
PYTHON

TURTLE

홍익대학교

5. 터틀 라이브러리(Turtle Library)

01 그래픽 활용(Turtle 라이브러리)
02 터틀 움직이기
03 goto()와 pendown(), penup()
04 반복문으로 다각형 그리기
P1 반복문으로 다각형 그리기 실습

05 펜 모양 다양하게 만들기
06 터틀 애니메이션 만들기
07 터틀 명령어 정리 및 약어

P2 터틀 실습
P3 터틀 실습
P4 터틀 실습
P5 터틀 실습



그래픽 활용(Turtle 라이브러리)

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> turtle.forward(100)
>>>
```

Ln: 6 Col: 4



라이브러리 불러오기

```
>>> import 라이브러리이름
```

그래픽 활용(Turtle 라이브러리)



라이브러리 불러오기

```
>>> import 라이브러리이름
```

터틀 이동하기

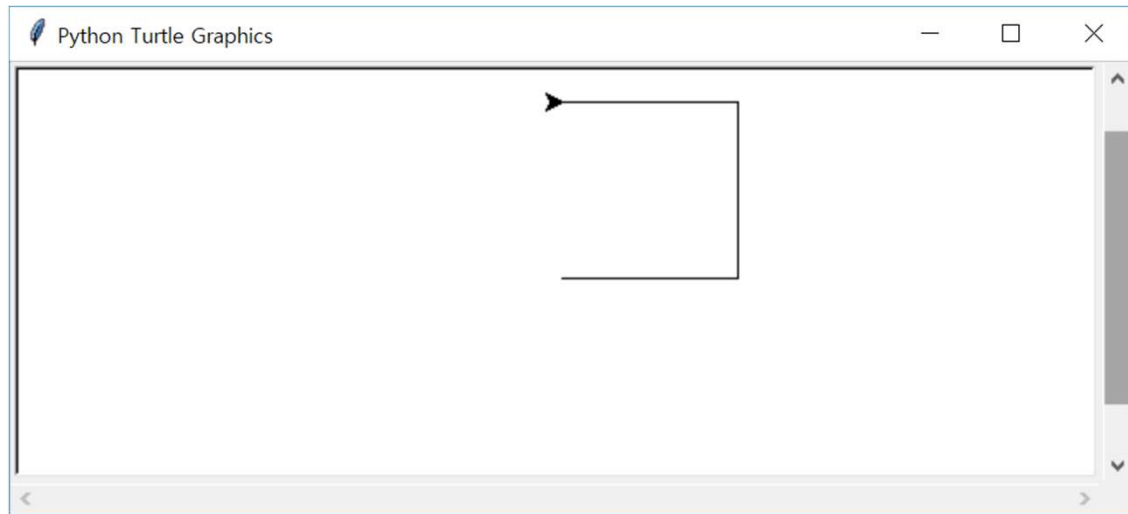
```
>>> turtle.forward(이동거리)
```

1. 터틀 라이브러리를 불러온 후 앞으로 가기 200만큼 실행하라.
2. 터틀을 앞으로 가기 -300만큼 실행해보자.
3. 앞으로 가기(fd) 명령으로부터 추론하여 터틀을 뒤로 가기 150만큼 실행하라.

터틀 움직여보기



```
import turtle
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.right(90)
turtle.back(100)
```



```
>>> import turtle
>>> turtle.명령어(값)
```

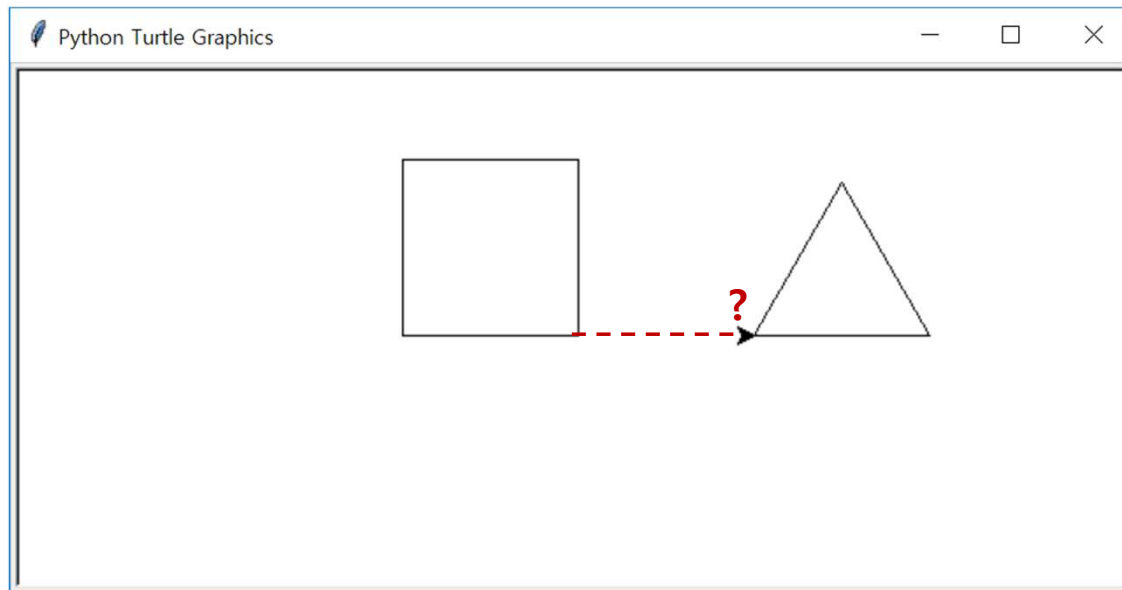
터틀 움직여보기



명령어	설명	예시
forward()	앞으로 ()만큼 가기	<code>turtle.forward(100)</code>
back()	뒤로 ()만큼 가기	<code>turtle.back(100)</code>
left()	왼쪽으로 ()도만큼 회전하기	<code>turtle.left(90)</code>
right()	오른쪽으로 ()도만큼 회전하기	<code>turtle.right(90)</code>

1. 터틀 라이브러리를 이용하여 사각형을 그려보자.
2. 터틀 라이브러리를 이용하여 삼각형을 그려보자.
3. 터틀 라이브러리를 이용하여 원을 그려보자.

자취 그리기 기능 조절하기

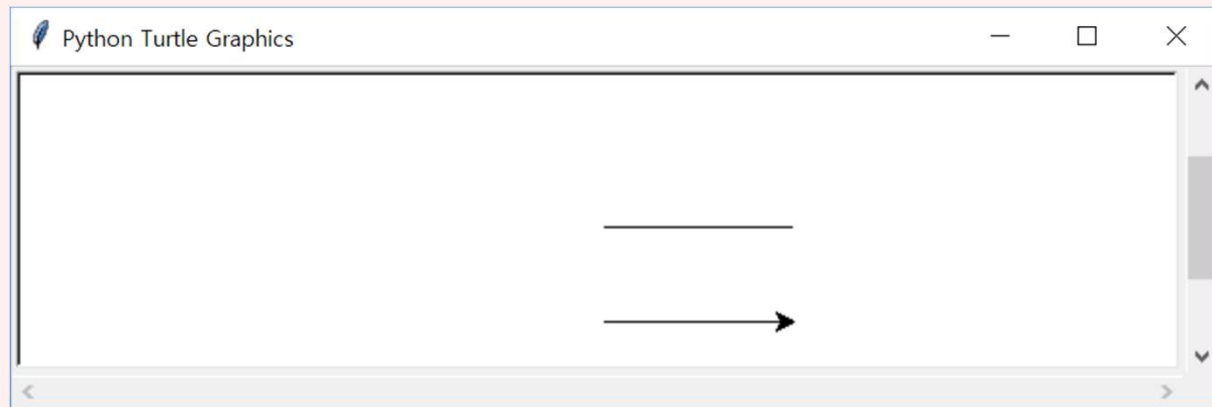


`goto(x, y)`
(x, y) 좌표로 이동하기

`pendown()`
펜을 들어 그리지 않기

`penup()`
펜을 내려 그림 그리기

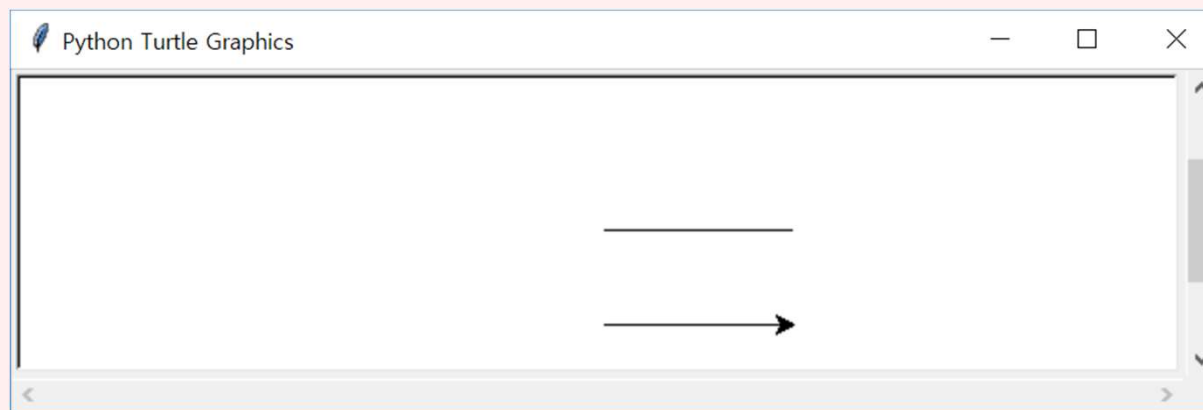
자취 그리기 기능 조절하기



503 goto(), pendown(), penup()

CH5 터틀 라이브러리

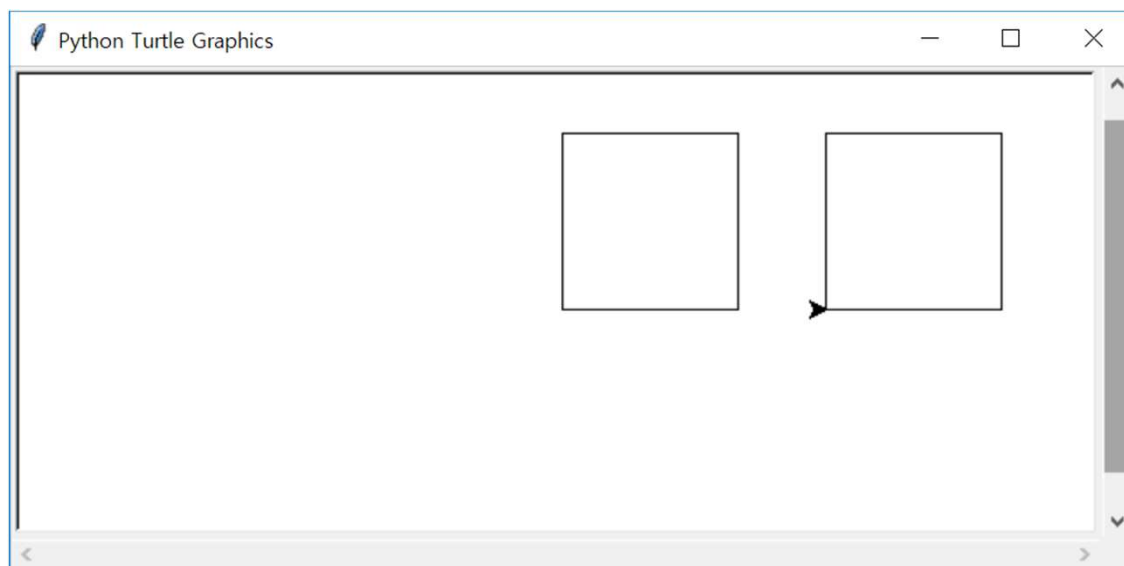
자취 그리기 기능 조절하기



