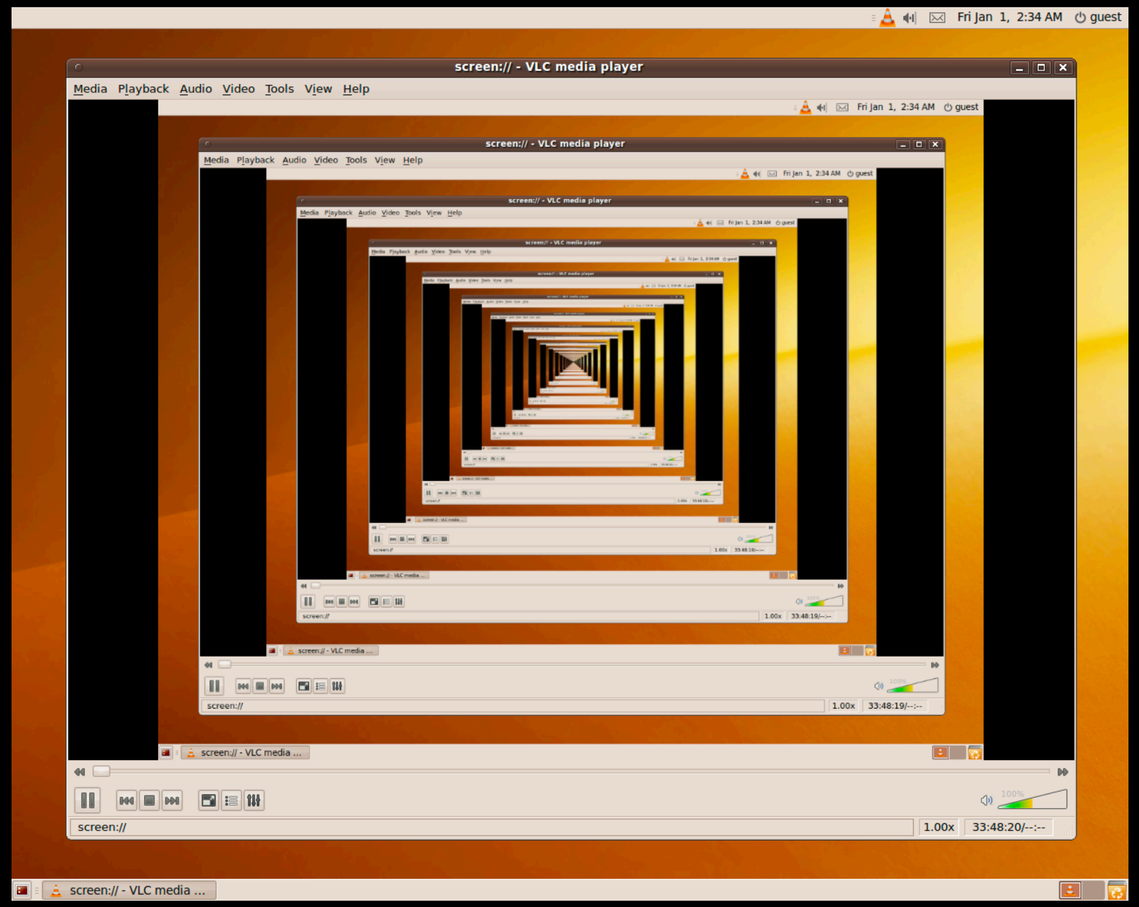
재귀함수

[AI/6] 1반 김도엽

재귀함수가 무엇인가요?

재귀함수란 무엇인가?

함수 정의 안에서 자기 자신을 참조하는 함수.
사실 이게 재귀함수의 정의 전부이다.



