



# CNU 데이터 분석 교육



3<sup>rd</sup> lecture

"Loops, Control Statements and Functions"

2021 - 10 - 19

# 지난시간?

1. Operations : Arithmetic, Relational, Logical
2. Data type : int, float, str, list, tuple, dict
3. 선형대수 라이브러리 : Numpy (as np)

# 오늘은 무엇을?

1. 반복문(`for`, `while`)에 대하여 학습
2. 조건문(`if`, `else`, `elif`)에 대하여 학습
3. 연습문제

# for

for 변수 in 데이터: (콜론으로 마무리!)  
    들여쓰기 (tab) 후 작성

Ex) For k in [1, 2, 3, 4, 5]:  
    print(k\*\*2)

① k = 1 대입 → print(1\*\*2) 실행 → 1 출력

② k = 2 대입 → print(2\*\*2) 실행 → 4 출력

③ k = 3 대입 → print(3\*\*2) 실행 → 9 출력

...

⑤ k = 5 대입 → print(5\*\*2) 실행 → 25 출력

종료

# for

Ex) 1 부터 10까지의 합은?

$a = 0$

for x in range(1,10):

$a = a + x$

①  $x = 1$  대입  $\rightarrow a = 0 + 1$ , 즉  $a = 1$

②  $x = 2$  대입  $\rightarrow a = 1 + 2$ , 즉  $a = 3$

③  $x = 3$  대입  $\rightarrow a = 3 + 3$ , 즉  $a = 6$

...

⑩  $x = 10$  대입 후... 종료

# for

Ex)  $a_{n+1} = 3a_n - 2a_{n-1}$ ,  $a_0 = 1, a_1 = 2$  수열 구하기

$a = [1, 2]$

for x in range(10):

$a\_new = 3*a[x+1] - 2*a[x]$

$a.append(a\_new)$

①  $x = 0$  대입  $\rightarrow a\_new = 3*a[0+1] - 2*a[0] = 3*2 - 2*1 = 4$

$a.append(a\_new)$  :  $[1, 2]$  에 4를 추가  $\rightarrow a = [1, 2, 4]$

# for

Ex)  $a_{n+1} = 3a_n - 2a_{n-1}$ ,  $a_0 = 1, a_1 = 2$  수열 구하기

```
a = [1, 2]
```

```
for x in range(10):
```

```
    a_new = 3*a[x+1] - 2*a[x]
```

```
    a.append(a_new)
```

②  $x = 1$  대입  $\rightarrow a_{\text{new}} = 3*a[1+1] - 2*a[1] = 3*4 - 2*2 = 8$

$a.append(a_{\text{new}})$  :  $[1, 2, 4]$  에 8를 추가  $\rightarrow a = [1, 2, 4, 8]$

# for

Ex)  $a_{n+1} = 3a_n - 2a_{n-1}$ ,  $a_0 = 1, a_1 = 2$  수열 구하기

```
a = [1, 2]
```

```
for x in range(10):
```

```
    a_new = 3*a[x+1] - 2*a[x]
```

```
    a.append(a_new)
```

③  $x = 2$  대입  $\rightarrow a_{\text{new}} = 3*a[2+1] - 2*a[2] = 3*8 - 2*4 = 16$

$a.append(a_{\text{new}})$  :  $[1, 2, 4, 8]$  에 16를 추가  $\rightarrow a = [1, 2, 4, 8, 16]$



# If, else, elif

if 조건문:

조건문이 참(True)이면 실행, 거짓이면 패스.

if 조건문:

조건문이 참(True)이면 실행.

else:

조건문이 거짓(False)이면 실행.

# If, else, elif

if 조건문 1:

조건문 1이 참(True)이면 실행

elif 조건문 2:

조건문 1이 거짓(False)이고, 조건문 2가 참 (True) 이면 실행

else:

조건문 1이 거짓 (False) 이고, 조건문 2가 거짓 (False) 이면 실행

# If, else, elif

Ex)

x = 8

```
if x < 10:
```

```
    print( "x는 10보다 작습니다." )
```

```
else:
```

```
    print( "x는 10보다 크거나 같습니다" )
```

→ x는 10보다 작습니다. 출력

# While

while 조건문:

조건문이 참(True) 이면 계속 실행, 거짓이면 정지.

while True:

무한 loop..., 이럴 때 사용하는 것이 *break*

## 문제 1.

다음을 이용하여  $\pi$ 의 근사값을 구하여라. (n까지의 합을 구하면 됨)

$$\frac{1}{1+x^2} = \sum_{k=0}^{\infty} (-x^2)^k = 1 - x^2 + x^4 - x^6 + x^8 \dots$$

$$\sum_{k=0}^n \frac{1}{2k+1} (-1)^k$$

$$\int \frac{1}{1+x^2} = \int \sum_{k=0}^{\infty} (-x^2)^k = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9 \dots = \arctan(x)$$

$$\arctan(1) = \frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots$$

## 문제 1.

정답

```
x = 0
n = 1000
for k in range(n):
    x += 4*(-1) ** k / (2*k+1)

print(x)
```

```
# 문제 1
# arctan(1) = np.pi/4 구하기

x = 0
n = 1000
for k in range(n):
    x += 4*(-1) ** k / (2*k+1)

print(x)
```

## 문제 2.

For, while과 if 문을 사용하여 아래 방법을 참고해  $\sqrt{2}$  를 소수 7번째 자리까지 구하여라.