```
*chapter5_3_1.py - C:/Python/chapter5_3_1.py (3.6.4)*
File Edit Format Run Options Window Help
import turtle
turtle.forward(100)
turtle.penup()
turtle.goto(0, -50)
turtle.pendown()
turtle.forward(100)
Ln: 8 Col: 0
```

503 goto(), pendown(), penup()

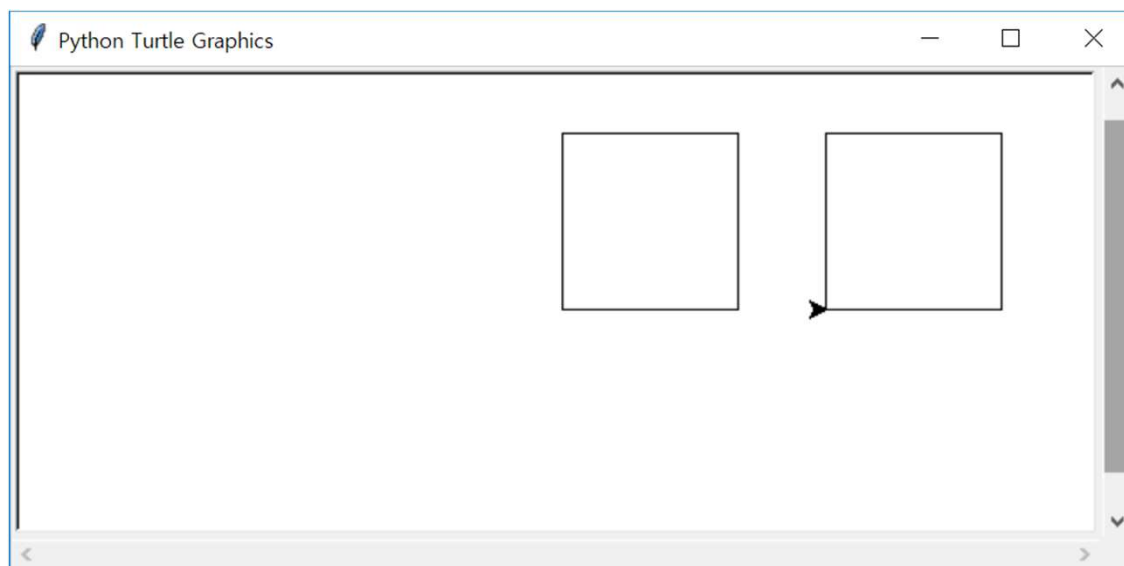
반복문 사용하기



```
import turtle
turtle.forward(100)
turtle.left(100)
```

·
·
·
·
·

반복문 사용하기

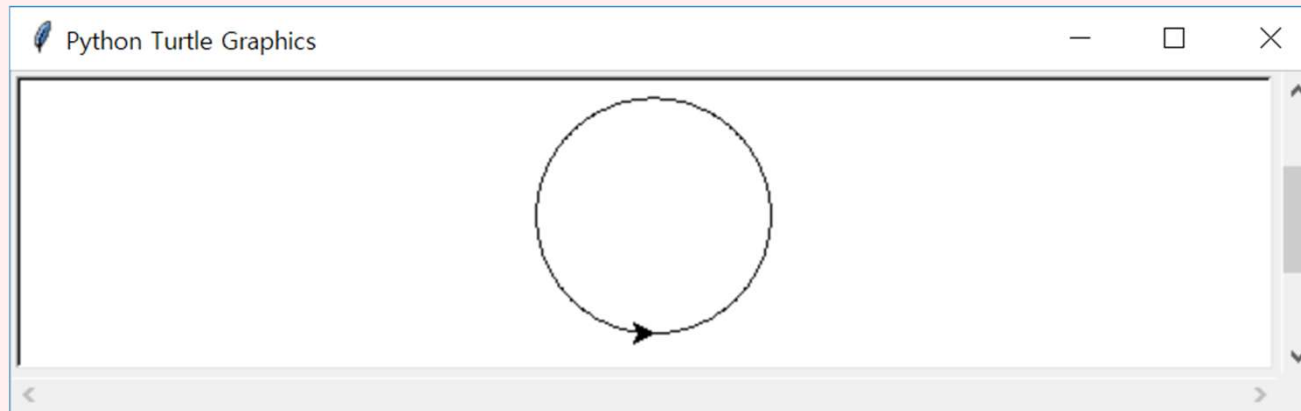


