파이썬과 함께하는 이세계 코딩 생활

3주차 : 자료형과 조건문

오늘 배울 내용

- 자료형 (data type)
- 조건문 : if, elif, else

파이썬은 동적 타이핑 언어 (from 첫 시간)

- 실행시간에 **자료형(data type)**을 검사하는 언어
- 반면, 정적 타이핑 언어는 실행 전에 **자료형**을 미리 지정해야 한다.

C언어	**파이썬**
`int num = 10;` 정적 타이핑 언어라서, int 라고 미리 지정해줘야 함	`num = 10` 동적 타이핑 언어라서, 따로 지정해주지 않음

둘 다 정수형 데이터! 정수형 말고, 다른 자료형은 어떤 게 있을까? 🤔

자료형

- Numeric Types
  - int, float, complex
- Boolean Type
  - boolean
- Sequence Types
  - list, tuple, range
- Text Sequence Types
  - str
- Binary Sequence Types
  - bytes, bytearray
- Set types
  - set, frozenset
- Mapping Types
  - dict

- Numeric Types
  - int, float
- Boolean Type
  - boolean
- Sequence Types
  - list, tuple, range
- Text Sequence Types
  - str
- Mapping Types
  - dict

자료형 (나머지는 나중에~!)

- 정수형 : int
- 실수형 : float
- 불리언형 : boolean
- 문자열형 : str

} 숫자형

type()

- 객체의 type을 return 해주는 함수

```
>> help(type)

Help on class type in module builtins:

class type(object)
 |  type(object_or_name, bases, dict)
 |  type(object) -> the object's type
 |  type(name, bases, dict) -> a new type
 …
```

```
>> num = 10
>> type(num)

int
```

숫자형

- 정수형 : int
- 실수형 : float

```
>> num = 518
>> type(num)

int
```

```
>> num = 5.18
>> type(num)

float
```

```
num = input() # num 입력
print(num) # num 출력
type(num) # a의 자료형 출력
```

```
>> 518
518
str
```

int() 숫자나 문자열을 정수로 바꿔주는 함수

```
num = int(input())
print(num)
type(num)
```

```
>> 518
518
int
```

float() 숫자나 문자열을 실수로 바꿔주는 함수 (가능하다면!)

```
num = float(input())
print(num)
type(num)
```

```
>> 5.18
5.18
float
```

숫자형 데이터에 사용할 수 있는 연산들

연산	설명
x + y	x와 y의 합
x - y	x와 y의 차
x * y	x와 y의 곱
x / y	x를 y로 나눈 값
x // y	x를 y를 나눴을 때 몫
x % y	x를 y로 나눴을 때 나머지
-x	x의 부호를 바꾼 것
+x	x 그대로 임

숫자형 데이터에 사용할 수 있는 연산들

연산	결과
abs(x)	x의 절댓값 (x의 크기)
int(x)	x를 정수화
float(x)	x를 실수화

```
>> print(abs(1), abs(-2.0))
1 2.0

>> print(int(2.3), int(-4.5), int(6.7))
2 -4 6

>> print(float(8))
8.0
```

숫자형 데이터에 사용할 수 있는 연산들

연산	설명
divmod(x, y)	x // y 과 x % y 쌍
pow(x, y)	x의 y승
x ** y	x의 y승

2^2

pow(2,2)
2**2

```
>> x = 5
>> y = 2

>> print(divmod(x,y))
(2, 1)

>> print(pow(x,y))
25

>> print(x**y)
25

>> print(x^y) #🤔🤔🤔
7
```

주의할 점! '^'는 제곱 연산이 아니다.
비트연산 할 때 쓰이는, 비트 연산자이다.

비트 연산자 (2진수를 다루는 연산자)

논리 연산자

- 논리 연산이란?
  - 참, 거짓 이렇게 2가지 값 만을 가지고 행하는 연산!

연산자	뭐라고 부르는지	예시	설명
and	논리곱	a and b	둘 다 True 면 True
or	논리합	a or b	둘 중 하나만 True 면 True
not	부정	not a	아니면 True

비트 연산자 (2진수를 다루는 연산자)

연산자	설명
&	비트 AND 연산
|	비트 OR 연산
^	비트 XOR 연산
~	비트 NOT 연산
<<	지정한 수만큼 left shift 연산
>>	지정한 수만큼 right shift 연산

5^2 == 7

8 4 2 1

0101 == 5
0010 == 2
0111 == 4 + 2 + 1 == 7

이건 디지털시스템 같은 과목 배울 때 공부하면 됨...!
그냥 이런 것도 있구나~
제곱할 때 ^ 쓰지 말고 ** 쓰자~ 이런 느낌으로 소개해봤어용

숫자형 데이터에 사용할 수 있는 연산들

연산	설명
math.trunc(x)	x의 정수 부분
round(x[, n])	x를 소수점 아래 n자리까지 보이도록 반올림 n 은 default 0
math.floor(x)	x 이하의 가장 큰 정수
math.ceil(x)	x 이상의 가장 작은 정수

```
import math
>> print(math.trunc(2.3), math.trunc(-4.5), math.trunc(6.7))
2 -4 6

>> print(round(2.345, 1), round(-4.567, 2), round(6.7, 0))
2.3 -4.57 7.0

>> print(math.floor(2.3), math.floor(-4.5), math.floor(6.7))
2 -5 6

>> print(math.ceil(2.3), math.ceil(-4.5), math.ceil(6.7))
3 -4 7
```

문자열

```
>> type("Hello interface")

str
```

- str (string)
  - 문자로 이루어진 데이터의 집합
  - 'Hi', "Hello, interface!" 처럼 작은/큰 따옴표로 감싸서 표현한다.