$x = 1 \rightarrow x^2 = 1, x^2 < 2$  이므로  $x = 2$  일때와 비교

$x = 2 \rightarrow x^2 = 4, x^2 > 2$  이므로  $x = 1.1$  일때와 비교,

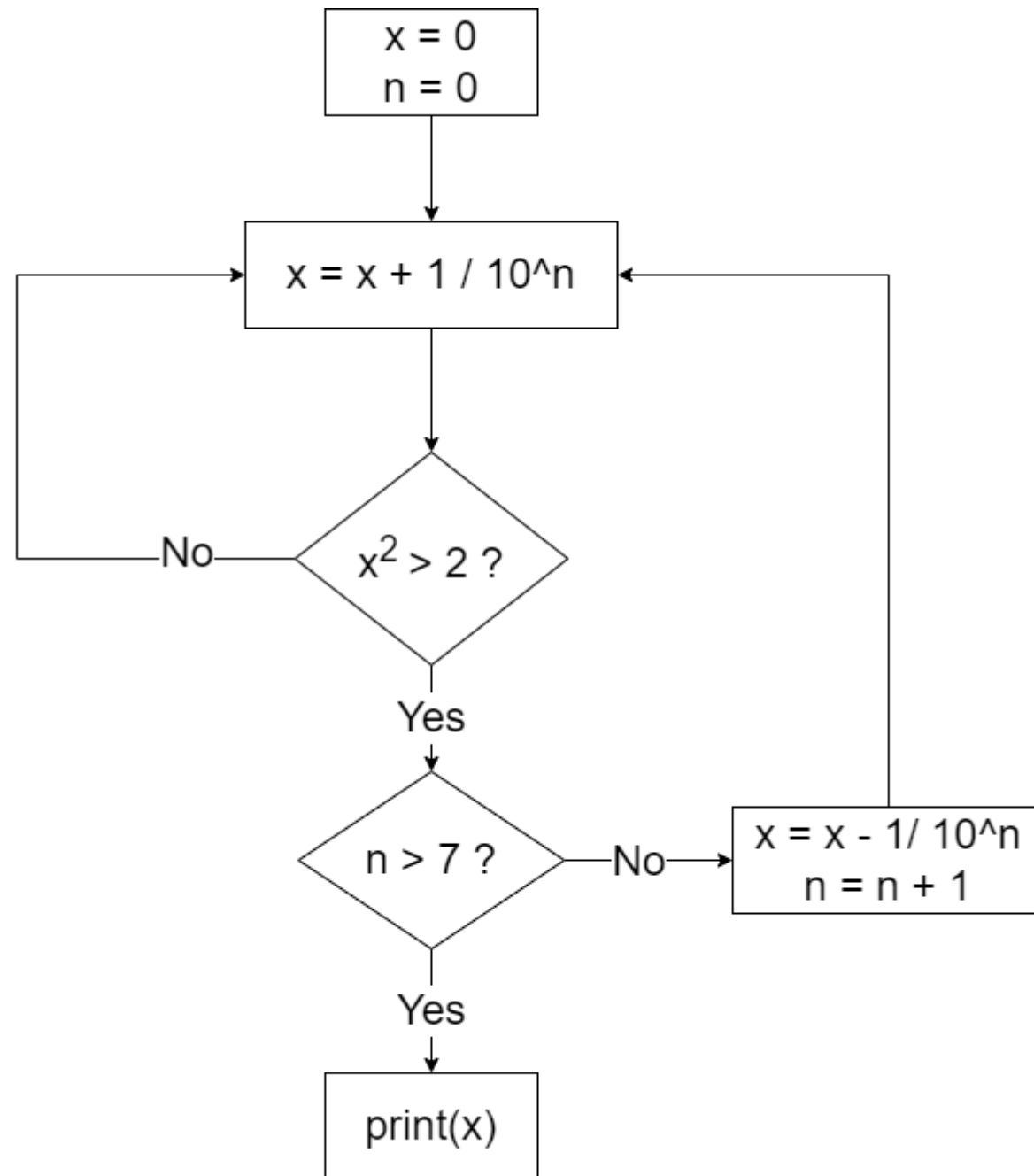
$x = 1.1 \rightarrow x^2 = 1.21, x^2 < 2$  이므로  $x = 1.2$  일때와 비교,

...

