

# Turtle 라이브러리를 이용한 시각적 프로그래밍

천양하 교수

# 목차

---

1. 거북이 생성과 간단한 이동
2. 거북이 모양과 크기 변경하기
3. 반복문 활용하여 도형 그리기
4. 멋진 애니메이션
5. 움직이는 거북이
  - 혼자 움직이는 거북이
  - 키보드 방향키에 따라 움직이는 거북이

# 오늘의 학습 목표

---

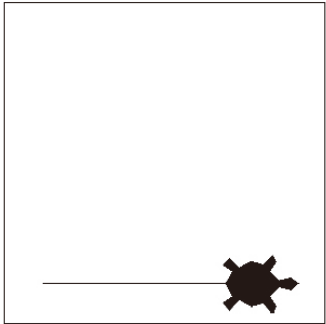
- Turtle 라이브러리(모듈)을 사용하여 거북이를 표현할 수 있다.
  - `import turtle`
  - `shape, shapesize` 함수
- Turtle 라이브러리(모듈)의 기본 함수를 설명할 수 있다.
  - `forward, backward, left, right, home, clear` 등
- Turtle 모듈을 사용하여 기본 도형을 그릴 수 있다.
  - 사각형, 삼각형, 원 등의 도형

# 1. 거북이 생성과 간단한 이동

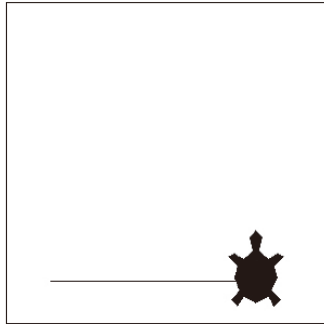
- ❖ 거북이를 생성하고 크기를 변경합니다.
- ❖ 거북이가 이동하며 그림을 그리도록 합니다.

# 거북이 생성하기

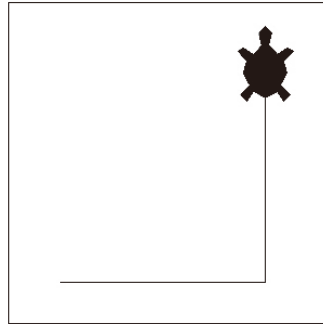
p.110 ~ 111



forward(200)



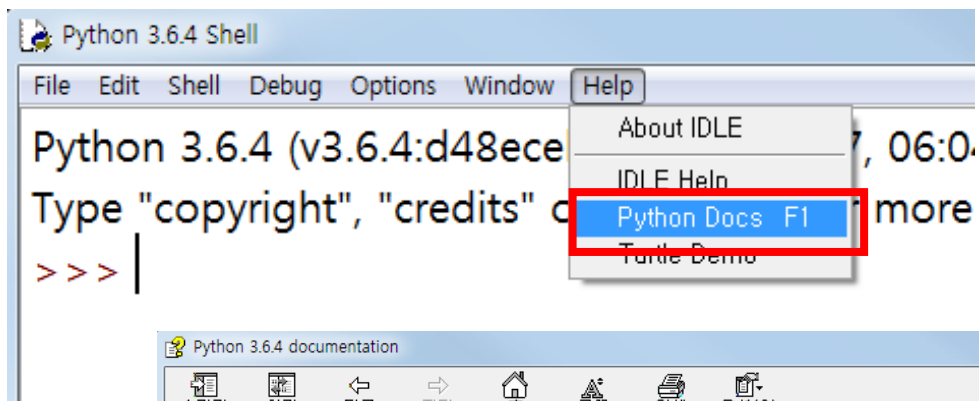
left(90)



