

---

# 재귀 함수의 이해

9주차\_03

한 동 대 학 교  
김 경 미 교수

# 재귀함수(Recursive function)

---

- **재귀(Recursion)**

- 함수가 바디에서 자기 자신을 호출하는 프로그래밍의 메소드 혹은 함수

- **재귀함수**

- 재귀로 정의된 함수를 지칭(비공식적으로 종종 사용됨)
- 재귀 함수에는 **종료 조건을 반드시 명시해야 함**
  - 그렇지 않은 경우에 무한 루프로 빠질 수 있음

# 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, $f(n) = 3 * n$

- $f(n) = 3 * n$ , 즉, 3의 배수의 재귀 함수 버전을 생각해 보자

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

```
>>>  
===== RESTART :  
3  
6  
9  
12  
15  
18  
21  
24  
27  
>>> |
```

# 연습문제 1, 문제와 코드

---

- 바로 이전 슬라이드의 재귀함수를 반복문을 사용하여 실행 되도록 수정해 보시오!

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

---

- 피보나치 수열을 재귀 함수로 만드시오
- 다음과 같이 항이 생성되는 수열

$$f_1 = 1$$

$$f_2 = 1$$

$$f_3 = f_1 + f_2$$

$$f_4 = f_2 + f_3$$

...

$$f_n = f_{n-1} + f_{n-2}$$

# 연습문제 2 코드

```
def fibo(n) :  
    if n == 1 or n == 2 :  
        return 1  
    else :  
        return fibo(n-1) + fibo(n-2)  
  
for i in range(1,15) :  
    print(fibo(i))
```

```
>>>  
===== RESTART :  
1  
1  
2  
3  
5  
8  
13  
21  
34  
55  
89  
144  
233  
377
```

# 숙제

---

- 연습문제 1, 2를 입력한 코드와
- 실행 결과를 캡처하여 게시판에 올리시오

# 요약

---

- 재귀 함수가 무엇인지 이해한다
- 재귀 함수를 반복문으로 변경 해 본다
- 재귀함수에서는 종료 조건이 중요하다

---

# 감사합니다

9주차\_03 재귀 함수의 이해