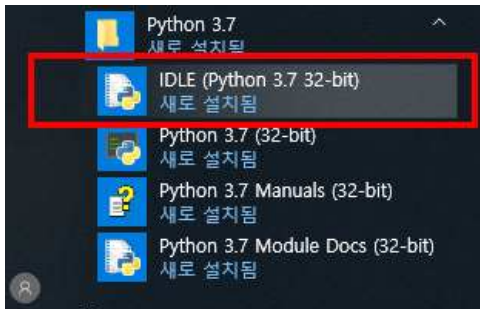
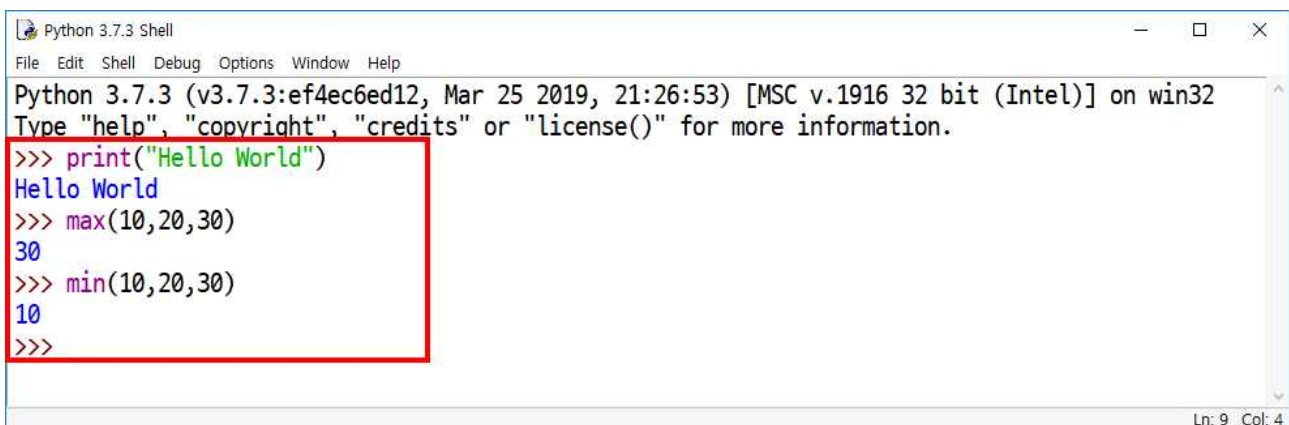


1. 파이썬(Python) 설치하고 Shell 사용하기

- 1) <http://www.python.org/> 파이썬 3.x.x 윈도우 인스톨러를 다운로드 하여 설치한다.
- 2) 파이썬 통합 개발환경인 IDLE(Python GUI)을 실행한다.



- 3) 세 개의 꺾은 괄호(>>>)를 프롬프트라고 한다.
- 4) >>>print("Hello World") 입력후 엔터를 누른다. => 실행확인



- 5) 셸에서는 한 줄 한 줄 바로 결과를 알 수 있고 **Ctrl+C** => **Ctrl+V**를 사용할 수 있다.

※ 프로그래밍시 유의할 점

- 1) 영어 대소문자를 구분한다. python ≠ Python ≠ PyThon ≠ PYTHON
- 2) 문자입력시 “ ” 혹은 ‘ ’로 묶어서 사용한다.
- 3) 들여쓰기가 중요하다.



1. 파이선은 배우고 사용하기 쉬운 언어이다.

파이선은 다른 프로그래밍 언어에 비해 간결하고 이해하기 쉽다.

예를 들어 `print("Hello!")`라는 명령어는 'Hello!'라는 문자열을 출력하는 명령어로, 간단하게 문자열을 출력할 수 있다.

```
>>> print("Hello!")
Hello!
```

2. 파이선은 바로 사용이 가능한 명령어들이 많이 저장된 언어이다.

파이선은 사용자들이 원하는 기능을 바로 사용할 수 있도록 많은 명령어를 충실하게 제공하고 있다.

예를 들어 최댓값을 구하고 싶을 때, 파이선에서 미리 만들어 저장한 내장 함수(built-in function)인 `max()`를 이용하면 쉽게 구할 수 있다.

```
>>> max(10, 30)
30
>>> max(3, 10, 200, 50)
200
>>> max(10.5, 20, -30)
20
```

3. 파이선은 다양한 프로그램을 만들 수 있는 강력한 언어이다.

많은 연구 기관과 기업이 파이선을 사용하고 있고, 우리가 사용하는 주변의 많은 프로그램도 파이선으로 만들어지거나 파이선의 기능을 사용하였다.



2. 파이썬의 거북이(turtle)를 이용하여 그림을 그리는 방법

1) 캔버스에 거북이 불러오기

명령어	설명
<code>import turtle</code>	거북이 모듈 불러오기
<code>t=turtle.Turtle()</code> 또는 <code>t=turtle.Pen()</code>	거북이 만들기 (거북이 이름: t)

2) 거북이 움직이기

명령어	설명
<code>t.forward()</code>	거북이 전진하기
<code>t.backward()</code>	거북이 후진하기
<code>t.left()</code>	거북이 왼쪽으로 회전하기
<code>t.right()</code>	거북이 오른쪽으로 회전하기

거북이 움직이기

```
>>> import turtle
>>> t=turtle.Pen()
>>> t.forward(50)
>>> t.left(90)
>>> t.forward(50)
>>> t.left(90)
```



실행 결과

3) 2개의 선 그리기

명령어	설명
t.penup() up() pu()	거북이 위로 올리기(그림을 그리지 않게 하기)
t.pendown() down() pd()	거북이 내리기(그림을 그리게 하기)

2개의 선 그리기

```
>>> import turtle
>>> t=turtle.Pen()
>>> t.forward(200)
>>> t.penup()
>>> t.left(90)
>>> t.forward(30)
>>> t.right(90)
>>> t.pendown()
>>> t.forward(50)
```

