
Chapter 4

제어문





제어문의 종류

1. 조건문
2. 반복문 (중첩반복문)
3. 탈출문

제어문 if

- 조건문에는 if 라는 키워드를 사용
- if 다음에는 '조건식'이 존재하는데 이 '조건식'이 참(True)이면 들여쓰기 한 문장 실행
- if 문장 끝에는 콜론(:) 을 입력
- 콜론은 블록의 시작을 의미
- if 문의 '조건식'이 참(True)일 때 실행되는 문장은 들여쓰기를 해야 합니다.

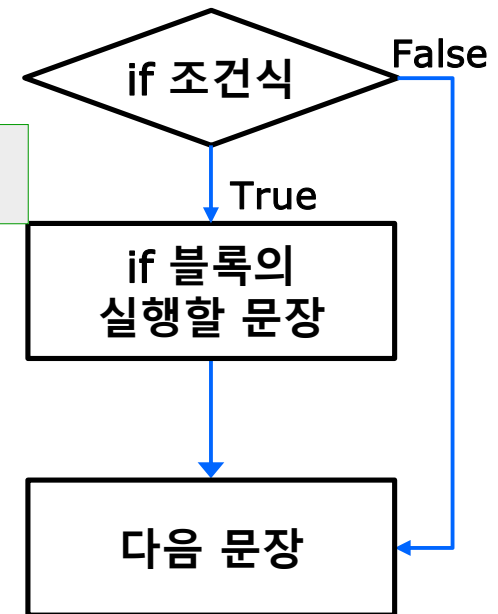
❖ 조건식은 반드시 True 또는 False 로 판별될 수 있어야 합니다.

if 조건식 :

❖ 조건문 뒤에는 반드시 콜론(:) 기호가 있어야 합니다.

조건식이 참(True)일 경우 실행할 문장 1

- ❖ 들여쓰기 간격은 같아야 합니다.
- ❖ 일반적으로 공백 4개 또는 탭(Tab)을 사용합니다.
- ❖ 들여쓰기 되어 있는 문장들을 블록이라고 합니다.



제어문 if~else

- 조건식이 참일 경우에 실행할 문장과 거짓일 경우 실행할 문장이 다를 경우 if 구문에 else 구문을 추가
- else 구문은 if 문의 조건식이 False일 경우 실행하는 블록을 정의 합니다.
- else는 단독으로 사용될 수 없으며 반드시 if와 같이 사용됩니다.

if 조건식 :

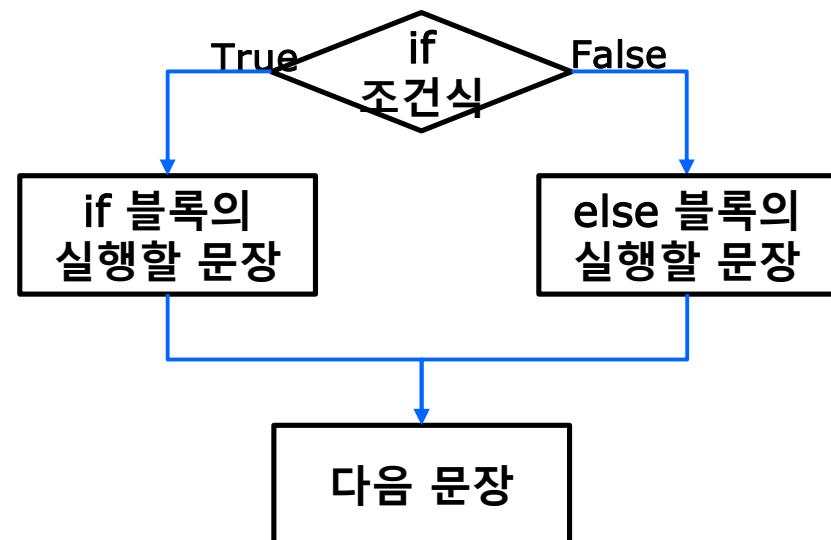
조건식이 참(True)일 경우 실행할 문장

else :

❖ else 뒤에도 반드시 콜론 (:) 기호가 있어야 합니다.

조건식이 거짓(False) 일 경우 실행할 문장

❖ if 구문 다음의 들여쓰기 간격과 else 구문 다음의 들여쓰기 간격은 같아야 합니다.



제어문 if~elif~else

- 여러 개 조건식을 사용하려면 elif 구문을 이용
- elif는 단독으로 사용 안됨

if 조건식1 :

조건식1이 참(True)일 경우 실행할 문장

...

elif 조건식2 :

조건식1이 거짓(False)이고 조건식2가 참(True)일 경우 실행할 문장

...

elif 조건식3 :

조건식1, 2가 거짓(False)이고 조건식3이 참(True)일 경우 실행할 문장

...

else :

모든 조건식이 거짓(False)일 경우 실행할 문장

❖ elif 구문은 여러 개 넣을 수 있습니다.
❖ if 구문 없이 elif 구문만 사용할 수 없습니다.

❖ else 구문은 선택사항 입니다.

❖ if 구문 다음의 들여쓰기 간격과 elif 구문 다음의 들여쓰기 간격은 같아야 합니다.

반복문 while

- while 반복문
- 조건문이 참일 동안 계속 실행

- ❖ 조건문을 검사해서 결과가 참(True)일 동안 블록을 반복 실행합니다.
- ❖ 조건문의 결과가 거짓(False)이면 else 블록을 1회 실행하고 빠져 나옵니다.

while **조건문** :

- ❖ while 조건문 뒤에는 반드시 콜론(:) 기호가 있어야 합니다.