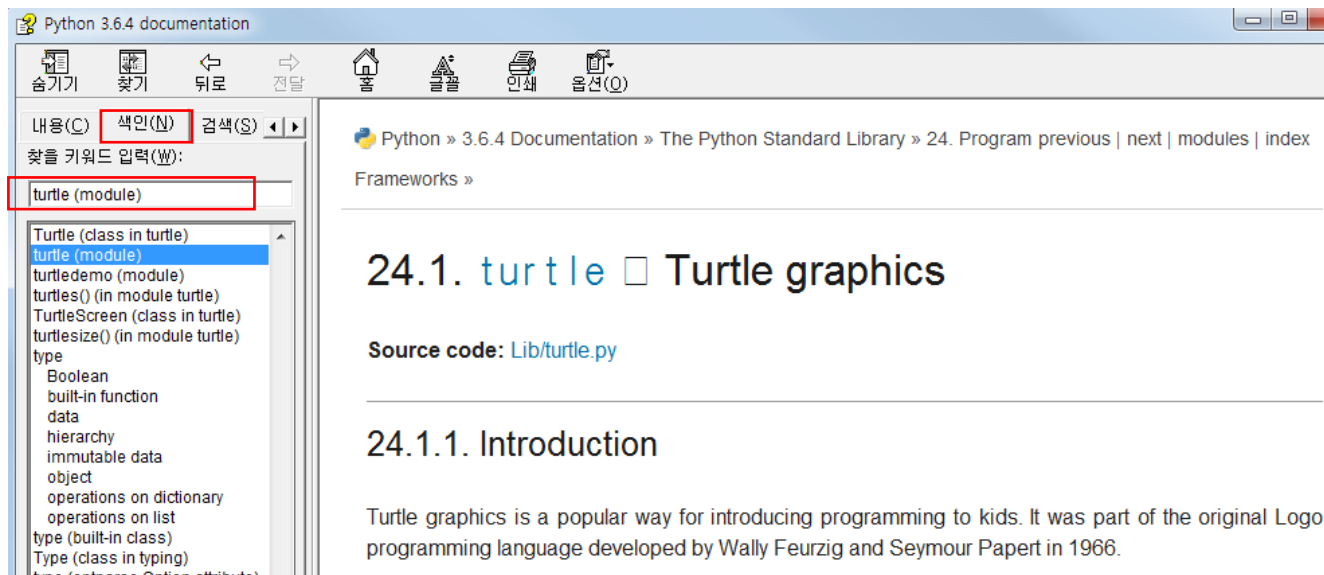
forward(200)

- 거북이 표시하기
- 거북이 크기 변경하기
- 거북이 이동하기
- 원 위치로 이동하기
- 모두 지우기

# 더 자세한 내용 살펴보기



Help – Python Docs F1 – 색인 –  
- 'turtle' 검색 – turtle(module)



OFF

# Turtle 객체의 함수 줄임말

교재 p.111 기록해두기

원래 이름	줄임말
forward	fd
backward	bk 혹은 back
right	rt
left	lt

# 간단한 그림 그리기

```
import turtle as t
```

```
t.forward(300)
```

```
t.right(90)
```

```
t.forward(200)
```

```
import turtle as t
```

```
t.fd(300)
```

```
t.rt(90)
```

```
t.fd(200)
```



거북이가 표시되도록 코드를 추가해보세요.



## 2. 거북이 모양과 크기 변경하기

- ❖ 거북이 모양을 변경합니다.
- ❖ 거북이 크기를 변경합니다.

# 거북이 모양, 크기 변경하기

교재 p.113

```
>>> import turtle as t
>>>
>>> t.shape('triangle')
>>> t.shapesize(5)
>>>
>>> t.shape('turtle')
>>>
>>> t.shape('circle')
>>> t.shape('square')
>>>
>>> t.shapesize(3)

>>> t.color('blue')
>>> t.color('red')
>>> t.color('green')

>>> t.setheading(0)
>>>
>>> t.setheading(90)
>>> t.setheading(180)
>>> t.setheading(270)
```

온라인에서는 설명만하고 오프라인에서 실습

# 코드 편집기에서 수행하기

```
import turtle as t

t.shape('triangle')
t.shapesize(5)

t.shape('turtle')
t.shape('circle')
t.shape('square')

t.shapesize(3)
```

```
import turtle as t
import time

t.shape('triangle')
t.shapesize(5)

time.sleep(1)
t.shape('turtle')
time.sleep(1)
t.shape('circle')
time.sleep(1)
t.shape('square')
time.sleep(1)
t.shapesize(3)
t.color('blue')
```