```
*chapter5_4_1.py - C:/Python/chapter
File Edit Format Run Options W
import turtle
i = 0
while i < 4:
    turtle.forward(100)
    turtle.left(90)
    i = i+1

turtle.penup()
turtle.goto(150,0)
turtle.pendown()

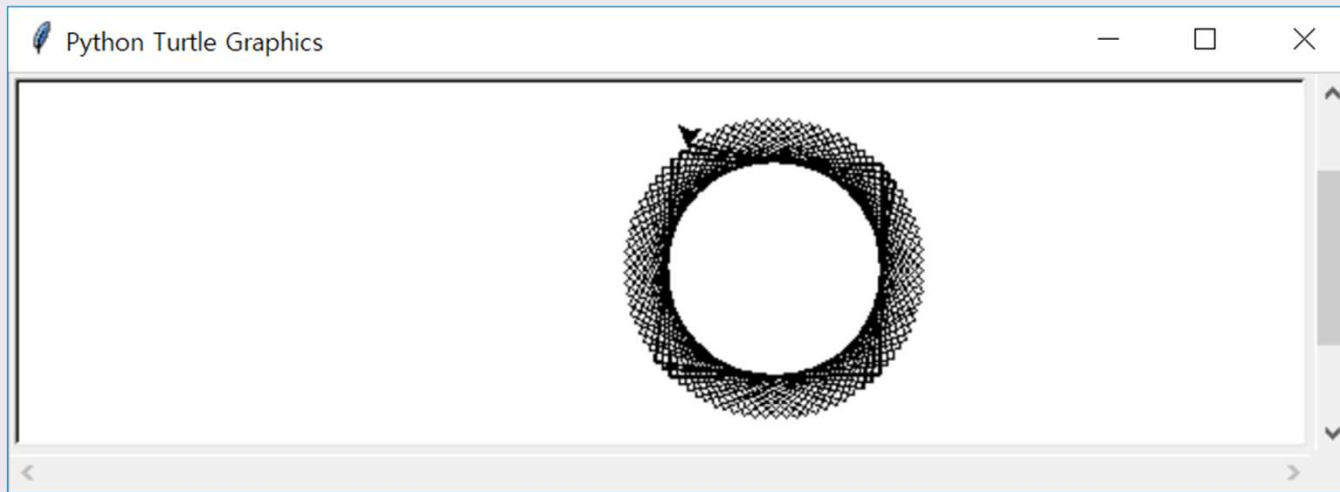
for j in range(4):
    turtle.forward(100)
    turtle.left(90)
```

반복문 사용하기



```
import turtle
for i in range(360):
    turtle.forward(1)
    turtle.left(1)
```

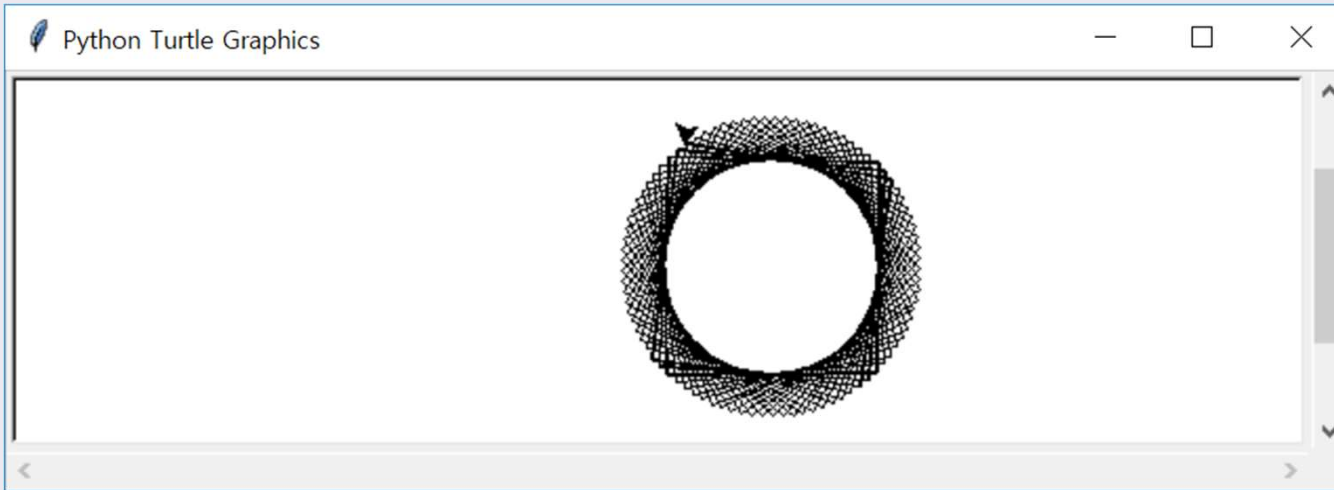
반복으로 다양한 모양 그리기



5P1 반복문으로 다각형 그리기 실습

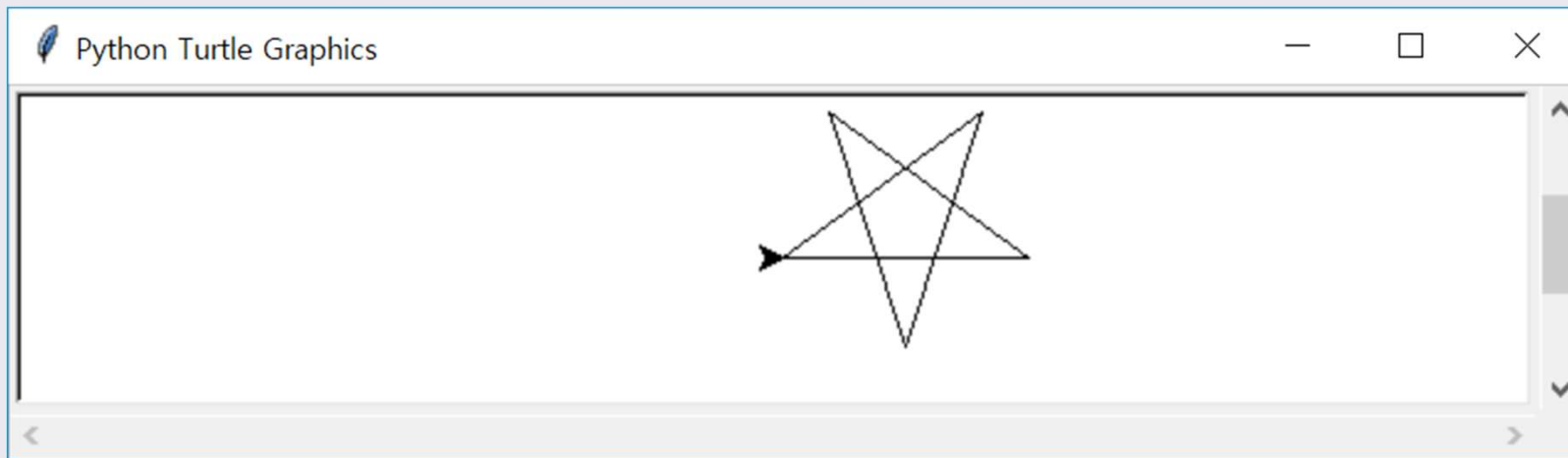
CH5 터틀 라이브러리

반복으로 다양한 모양 그리기



