

CHAPTER 03

그래픽에 대해 알아보시다.

문제해결을 위한 파이썬 첫걸음

이미향 교수

smilequeen@gmail.com

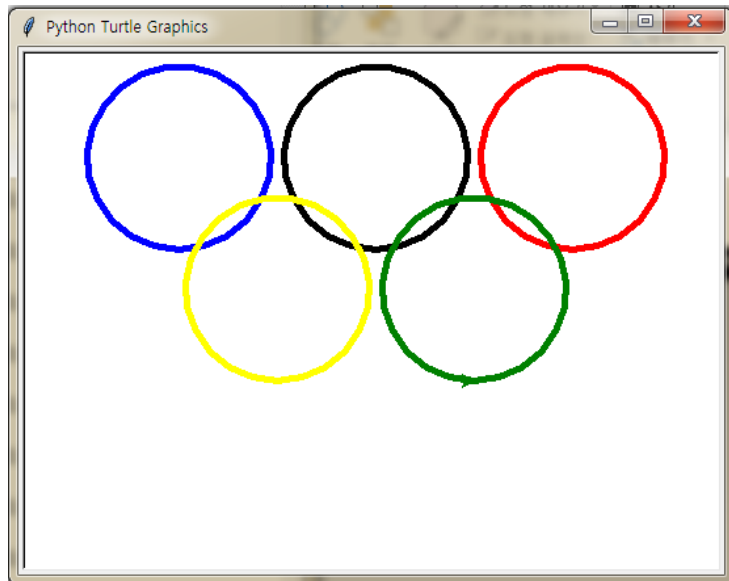
학습 목표

- 터틀 그래픽 환경과 터틀 모듈의 개념을 이해합니다.
- 터틀 모듈에서 제공하는 함수의 의미와 작성규칙을 이해합니다.
- tkinter를 이용하여 간단한 윈도우 위젯 프로그램을 만들 수 있습니다.