온라인에서는 설명만하고 오프라인에서 실습

### 3. 반복문 활용하여 도형 그리기

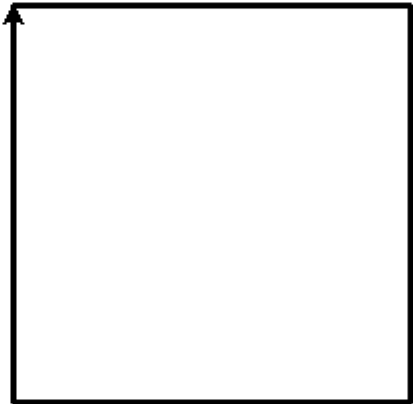
- ❖ 반복문을 활용하여 그림을 그려봅시다.
- ❖ 반복문은 추후 수업에서 자세히 배울 예정입니다.

# 다각형 그리기

---

- 사각형 그리기

교재 p.116



width 함수  
color 함수  
pencolor 함수

- 삼각형 그리기

교재 p.118

- 오각형 그리기

교재 p.119

OFF

	black
	grey
	silver
	whitesmoke
	rosybrown
	firebrick
	r
	darksalmon
	sienna
	sandybrown
	bisque
	tan
	moccasin
	floralwhite
	gold
	darkkhaki
	lightgoldenrodyellow
	olivedrab
	chartreuse
	darksage
	lightgreen
	green
	mediumseagreen
	mediumaquamarine
	mediumturquoise
	darkslategray
	c
	cadetblue
	skyblue
	dodgerblue
	slategray
	ghostwhite
	darkblue
	slateblue
	blueviolet
	mediumorchid
	purple
	magenta
	hotpink
	pink

	k
	gray
	lightgrey
	white
	lightcoral
	maroon
	mistyrose
	coral
	seashell
	peachpuff
	darkorange
	navajowhite
	orange
	darkgoldenrod
	lemonchiffon
	ivory
	olive
	yellowgreen
	lawngreen
	honeydew
	forestgreen
	g
	springgreen
	aquamarine
	azure
	darkslategrey
	cyan
	powderblue
	lightskyblue
	lightslategrey
	lightsteelblue
	lavender
	mediumblue
	darkslateblue
	indigo
	thistle
	darkmagenta
	orchid
	lavenderblush
	lightpink

	dimgrey
	darkgray
	lightgray
	w
	indianred
	darkred
	salmon
	orangered
	chocolate
	peru
	burlywood
	blanchedalmond
	wheat
	goldenrod
	khaki
	beige
	y
	darkolivegreen
	sage
	darkseagreen
	limegreen
	lime
	mintcream
	turquoise
	lightcyan
	teal
	aqua
	lightblue
	steelblue
	lightslategray
	cornflowerblue
	midnightblue
	blue
	mediumslateblue
	darkorchid
	plum
	m
	mediumvioletred
	palevioletred

	dimgray
	darkgrey
	gainsboro
	snow
	brown
	red
	tomato
	lightsalmon
	saddlebrown
	linen
	antiquewhite
	papayawhip
	oldlace
	cornsilk
	palegoldenrod
	lightyellow
	yellow
	greenyellow
	lightsage
	palegreen
	darkgreen
	seagreen
	mediumspringgreen
	lightseagreen
	paleturquoise
	darkcyan
	darkturquoise
	deepskyblue
	aliceblue
	slategray
	royalblue
	navy
	b
	mediumpurple
	darkviolet
	violet
	fuchsia
	deeppink
	crimson

## Matplotlib 라이브러리 지원 (일부 turtle에서는 지원 안됨)

red : r  
green : g  
blue : b

cyan : c  
magenta : m  
yellow : y  
black : b

pink  
lightpink  
brown  
navy  
grey  
purple  
orange  
lightgreen  
violet



# RGB 코드로 색상 지정하기

```
import turtle as t
t.width(5)

t.color('#ff0000')
t.fd(200)
t.rt(90)

t.pencolor('#0000ff')
t.fd(200)
```