```
import turtle
for i in range(100):
    turtle.forward(100)
    turtle.left(89)
```

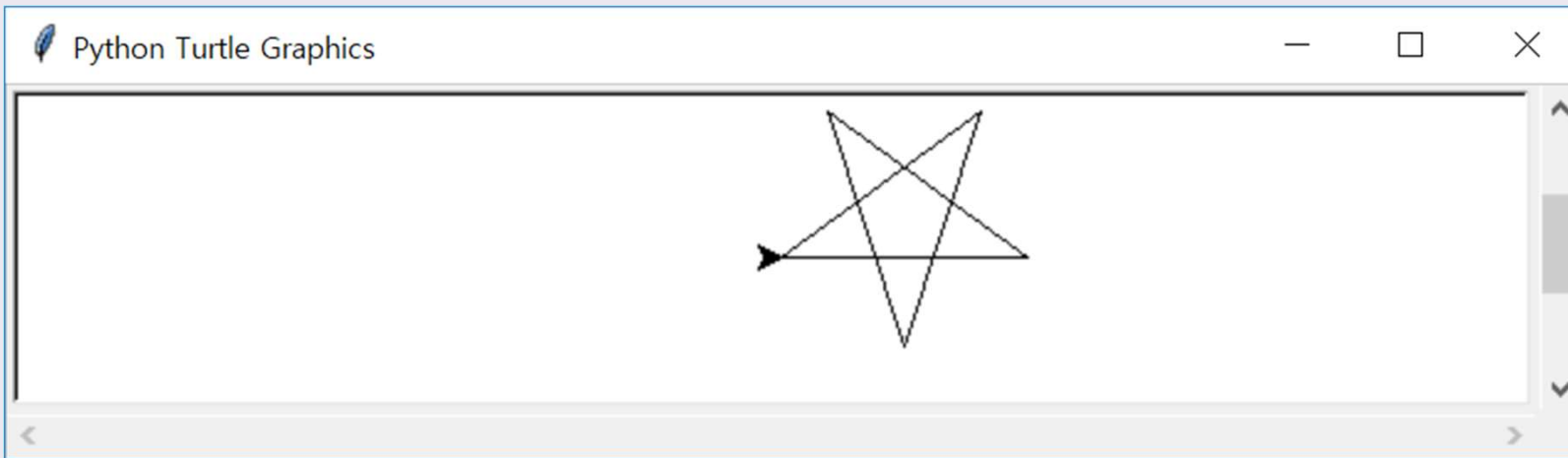
별 모양 그리기



5P1 반복문으로 다각형 그리기 실습

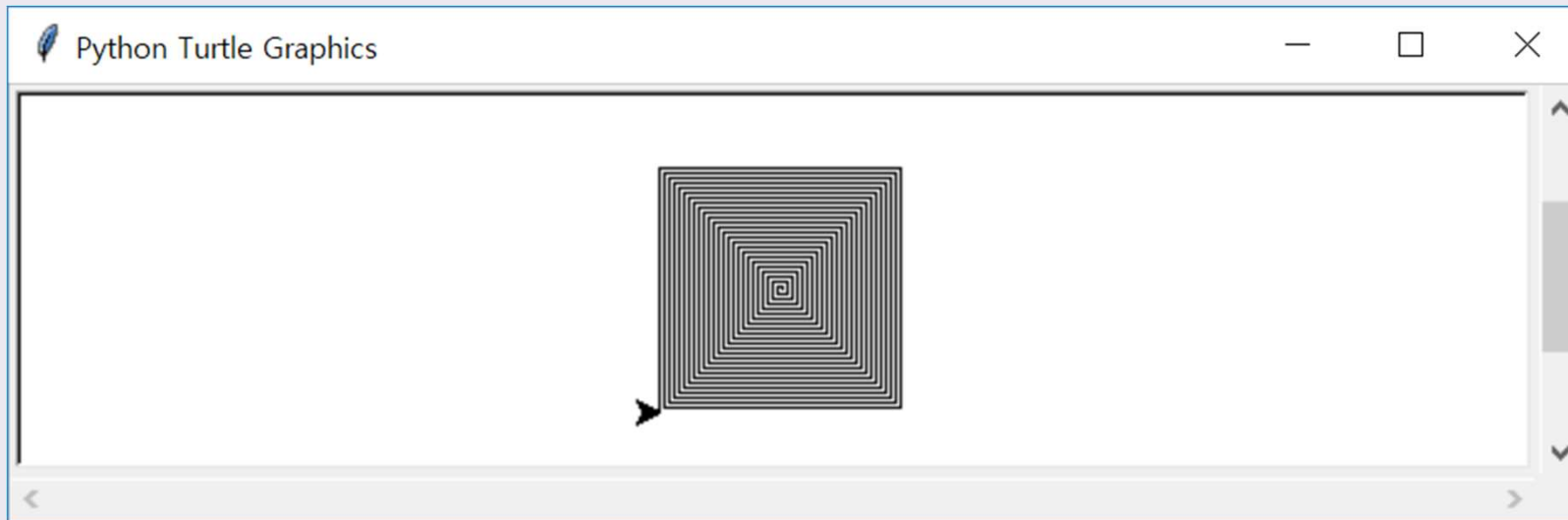
CH5 터틀 라이브러리

별 모양 그리기



```
import turtle
for i in range(5):
    turtle.forward(100)
    turtle.left(144)
```

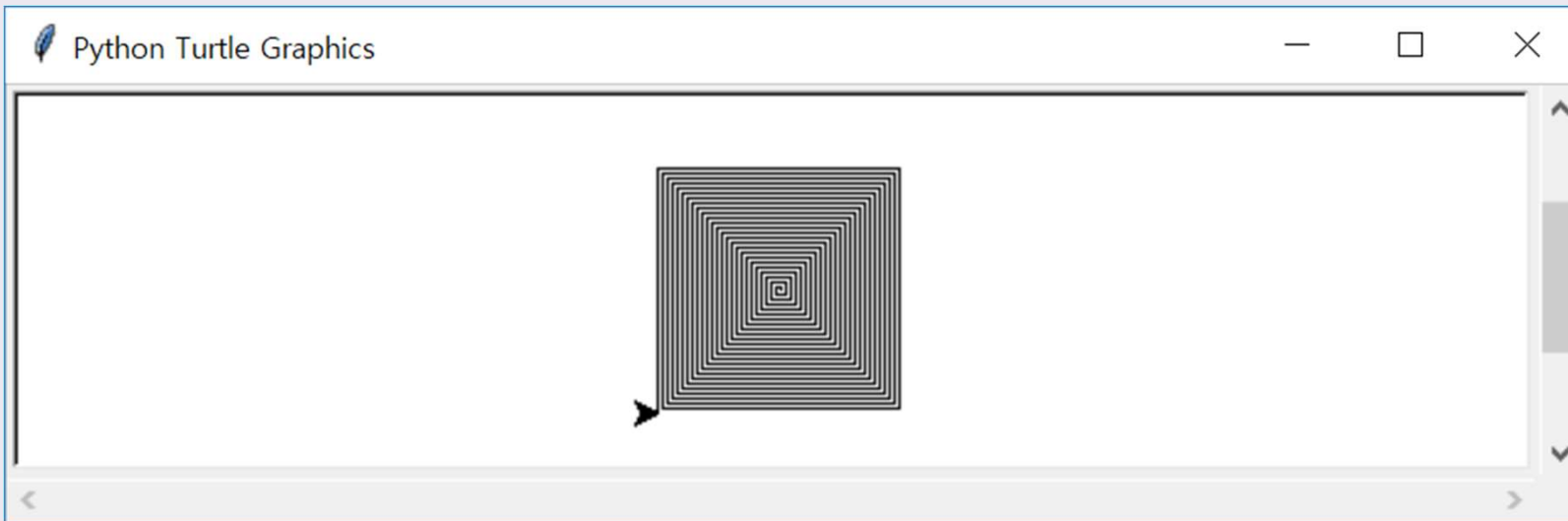

미로 그리기



5P1 반복문으로 다각형 그리기 실습

CH5 터틀 라이브러리

미로 그리기