재귀함수의 동작 구조

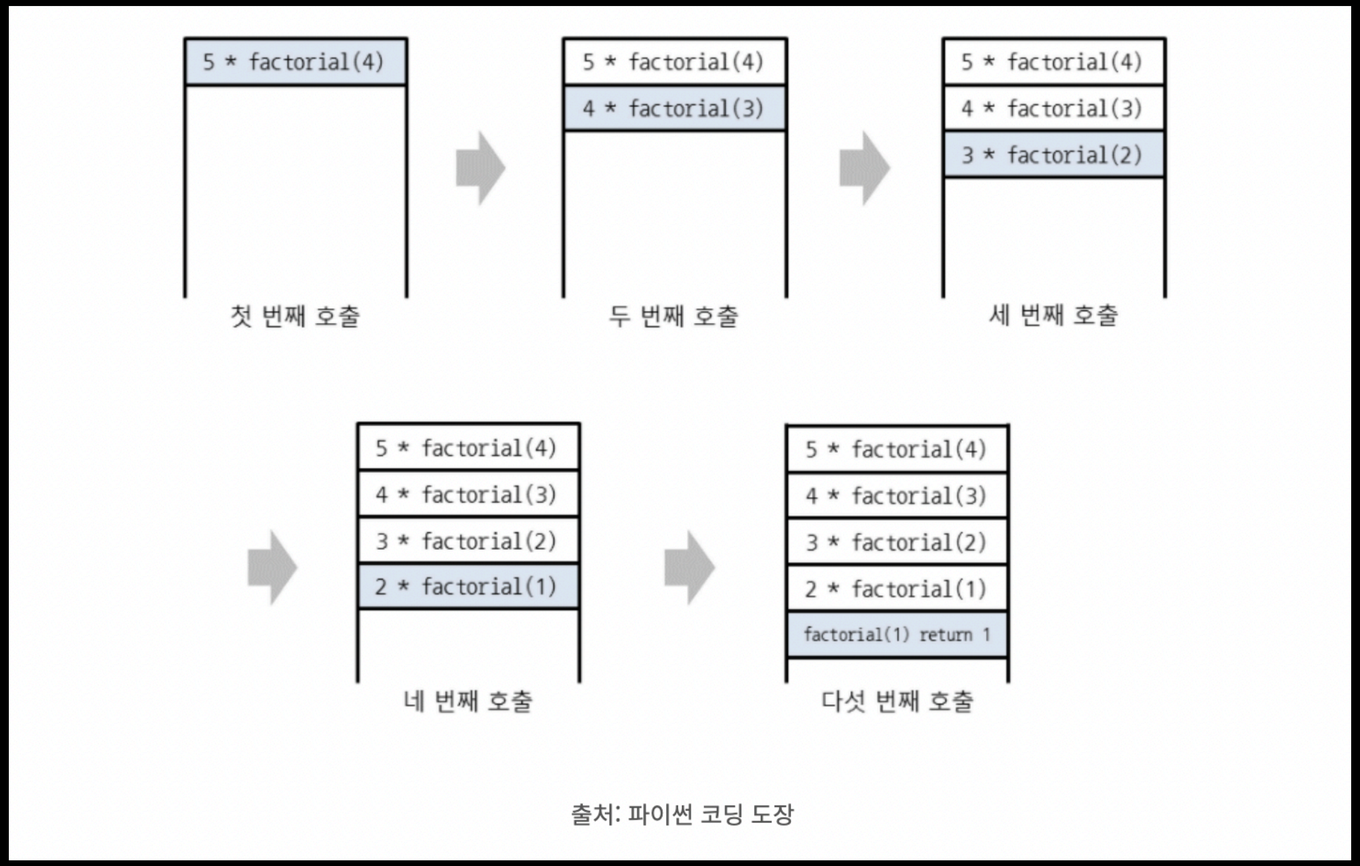
모든 함수는 스택영역을 이용해서 실행됩니다.

- 재귀함수는 함수 내부에서 자기 자신을 참조하기 때문에, 재귀함수의 탈출 조건을 만나기 전까지는 계속해서 함수의 호출을 스택에다가 쌓아둡니다.
- 탈출 조건을 만나게 되면, 해당 함수를 리턴시키면서 스택에서 하나씩 제거합니다.
- 스택 상에서 재귀함수로 인해 쌓인 모든 함수 참조가 없어지게 되면 비로소 재귀함수가 종료되는 방식입니다.