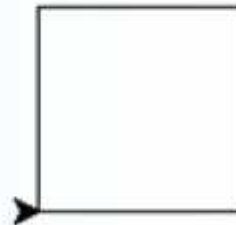


실행 결과

4) 정사각형 그리기

정사각형 그리기

```
>>> import turtle
>>> t=turtle.Pen()
>>> t.forward(80)
>>> t.left(90)
>>> t.forward(80)
>>> t.left(90)
>>> t.forward(80)
>>> t.left(90)
>>> t.forward(80)
>>> t.left(90)
```



실행 결과

3. 파이선의 거북이(turtle)를 이용하여 색칠하는 방법

1) 컬러 비트맵 이미지

(R, G, B) 값	색상
(0, 0, 0)	BLACK
(0, 0, 255)	BLUE
(0, 255, 255)	CYAN
(255, 255, 255)	WHITE
(255, 0, 255)	MAGENTA
(255, 0, 0)	RED
(0, 255, 0)	GREEN
(255, 255, 0)	YELLOW



2) 도형에 색칠하기

원 테두리 색칠하기

```
>>> import turtle
>>> t=turtle.Pen()
>>> t.color("green") # 녹색
>>> t.circle(50)
```



실행 결과

▶ RGB 색상 비율 이용하기

color(0, 0, 0) ~ color(1, 1, 1) 예 t.color(1, 0.5, 1)

▶ RGB 색상값 이용하기

color("#000000") ~ color("#ffffff") 예 t.color("#ff00ff")

● 도형 색칠하기

도형 안에 색을 칠하기 위해서는 `color()` 함수로 색을 지정해야 할 뿐 아니라, `begin_fill()`, `end_fill()` 명령어를 이용해야 한다.

명령어	설명
<code>t.begin_fill()</code>	RGB 색상으로 도형 안을 채우기 위해 시작하는 함수
<code>t.end_fill()</code>	<code>begin_fill()</code> 함수 호출 후 색을 채우고 종료하는 함수

원 색칠하기

```
>>> import turtle
>>> t=turtle.Pen()
>>> t.color("orange") # 주황색
>>> t.begin_fill()
>>> t.circle(50)
>>> t.end_fill()
```



실행 결과



기본 거북이
`t.shape("classic")`



진짜 거북이
`t.shape("turtle")`



세모 거북이
`t.shape("triangle")`

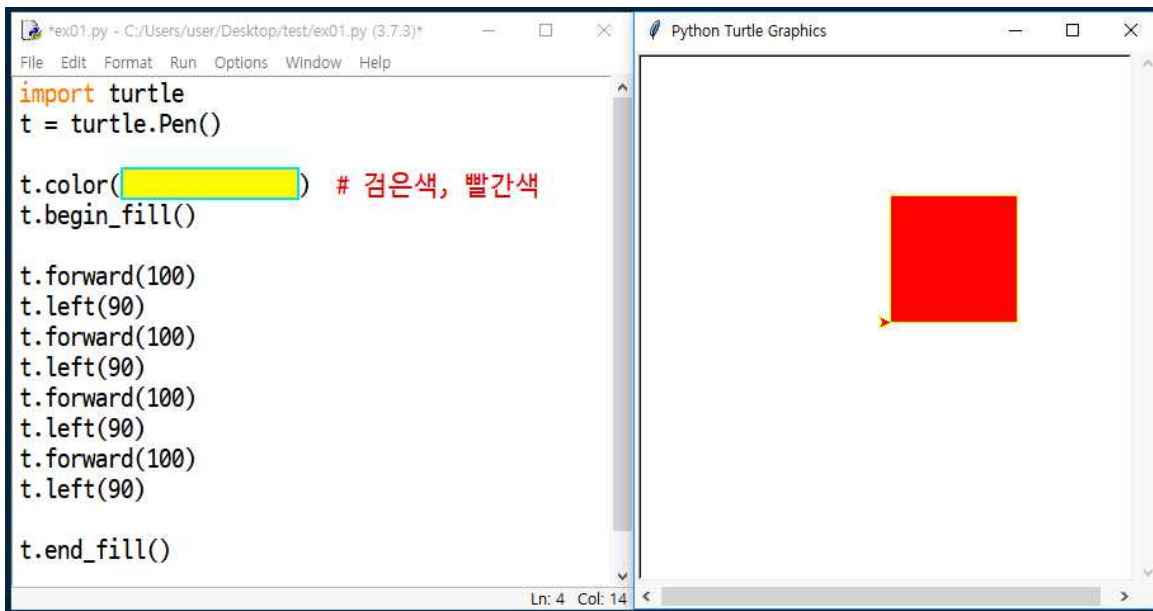
Copyright © Corbis Inc. All rights reserved.

그림 6-2 다양한 거북이 모양을 지정할 수 있습니다.