이번 장에서 만들 프로그램

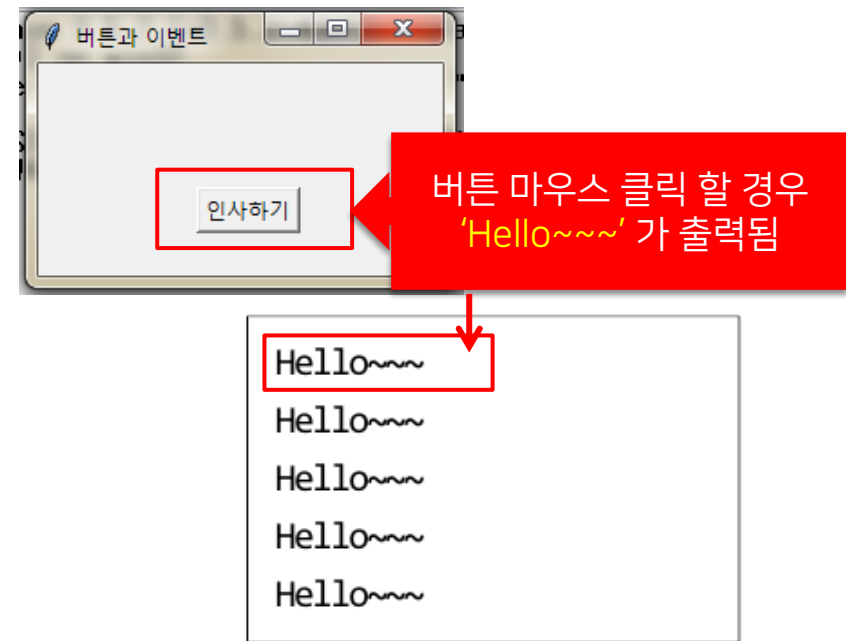
오륜기 그리기 프로그램

- turtle() 모듈 이용



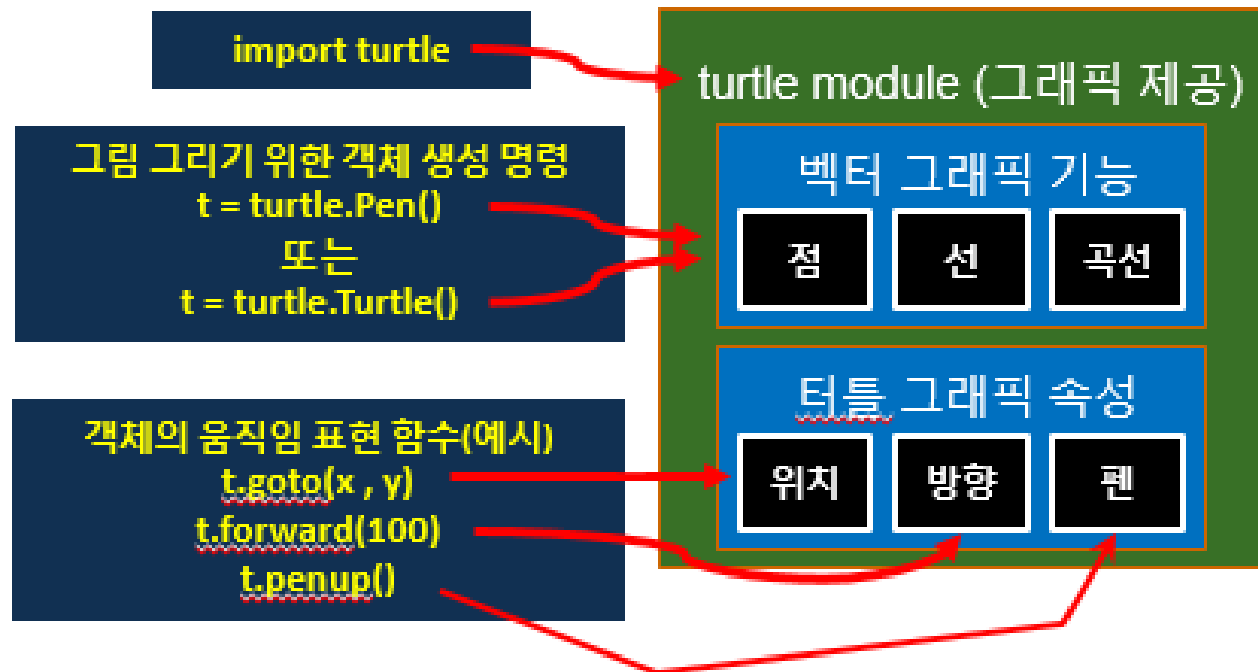
이벤트 프로그램

- tkinter() 모듈 이용



3.1 터틀 그래픽(turtle graphic)

- 터틀 그래픽(turtle graphic)
 - 파이썬에서 마우스나 키보드를 이용하여 그림을 그리도록 지원하는 모듈
 - 화면에서 커서가 지나간 흔적을 통해 그림을 그림
 - 점, 선, 곡선을 그릴 수 있는 벡터 기반 그래픽 기능 제공



이미지 표현 방법_래스터(비트맵)

- 픽셀(pixel)
 - 그림을 표현하는 최소 단위 점
 - 화소
 - 같은 그림을 표현할 때

픽셀 수 많을 수록, 한 픽셀에 표현되는 값(비트 수)가 많을 수록 그림이 선명하고 파일의 용량은 커짐

- jpg, bmp, gif, png, psd



원본



픽셀 수 1/3로 줄인 그림을 확대

이미지 표현 방법_벡터

- 벡터(Vector)
 - 이미지를 수학 함수로 표현하는 방법
 - 장점_그림을 확대, 축소해도 그림의 선 명도에 변화가 없이 깨끗하게 나타남
 - 단점_색상 표현이나 모양이 복잡한 실물 사진 등을 표현하기에는 복잡한 수학 함수가 요구되므로, 처리 시간이 많이 걸림
 - 일러스트레이터(ai), 플래시(fla)



