

※프로그램의 코드의 주석(comment)과 실행 결과 화면을 보고 코드의 빈 칸을 채우세요.

9차시_사용자정의함수활용하기

1. Ex1_사각형을그리는함수

[조건]
1. square함수 작성 : 사각형 그리기 100픽셀만큼 앞으로 가고 왼쪽으로 90도 회전하는 동작을 4회 반복한다.
2. (0,0), (-150,150),(150,-150)위치에 square함수를 이용하여 사각형을 그린다.

코드
<pre>import turtle t=turtle.Turtle() t.shape("turtle") turtle.title("사각형을 그리는 함수") def square(): #사각형을 그리는 함수 정의 for i in range((가)): #4회 반복 t.forward(100) t.left(90) #(0,0)에 사각형 그리기 t.write(t.position()) #위치 출력 (나) #함수 호출 : 사각형 그리기 #(-150,150)에 사각형그리기 t.penup() #펜을 든다 t.goto(-150,150) #(-150,150)으로 이동 t.pendown() #펜을 내린다 t.write(t.position()) #위치 출력 square() #함수호출 : 사각형 그리기 #(150,-150)에 사각형그리기 t.penup() t.goto(150,-150) t.pendown() t.write(t.position()) #위치 출력 square() #함수호출 : 사각형 그리기 turtle.done()</pre>

실행 결과 화면

(가)	4
(나)	square()

2. Ex2_서로다른크기의사각형을그리는함수

[조건]
1. square함수 작성 : 사각형 그리기 앞으로(거리 : line) 가고 왼쪽으로 90도 회전하는 동작을 4회 반복한다.
2. (0,0), (-150,150),(150,-150)위치에 square함수를 이용하여 사각형을 그린다.

코드
<pre>import turtle t=turtle.Turtle() t.shape("turtle") turtle.title("서로 다른 크기의 사각형을 그리는 함수") def (가): #line의 길이로 사각형을 그리는 square함수 정의 for i in range(4) : (나) #line만큼 앞으로 이동 t.left(90) #(0,0)에 사각형 그리기 t.write(t.position()) (다) #함수 호출 : 사각형 그리기-한 번 길이 50픽셀 #(-150,150)에 사각형그리기 t.penup() #펜을 든다 t.goto(-150,150) #(-150,150)으로 이동 t.pendown() #펜을 내린다 t.write(t.position()) square(100) #함수호출 : 사각형 그리기-한 번 길이 100픽셀 #(150,-150)에 사각형그리기 t.penup() t.goto(150,-150) t.pendown() t.write(t.position()) square(150) #함수호출 : 사각형 그리기-한 번 길이 150픽셀 turtle.done()</pre>

실행 결과 화면

(가)	square(line)
(나)	t.forward(line)
(다)	square(50)

3. Ex3_별을그리는함수

[조건]

- draw_star함수 작성 : 사각형 그리기
앞으로(거리 : line) 가고 오른쪽으로 144도 회전하는 동작을 5회 반복한다.
- (0,0), (-150,150),(150,-150)위치에 draw_star함수를 이용하여
별의 한 변의 길이가 각각 50,100,150인 별을 그린다.
- 별의 색은 colorList에서 랜덤으로 발생시켜 설정한다.

코드

```
import turtle
import random
t=turtle.Turtle()
t.shape("turtle")
turtle.title("별그리기")
colorList=["red","orange","yellow","green","blue","purple"] #색 리스트
def (가) : #길이(line)를 입력받아 별그리는 함수
    for x in range((나)) : #5회 반복
        t.forward(line) #길이(line)만큼 앞으로 이동
        t.right(144) # 144도 오른쪽 회전하기
#(0,0) 위치에 별 그리기
t.color((다)) #색 설정 : colorList에서 랜덤으로 하나를 선택
t.begin_fill()
(라) #함수 호출 : 한 변의 길이가 50픽셀인 별 그리기
t.end_fill()
#(-150,150)위치에 별 그리기
t.penup() #펜을 든다
t.goto(-150,150) #(-150,150)으로 이동
t.pendown() #펜을 내린다
t.begin_fill()
t.color(random.choice(colorList)) #색 설정 : colorList에서 랜덤으로 하나를 선택
draw_star(100) #함수호출 : 한 변의 길이가 100픽셀인 별 그리기
t.end_fill()
#(150,-150) 위치에 별 그리기
t.penup()
t.goto(150,-150)
t.pendown()
t.color(random.choice(colorList)) #색 설정 : colorList에서 랜덤으로 하나를 선택
t.begin_fill()
draw_star(150) #함수호출 : 한 변의 길이가 150픽셀인 별 그리기
t.end_fill()
turtle.done()
```

실행 결과 화면



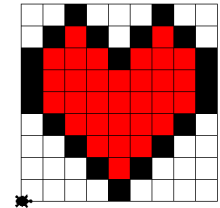
(가)	draw_star(line)	(나)	5
(다)	random.choice(colorList)	(라)	draw_star(50)

10차시_도트그림그리기

1.Ex1_하트그리기

코드

```
ver=[
    [0,0,1,0,0,0,1,0,0], #1
    [0,1,2,1,0,1,2,1,0], #2
    [1,2,2,2,1,2,2,2,1], #3
    [1,2,2,2,2,2,2,2,1], #4
    [1,2,2,2,2,2,2,2,1], #5
    [0,1,2,2,2,2,2,1,0], #6
    [0,0,1,2,2,2,1,0,0], #7
    [0,0,0,1,2,1,0,0,0], #8
    [0,0,0,0,1,0,0,0,0] #9
]
length=30 #사각형의 한 변의 길이 설정
def (가) : #square함수 정의 : 사각형 그리기
    for i in range(4) :
        t.forward(length)
        t.right(90)
def (나) : #drawRec함수 정의 : 채우기 색 지정하여 사각형그리기
    if(num==1) :
        t.fillcolor("black") #채우기 색을 검정색으로 설정
        t.begin_fill()
        square() #사각형 그리기
        t.end_fill()
    elif(num==2) :
        t.fillcolor("red") #채우기 색을 빨간색으로 설정
        t.begin_fill()
        square() #사각형 그리기
        t.end_fill()
    else :
        square()
        t.forward(length)
import turtle
turtle.title("도트 그림그리기 : 9X9 하트")
t=turtle.Turtle()
t.shape("turtle")
t.speed(0)
t.penup()
t.goto(-300,300)
t.pendown()
#도형그리기
for j in range(len(ver)):
    for i in ver[j]:
        drawRec(i)
        t.left(180)
        t.forward(length*len(ver[0]))
        t.left(90)
        t.forward(length)
        t.left(90)
turtle.done()
```



(가)	square()
(나)	drawRec(num)