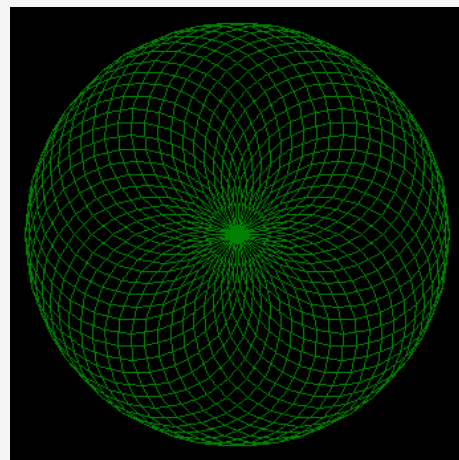
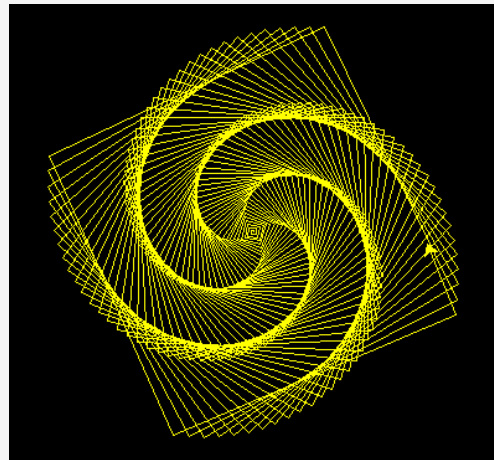
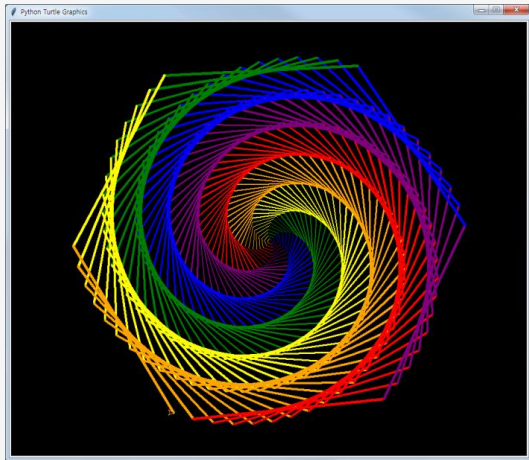


[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)

OFF

## 4. 멋진 애니메이션

❖ Turtle 모듈을 활용하여 화려한 그래픽 만들기



OFF



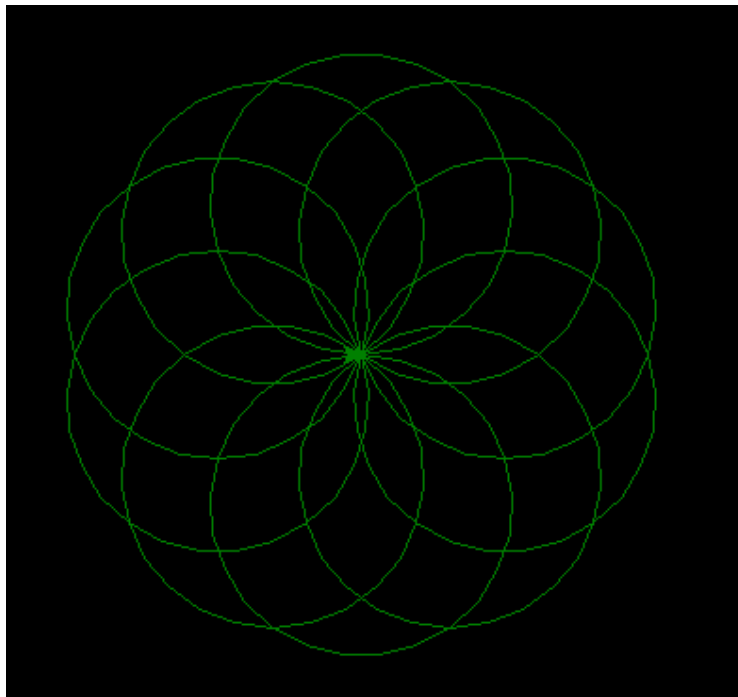
# 회전하는 원 그리기

```
import turtle as t

t.bgcolor("black")
t.color("green")
t.speed(5)

for x in range(10) :
    t.circle(150)
    t.left(360/10)
```