```
import turtle
for i in range(100):
    turtle.forward(i)
    turtle.left(144)
```

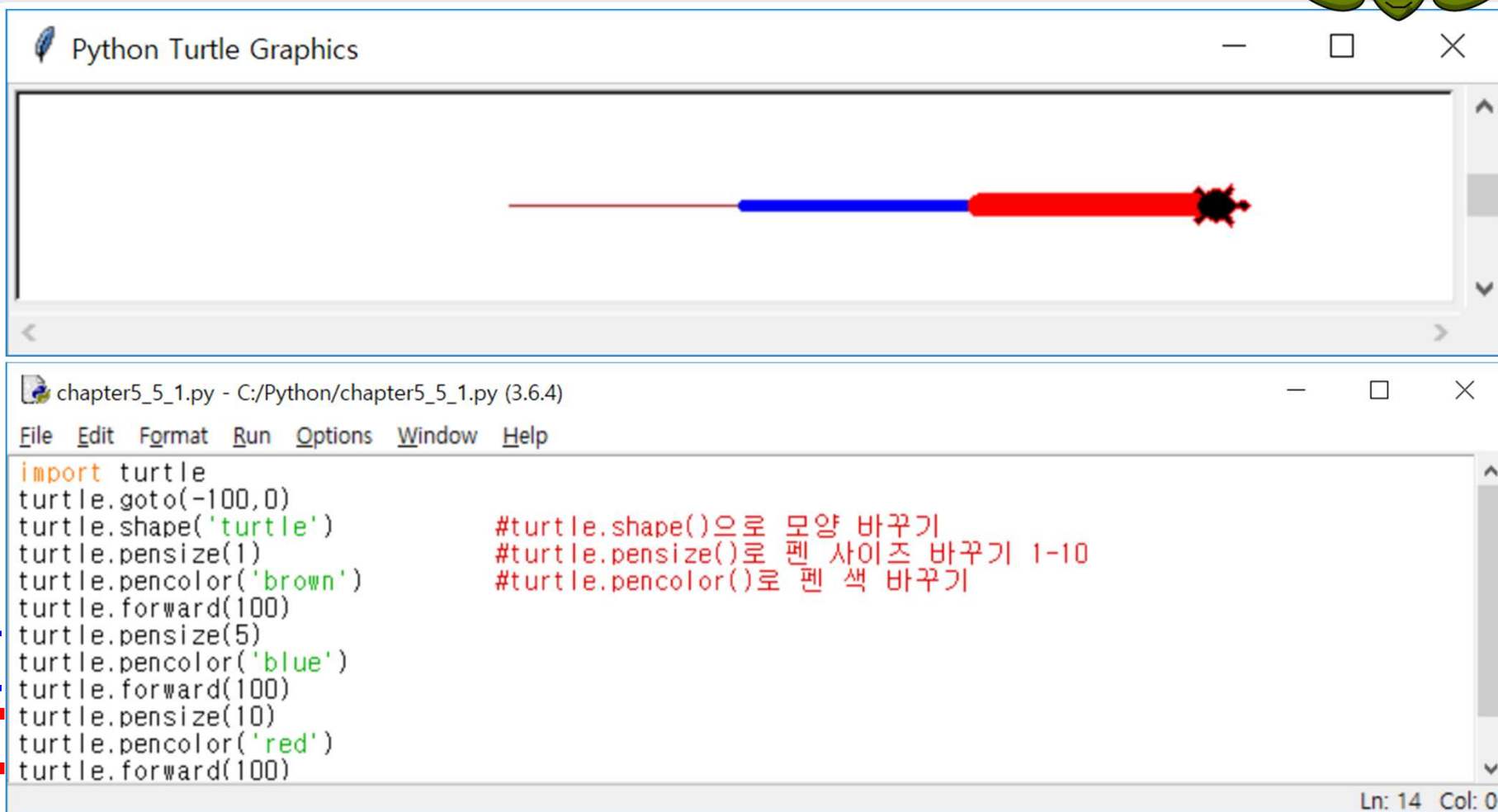


Tip 라이브러리(모듈) 이름 지정하기

라이브러리를 불러들일 때 `import turtle as t`라고 입력하면, `t.forward(100)`처럼 사용할 수 있다. 이는 `turtle`을 `t`라는 변수에 담아 사용하는 것과 같다. 또는 `t = turtle.Turtle()`처럼 입력하면 위와 같이 사용할 수 있다.

