

재귀함수 개요

10주차_03_01

한 동 대 학 교
김경미 교수

학습목표

2

- ▶ 재귀 함수가 무엇인지 이해하기
- ▶ 재귀 함수를 반복문으로 변경해보기

재귀 함수(Recursive function)

- ▶ 재귀(Recursion)
 - ▶ 함수가 바디에서 자기 자신을 호출하는 프로그래밍의 메소드 혹은 함수
- ▶ 재귀 함수
 - ▶ 재귀로 정의된 함수를 지칭(비공식적으로 종종 사용됨)
 - ▶ 재귀 함수에는 종료 조건을 반드시 명시해야 함
 - ▶ 그렇지 않은 경우에 무한 루프로 빠질 수 있음

재귀 함수(Recursive function)

- ▶ 재귀 함수 조건
 - ▶ Recursive Step : 자기자신을 호출하는 단계
 - ▶ Termination Step : 종료 단계

스스로 부르는 함수

```
def countdown(n):  
    print(n)  
    if n > 1:  
        countdown(n-1)  
  
countdown(5)
```

```
>>>  
===== RESTART: E:/1_Works  
5  
4  
3  
2  
1  
>>>
```

재귀 함수, pow

```
def pow(n1, n2) :  
    if n2 == 0 :  
        return 1  
    else:  
        return( n1 * pow(n1, n2-1) )
```

```
print(pow(3, 8))
```

```
>>>
```

```
===== RESTART:
```

```
6561
```

```
>>>
```

재귀 함수, $f(n) = 3$ 의 n 승

7

- ▶ $f(n) = 3$ 의 n 승, 재귀 함수 버전을 생각해보자

```
def mult3(n) :  
    if n >= 1 :  
        return 3*mult3(n-1)  
    else:  
        return 1  
  
for i in range(1,10) :  
    print(mult3(i))
```

```
3  
9  
27  
81  
243  
729  
2187  
6561  
19683  
➤ □
```

연습문제 1, 문제와 코드

- ▶ 바로 이전 슬라이드 코드를 재귀함수 아닌 문장으로 수정해 보시오!

```
def mult3(n) :  
    if n == 1 :  
        return 3  
    else:  
        return 3**n  
  
for i in range(1,10) :  
    print(mult3(i))
```


재귀 함수, pattern()

```
def pattern(n) :  
    if n == 0 :  
        print(0, end=' ')  
    else:  
        pattern(n-1)  
        print(n, end=' ')
```

```
pattern(5)  
print()
```

```
pattern(11)
```

```
>>> =====  
>>>  
0 1 2 3 4 5  
0 1 2 3 4 5 6 7 8 9 10 11  
>>>
```

재귀 함수, factorial

```
def factorial(n) :  
    print("factorial has been called with n = " + str(n))  
    if n == 1 :                                # terminate condition  
        return 1  
    else :  
        res = n * factorial(n-1)  
        print("intermediate result for ", n, " * factorial(", n-1, "): ", res)  
        return res
```

```
i = int(input("input an integer : "))  
print(factorial(i))
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise  
input an integer : 5  
factorial has been called with n = 5  
'factorial has been called with n = 4  
factorial has been called with n = 3  
factorial has been called with n = 2  
factorial has been called with n = 1  
intermediate result for 2 * factorial( 1 ): 2  
intermediate result for 3 * factorial( 2 ): 6  
intermediate result for 4 * factorial( 3 ): 24  
intermediate result for 5 * factorial( 4 ): 120  
120  
>>> |
```

연습문제 2, 문제와 코드

11

- ▶ 2개의 양의 정수 값을 파라미터로 power 함수를 재귀함수로 만드시오

```
def pow(n1, n2) :  
    if n2 == 0 :  
        return 1  
    else:  
        return( n1 * pow(n1, n2-1) )  
  
print(pow(3, 4))
```

- ▶ 재귀 함수가 무엇인지 이해하기
 - ▶ 함수가 body에서 자기 자신을 호출하는 프로그래밍의 메소드 혹은 함수
 - ▶ 반드시 종료조건을 명시해야 함
- ▶ 재귀 함수를 반복문으로 변경해보기

목표 달성 질문

13

- ▶ 재귀함수가 무엇인지 설명하시오
- ▶ 재귀함수로 만들어진 코드를 반복문으로 바꿔서 코딩 가능할까요?

감사합니다