pencolor 함수



속도 : 5 -> 0  
반복횟수 : 10 -> 50

OFF

# 회전하는 원 그리기

```
import turtle as t
```

```
n=50
```

```
t.bgcolor("black")
```

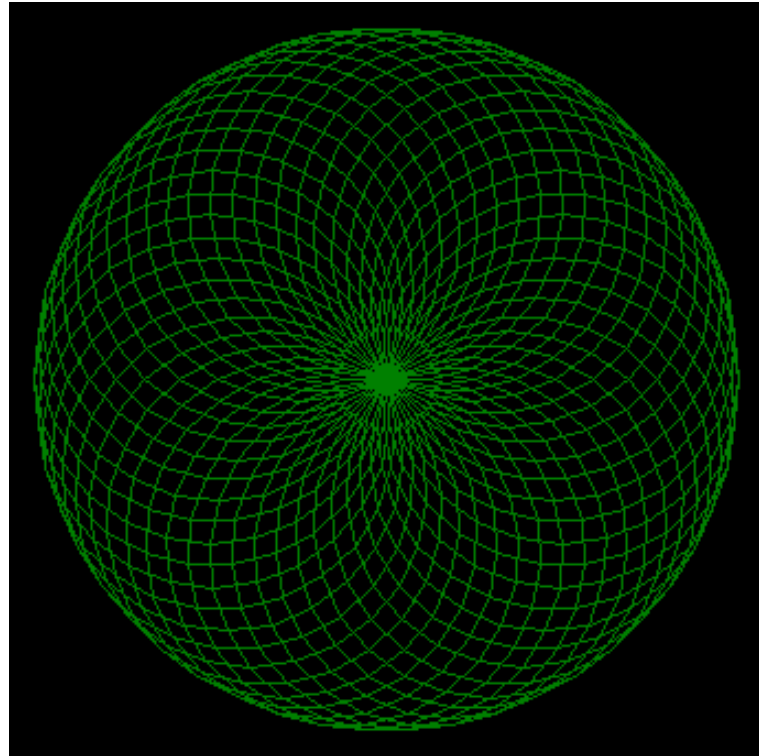
```
t.color("green")
```

```
t.speed(0)
```

```
for x in range(n) :
```

```
    t.circle(150)
```

```
    t.left(360/n)
```



