

# 데이터 처리를 위한 Python 프로그래밍 입문

## 4-1강. 조건문

ERICA 2018-2

# 강의 내용

- ▶ 데이터 타입 : boolean
- ▶ 비교 연산자
- ▶ 조건문 : if
- ▶ 조건문 : if – else

# 데이터 타입 : boolean

## ▶ boolean형

- ▶ True : 참(맞음, 같음)
- ▶ False : 거짓(틀림, 다름)

```
>>>  
>>> a = True  
>>> type(a)  
<class 'bool'>  
>>>
```

```
>>>  
>>> b = TRUE  
Traceback (most recent call last):  
  File "<pyshell#6>", line 1, in <module>  
    b = TRUE  
NameError: name 'TRUE' is not defined  
>>> c = true  
Traceback (most recent call last):  
  File "<pyshell#7>", line 1, in <module>  
    c = true  
NameError: name 'true' is not defined  
>>>
```

**대소문자 유의하여야 합니다.**

# 비교 연산자(1/6)

## ▶ 비교 연산자 기호

비교 연산자	내 용
$x == y$	x와 y의 값이 같으면 True를 반환하는 연산자
$x != y$	x와 y의 값이 다르면 True를 반환하는 연산자
$x > y$	x의 값이 y의 값보다 크면 True를 반환하는 연산자
$x < y$	x의 값이 y의 값보다 작으면 True를 반환하는 연산자
$x >= y$	x의 값이 y의 값보다 크거나 같으면 True를 반환하는 연산자
$x <= y$	x의 값이 y의 값보다 작거나 같으면 True를 반환하는 연산자

# 비교 연산자(2/6)

## ▶ 비교 연산자 활용

```
>>>  
>>> n = 9 - 3  
>>> n == 6  
True  
>>> n == 4  
False  
>>>
```

할당연산자 = 와  
비교연산자 == 구분하여야 합니다.

## ▶ 숫자의 짝수인지 홀수 인지 확인

```
>>> N = input("숫자를 입력하세요 : ")  
숫자를 입력하세요 : 11  
>>> flag = int(N) % 2  
>>> flag == 0  
False  
>>>  
  
>>> N  
'11'  
>>>
```

정수로 형변환 하여야 함을 유의하여야 합니다.

# 비교 연산자(3/6)

## ▶ 비교 연산자 활용

### ▶ 숫자 비교

```
>>> math_score = 92
>>> math_score > 90
True
>>>
>>> math_score = 89
>>> math_score > 90
False
>>>
>>>
```

90점 초과인지 확인

### ▶ 문자열 비교

```
>>> blood_type = 'A'
>>> blood_type == 'A'
True
>>> blood_type != 'A'
False
>>> blood_type != 'a'
True
>>> blood_type != 'B'
True
>>>
>>> type(blood_type)
<class 'str'>
>>>
```

대소문자 구분함을 유의 하여야 합니다.

# 비교 연산자(4/6)

## ▶ 비교 연산자 : boolean 식

Boolean식	내 용
x and y	x와 y가 모두 True인 경우 True를 반환하는 연산자 (x와 y중 하나라도 False인 경우 False를 반환하는 연산자)
x or y	x와 y 중 하나라도 True인 경우 True를 반환하는 연산자 (x와 y가 모두 False인 경우 False를 반환하는 연산자)
not x	X가 False인 경우 True를 반환하는 연산자 X가 True인 경우 False를 반환하는 연산자

# 비교 연산자(5/6)

## ▶ 비교 연산자 활용

```
>>> a = True
>>> b = False
>>> a and b
False
>>> a or b
True
>>>
>>> not a
False
>>> not b
True
>>>
```

```
>>> a = 3
>>> b = 6
>>> c = 9
>>>
>>> (a < b ) and ( b > c )
False
>>>
>>> ( a < b) and (b < c)
True
>>>
```

and 전 후 boolean값이 모두 True이어야 True 반환합니다.



# 비교 연산자(6/6)

## ▶ 비교 연산자 활용

```
>>> score = 96
>>> grade = 'C'
>>> (score > 90) and (grade == 'A')
False
>>> (score > 90) and (grade == 'C')
True
>>>
>>> score > 90 and grade == 'A'
False
>>> score > 90 and grade == 'C'
True
>>>
>>> score > 90 and grade >= 'C'
True
>>> score > 90 and grade >= 'B'
True
>>>
```

괄호를 없어도 boolean값이 반환되지만  
and 전 후 괄호 사용을 권장합니다.

문자를 비교연산자로 잘못 사용하면 안됩니다.

# 조건문 : if(1/4)

## ▶ 조건문 표현

### ▶ If 조건

표기 방법	내 용
if 조건: True_statements Next_statements	조건이 "True"이면 True_statements를 실행하고, 조건이 "False"이면 Next_statements를 실행

# 조건문 : if(2/4)

## ▶ 조건문 활용

- ▶ 상황 : 사용자로 부터 숫자를 입력받아 만약 숫자가 7이면,  
“Lucky Number”라는 메시지를 출력하는 프로그램

```
File Edit Format Run Options Window Help
num = int(input('Choose number between 1 to 10 : '))
if num == 7:
    print('Lucky number')
```

```
Choose number between 1 to 10 : 7
Lucky number
>>>
```

```
Choose number between 1 to 10 : 3
>>>
```

: 을 잊지 마세요.  
들여쓰기(4칸) 잊지 마세요.

7을 입력하면  
Lucky number가 출력됩니다.