```
import turtle as t  # turtle를 t로 대체
t.fd(100)           # turtle.forward(100)
t.lt(90)            # turtle.left(90)
t.fd(100)
t.rt(90)            # turtle.right(90)
```

펜 옵션 사용하기



505

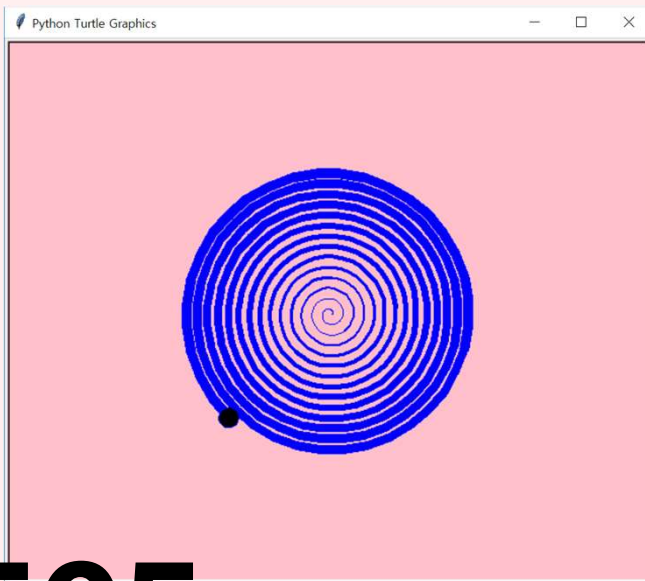
펜 모양 다양하게 만들기

CH5 터틀 라이브러리

펜 옵션 사용하기



명령어	설명	예시	범위
shape()	펜 모양	<code>turtle.shape('turtle')</code>	turtle, arrow, circle, square, triangle
pensize()	펜 사이즈	<code>turtle.pensize(5)</code>	1 - 10
pencolor()	펜 색	<code>turtle.pencolor('red')</code>	red, blue, brown, pink 등
bgcolor()	배경 색	<code>turtle.bgcolor('beige')</code>	red, blue, brown, pink 등



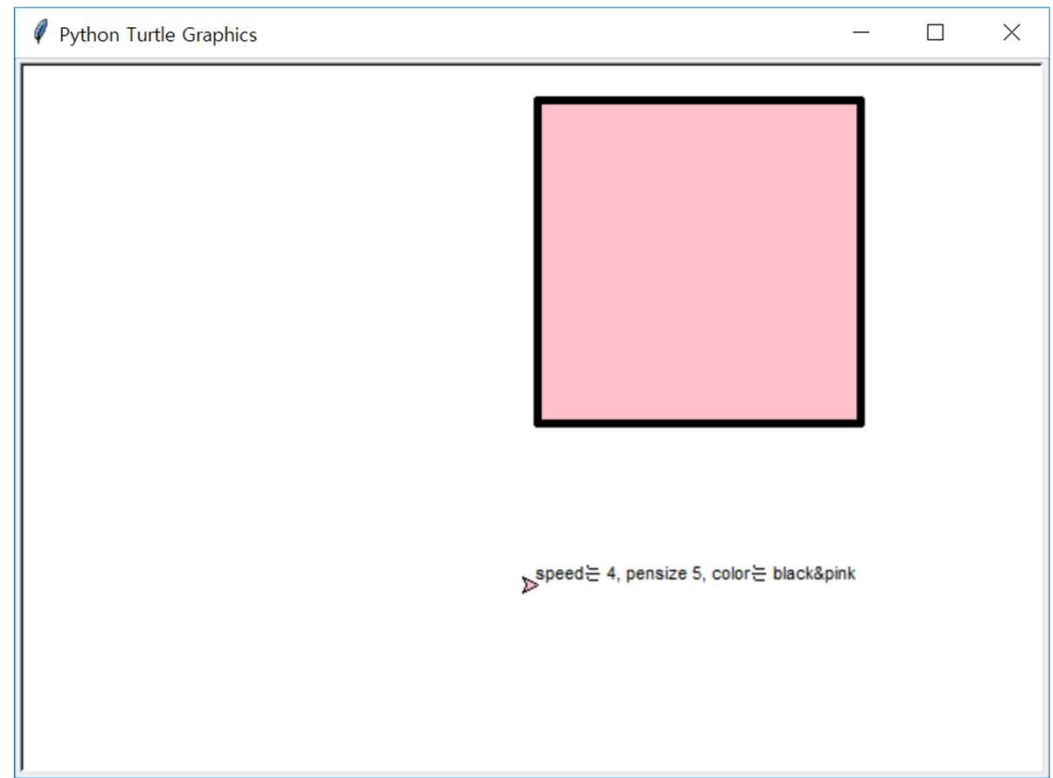
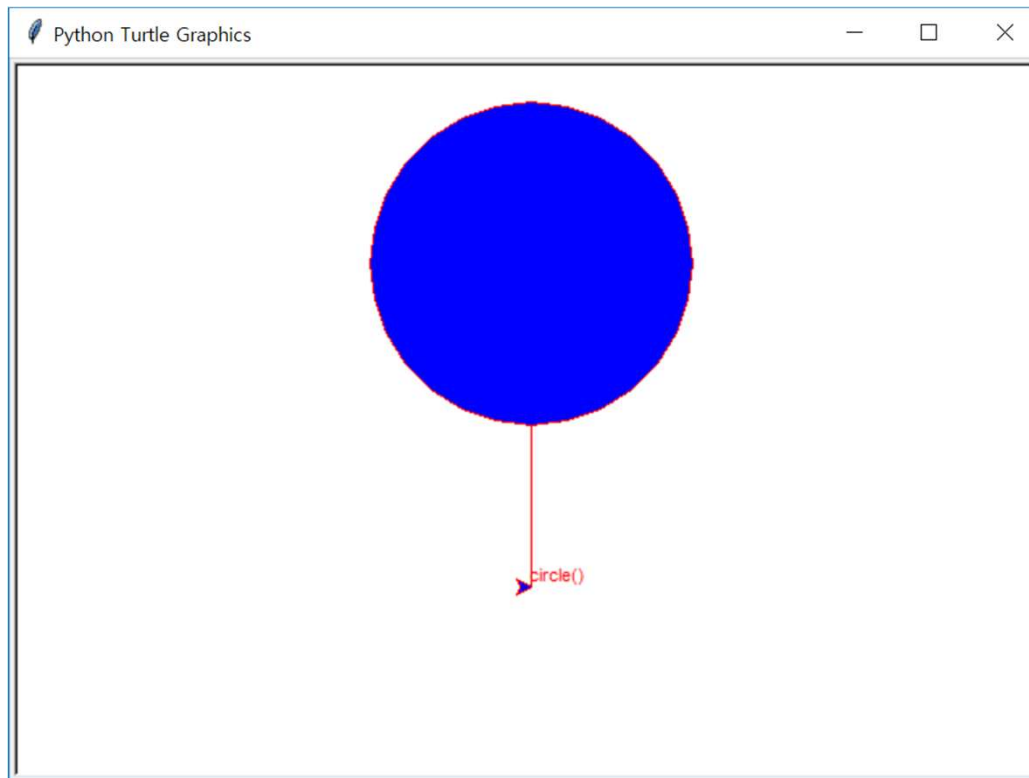
- 왼쪽과 같은 나선형의 모양을 만들어보자.
- 배경색은 핑크, 펜색은 파란이며 나선형으로 커지면서 펜 사이즈가 점점 커지도록 한다.
- 펜 모양은 원 모양으로 한다.

다양한 옵션 배우기



명령어	설명	예시
<code>clear()</code>	화면에 그려진 펜을 모두 지운다.	<code>turtle.clear()</code>
<code>reset()</code>	펜의 위치와 화면을 맨 처음 상태로 되돌린다.	<code>turtle.reset()</code>
<code>write()</code>	해당 위치에 글자를 쓴다.	<code>turtle.write("ABCDEFGH")</code>
<code>circle()</code>	반지름으로 원을 그립니다.	<code>turtle.circle(100)</code>
<code>color()</code> <code>begin_fill()</code> <code>end_fill()</code>	색을 설정해둔 후 도형이 만들어지면 도형에 색을 채운다.	<code>turtle.color('red','black')</code> #펜색, #채우기색 <code>turtle.begin_fill()</code> <code>turtle.circle(100)</code> <code>turtle.end_fill()</code>
<code>speed()</code>	그리는 속도	<code>turtle.speed(10)</code> #(0, 10:Max, 1:Min)

다양한 옵션 배우기



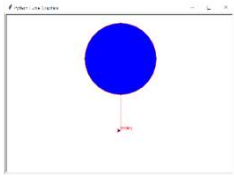
원 생성 후 내려와서 글자 쓰기 → 화면 삭제 → 사각형 생성 후 밑에서 글자 쓰기

506

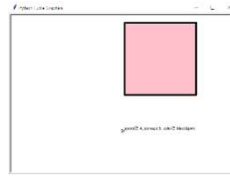
터틀 애니메이션 만들기

CH5 터틀 라이브러리

다양한 옵션 배우기



```
import turtle
turtle.speed(4)
turtle.pensize(5)
turtle.color('black', 'pink')
turtle.begin_fill()
for i in range(4): #사각형
    turtle.forward(200)
    turtle.left(90)
turtle.end_fill()
turtle.penup()
```