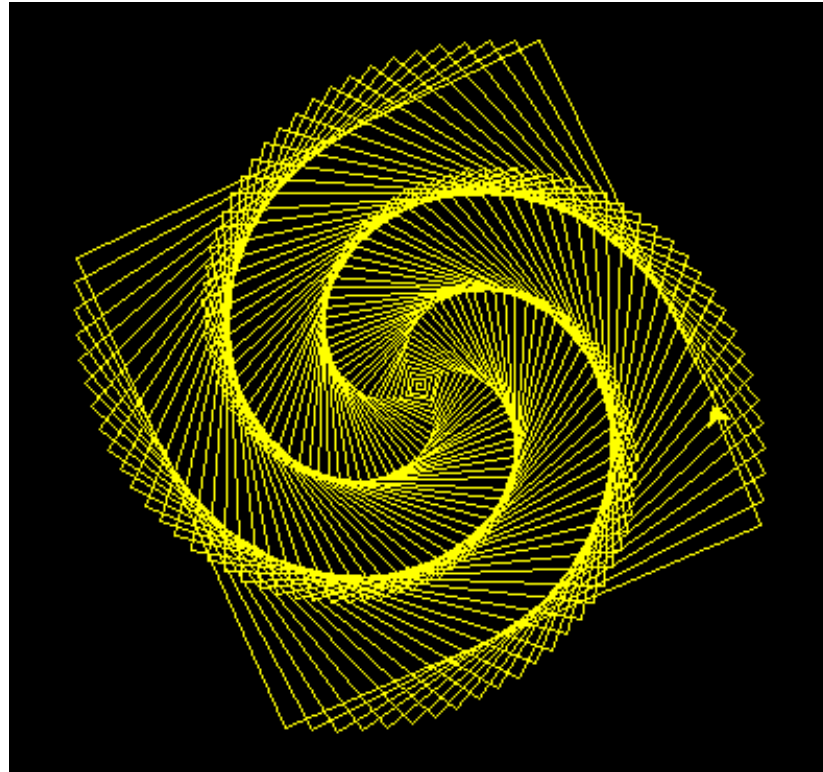
OFF

# 사각형 회오리

```
import turtle as t

angle = 89

t.bgcolor("black")
t.color("yellow")
t.speed(0)
for x in range(200) :
    t.forward(x)
    t.left(angle)
```



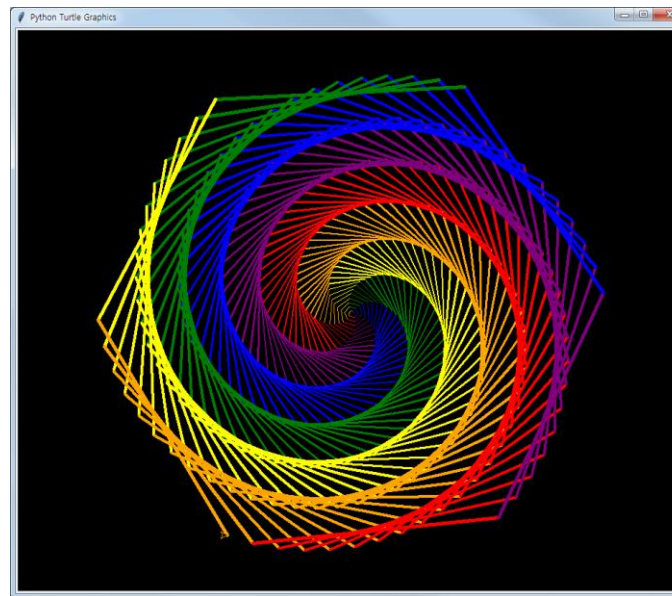
# 0은 최고 빠름, 1~10(느림~빠름)

OFF

```
import turtle
t = turtle.Turtle()
colors = ['red', 'purple', 'blue', 'green', 'yellow', 'orange']

# t.bgcolor('black')
t.pencolor('white')
t.speed(0)
turtle.bgcolor('black')

for x in range(360) :
    t.pencolor( colors[x%6] )
    t.width( x/100 + 1 )
    t.forward( x )
    t.left( 59 )
```



파이썬 공식 홈페이지에서 제공하는 예제 소스코드

## 5. 움직이는 거북이

- ❖ 방향키에 따라 움직이는 거북이
- ❖ 혼자 돌아다니는 거북이

OFF

## 방향키로 거북이 움직이게 하기 (단순화: 좌우)

```
import turtle as t

def goRight() :
    t.setheading(0)
    t.fd(50)

def goLeft() :
    t.setheading(180)
    t.fd(50)

t.shape('turtle')
t.shapesize(3)

screen = t.Screen()
screen.listen()
screen.onkey(goLeft, 'Left')
screen.onkey(goRight, 'Right')
```

위의 코드를 수정하여 상하좌우 이동하는 프로그램을 만들어 봅시다.

OFF

# 상하좌우 움직이는 거북이 (완성)

```
import turtle as t
```

```
def goUp() :  
    t.setheading(90)  
    t.fd(50)
```

```
def goDown() :  
    t.setheading(270)  
    t.fd(50)
```

```
def goRight() :  
    t.setheading(0)  
    t.fd(50)
```

```
def goLeft() :  
    t.setheading(180)  
    t.fd(50)
```

```
t.shape('turtle')  
t.shapesize(3)
```

```
screen = t.Screen()  
screen.listen()  
screen.onkey(goLeft, 'Left')  
screen.onkey(goRight, 'Right')  
screen.onkey(goUp, 'Up')  
screen.onkey(goDown, 'Down')
```

상하좌우로 이동하는 완성된 코드

OFF

# 혼자 돌아다니는 거북이

```
import turtle as t
import time
import random
```

거북이를 더 빨리 움직이게 하려면?

