



조건문

if문

```
기본형 if boolean :  
    statement 1  
elif  
    statement 2  
else  
    statement 3
```



파이썬에서 if문에는 ':' 콜론을 붙여야 한다.



해당 조건에서 실행될 statement는 tab/space로 들여쓰기

비교연산자

↳ 조건문을 생성하기 위한 연산자.

<, >, ==, !=, >=, <=

and, or, not

↳ &는 비트연산자이다! ⚠ 파이썬에 &&는 없다!

포함확인연산자

↳ in / not in 이 존재한다.

x in 리스트/튜플/문자열이 사용가능하다.

pass

if의 조건이 참일때 아무것도 실행하고 싶지 않다면 pass 키워드를 사용한다.

```
ex) if 1 :  
    pass  
else print("0")
```

반복문

while문

C언어와 똑같다. break와 continue도 똑같이 사용 가능
다른 점도 들어쓰기와 콜론이 있다는 것!

for문

기본구조

```
for index in 반복가능한 객체:  
    statement1
```

⚠ 반복가능한 객체: 리스트, 튜플, 딕셔너리 등

```
ex) x = [1, 2, 3, 4, 5]  
for i in x:  
    print(i)
```

for i in range(10, -1, -1)?

```
ex2) x = [(1, 2), (3, 4), (5, 6)]  
for (first, second) in x:  
    print(first + second)
```

continue Statement

↳ 다음 iteration으로 넘어감

range function

↳ 정수로 이루어진 range 타입의 리스트 생성 (너장뽕)

range(x) : 0부터 x-1까지의 정수 생성

range(x, y) : x부터 y-1까지의 정수 생성

range의 결과를 list로 받고싶다 → list(range(x))

append method

↳ 리스트 누적어 사용되는 메서드.

ex) $x = [1, 2, 3, 4]$

$y = []$ # 빈 리스트 생성 $y = list()$ 도 가능

for i in x :

$y.append(i*3)$

#또는 $y = y + [i*3]$

↳ 반드시 리스트 리팅

⇒ $[3, 6, 9, 12]$ 저장

list comprehension

↳ 반복문을 리스트 내에서 실행하여, 직관적 프로그래밍

$list_name = [statement \text{ for } index \text{ in } 반복가능한 객체 \text{ if 조건}]$

ex) $x = [1, 2, 3, 4]$

$y = [num*3 \text{ for } num \text{ in } x]$

⇒ $[3, 6, 9, 12]$ 저장

ex) $y = [num*3 \text{ for } num \text{ in } x \text{ if } num \% 2 == 0]$

⇒ $[6, 12]$ 저장