팩토리얼 함수를 이용해서 간단히 동작을 알아봅시다.

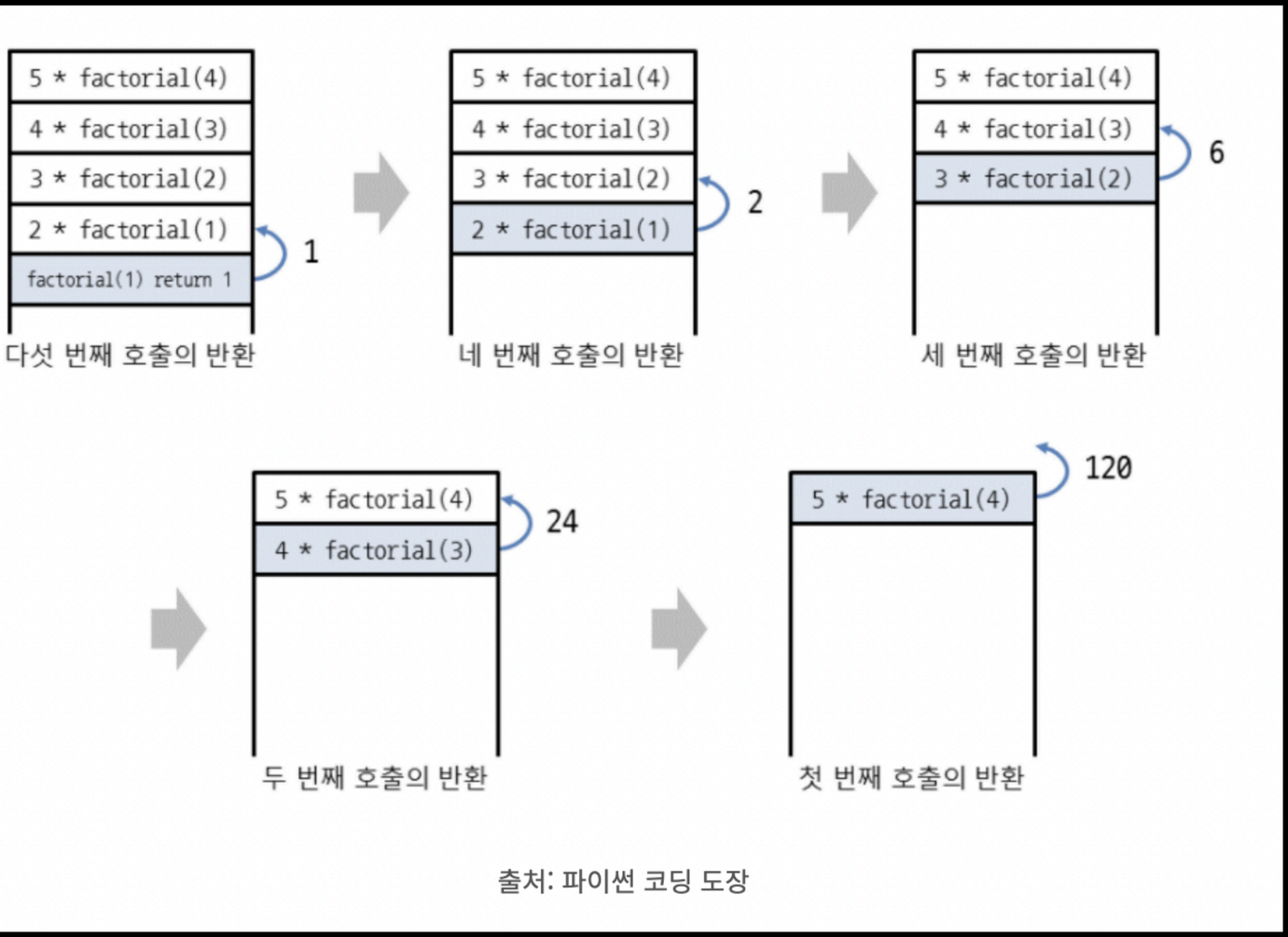
```
def factorial(n):
    if n == 1:                          # n이 1일 때
        return 1                        # 1을 반환하고 재귀호출을 끝냄

    else:
        return n * factorial(n - 1)     # n과 factorial 함수에 n - 1을 넣어서 반환된 값을 곱함

print(factorial(5))
```

1. 탈출조건을 만날 때까지 스택에다가 함수를 쌓습니다.