💡 str() : 주어진 객체를 이용하여 string 객체를 만든다.

```
num = int(10)
type(num)
type(str(num))

str
```

```
>> help(str)

Help on class str in module builtins:
class str(object)
| str(object='') -> str
| str(bytes_or_buffer[, encoding[, errors]]) -> str
|
| Create a new string object from the given object.
…
```

문자열

```
s = "포자춤~ 포자춤~ 기쁨의~ 포자춤~"
print(s)
```

```
포자춤~ 포자춤~ 기쁨의~ 포자춤~
```

```
s = "포자춤~ 포자춤~ 기쁨의~ 포자춤~ '버섯머겅~' 얍"
print(s)
```

```
포자춤~ 포자춤~ 기쁨의~ 포자춤~ '버섯머겅~' 얍
```

문자열 안에 “ “, 또는 ‘ ‘ 사용 가능!
단, “ ‘ ‘ “, 또는 ‘ “ “ ‘ 이렇게 서로 다르게 감싸줘야 한다.

불리언형 (boolean)

- bool
  - 논리값, True 또는 False 만을 가지는 자료형
  - True 와 False는 예약어로 지정되어 있다.
  - 파이썬에서는 대소문자를 구분하므로 꼭 True, False로 써줘야한다.

```
>> type(True)

bool
```

bool() : 참이면 True를 반환하고, 거짓이면 False를 반환하는 함수

```
>> print(bool(51>8))
True

>> print(51>8)
True

>> print(51<8)
False
```

```
>> help(bool)

Help on class bool in module builtins:
class bool(int)
| bool(x) -> bool
|
| Returns True when the argument x is true, False otherwise.
…
```

불리언형

bool() : 참이면 True를 반환하고, 거짓이면 False를 반환하는 함수

```
>> print(bool(0))
False
>> print(bool(1))
True
>> print(bool(2))
True

>> print(bool(""))
False
>> print(bool("Hi"))
True
```

숫자형
- 0이면 False
- 다른 값들은 True

문자열
- 빈 문자열이면 False
- 채워져 있으면 True

조건문

도대체 농담곰은 왜 이랬을까...?🤔

조건문

if 아보카도가 있다 :
    우유 6개 사오기

- 아보카도가 있다 == True
- 우유 6개 사오기 **실행**

- 아보카도가 있다 == False
- 우유 6개 사와 **실행안함**

if-else 조건문

if 조건식 :

조건식이 결과가 True일 때 실행

else :

조건식의 결과가 False일 때 실행

if-elif-else 조건문 (elif는 여러 개 가능)

if 조건식1 :

    조건식1의 결과가 True일 때 실행

elif 조건식2 :

    조건식2의 결과가 True일 때 실행

else :

    조건식1의 결과가 False이고 조건식2의 결과가 False일때 실행

if-elif-else 조건문

if 조건식1 :

    조건식1의 결과가 True일 때 실행

elif 조건식2 :

    조건식2의 결과가 True일 때 실행

else :

    위에 있는 조건식들이 다 False일 때 실행

if 문은 여러 개 중첩 가능

```
if a는 사람 :
        if a는 여자 :
                if a는 초등학생 :
                        print(a)
                elif a는 중고등학생 :
                        print(a)
                else :
                        print(a)
        else :
                if a는 학생 :
                        print(a)
else :
        if a는 동물:
                print(a)
        elif a는 식물:
                print(a)
        else :
                print(a)
```

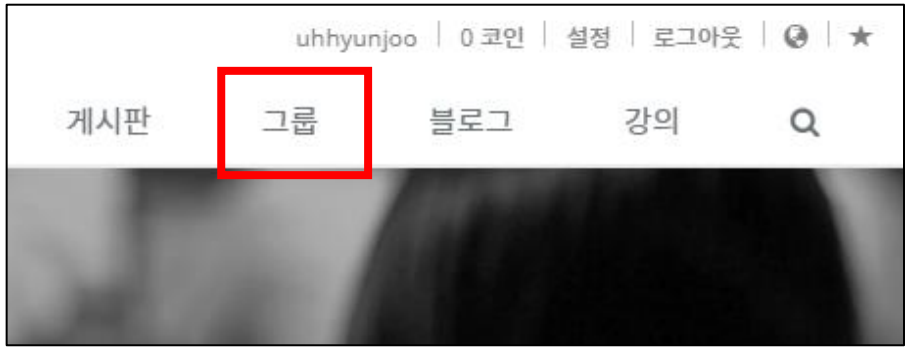
정리한 다음에 문제 푸는 거 추천!
순서도를 그리는 것도 좋다.

a와 b의 대수관계 비교하기

```
a, b = map(int, input().split())
if a > b:
  print("a가 b보다 크다")
elif a < b:
  print("a가 b보다 작다")
else :
  print("a와 b가 같다")
```

백준으로 파이썬 맛보기

- 3주차 문제집



문제 번호	제목
1330	4 두 수 비교하기
9498	4 시험 성적
2753	4 윤년
14681	4 사분면 고르기
2884	3 알람 시계