

재귀호출함수(recursion, recursive function)

- 되부름 함수라고도 함
- 함수가 자기 자신을 부르게 작성된 함수
- 재귀호출 함수를 작성할 때는 함수가 종료되는 조건을 반드시 포함하도록 해야 함
- 그렇지 않으면 계속 자기 자신을 불러 시스템 자원을 소진시켜 비정상적으로 끝나게 됨
- 재귀호출을 사용한다고 해서 기억장소(storage)가 절약되거나 속도가 빨라지지 않는
- 하지만 프로그램이 간결해지고 이해하기가 쉬운 장점

재귀호출 프로그램의 전형적인 예 - 하노이 탑 프로그램

- 그림과 같이 놓인 구멍이 뚫린 원판(디스크)를 움직이려고 함
- 초기 상태는 A, B, C 세 개의 말뚝이 있고, A에 64개의 원판이 차례대로 놓여 있는 데, 위의 원판은 바로 아래보다 작다.
- 주어진 임무는 두번째 말뚝 B를 이용하여 모든 원판을 말뚝 C로 옮기는 것이다. 단 조건은 원판은 한 번에 하나씩 옮기며, 어떤 경우에라도 위에 놓인 원판은 아래 원판보다 클 수 없다.



A

B

C

factorial(!)의 계산 함수를 재귀함수로 작성 _

- 다음 조건을 이용해 임의의 자연수 n 이 주어지면 $n!$ 을 계산하는 재귀호출함수 `factorial(n)`을 작성해 보자
 - $n! = n * (n-1)!$
 - $0! = 1$

```
def factorial(n):  
    if n == 0:  
        return ____ # 재귀 함수 종료 조건. n=0이면 종료  
    else:  
        return _____ #  $n! = n * (n-1)!$   
  
n = int(input('자연수='))  
print(factorial(n))
```

Koch curve 그리기 프로그램 _

- 터틀 그래픽스와 재귀 호출을 이용해 다음과 같은 Koch curve 그리기 프로그램을 작성
- Koch curve를 그리려면 한 변의 길이가 일정한(예:300)인 선분에서 출발
- 선분을 3등분하여 가운데의 선분을 한 변으로 하는 정삼각형을 그린다(단 밑변은 안 그림)
- 이 과정을 계속 반복하면 아래 그림을 얻을 수 있다.
- 이런 커브를 자기 유사성(닮음)이 있는 프랙털(fractal)이라 한다.



Koch snow 그리기 프로그램 _

- 앞의 Koch curve 그리기 프로그램을 이용해 다음과 같은 Koch snow 그리기 프로그램을 작성
- Koch curve를 이용해 삼각형의 각 선분에 대해 재귀 호출을 반복