[응용과제-1] 사각형의 테두리와 내부 색칠하기

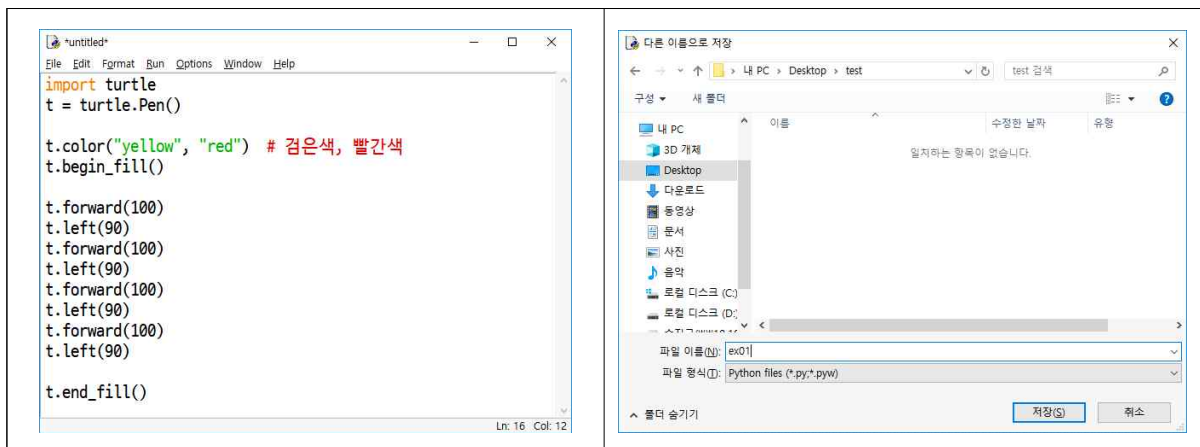
도형의 테두리와 도형 내부의 색을 다르게 지정하려면 다음과 같이 `color()` 함수 안에 색을 2개 지정해 준다.

명령어	설명
<code>t.color(색1, 색2)</code>	색1(테두리 색), 색2(도형 내부의 색)를 지정하는 함수



※ 파이썬 에디터 사용하기

- 바탕화면에 “test” 폴더 만들기
- [Python 3.7.3 Shell] → [File] → [New File]
- [File] → [Save As...] : ex01

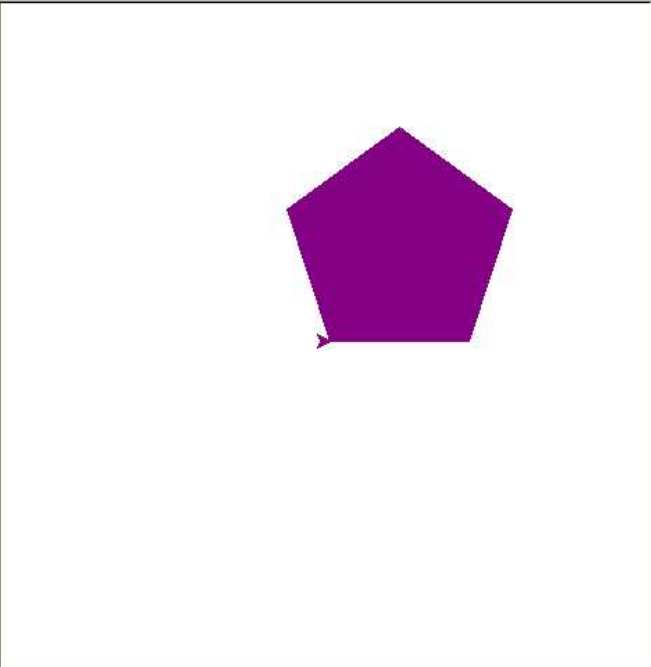


- [Run] → [Run Module F5]

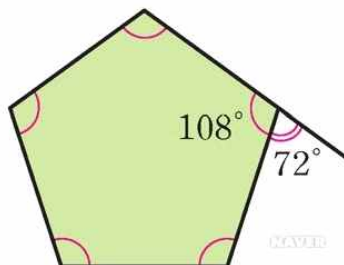
[응용과제-2] 정오각형 그리기

```
ex02.py - C:/Users/user/Desktop/test/ex02.py (...)  
File Edit Format Run Options Window Help  
import turtle  
t = turtle.Pen()  
  
t.color("purple")  
t.begin_fill()  
  
t.forward(100)  
t.left(72)  
t.forward(100)  
t.left(72)  
t.forward(100)  
t.left(72)  
t.forward(100)  
t.left(72)  
t.forward(100)  
t.left(72)  
  
t.end_fill()
```

Ln: 10 Col: 0



- 정오각형의 내각은?



[응용과제-3] 정팔각형 그리기

```
ex03.py - C:/Users/user/Desktop/test/ex03.py (...)
```

```
File Edit Format Run Options Window Help
```

```
import turtle
t = turtle.Pen()


n = 8

t.color("purple")
t.begin_fill()

t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )
t.forward(50)
t.left( )

t.end_fill()
```

Ln: 24 Col: 15



- 정n각형의 내각은?

[응용과제-4] 도형 그리기

```
ex04.py - C:/Users/user/Desktop/test/ex04.py (...)
```

```
File Edit Format Run Options Window Help
```

```
import turtle
t=turtle.Turtle()

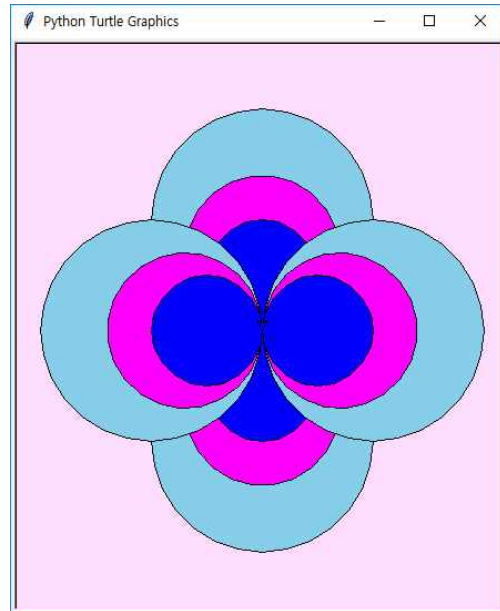
ts = turtle.getscreen()
ts.bgcolor("#FFDDFF")

t.fillcolor('sky blue')
t.begin_fill()
t.circle(100)
t.end_fill()

t.fillcolor('magenta')
t.begin_fill()
t.circle(70)
t.end_fill()

t.fillcolor('blue')
t.begin_fill()
t.circle(50)
t.end_fill()
```

