

알고리즘과 수능 수학 2

2019.06.01
정보컴퓨터공학부
알고리즘 학술통아리 AICall

학습목표

프로그래밍 언어 Python3를 이용해서

1. 고등학교 수학시간에 배우는 극한 개념과 밀접하고, 유명한 프랙탈 도형들을 그리는 법을 알아보자!
2. 초등학교 때 부터 친숙했던 정삼각형, 정사각형부터 그것을 일반화한 정N각형을 그리는 법을 알아보자!

파이썬 실행하기

실습

바탕화면에 있는 PyScripter 실행

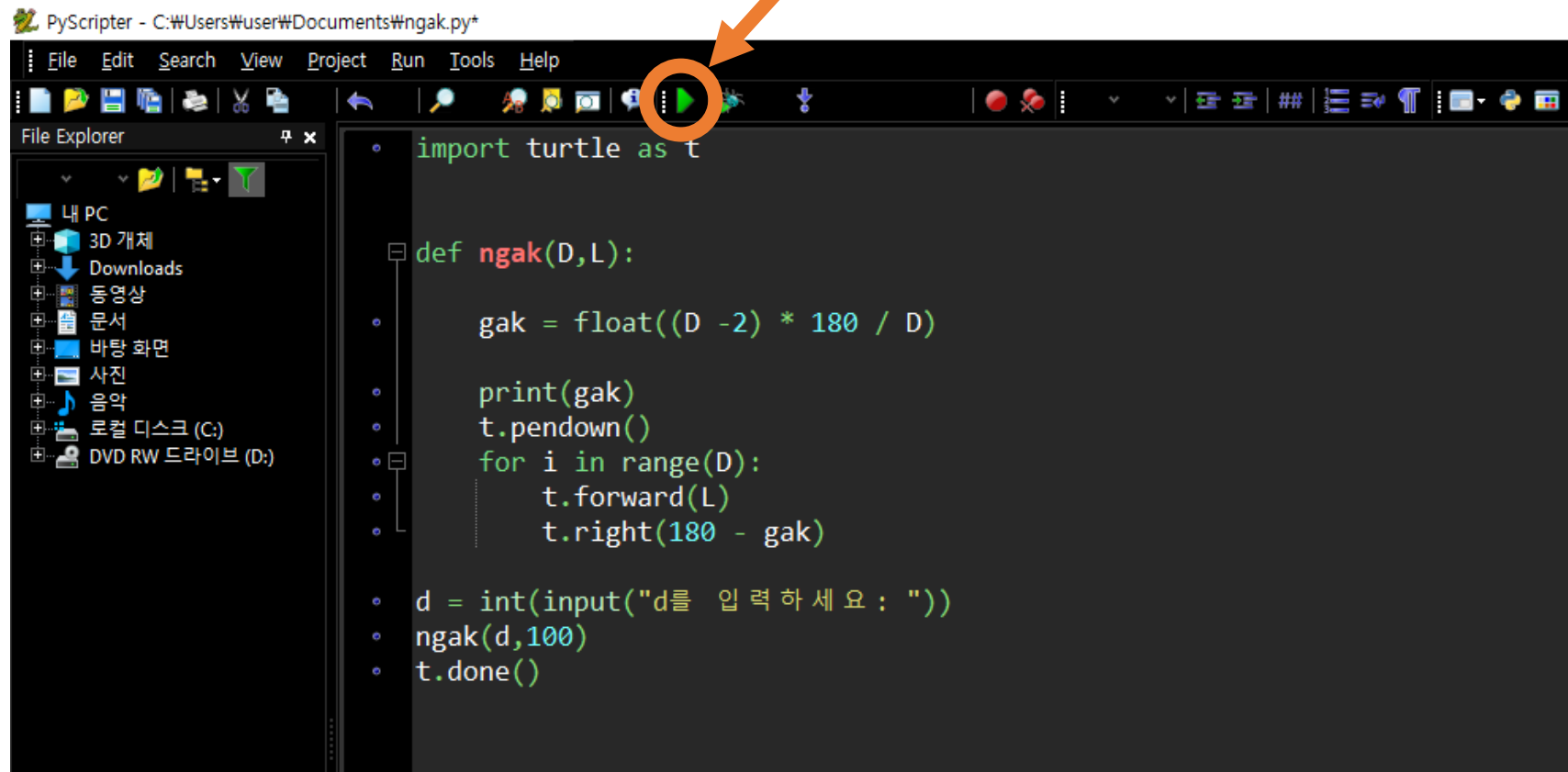


PyScripter

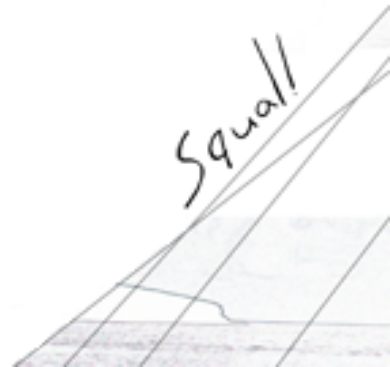
코드 실행하기

실습

Run(초록 삼각형) 클릭



4. 프랙탈



프랙탈이란?

- 프랙탈(영어: fractal) 또는 프랙털은 일부 작은 조각이 전체와 비슷한 기하학적 형태