거북이가 바깥쪽으로 나가면 중앙으로 이동

```
t.shape('turtle')
t.shapesize(3)
```

```
while True :
    t.fd(50)
    angle = random.randint(-90, 90)
    t.rt( angle )
    time.sleep(1)
```

OFF



# 혼자 돌아다니다 집으로 돌아오는 거북이

---

```
while True :  
    t.fd(50)  
    angle = random.randint(-90, 90)  
    t.rt( angle )  
    time.sleep(0.3)  
  
    xyPoint = t.position()  
    print(xyPoint)  
  
    if -500<= xyPoint[0] <500 and -500 <= xyPoint[1] <= 500 :  
        pass  
    else :  
        t.home()  
        t.clear()
```

OFF

# 스스로 해보기

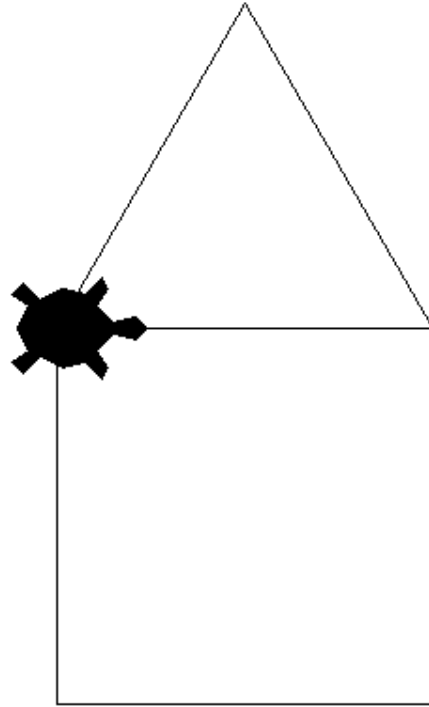
- ❖ 오늘 배운 내용들을 토대로
- ❖ 몇가지 실습을 수행해봅시다.

OFF

# 문제1

---

- 사각형과 삼각형을 그려서  
오른쪽과 같이 집 모양을  
그려봅시다.



- 참고 :