Ln: 20 Col: 12



[힌트]

t.circle(100) : 터틀이 바라보고 있는 방향에서 반시계 방향으로 반지름이 100인 원을 그림

t.circle(-100) : 터틀이 바라보고 있는 방향에서 시계 방향으로 반지름이 100인 원을 그림

배경색 지정하기

```
ts = turtle.getscreen()
ts.bgcolor("#FFDDFF")
```

아래쪽에 원 그리기

```
t.circle(-100)
```

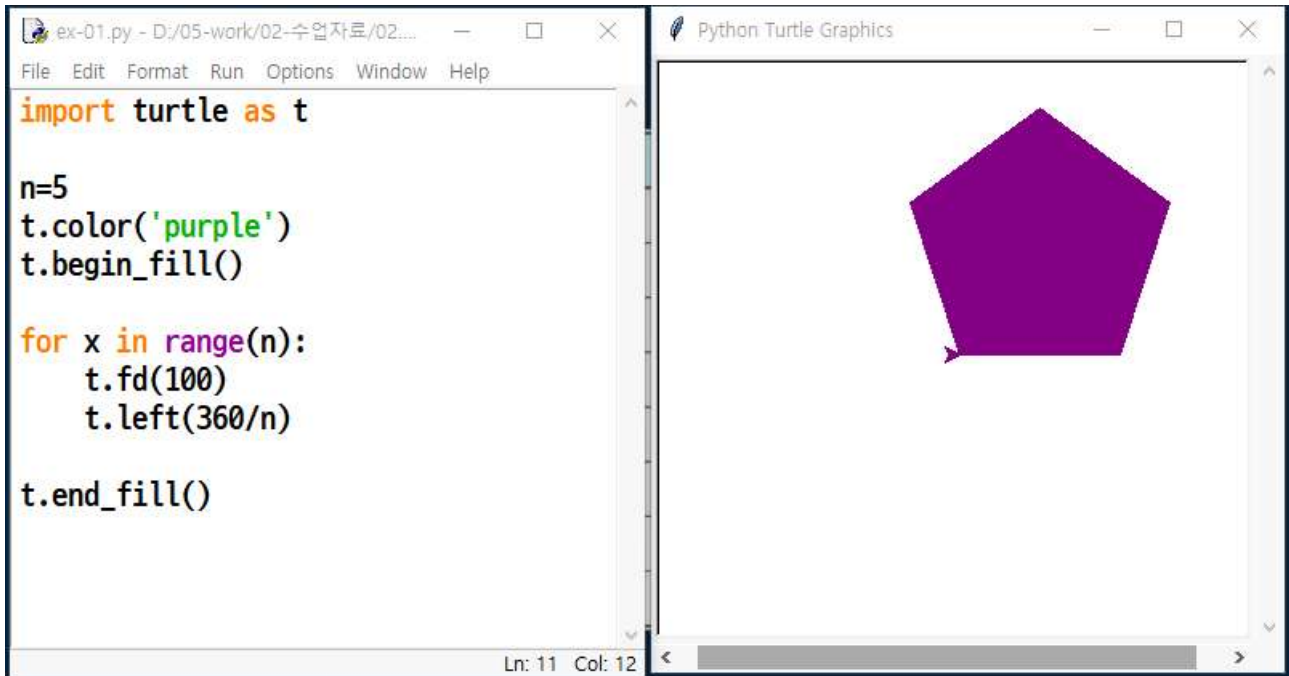
왼쪽에 원 그리기

```
t.left(90)
```

오른쪽에 원 그리기

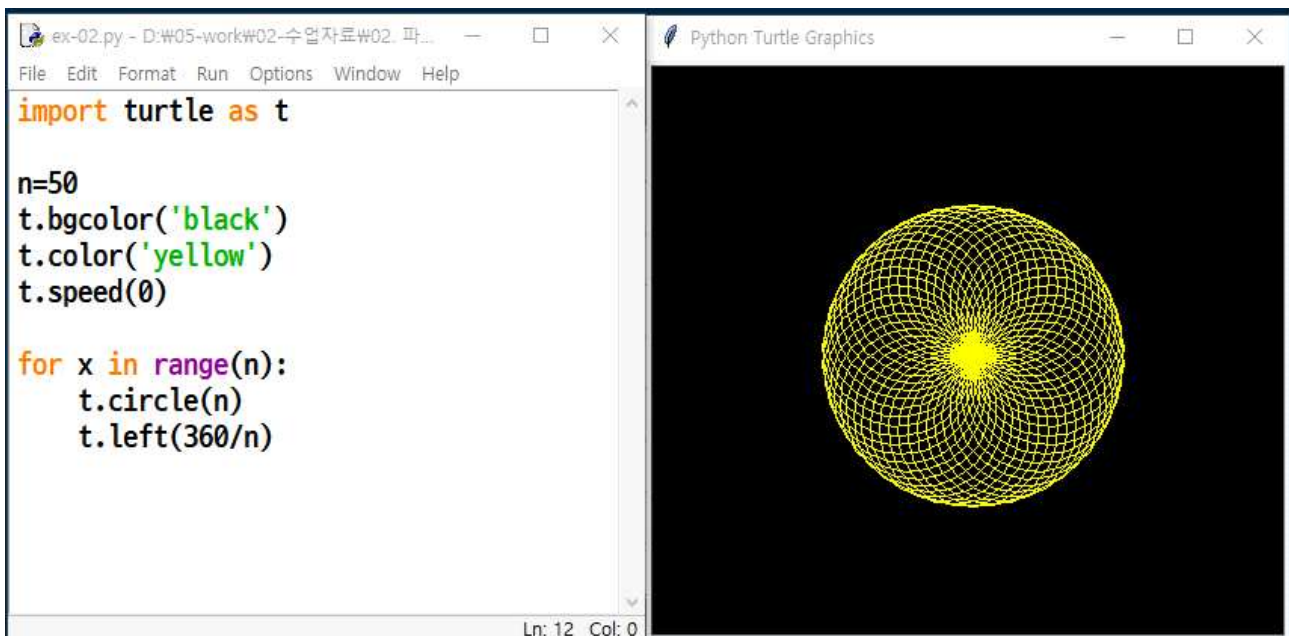
```
t.circle(-100)
```

[도전과제-1] 거북이 그래픽으로 그림그리기 - 정오각형 그리기



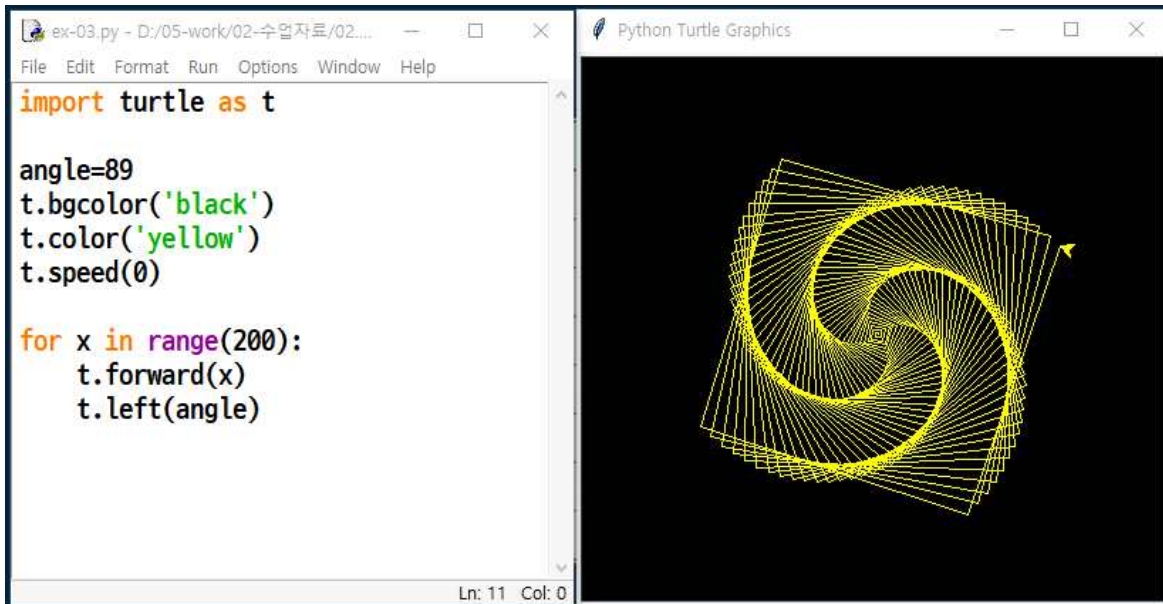
- import turtle as t : turtle 모듈을 불러오고 turtle 대신 t를 사용한다.