- 프랙탈 도형은 종종 컴퓨터 소프트웨어를 이용한 **재귀**적이거나 반복적인 작업에 의한 반복되는 패턴으로 만들어짐
- 망델브로 집합, 칸토어 집합, 시어핀스키 삼각형, 페아노 곡선, 코흐 곡선 등

프랙탈을 그리기 위한 알고리즘 - 재귀

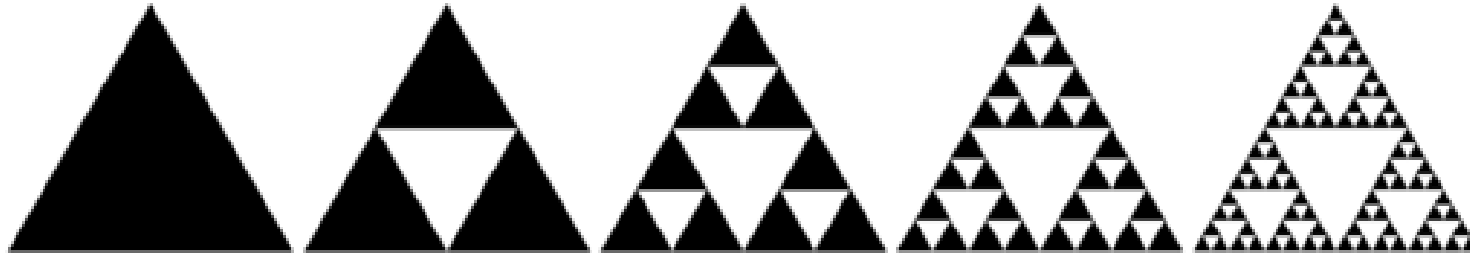
자기 자신(함수)을 재 참조 하는 방법

```
fact(n):
    if(n>1) return n*fact(n-1)
    else return 1
```

파스칼의 삼각형

$n-1C_{r-1} + n-1C_r = nC_r$

시어핀스키 삼각형



그리는 법

1. 커다란 삼각형을 그림
2. 중점 끼리 선을 연결
3. 연결된 선으로 만들어진 흰 삼각형은 제외하고
4. 검은 삼각형에 대해 2번으로 돌아가 N번 만큼 반복