래스터이미지



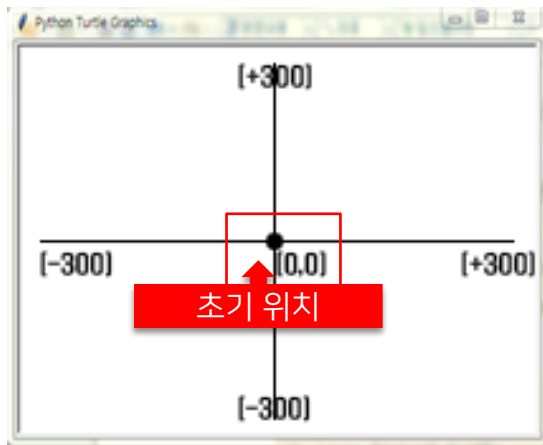
벡터이미지

3.2 터틀 그래픽(turtle graphic) 활용

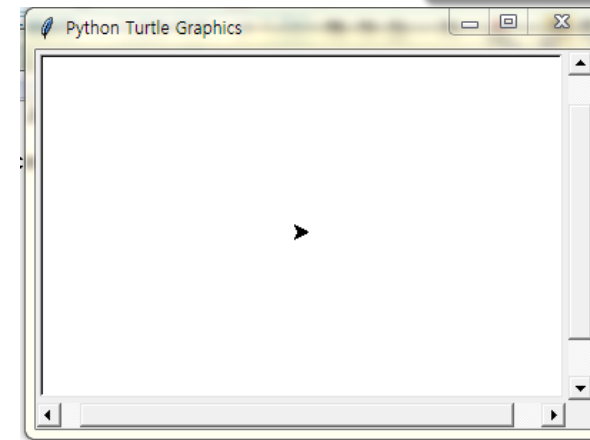
- 터틀 그래픽으로 그림 그리기
 - import 명령을 통해 turtle 모듈을 작업 환경으로 가지고 올

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

- 초기 커서의 위치 (x좌표=0, y좌표=0)









[실행결과]



3.2 터틀 그래픽 활용 - 커서 모양 변경

- 커서의 모양 변경하기
 - `turtle.shape("커서모양")`

종류	형태	명령	종류	형태	명령
arrow		<code>turtle.shape("arrow")</code>	square		<code>turtle.shape("square")</code>
circle		<code>turtle.shape("circle")</code>	triangle		<code>turtle.shape("triangle")</code>
classic		<code>turtle.shape("classic")</code>	turtle		<code>turtle.shape("turtle")</code>

