. 복잡한 도형 그리기

## ❖ 색상을 변경하며 별 그리기

- 터틀 그래픽을 사용하여 각 꼭지점에서 선 색상이 변경되는 5각 별을 그리는 함수를 정의 및 호출하는 프로그램을 작성하시오.
  - 선 색상: 'red','yellow','blue','green','purple'
  - 속도: 1
  - 펜 사이즈: 2
  - 별 함수
    - 입력: 없음
    - 선 길이: 100



# 4. 복잡한 도형 그리기

## ❖ 색상을 변경하며 별 그리기

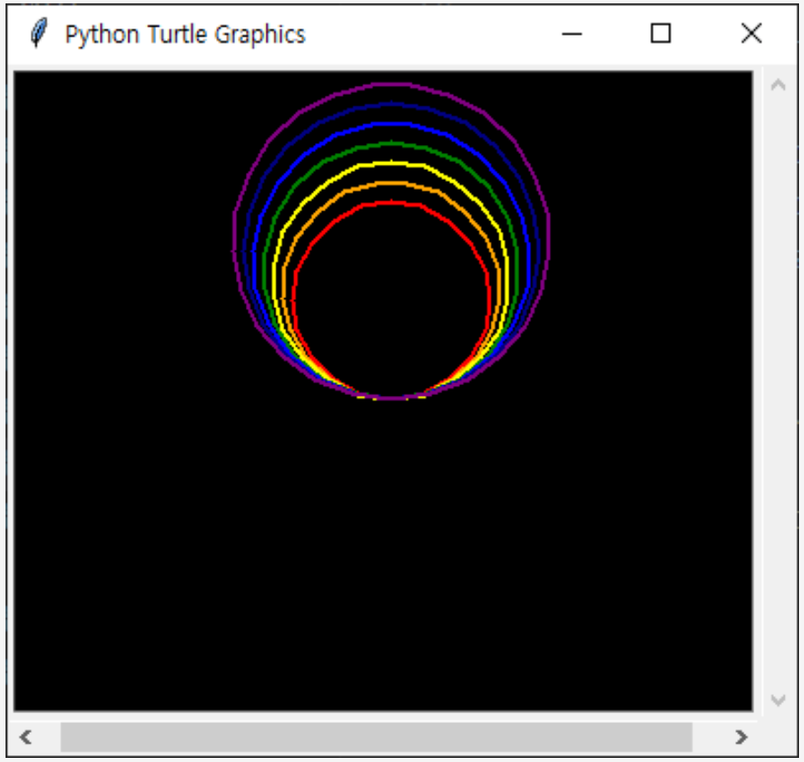
- 터틀 그래픽을 사용하여 각 꼭지점에서 선 색상이 변경되는 5각 별을 그리는 함수를 정의 및 호출하는 프로그램을 작성하시오.
  - 선 색상: 'red','yellow','blue','green','purple'
  - 속도: 1
  - 펜 사이즈: 2
  - 별 함수
    - 입력: 없음
    - 선 길이: 100

```
1  import turtle
2  turtle.setup(410, 310)
3  turtle.bgcolor('black')
4
5  star = turtle.Turtle()
6  star.speed(1)                    # 속도(slowest)
7  star.pensize(2)
8
9  colors = ['red','yellow','blue','green','purple']
10
11 def drawStar():
12     n = 5                        # 반복횟수
13     for i in range(n):
14         star.color(colors[i])
15         star.forward(100)        # 선(이동) 길이
16         star.right(144)          # 오른쪽 144도 회전
17
18 drawStar()
19 star.hideturtle()                # 모양 숨기기
20
21 turtle.exitonclick()
```

# 4. 복잡한 도형 그리기

## ❖ 무지개 색 원 그리기

- 터틀 그래픽을 사용하여 무지개 색상의 원을 겹쳐 그리는 함수를 정의 및 호출하는 프로그램을 작성하시오.
  - 선 색상: 'red','orange','yellow','green','blue','navy','purple'
  - 속도: 0
  - 펜 사이즈: 2
  - 무지개색 원 함수
    - 기능: 반지름이 5씩 증가하는 무지개색 원 여러개 그리기
    - 입력: 기본 반지름(50)
    - 선 길이: 100



# 4. 복잡한 도형 그리기

## ❖ 무지개 색 원 그리기

- 터틀 그래픽을 사용하여 무지개 색상의 원을 겹쳐 그리는 함수를 정의 및 호출하는 프로그램을 작성하시오.
  - 선 색상: 'red','orange','yellow','green','blue','navy','purple'
  - 속도: 0
  - 펜 사이즈: 2
  - 무지개색 원 함수
    - 기능: 반지름이 5씩 증가하는 무지개색 원 여러개 그리기
    - 입력: 기본 반지름(50)
    - 선 길이: 100

```
1  import turtle
2  turtle.setup(400, 350)
3  turtle.bgcolor('black')
4
5  rainbow = turtle.Turtle()
6  rainbow.speed(0)            # 속도
7  rainbow.pensize(2)
8  colors = ['red','orange','yellow',
9            'green','blue','navy','purple']
10
11 def drawRainbow(radius):
12     for color in colors:
13         rainbow.color(color)
14         rainbow.circle(radius)
15         radius += 5          # 원 반지름 증가
16
17 drawRainbow(50)
18 rainbow.hideturtle()        # 모양 숨기기
19
20 turtle.exitonclick()
```

# 5. 연습 문제

## ❖ 무지개 그리기

- 터틀 그래픽을 사용하여 특정 좌표에 무지개를 그리는 함수를 정의 및 호출하는 프로그램을 작성하시오.
  - 무지개 함수
    - 기능
      - » 특정 좌표에 무지개 그리기
      - » 그리기 완료 후 모양 숨기기
      - » 속도: 0
      - » 펜크기: 5
    - 입력: 반지름, x좌표, y좌표

# Notice

## ❖ 과제

- 1. 터틀 그래픽스 함수 사용하여 코드 작성하기
- 2. 반복문을 활용한 그림 그리기
- 3. 다양한 도형 그리며 필요한 기능 탐색하기
- 4. 터틀 그래픽스 추가 함수 검색해서 사용하기
- 5. 다수의 객체 생성 및 활용하기

## ❖ 다음 수업 내용

- 딥러닝 파이썬 패키지(1)
  - pandas, numpy, matplotlib