```
turtle.goto(0, -100)
turtle.write("speed 는 4, pensize 5,
color는 black&pink")
turtle.forward(200)
turtle.reset() #다시 시작, 전부 reset
turtle.color('red', 'blue')
turtle.begin_fill()
turtle.circle(100)
turtle.end_fill()
turtle.goto(0, -100)
turtle.write("circle()")
```


다양한 명령어 살펴보기



명령어	설명	예시
터틀 움직임		
forward()	앞으로 ()만큼 가기	turtle.forward(100) / turtle.fd(100)
back()	위로 ()만큼 가기	turtle.back(100) / turtle.bk(100)
left()	왼쪽으로 ()도만큼 회전하기	turtle.left(90) / turtle.lt(100)
right()	오른쪽으로 ()도만큼 회전하기	turtle.right(90) / turtle.rt(100)
speed()	그리는 속도	turtle.speed(10) #(0, 10:Max, 1:Min)
goto()	해당 좌표로 이동하기	turtle.goto(100,100)

다양한 명령어 살펴보기



펜 컨트롤

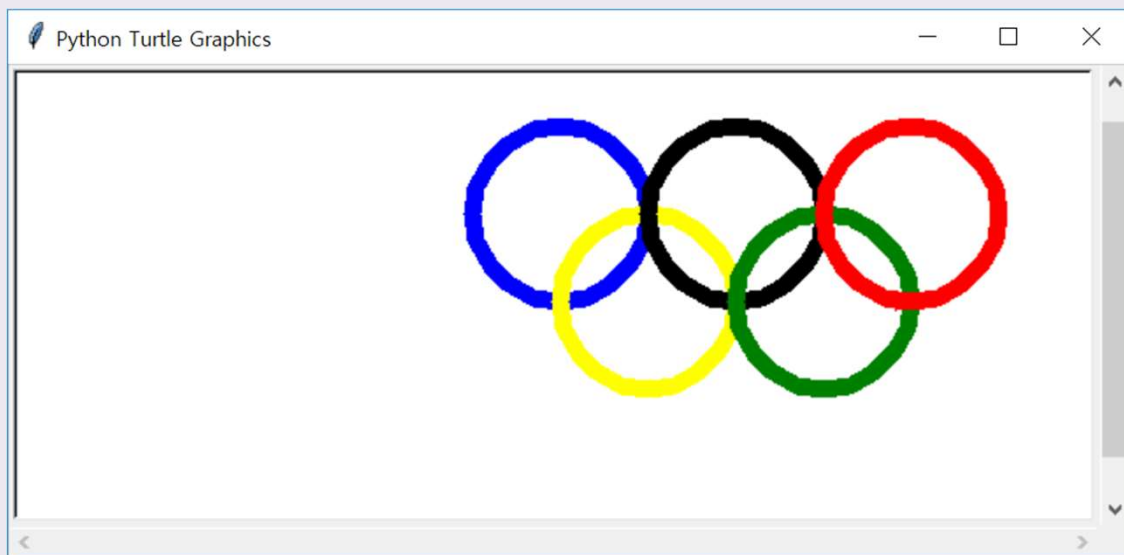
pendown()	펜을 내려 그리기	turtle.pendown(), pd(), down()
penup()	펜을 떼고 그리지 않기	turtle.penup(), pu(), up()
shape()	펜 모양	turtle.shape('turtle')
pensize()	펜 사이즈	turtle.pensize(5), width()
pencolor()	펜 색	turtle.pencolor('red')
color() begin_fill() end_fill()	색을 설정해둔 후 도형이 만들어 지면 도형에 색을 채운다.	turtle.color('red', 'black') #펜색, #채우기색 turtle.begin_fill() turtle.circle(100) turtle.end_fill()
write()	해당 위치에 글자를 쓴다.	turtle.write("ABCDEFGH")
circle()	반지름으로 원을 그립니다.	turtle.circle('100')

다양한 명령어 살펴보기



스크린		
bgcolor()	배경 색	turtle.bgcolor('beige')
clear()	화면에 그려진 펜을 모두 지운다.	turtle.clear()
reset()	펜의 위치와 화면을 맨 처음 상태로 되돌린다.	turtle.reset()
title()	터틀 그래픽 실행창의 제목을 정한다.	turtle.title("제목표시줄 제목")
기타		
listen()	키 이벤트 등을 혼용할 경우 터틀 그래픽을 대기시킨다.	turtle.listen()
mainloop()	터틀 그래픽 프로그램을 종료하지 않고 지속한다.	turtle.mainloop()

오륜기 그리기



오륜기 그리기



```
import turtle
turtle.pensize(10) #첫 번째 원
turtle.color("blue")
turtle.circle(50)
```

```
turtle.penup() #두 번째 원
turtle.goto(50,-50)
turtle.pendown()
turtle.color("yellow")
turtle.circle(50)
```

```
turtle.penup() #세 번째 원
turtle.goto(100,0)
turtle.pendown()
turtle.color("black")
turtle.circle(50)
```

```
turtle.penup() #네 번째 원
turtle.goto(150,-50)
turtle.pendown()
turtle.color("green")
turtle.circle(50)
```

```
turtle.penup() #다섯 번째 원
turtle.goto(200,0)
turtle.pendown()
turtle.color("red")
turtle.circle(50)
```

오륜기 그리기



```
import turtle
turtle.pensize(10)
XYC = [(0,0,"blue"), (50,-50,"yellow"), (100,0,"black"), (150,-50,"green"),
(200,0,"red")]
for i,j,k in XYC:
    turtle.penup()
    turtle.goto(i,j)
    turtle.pendown()
    turtle.color(k)
turtle.circle(50)
```

도형 자판기



```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Python/chapter5_8_2.py =====
도형을 몇 번 만드시겠습니까? 3
만들고 싶은 도형은? (x)각형 5
각 변의 길이는 얼마로 하시겠습니까? 100
선의 굵기는 얼마로 하시겠습니까? 10
채우기 색은 무엇으로 설정하시겠습니까? red
배경 색은 무엇으로 설정합니까? pink
>>>
```

Ln: 116 Col: 4

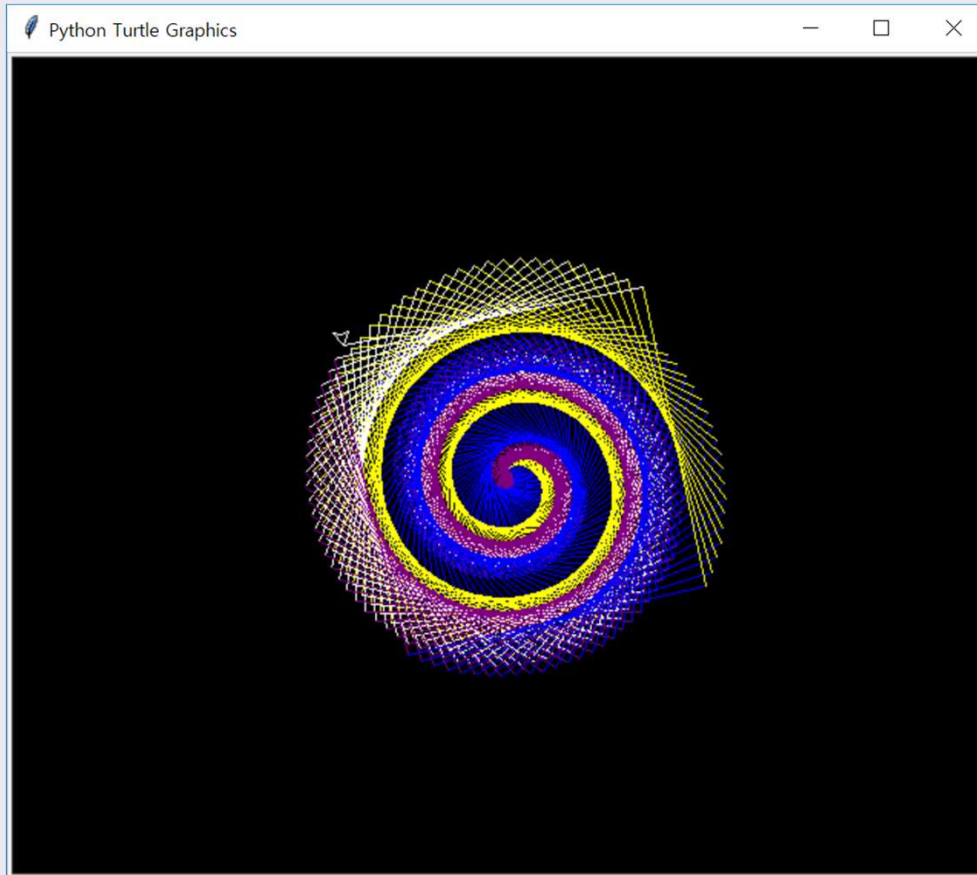
도형 자판기



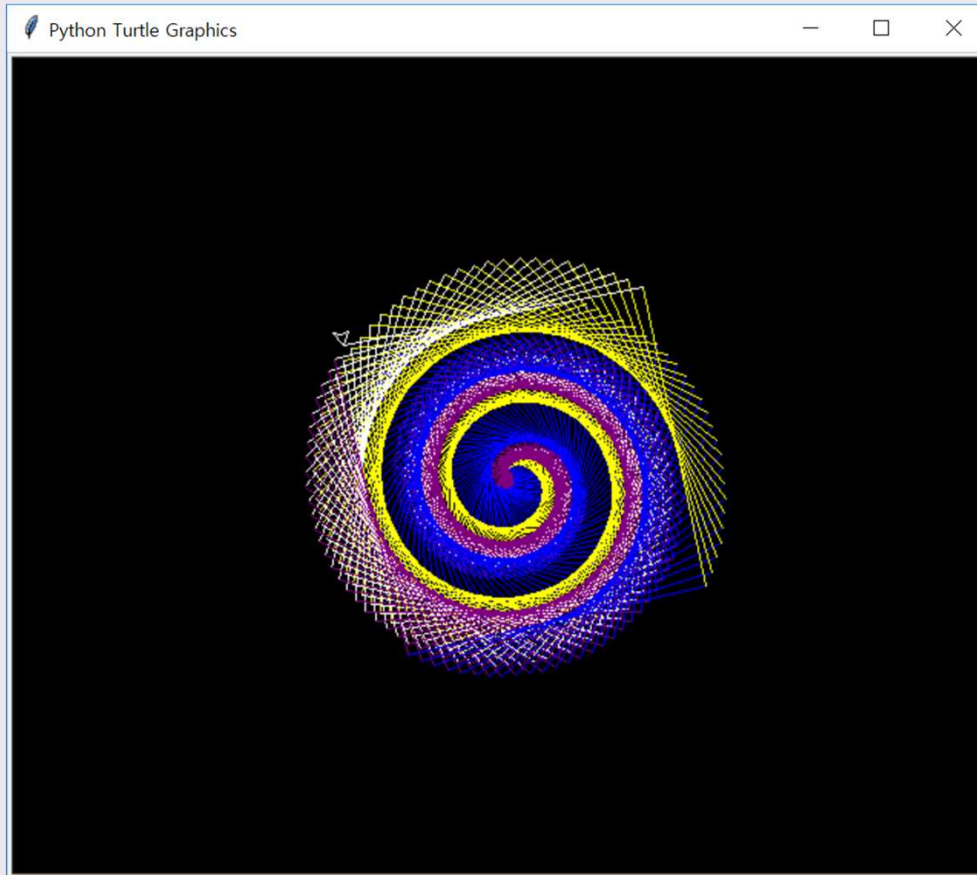
```
import turtle
m1 = int(input("도형을 몇개 만드시겠습니까? "))
m2 = int(input("만들고 싶은 도형은? (x)각형 "))
m3= int(input("각 변의 길이는 얼마로 하시겠습니까? "))
m4 = int(input("선의 굵기는 얼마로 하시겠습니까? "))
m5 = input("채우기 색은 무엇으로 설정하시겠습니까? ")
m6 = input("배경 색은 무엇으로 설정합니까? ")
turtle.bgcolor(m6)
```