[도전과제-2] 거북이 그래픽으로 그림그리기 - 원을 반복해서 그리기

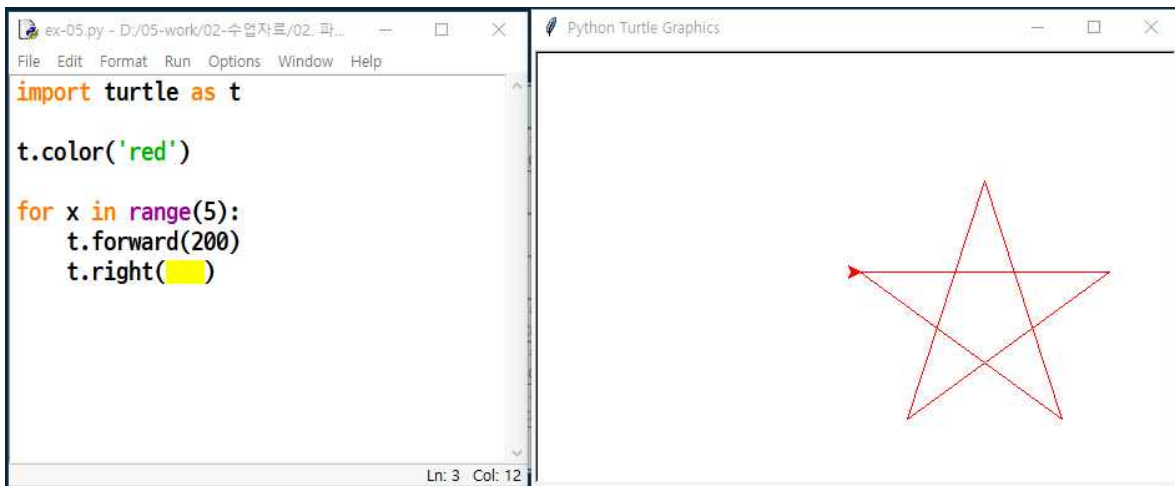


- t.speed(속도)
거북이가 움직이는 속도 : 1이 느리고 10이 빠름, 0이 최고속도

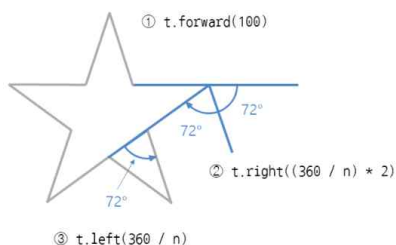
[도전과제-3] 거북이 그래픽으로 그림그리기 - 선을 반복해서 그리기



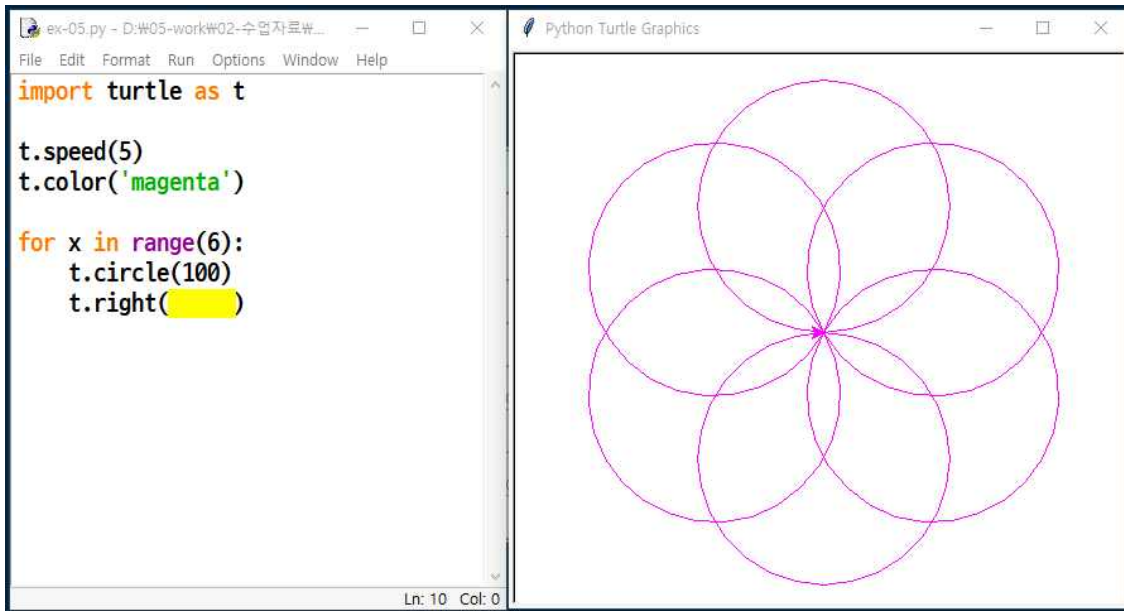
[도전과제-4] 거북이 그래픽으로 그림그리기 - 별 그리기



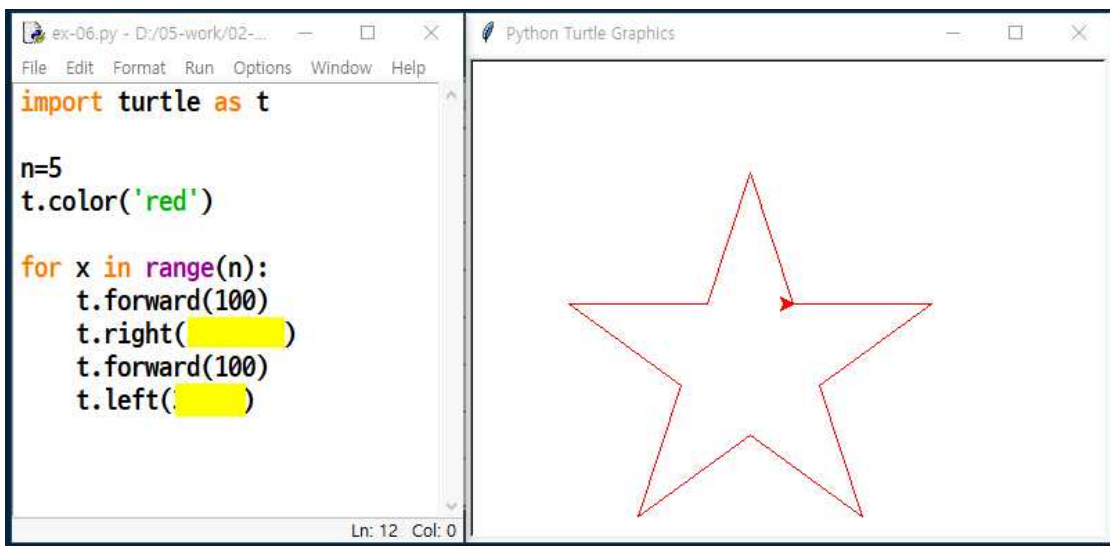
힌트



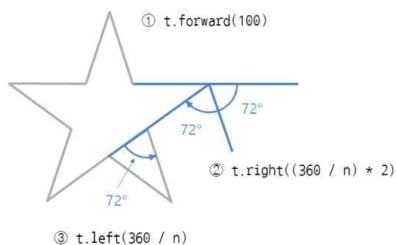
[도전과제-5] 거북이 그래픽으로 그림그리기 - 6개의 원 그리기



[도전과제-6] 거북이 그래픽으로 그림그리기 - 오각별 그리기



힌트



[도전과제-7] 태극 문양 그리기

```