turtle 사용법

프로그램 앞부분에 `import turtle as t`로 거북이 생성

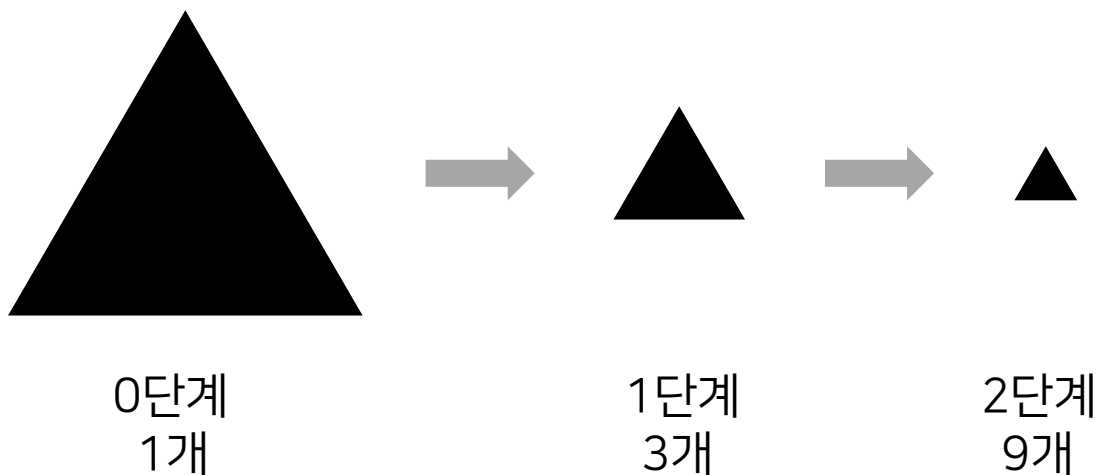
<code>t.forward(숫자)</code>	숫자만큼 앞으로 직진
<code>t.left(숫자)</code>	숫자만큼 왼쪽으로 회전
<code>t.right(숫자)</code>	숫자만큼 오른쪽으로 회전
<code>t.pendown()</code>	펜을 내림
<code>t.penup()</code>	펜을 올림

프로그램 끝부분에 `t.done()`

시어핀스키 삼각형 그리는 법

한 삼각형에서 만들어진 3개의 삼각형에 대해
다시 삼각형을 만드는 형태

재귀함수로 구현할 수 있다!



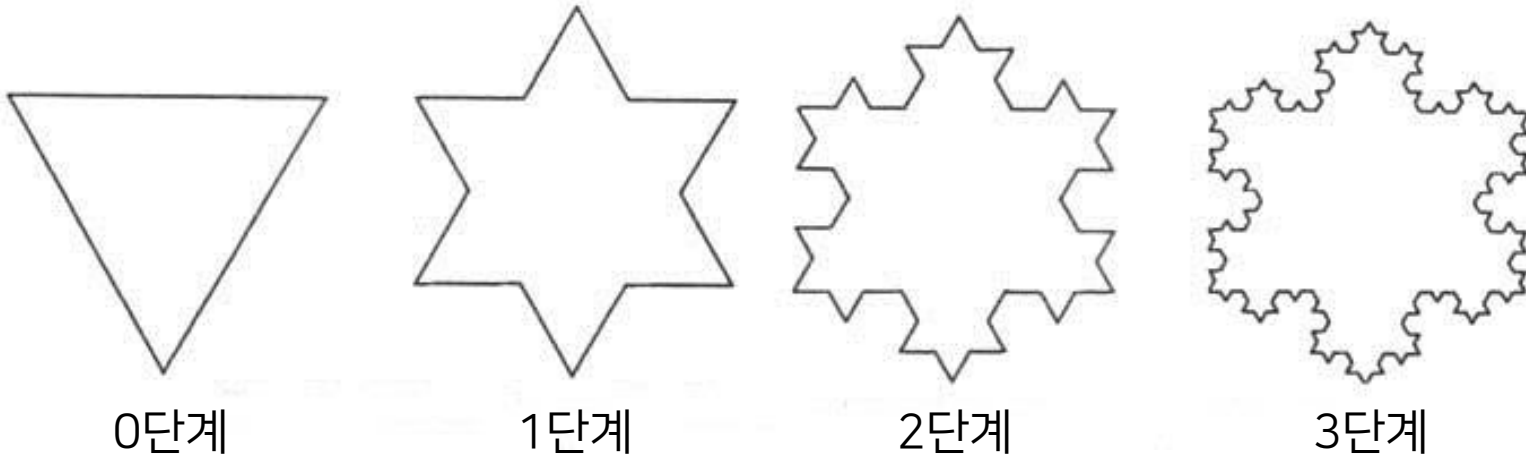
```
import turtle as t
```

```
def tri(tri_len):  
    if tri_len <= 40: # 삼각형 크기 조절  
        for i in range(0, 3):  
            t.forward(tri_len)  
            t.left(120)  
        return
```

```
new_len = tri_len / 2  
tri(new_len)  
t.forward(new_len)  
tri(new_len)  
t.backward(new_len)  
t.left(60)  
t.forward(new_len)  
t.right(60)  
tri(new_len)  
t.left(60)  
t.backward(new_len)  
t.right(60)
```

```
i = 3 # 삼각형 단계의 수  
t.speed(0)  
tri(40*2**i)  
t.hideturtle()  
t.done()
```