7이 아닌 숫자를 입력하면  
프로그램이 종료됩니다.

# 조건문 : if(3/4)

## ▶ 조건문 활용 : 할인 계산

▶ 상황 : 하나에 1,000원하는 연필과 하나에 2,000원하는 펜이 있다.

구입시 10,000원 이상이면 10%할인해주는 프로그램 코딩

- 구매하고자 하는 연필과 펜의 개수를 사용자로 부터 입력받는다.

- 조건문을 활용하여 할인율을 적용한다

## ▶ 결과 화면

```
연필 개수 입력 : 3  
펜 개수 입력 : 3  
총합 : 9000 원  
>>>  
>>>
```

```
연필 개수 입력 : 6  
펜 개수 입력 : 6  
10% 할인되었습니다.  
총합 : 16200 원
```

# 조건문 : if(4/4)

- ▶ 조건문 활용 : 할인 계산
  - ▶ 해결 코드

```
File Edit Format Run Options Window Help
price_pencil = 1000
price_pen = 2000
num_pencil = int(input("연필 개수 입력 : "))
num_pen = int(input("펜 개수 입력 : "))

total_price = (price_pencil * num_pencil) + (price_pen * num_pen)

if total_price >= 10000:
    total_price = total_price * 0.9
    print('10% 할인되었습니다.')

print("총합 : ", int(total_price), "원")
```

# 조건문 : if – else(1/8)

- ▶ if – else 조건문 표현
  - ▶ If – else 조건

표기 방법	내 용
if 조건: True_statements else : False statements Next_statements	조건이 "True"이면 True_statements를 실행하고, 조건이 "False"이면 False_statements를 실행한 후 Next_statements로 이동

# 조건문 : if – else(2/8)

## ▶ if – else 조건문 활용

▶ 상황 : 나이를 입력받아 20 초과이면,

‘You are an adult.’를 출력하고 그렇지 않으면

‘You are a baby.’를 출력하는 프로그램

```
File Edit Format Run Options Window Help
your_age = int(input('How old are you?'))

if your_age >20:
    print('You are an adult.')
else:
    print('You are a baby.')

How old are you?21
You are an adult.
>>>
How old are you?19
You are a baby.
```

# 조건문 : if – else(3/8)

- ▶ if – else 조건문 활용 : 영어 단어 맞추기
  - ▶ 상황 : 내일이라는 뜻의 영어단어를 입력하고  
철차가 맞으면 '맞습니다.'를 출력하고 그렇지 않으면  
'틀렸습니다.'를 출력하는 프로그램
  - ▶ 결과 화면

```
내일의 뜻을 갖은 영어 단어를 쓰시오 : today
틀렸습니다.
>>>
내일의 뜻을 갖은 영어 단어를 쓰시오 : tomorrow
맞습니다.
>>>
```



# 조건문 : if – else(4/8)

- ▶ if – else 조건문 활용 : 영어 단어 맞추기
  - ▶ 해결 코드

```
File Edit Format Run Options Window Help
word = input('내일의 뜻을 갖은 영어 단어를 쓰시오 : ')

if (word == "tomorrow"):
    print('맞습니다.')
else:
    print('틀렸습니다.')
```

# 조건문 : if – else(5/8)

- ▶ if – else 조건문 활용 : 저녁 메뉴
  - ▶ 상황 : 동전의 '앞' 또는 '뒤'라는 단어를 입력하고  
앞이면 '중국요리'메뉴를 결과화면과 같이 출력하고 그렇지 않으면  
뒤이면 '일본요리'를 결과화면과 같이 출력하는 프로그램
  - ▶ 결과 화면

```
동전의 면을 입력하시오(앞/뒤) : 앞
오늘 저녁은 중국 요리 로 결정되었습니다.
>>>
>>>
동전의 면을 입력하시오(앞/뒤) : 뒤
오늘 저녁은 일본 요리 로 결정되었습니다.
```

# 조건문 : if – else(6/8)

- ▶ if – else 조건문 활용 : 저녁 메뉴
- ▶ 해결 코드

```
File Edit Format Run Options Window Help
coin = input('동전의 면을 입력하시오(앞/뒤) : ')

if (coin == "앞"):
    dinner_menu = '중국 요리'
else:
    dinner_menu = '일본 요리'

print('오늘 저녁은 ',dinner_menu, '로 결정되었습니다.')
```

# 조건문 : if – else(7/8)

- ▶ if – else 조건문 활용 : 놀이기구 입장 허가
  - ▶ 상황 : 놀이공원 놀이기구는 어린이용이다. 그래서  
키는 155이하, 몸무게는 50이하인 경우에만 입장하여 사용가능  
키와 몸무게를 입력 받아 놀이기구 사용여부를  
결과 화면과 같이 출력하는 프로그램
  - ▶ 결과 화면

```
키를 입력하세요 : 170  
몸무게를 입력하세요 : 70  
입장 불가능  
>>>  
>>>  
키를 입력하세요 : 120  
몸무게를 입력하세요 : 29  
입장 가능  
>>>
```

# 조건문 : if – else(8/8)

- ▶ if – else 조건문 활용 : 놀이기구 입장 허가
  - ▶ 해결 코드

```
File Edit Format Run Options Window Help
height = int(input('키를 입력하세요 : '))
weight = int(input('몸무게를 입력하세요 : '))

if (height <= 155) and (weight <=50):
    print('입장 가능')
else:
    print('입장 불가능')
```

**Thank you**