taeguk-1.py - C:/Users/user/Desktop/test/...
File Edit Format Run Options Window Help
import turtle as t

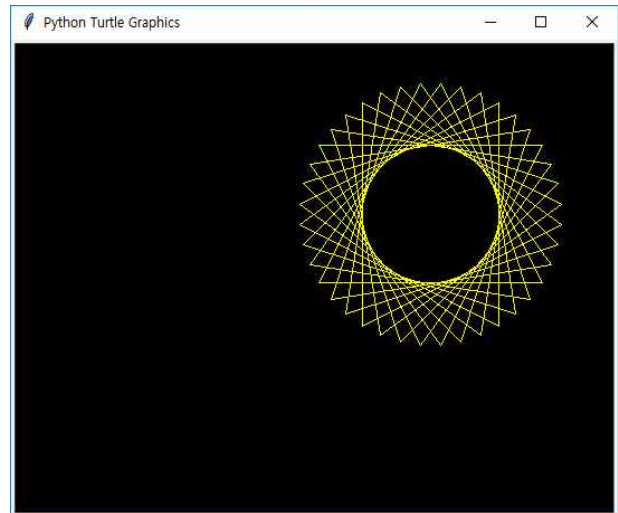
내각크기 = 63
외각크기 = 180 - 내각크기
선분개수 = 40
선분길이 = 200

t.speed(0)
t.bgcolor('black')
t.pencolor('yellow')

for i in range(선분개수):
    t.fd(선분길이)
    t.left(외각크기)

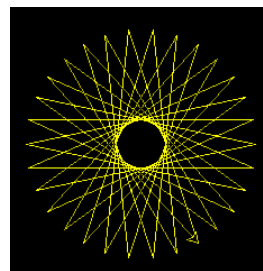
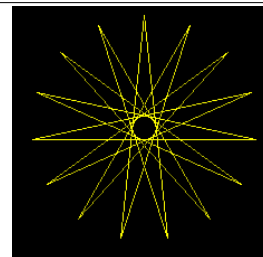
t.hideturtle()
t.done()
Ln: 18 Col: 0

```



선분의 개수를 400으로 하고,
내각의 크기를 60이 아닌 59, 61, 62 등으로 변경
해 보자.