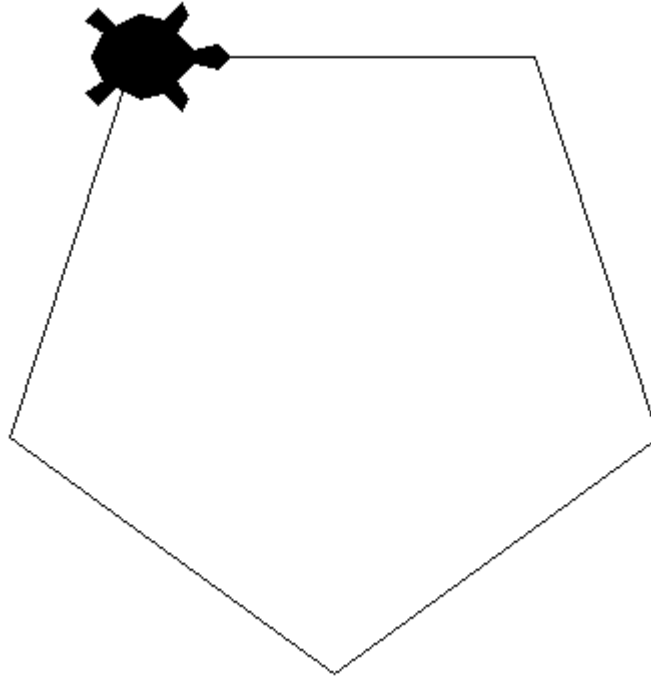
**p.116** 사각형 그리기

**p.118** 삼각형 그리기

## 문제2

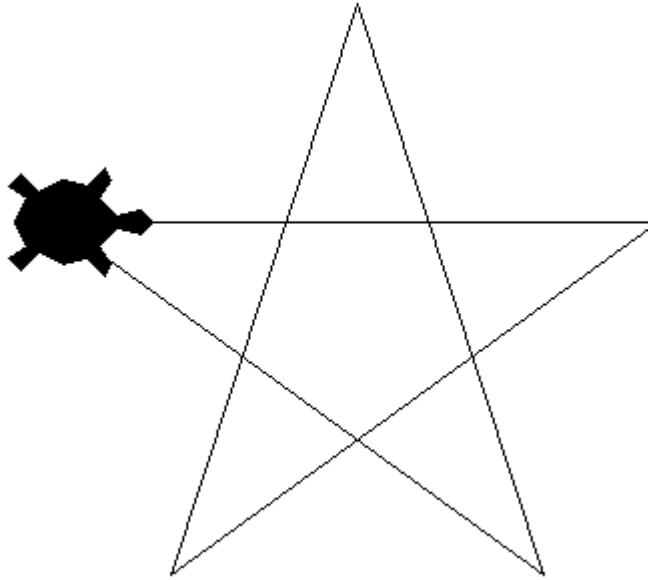
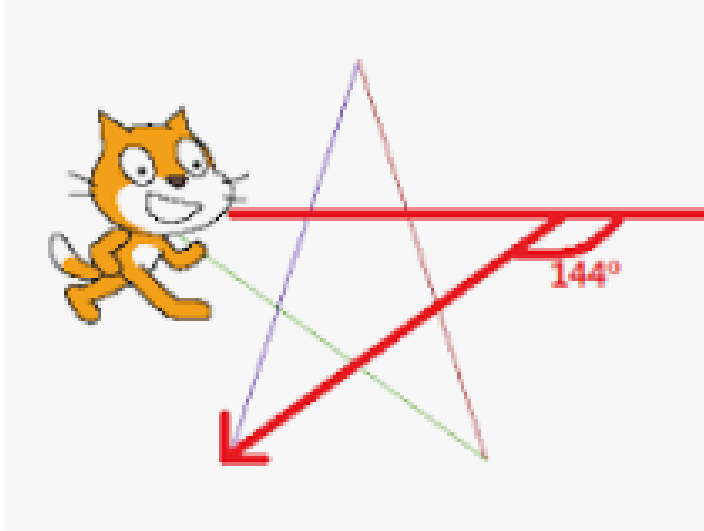
---

- 다음과 같이 한 변의 길이가 200인 오각형을 그려봅시다.
- 참고 : **p.119**



## 문제3

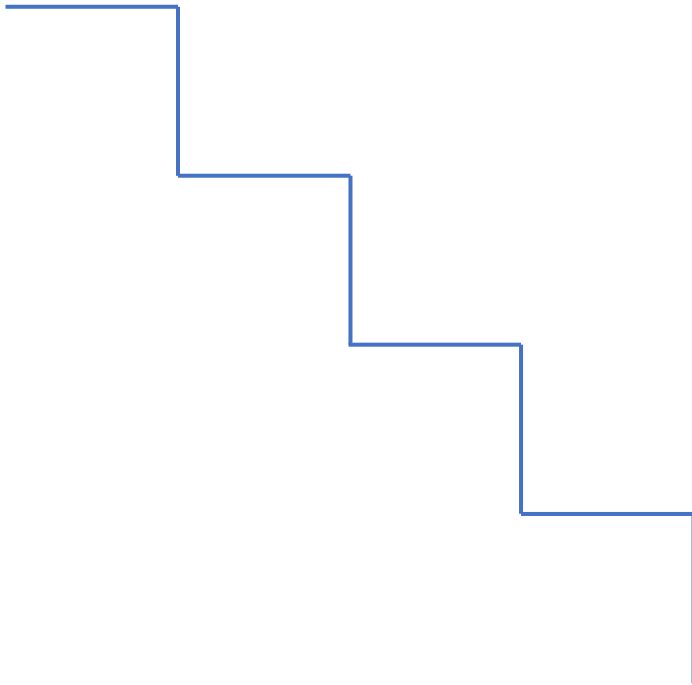
- 한 변의 길이가 300인 별 그림을 그려봅시다.
- 힌트



## 문제4

---

- 계단 모양 그리기



수고하셨습니다.