코흐의 눈송이



프랙탈 도형의 한 종류

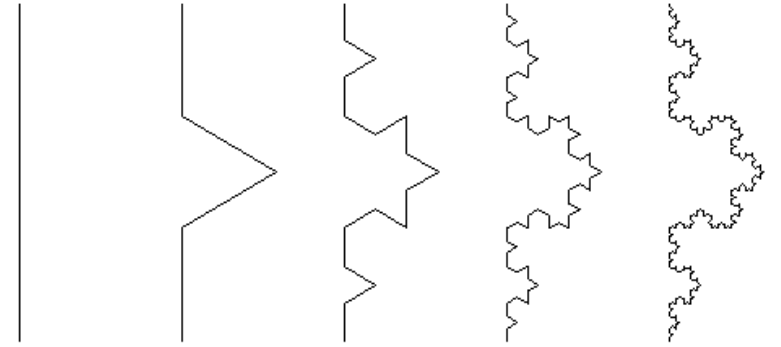
프랙탈 단계를 거칠수록 도형의 둘레가 무한하게 늘어나는 신기한 도형.

특징 : 단계를 거칠수록 도형을 둘러싸고 있는 선분의 개수가 4배씩 늘어남
(0단계 3, 1단계 12, 2단계 48... N단계 3×4^n)

코흐 곡선과 눈송이

코흐 곡선 그리기!

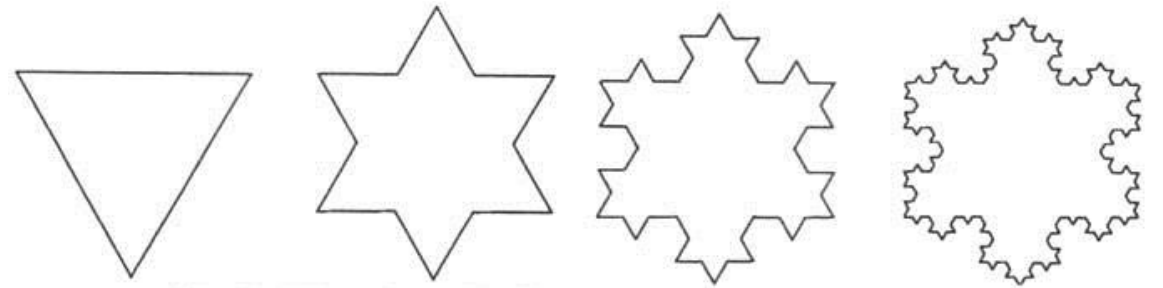
처음에 곡선의 베이스가 되는 일정한 길이의 선분을 그림
각 단계를 거칠 때 마다, 모든 선분을 3등분
가운데에 삼각형 모양 홈을 파냄



코흐 곡선

코흐 눈송이 그리기!

코흐 곡선은 0단계가 선분인 곡선이지만
코흐의 눈송이는 0단계가 정삼각형인 도형



코흐 눈송이

즉 N단계의 코흐의 눈송이는 N단계의 코흐 곡선 3개로 이루어짐!

코흐의 눈송이 code

```
import turtle
import math

koch_size = 300
koch_step = int(turtle.numinput('Koch Snowflake', '몇 단계 까지 그리겠습니까?'))

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

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)
```

```
turtle.tracer(3)
turtle.penup()
turtle.backward(koch_size/math.sqrt(3))
turtle.left(0)
turtle.pendown()
turtle.begin_fill()
for i in range(3):
    koch_line(koch_size, koch_step)
    turtle.left(-120)

turtle.end_fill()
turtle.update()
turtle.done()
```

5. 정다각형 그리기



다각형? 정다각형?