내각의 크기를 360의 약수인 12의 배수를 이용하
여 실행해 보자. (12, 24, 36)




```

taeguk-3.py - C:/Users/user/Desktop/t...
File Edit Format Run Options Window Help
import turtle as t

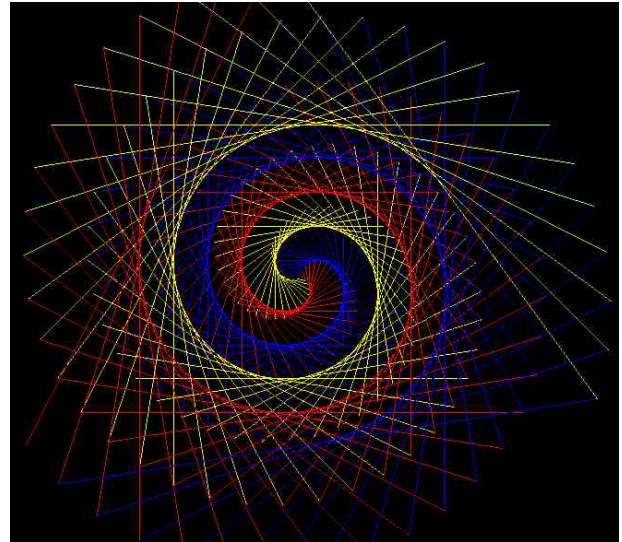
내각크기 = 63
외각크기 = 180 - 내각크기
선분개수 = 200

t.speed(0)
t.bgcolor('black')
t.pencolor('yellow')
t.fillcolor('orange')

for i in range(선분개수):
    선분길이 = i*3
    if i%3 == 1:
        t.pencolor('red')
    elif i%3 == 2:
        t.pencolor('blue')
    else:
        t.pencolor('yellow')
    t.fd(선분길이)
    t.left(외각크기)

t.hideturtle()
t.done()
Ln: 25 Col: 0

```



```

taeguk-3.py - C:/Users/user/Desktop/t...
File Edit Format Run Options Window Help
import turtle as t

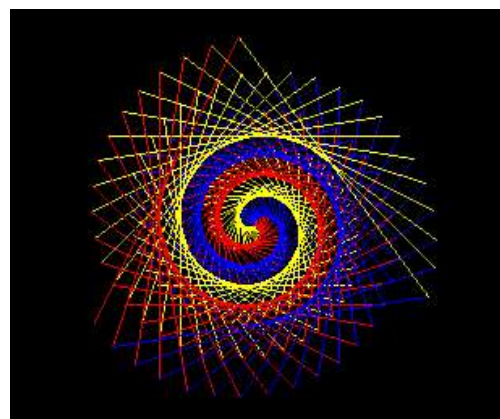
내각크기 = 63
외각크기 = 180 - 내각크기
선분개수 = 200

t.speed(0)
t.bgcolor('black')
t.pencolor('yellow')
t.fillcolor('orange')

for i in range(선분개수):
    선분길이 = i*1.0
    if i%3 == 1:
        t.pencolor('red')
    elif i%3 == 2:
        t.pencolor('blue')
    else:
        t.pencolor('yellow')
    t.fd(선분길이)
    t.left(외각크기)

t.hideturtle()
t.done()
Ln: 25 Col: 0

```



[도전과제-8] 거북이 그래픽으로 프랙탈 도형 그리기

```
import turtle
from math import *

koch_size = 600
koch_step = 0
message = ""마우스를 클릭할 때마다 코흐의 눈송이 곡선을 더 자세히 그립니다.
```

주의 : 그림이 모두 그려지기 전에 마우스를 클릭하면 오류가 발생합니다.

마우스 클릭으로 시작 : 빠른 그림 그리기

스페이스바 버튼으로 시작 : 느린 그림 그리기""

```
turtle.bgcolor('black')
turtle.color('sky blue', 'white')
turtle.hideturtle()
```

```
def koch_line(length, step):
    if step > 0:
        length = length/3
        for angle in [60, -120, 60, 0]:
            koch_line(length, step-1)
            turtle.left(angle)
    else:
        turtle.forward(length)
```

