

제5장 5-2 함수의 활용 마무리

▶ 3가지 키워드로 정리하는 핵심 포인트

1. ()는 내부에서 자기 자신을 호출하는 함수를 의미합니다.
2. ()는 한 번 계산한 값을 저장해 놓은 후, 이후에 다시 계산하지 않고 저장된 값을 활용하는テクニック입니다.
3. ()은 함수의 흐름 중간에 return 키워드를 사용해서 코드 들여쓰기를 줄이는 등의 효과를 가져오는テクニック입니다.

정답 : 1. 재귀 함수, 2. 메모화, 3. 조기 리턴

▶ 확인문제

1. 다음 빈칸을 재귀 함수로 만들어서 리스트를 평탄화하는 함수를 만들어 보세요. 리스트 평탄화는 중첩된 리스트가 있을 때 중첩을 모두 제거하고 풀어서 1차원 리스트로 만드는 것을 의미합니다.

```
def flattern(data):  
    ()
```

```
example = [[1, 2, 3], [4, [5, 6]], 7, [8, 9]]  
print("원본", example)  
print("변환", flattern(example))
```

정답

```
output = []  
for item in data:  
    if type(item) == list:  
        output += flattern(item)  
    else:  
        output.append(item)
```

return output

2. 패밀리 레스토랑에서 여러 개의 테이블에 나누어 앉으려고 합니다. 이때 한 사람만 앉는 테이블이 없게 그룹을 지어야 합니다. 인원 수를 나누는 패턴만 구하면 되며, 누가 어디에 앉는지 등은 고려하지 않아도 괜찮습니다. 예를 들어 6명이라면 다음과 같은 네 가지 경우를 생각할 수 있습니다.

2명 + 2명 + 2명, 2명 + 4명, 3명 + 3명, 6명

앉힐 수 있는 최소 사람 수 = 2

앉힐 수 있는 최대 사람 수 = 10

전체 사람의 수 = 100

memo = {}

def 문제(남은 사람 수, 앉힌 사람 수):

key = str([남은 사람 수, 앉힌 사람 수])

if key in memo:

()

if 남은 사람 수 < 0:

()

if 남은 사람 수 == 0:

()

()

()

()

print(문제(전체 사람의 수, 앉힐 수 있는 최소 사람 수))

정답

```
return memo[key]
return 0, return 1
count = 0
    for i in range(앉힌 사람 수, 앉힐 수 있는 최대 사람 수 + i):
        count += 문제(남은 사람 수 - i, l)
memo[key] = count
return count
```