3.2 터틀 그래픽 활용 - 이동과 회전

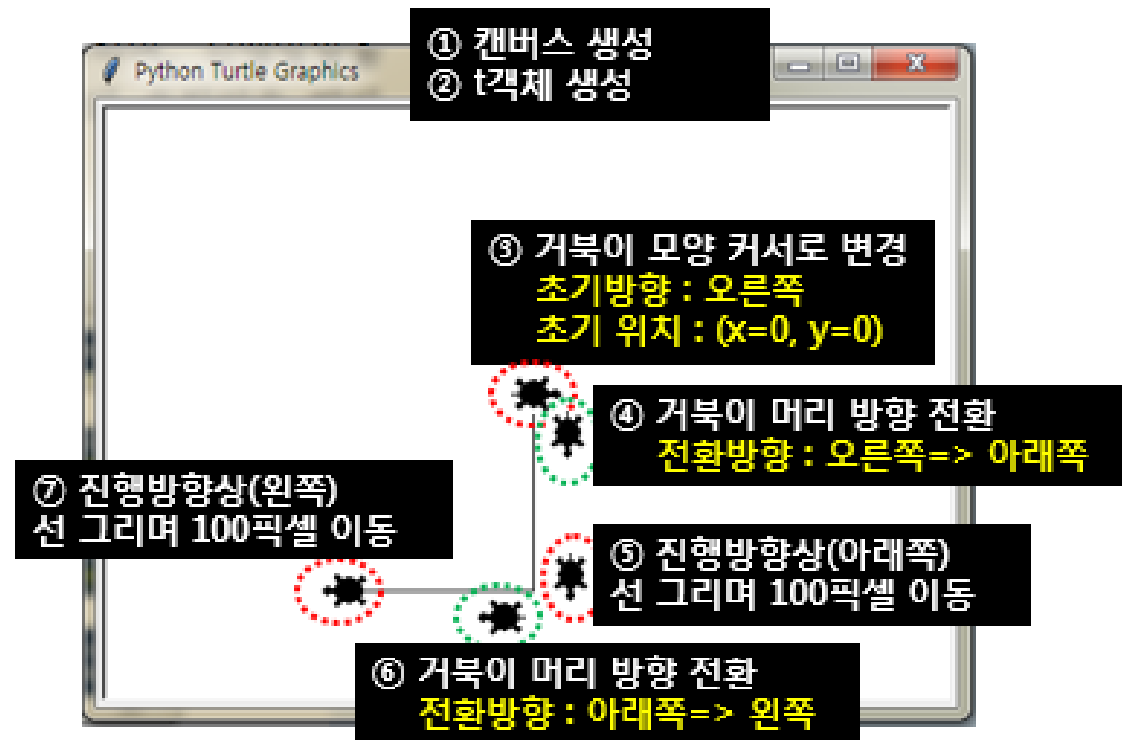
● 이동과 회전

- `forward(이동길이)` : 진행방향으로 이동
- `left(회전각도)` : 왼쪽 방향으로 회전
- `right(회전각도)` : 오른쪽 방향으로 회전

python shell

```

❶ >>> import turtle
❷ >>> t=turtle.Turtle()
❸ >>> t.shape("turtle")
❹ >>> t.right(90)
❺ >>> t.forward(100)
❻ >>> t.right(90)
❼ >>> t.forward(100)
  
```

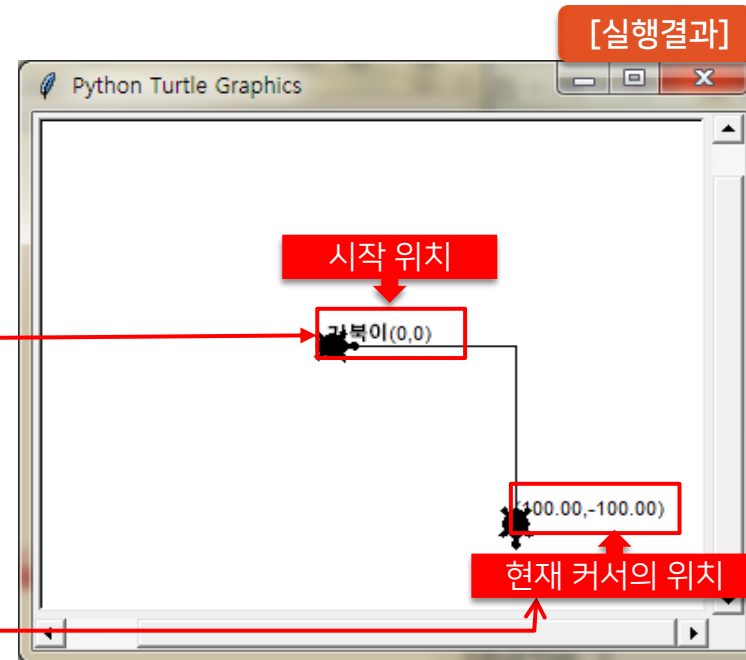


3.2 터틀 그래픽 활용 - 모양과 문자열 출력

- 커서 모양과 문자열 출력
 - `stamp()`: 커서 모양 출력
 - `write("출력문자열")`: 문자열 출력
 - `position()`: 현재 위치 좌표 출력

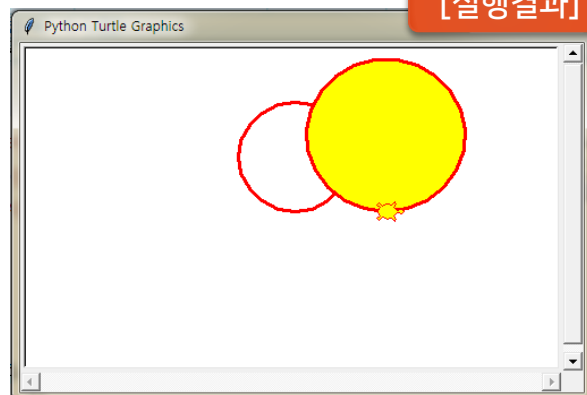
python shell

```
>>> import turtle
>>> t=turtle.Turtle()
>>> t.shape("turtle")
>>> t.stamp()
>>> t.write("거북이(0,0)")
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.write(t.position())
```