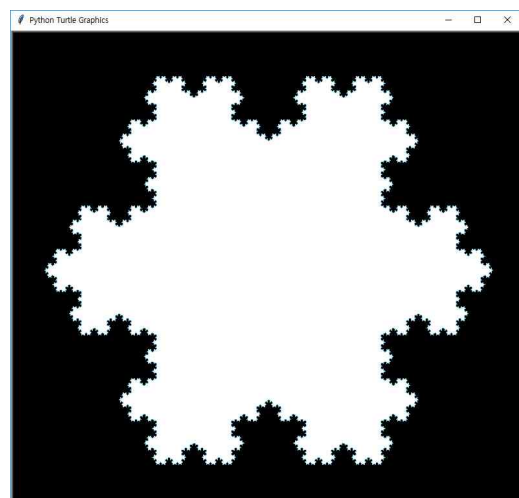
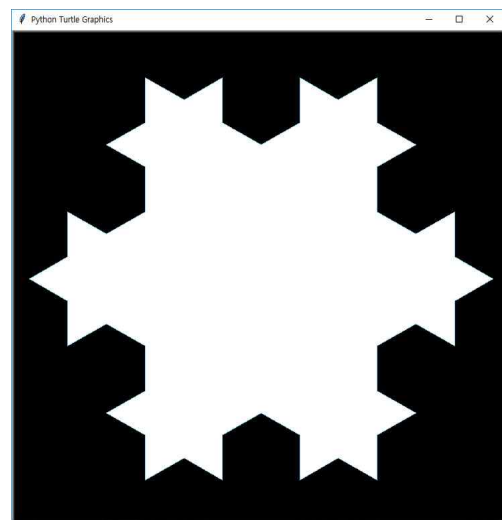
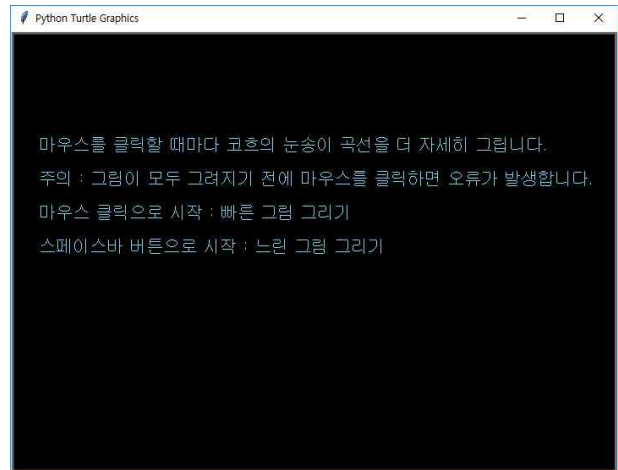
코흐의 눈송이의 중심이 원점에 오도록 위치를 설정

```
turtle.penup()
turtle.write(message, False, 'center', ('',15))
turtle.backward(koch_size/sqrt(3))
turtle.left(30)
turtle.pendown()
```

코흐 눈송이 그리는 코드

```
def draw_koch(x, y):
    global koch_step

    turtle.clear()
    turtle.tracer(100)
    turtle.begin_fill()
    for i in range(3):
```




```
        koch_line(koch_size, koch_step)
        turtle.left(-120)
    turtle.end_fill()
    turtle.update()
    koch_step += 1

def draw_koch_slow():
    global koch_step

    turtle.clear()
    turtle.speed(0)
    turtle.begin_fill()
    for i in range(3):
        koch_line(koch_size, koch_step)
        turtle.left(-120)
    turtle.end_fill()
    koch_step += 1

turtle.onscreenclick(draw_koch)
turtle.onkeypress(draw_koch_slow, 'space')
turtle.listen()
turtle.done()
```

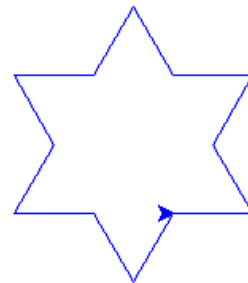
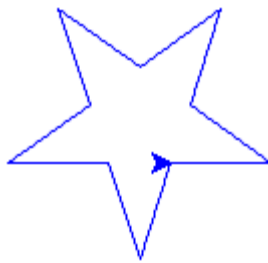
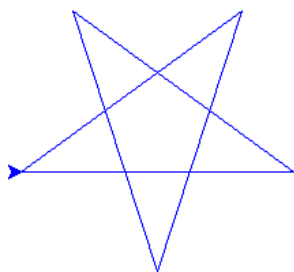
[도전과제-9] for문을 이용하여 한 변의 길이가 50인 정육각형을 그리시오.



[도전과제-10] for문을 이용하여 한 변의 길이가 50인 정십각형을 그리시오.



[도전과제-11] 다음 별 모양의 도형을 그리시오.



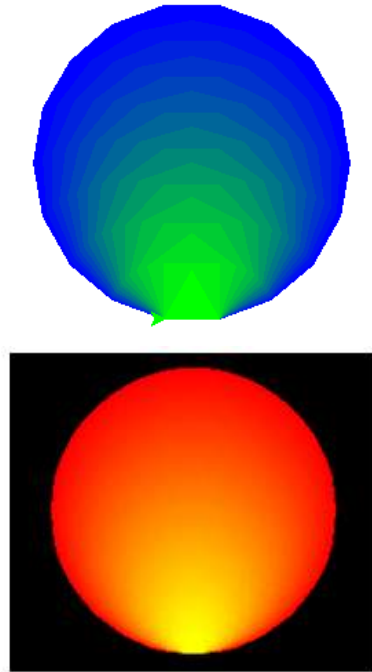
[도전과제-12] 다음 모양의 도형을 그리시오.

```
import turtle as t

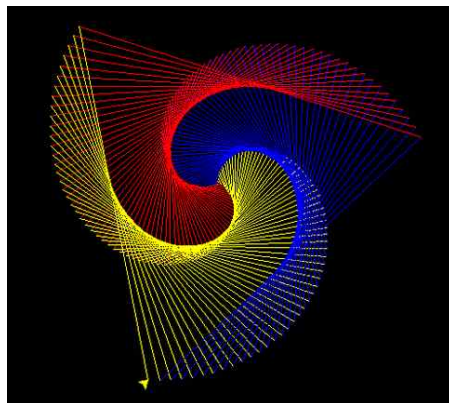
def draw_polygon(n):
    len = 40
    angle = 360/n
    for i in range(n):
        t.fd(len)
        t.left(angle)

t.speed(0)
g=0
b=255

for x in range(18, 2, -1):
    t.color(0, g/255, b/255)
    t.begin_fill()
    draw_polygon(x)
    t.end_fill()
    g=g+17
    b=b-17
```



[도전과제-13] 다음 모양의 도형을 그리시오.



선분의 길이를 2씩 증가하면서 200까지 ...
119도 왼쪽 회전하고 ..