출처: 파이썬 코딩 도장

2. 탈출조건을 만날 때부터 스택에서 하나씩 제거합니다.



출처: 파이썬 코딩 도장

재귀함수의 동작 비교

아래 두 개의 함수를 비교해봅시다.

singleDirection.py

```python
# singleDirection.py
def recursive(n, number_list):
    if len(number_list) == 5:
        return

    for i in range(5):
        if i not in number_list:
            number_list.append(i)
            print(n, number_list)
            recursive(n + 1, number_list)

recursive(0, [])
```

traverseExample.py

```python
# traverseExample.py
def recursive(n, number_list):
    if len(number_list) == 5:
        return

    for i in range(5):
        if i not in number_list:
            number_list.append(i)
            print(n, number_list)
            recursive(n + 1, number_list)
            number_list.pop()

recursive(0, [])
```

recursive() › for i in range(5) › if i not in number_list

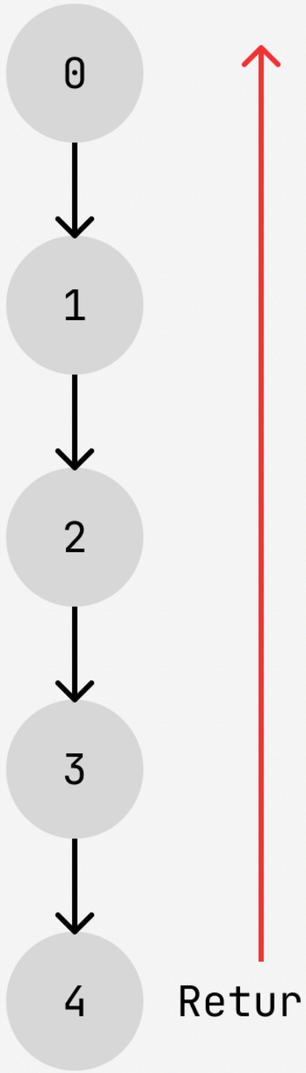
SingleDirection.py

```
0 [0]
1 [0, 1]
2 [0, 1, 2]
3 [0, 1, 2, 3]
4 [0, 1, 2, 3, 4]
```