```
for i in range(m1):
    turtle.clear()
    turtle.width(m4)
    turtle.fillcolor(m5)
    turtle.begin_fill()
    for s1 in range(m2):
        turtle.fd(m3)
        turtle.lt(360/m2)
    turtle.end_fill()
```


반복을 이용한 새로운 모양 만들기

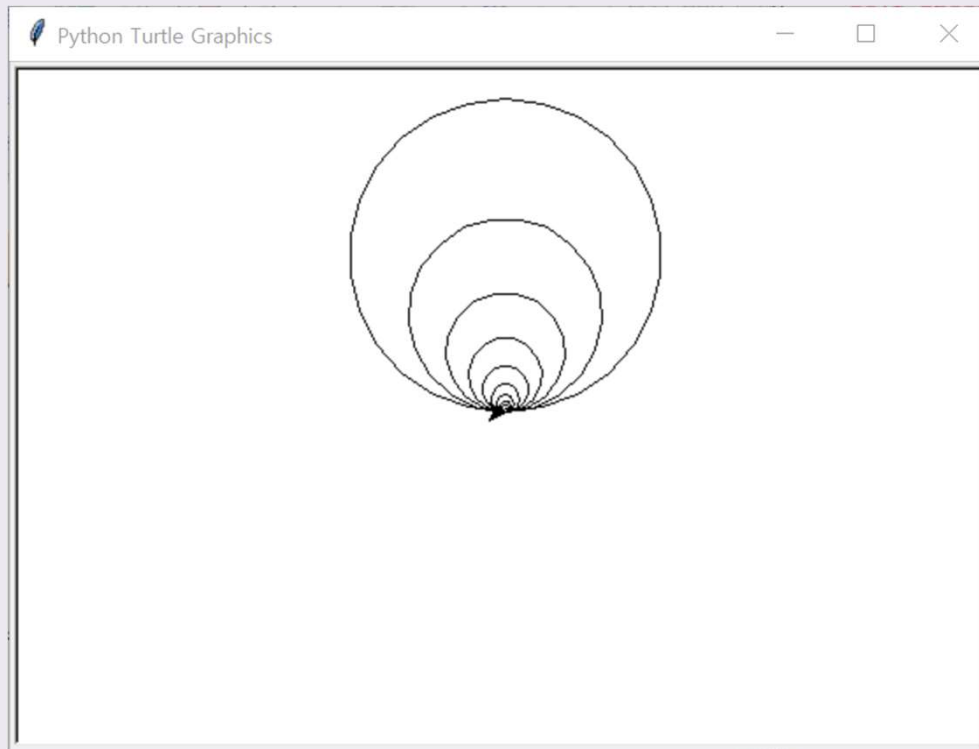


반복을 이용한 새로운 모양 만들기

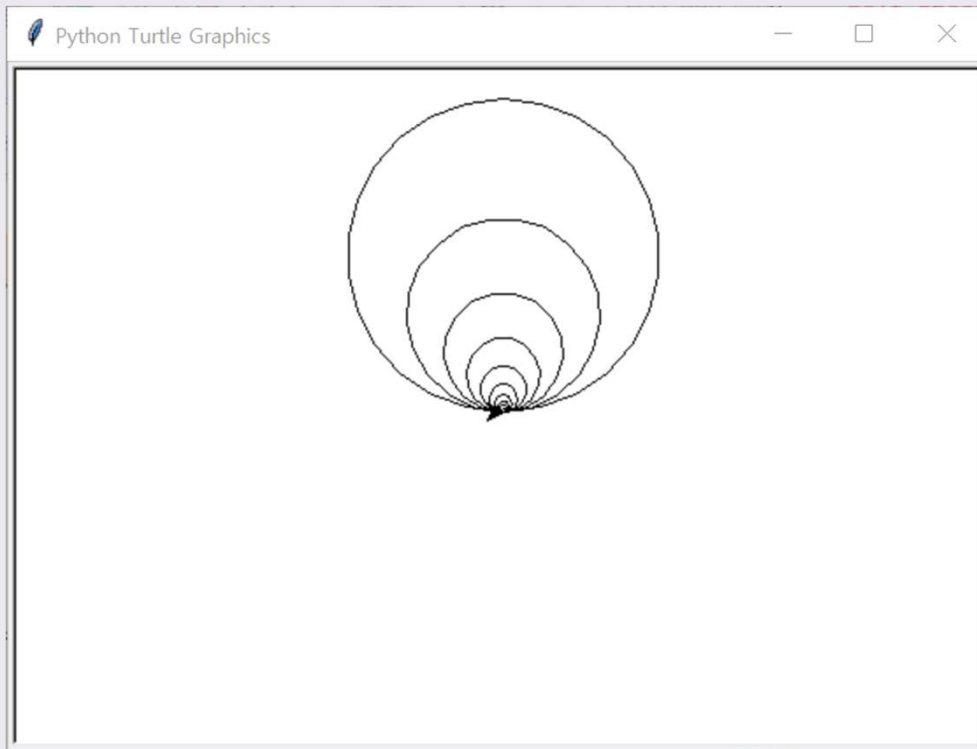


```
import turtle
color2=['purple','blue','yellow','white']
turtle.bgcolor("black")
turtle.speed(0)
for i in range(200):
    for j in color2:
        turtle.pencolor(j)
        turtle.forward(i)
        turtle.left(89)
```

피보나치형 원



피보나치형 원



```
import turtle
turtle.speed(0)
a=0; b=1

for i in range(10):
    c = a+b
    turtle.circle(c)
    a = b
    b = c
```

5. 터틀 라이브러리(Turtle Library)

01 그래픽 활용(Turtle 라이브러리)
02 터틀 움직이기
03 goto()와 pendown(), penup()
04 반복문으로 다각형 그리기
P1 반복문으로 다각형 그리기 실습

05 펜 모양 다양하게 만들기
06 터틀 애니메이션 만들기
07 터틀 명령어 정리 및 약어

P2 터틀 실습
P3 터틀 실습
P4 터틀 실습
P5 터틀 실습