$x = 1.5 \rightarrow x^2 = 2.25, x^2 > 2$  이므로  $x = 1.41$  일때와 비교,

## 문제 2.

힌트 2 :





## 문제 2.

정답

```
x = 0
```

```
n = 0
```

```
while n < 9:
```

```
    x = x + 1 / 10**n
```

```
    if x**2 > 2:
```

```
        x = x - 1 / 10**n
```

```
        n = n + 1
```

```
# 문제 2  
# sqrt(2) 구하기
```

```
x = 0
```

```
n = 0
```

```
while n < 9:
```

```
    x = x + 1 / 10**n
```

```
    if x**2 > 2:
```

```
        x = x - 1 / 10**n
```

```
        n = n + 1
```

```
print(round(x, 7))
```

```
import numpy as np
```

```
print(np.sqrt(2))
```

```
1.4142136
```

```
1.4142135623730951
```

# Def

def 함수이름(변수 1, 변수 2, ...)

변수 1, 변수 2... 를 이용한 코드 작성

return 결과

함수이름(변수 1, 변수 2, ...) 실행 → 결과 출력

# Def

Ex)

```
def my_function(x,y):  
    result = x**2 + y**2
```

```
    return result
```

```
my_function(1,2)
```

→  $1**2 + 2**2 = 5$  출력

# Def

기존 함수의 이름을 그대로 사용하지 않도록 조심!

```
ex) print(x,y):  
    return x+y
```

이러면 우리가 여태 쓰던 `print( "가나다" )` 이런 코드가 작동 불가...

## 문제 3.

다음 함수를 만들어라

1.  $f(x) = x(x + 1)(x + 2)$

2.  $g(x, y) = e^x \cos y$

힌트 : 지수함수와 삼각함수는 `numpy`를 이용하여 계산 가능

## 문제 3.

정답

```
def f(x):  
    return x * (x + 1) * (x + 2)  
  
import numpy as np  
def g(x,y):  
    return np.exp(x) * np.cos(y)
```

```
def f(x):  
    return x * (x + 1) * (x + 2)  
  
import numpy as np  
def g(x,y):  
    return np.exp(x) * np.cos(y)  
  
print(f(2))  
print(g(3,4))
```

## 문제 4.

### 편의점 쇼핑하기

상품	가격
바나나우유 (a	900
라면 (b	1400
맥주 (c	2300
삼각김밥 (d	1000
샌드위치 (e	2400

힌트 : Input 으로 되어야 하는 것은 pay, [a,b,c,d,e]

'pay' 원으로 각 상품을 a, b, c, d, e 개 구매했을 때의 총 가격과 거스름돈을 출력하고, 만약 지불한 금액이 넘는다면 돈이 부족하다고 출력하는 함수 `price_change()`를 만들어라.

## 문제 4.

편의점 쇼핑하기

힌트

1. def price\_change(**pay, wish\_list**):

2. if **내 돈 >= 총 가격**:

**총 가격, 거스름돈** = 내 돈 - 총 가격

else:

    돈이 부족해!



## 문제 4.

정답

```
def price_change(pay, wish_list):  
  
    banana_milk = 900  
    triangle_Bab = 1200  
    sandwich = 2400  
    bear = 2500  
    soda = 1500  
  
    a, b, c, d, e = wish_list  
  
    total_price = a*banana_milk + b*triangle_Bab + c*sandwich + d*bear + e*soda  
  
    if pay >= total_price:  
        print("총 가격은 {}원 이며, 잔돈은 {}원 입니다.".format(total_price, pay - total_price))  
    else:  
        print("돈이 부족합니다")  
  
price_change(10000, [1,1,1,1,1])  
  
총 가격은 8500원 이며, 잔돈은 1500원 입니다.
```

## 문제 5.

1부터 입력된 정수  $n$ 까지 사이의 수  $a, b, c$  ( $a < b < c$ ) 에 대하여 피타고라스 수의 pair를 출력하고, 총 몇 개가 있는지 알려주는 함수 `pythagorian_pair()`를 만들어라.

```
힌트 : n이 input
for a in range(1, n+1):
    for b in range(a, n+1):
        for c in range(b, n+1):
            if -----:
                -----
            else:
                -----
```

## 문제 5.

정답

```
def pythagorian_pair(n):  
    pythagorian_list = []  
  
    for a in range(1,n+1):  
        for b in range(a,n+1):  
            for c in range(b,n+1):  
                if a**2 + b**2 == c**2:  
                    pythagorian_list.append([a,b,c])  
                else:  
                    pass  
  
    print("1부터 {}까지의 피타고라스 수는 {}개가 있다.".format(n, len(pythagorian_list)))  
  
    return pythagorian_list  
  
pythagorian_pair(20)  
  
1부터 20까지의 피타고라스 수는 5개가 있다.  
[[3, 4, 5], [5, 12, 13], [6, 8, 10], [8, 15, 17], [9, 12, 15]]
```

# 공지

## 1. 시험 잘 봐!

고  
맴

고 맴뚱니다



담에뵈시당