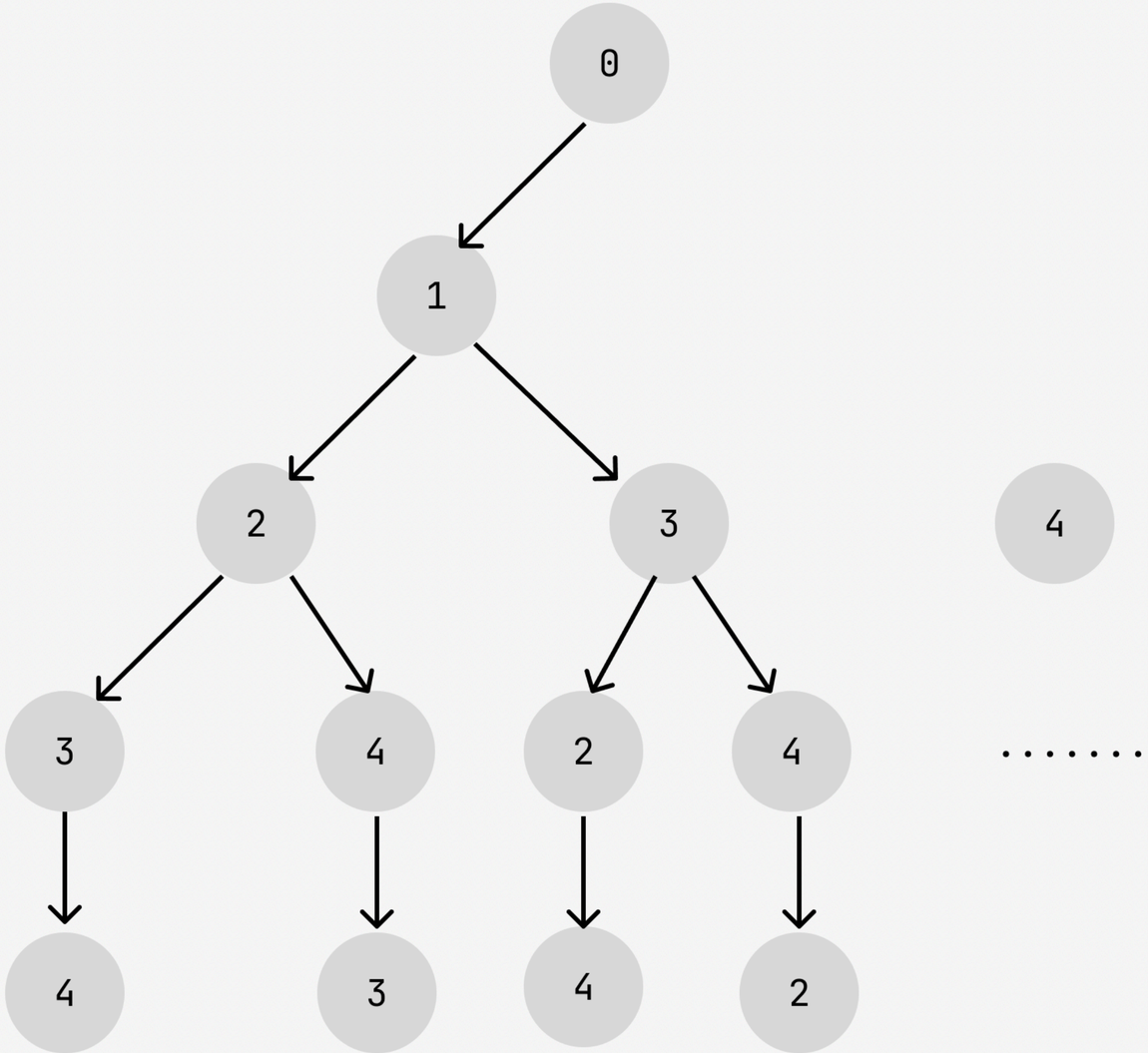
multiTraverse.py

```
0 [0]
1 [0, 1]
2 [0, 1, 2]
3 [0, 1, 2, 3]
4 [0, 1, 2, 3, 4]
3 [0, 1, 2, 4]
4 [0, 1, 2, 4, 3]
2 [0, 1, 3]
3 [0, 1, 3, 2]
4 [0, 1, 3, 2, 4]
3 [0, 1, 3, 4]
4 [0, 1, 3, 4, 2]
2 [0, 1, 4]
3 [0, 1, 4, 2]
4 [0, 1, 4, 2, 3]
3 [0, 1, 4, 3]
4 [0, 1, 4, 3, 2]
1 [0, 2]
2 [0, 2, 1]
3 [0, 2, 1, 3]
4 [0, 2, 1, 3, 4]
3 [0, 2, 1, 4]
4 [0, 2, 1, 4, 3]
2 [0, 2, 3]
3 [0, 2, 3, 1]
4 [0, 2, 3, 1, 4]
3 [0, 2, 3, 4]
4 [0, 2, 3, 4, 1]
2 [0, 2, 4]
3 [0, 2, 4, 1]
4 [0, 2, 4, 1, 3]
3 [0, 2, 4, 3]
4 [0, 2, 4, 3, 1]
```

singleDirection.py



multiTraverse.py



왜 이런 차이가 발생했을까요?

- [0, 1, 2, 3, 4]가 출력된 이후를 생각해봅시다