3.2 터틀 그래픽 활용 - 선 굵기와 색상, 도형

- 선의 굵기와 색상, 도형 지정
 - width(선굵기): 선의 굵기
 - color("색상명"): 선의 색상 지정
 - fillcolor("색상명") : 채우기 색상 지정
 - begin_fill() : 색상 채우기 준비
 - end_fill() : 색상 채우기 완료
 - up() : 펜 들기 (=penup())
 - down() : 펜 내리기 (=pendown())
 - circle(반지름) : 원 도형 그리기



python shell

```

>>> import turtle
>>> t=turtle.Turtle()
>>> t.shape("turtle")
>>> t.width(3)
>>> t.color("red")
>>> t.circle(50)
>>> t.up()
>>> t.forward(80)
>>> t.down()
>>> t.fillcolor("yellow")
>>> t.begin_fill()
>>> t.circle(70)
>>> t.end_fill()
  
```

펜 들기

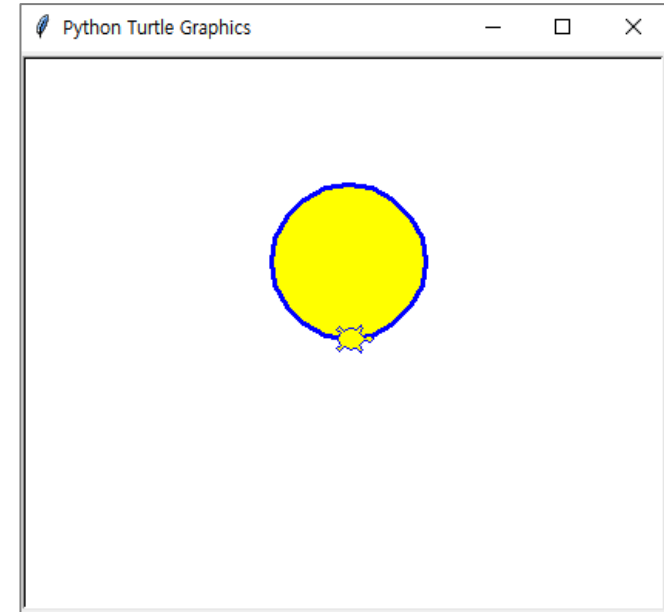
펜 내리기

색상 채우기 준비

색상 채우기 완료

터틀 그래픽 색상 표현 방법

- 빨간색 선을 표현할 경우
 - 색상명으로 표기 : `color("red")`
 - 16진수 값으로 표기 : `color("#ff0000")`
 - RGB color 값으로 표기 : `color(r,g,b)`
 - r,g,b 각 값의 범위는 1-255 사이의 값
 - RGB 방식의 모드로 설정하는 경우 다음과 같은 코드가 선행되어야 한다.
 - 만약 `turtle.colormode(255)`가 생략되면 0~1.0 사이의 값으로 설정되어야 한다.
- `turtle.colormode(255)`
`turtle.color(255,0,0)`



```
import turtle
t = turtle.Turtle()
t.shape("turtle")
turtle.colormode(255)
t.pensize(3)
t.color(0,0,255)      #파란색
t.fillcolor(255,255,0) #노란색
t.begin_fill()
t.circle(50)
t.end_fill()
```