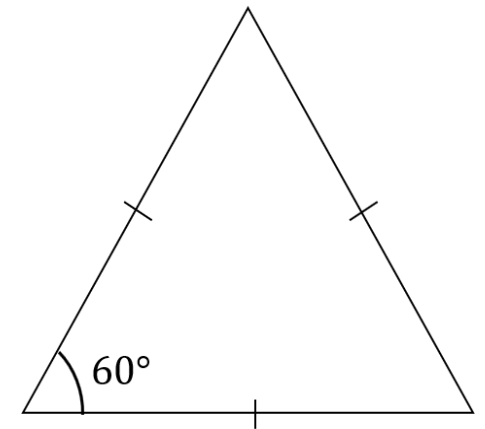
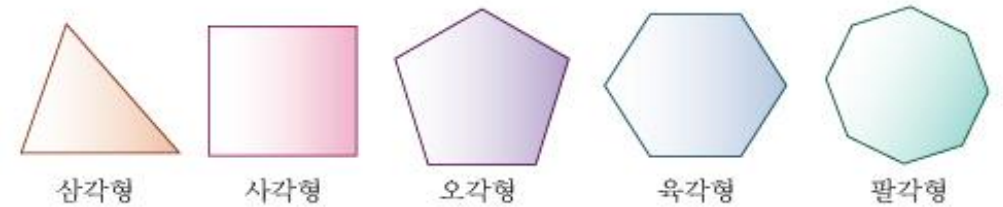
셋 이상의 선분으로 둘러싸인 평면 도형 선분의 수에 따라 삼각형, 사각형 등등으로 불림

정다각형

모든 각의 크기가 같고 모든 변의 길이도 같은 도형

정 n 각형의 한 내각의 크기는 $(180(n-2)/n)^\circ$

($n=3$ 인 정삼각형의 내각 크기 = $180 \cdot 1/3 = 60$)



정다각형 파이썬으로 그리기

D을 입력했을 때

D번만큼 일정 거리를 그린 후 외각만큼 회전함

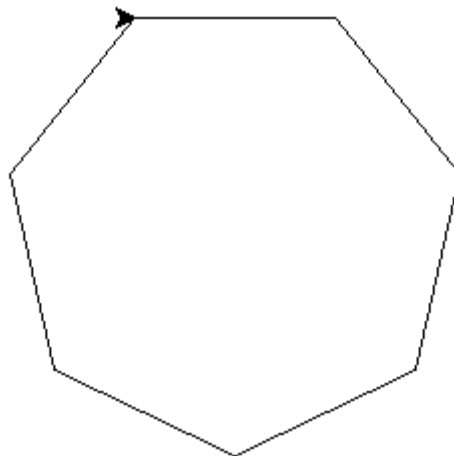
```
import turtle as t

def ngak(D,L):

    gak = float( (D -2) * 180 / D)

    print(gak)
    t.pendown()
    for i in range(D):
        t.forward(L)
        t.right(180 - gak)

ngak(7,100)
t.done()
```



예제 1. 정다각형 그리기

주의사항

- gak = float.. 앞에 Tab로 띄우기
- 대소문자 구별 중요!

```
import turtle as t

def ngak(D,L):

    gak = float((D - 2) * 180 / D)

    print(gak)
    t.pendown()
    for i in range(D):
        t.forward(L)
        t.right(180 - gak)

d = int(input("d를 입력하세요 : "))
ngak(d,100)
t.done()
```

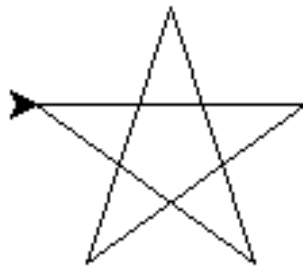
연습 1. 별 그리기

실습

(?)에 들어갈 알맞은 수는?

Hint

- 이등변삼각형
- 엇각의 성질



```
import turtle as t
```

```
def ngak(D,L):
```

```
    gak = float((D - 2) * 180 / D)
```

```
    print(gak)
```

```
    t.pendown()
```

```
    for i in range(D):
```

```
        t.forward(L)
```

```
        t.right(?)
```

```
ngak(5,100)
```

```
t.done()
```