34강. 재귀함수와 메모화

혼자 공부하는 파이썬 - 윤인성 -

https://www.youtube.com/playlist?list=PLBXuLgInP-5kr0PclHz1ubNZgESmliuB7

TABLE OF CONTENTS

1. 팩토리얼로 재귀함수 맛보기

1. 반복문으로 팩토리얼 구하기

```python
# 함수를 선언합니다.
def factorial(n):
    # 변수를 선언합니다.
    output = 1
    # 반복문을 돌려 숫자를 넣습니다.
    for i in range(1,n+1):
        output *= i
    return output

print(factorial(5))
```

어떤 값이라도 1을 곱하면 변화가 없기 때문에 초기값을 1로 설정했습니다.

2. 재귀함수로 팩토리얼 구하기

```python
# 함수를 선언합니다.
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

print(factorial(5))
```

팩토리얼의 경우는 아무 형태로 만들어도 크게 상관이 없습니다.

2. 피보나치 수열로 재귀함수 개념 익히기

```
def f(n):
    if n == 1:
        return 1
    if n == 2:
        return 1
    else:
        return f(n-1) + f(n-2)

print(f(6))
```

피보나치 수열을 재귀함수로 구현하는 코드는 왼쪽과 같습니다.

하지만 **재귀함수의 치명적인 단점**이 있는데 한번 계산했던 것도 다시 계산을 하고 있으니 함수가 비효율적으로 많이 호출된다는 것입니다.
(*아무리 컴퓨터라 하더라도 처리시간이 길어짐*)

f(6)을 구한다고 치면 f(5)와f(4)의 값을 알아야하고,

이 때 f(5)값을 구하려면 f(4)와 f(3)의 값을 알아야하고… 정답을 구하기 위해서 가지치듯 경우가 확장 되기 때문에 *시간복잡도가 너무 커짐.*

3. 메모화

이러한 재귀함수의 문제를 해결하기 위해서 메모화라는 것을 사용합니다.

메모화(memoization)는 한 번 계산한 값을 저장해 놓은 후, 이후에 다시 계산하지 않고 저장된 값을 활용하는 테크닉입니다.

딕셔너리에 값이 메모되어 있으면 처리를 수행하지 않고 곧바로 메모된 값을 돌려주면서 코드의 속도를 빠르게 만드는 것입니다.

앞서 살펴봤던 재귀함수(피보나치 수열)를 메모화를 사용하여 다시 만들어 보겠습니다.

```python
dictionary = {
    1: 1,
    2: 1
}

def f(n):
    if n in dictionary:
        # 메모가 되어 있으면 메모된 값을 리턴
        return dictionary[n]
    else:
        # 메모가 되어 있지 않으면 값을 구해서 딕셔너리에 넣음
        output = f(n-1) + f(n-2)
        dictionary[n] = output
        return output

print(f(200))
```

재귀함수와 많이 사용되는 기술이므로 꼭 기억해 주세요 !