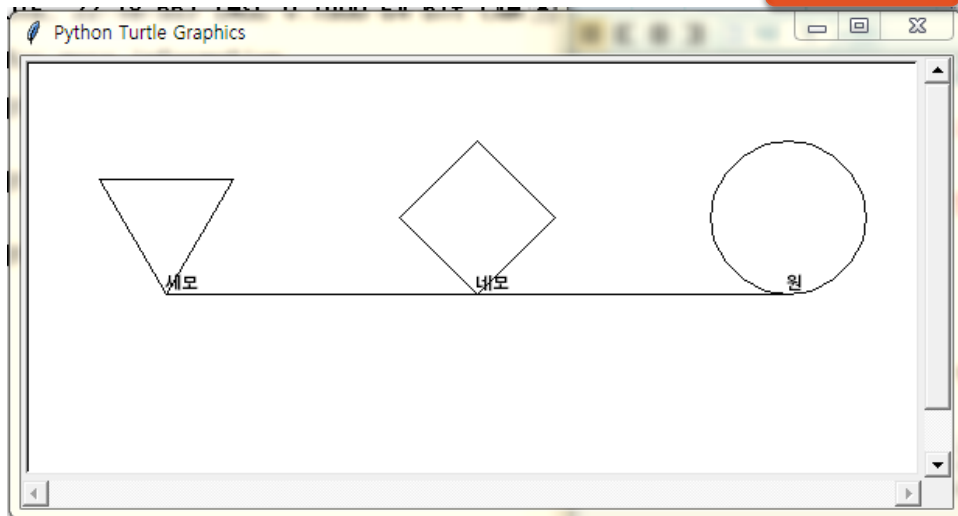
color example code

black	k	dimgray	dimgrey
gray	grey	darkgray	darkgrey
silver	lightgray	lightgrey	gainsboro
whitesmoke	w	white	snow
rosybrown	lightcoral	indianred	brown
firebrick	maroon	darkred	r
red	mistyrose	salmon	tomato
darksalmon	coral	orangered	lightsalmon
sienna	seashell	chocolate	saddlebrown
sandybrown	peachpuff	peru	linen
bisque	darkorange	burlywood	antiquewhite
tan	navajowhite	blanchedalmond	papayawhip
moccasin	orange	wheat	oldlace
floralwhite	darkgoldenrod	goldenrod	cornsilk
gold	lemonchiffon	khaki	palegoldenrod
darkkhaki	ivory	beige	lightyellow
lightgoldenrodyellow	olive	y	yellow
olivedrab	yellowgreen	darkolivegreen	greenyellow
chartreuse	lawngreen	honeydew	darkseagreen
palegreen	lightgreen	forestgreen	limegreen
darkgreen	g	green	lime
seagreen	mediumseagreen	springgreen	mintcream
mediumspringgreen	mediumaquamarine	aquamarine	turquoise
lightseagreen	mediumturquoise	azure	lightcyan
paleturquoise	darkslategray	darkslategrey	teal
darkcyan	c	aqua	cyan
darkturquoise	cadetblue	powderblue	lightblue
deepskyblue	skyblue	lightskyblue	steelblue
aliceblue	dodgerblue	lightslategray	lightslategrey
slategray	slategrey	lightsteelblue	cornflowerblue
royalblue	ghostwhite	lavender	midnightblue
navy	darkblue	mediumblue	b
blue	slateblue	darkslateblue	mediumslateblue
mediumpurple	rebeccapurple	blueviolet	indigo
darkorchid	darkviolet	mediumorchid	thistle
plum	violet	purple	darkmagenta
m	fuchsia	magenta	orchid
mediumvioletred	deeppink	pink	lavenderblush
palevioletred	crimson		lightpink

출처_ https://matplotlib.org/examples/color/named_colors.html

3.2 터틀 그래픽 활용 - 이동과 circle() 활용

- 이동과 circle() 함수의 활용
 - goto(x,y): 지정된 위치로 이동
 - circle(반지름): 원 도형 그리기
 - circle(반지름, 회전각도, 꼭지점갯수)
 - 다각형 도형 그리기 가능
 - hideturtle(): 커서 모양 숨기기



python shell

```

>>> import turtle
>>> t=turtle.Turtle()
>>> t.shape("turtle")
>>> t.goto(-200,0)
>>> t.write("세모")
>>> t.circle(50,360,3)
>>> t.goto(0,0)
>>> t.write("네모")
>>> t.circle(50,360,4)
>>> t.goto(200,0)
>>> t.write("원")
>>> t.circle(50)
>>> t.hideturtle()
  
```

x축 -200, y축 0으로 이동

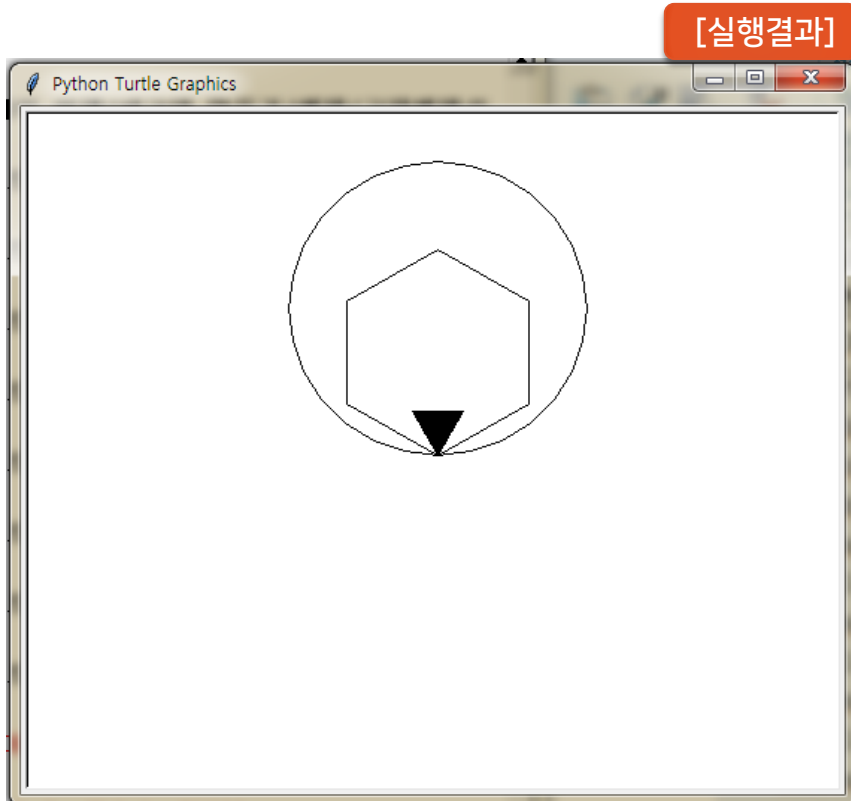
반지름 50, 회전각도 360, 꼭지점 3개 => 삼각형 도형

반지름 50, 회전각도 360, 꼭지점 4개 => 사각형 도형

커서 모양 숨기기

스스로 해보기

- 실행결과와 같은 그래픽 프로그램 만들기
 - 반지름의 길이가 100인 원, 70 인 육각형, 20인 삼각형 도형 그리기
 - 삼각형 도형은 검은색으로 채우기



```
1 import turtle
2 t = turtle.Turtle()
3 t.circle(100)
4 t.circle(70,360,6)
5 t.fillcolor("black")
6 t.begin_fill()
7 t.circle(20,360,3)
8 t.end_fill()
9 t.hideturtle()
```



Python

실습_코딩하기1

프로그램1_오륜기 그리기

- 터틀 그래픽으로 다양하게 표현하기
 - 오륜기 그리기
 - 원이 그려지는 시작 위치
 - x축 -150, y축 50
 - 이후 적당히 위치 변경하며 그리기
 - 원의 반지름은 70
 - 선의 굵기 5
 - 도형이 그려지는 속도 5
 - speed(속도) : 실행 속도



3.2 터틀 그래픽 활용 - 다양하게 표현하기

- 오륜기 그리기

시작

```
import turtle
t=turtle.Turtle()
t.speed(5)
t.width(5)

t.up()
t.goto(-150,50)
t.down()
t.color("blue")
t.circle(70)

t.up()
t.goto(0,50)
t.down()
t.color("black")
t.circle(70)
```

이어서

```
t.up()
t.goto(150,50)
t.down()
t.color("red")
t.circle(70)

t.up()
t.goto(-75,-50)
t.down()
t.color("yellow")
t.circle(70)

t.up()
t.goto(75,-50)
t.down()
t.color("green")
t.circle(70)
```