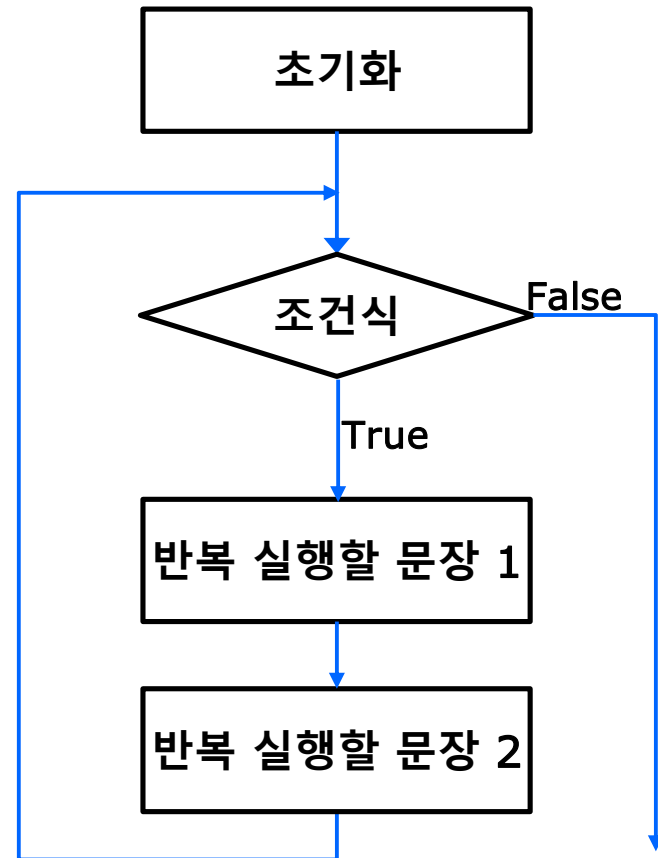
조건이 참(True)일 동안 반복 실행할 구문

...
else :

- ❖ else 구문은 선택사항입니다.
- ❖ 필요 없을 경우에는 사용하지 않아도 됩니다.

조건이 거짓(False)일 경우 실행할 문장

- ❖ 들여쓰기 되어 있는 문장들을 블록이라고 합니다.
- ❖ 들여쓰기 간격은 같아야 합니다.
- ❖ 일반적으로 공백 4개 또는 탭(Tab)을 사용합니다.



반복문 for

- for ~ in 반복문
- 나열 가능한 자료에서 자료를 모두 소비할 때 까지 처리

- ❖ 나열 가능한 자료에서 하나씩 빼내서 저장할 변수 또는 변수의 목록입니다.
- ❖ for 구문 블록에서 사용합니다.
- ❖ 단일 변수 또는 여러 개 변수가 올 수 있습니다.

- ❖ 나열 가능한 자료를 저장한 변수의 이름 또는 실행 결과입니다.
- ❖ 리스트, 튜플, 딕셔너리, 셋 등이 될 수 있습니다.

for 변수 in 나열가능한자료 :

변수의 값을 처리할 문장 1
변수의 값을 처리할 문장 2
...

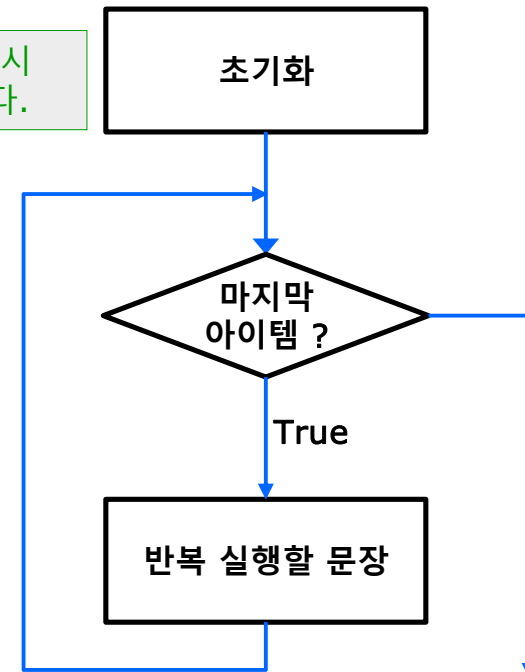
else :

- ❖ else 구분은 선택사항입니다.
- ❖ 필요 없을 경우에는 사용하지 않아도 됩니다.

변수에 값이 할당되지 않는 경우 실행할 문장

- ❖ 들여쓰기 되어 있는 문장들을 블록이라고 합니다.
- ❖ 들여쓰기 간격은 같아야 합니다.
- ❖ 일반적으로 공백 4개 또는 탭(Tab)을 사용합니다.

- ❖ for ~ in 반복문 뒤에는 반드시 콜론(:) 기호가 있어야 합니다.



반복문 range() 함수

- for문장의 나열 자료형을 쉽게 구성하게 해 주는 range() 함수.
- **range(begin, end, step)** 으로 구성됩니다.
- **End의 경우 미만**으로 해석되니 주의합니다.
- 때에 따라 step과 begin을 생략할 수 있습니다.

```
for i in range(0, 10):  
    print(i, end="\t")
```

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

- range(10) - 함수의 start가 생략되면 0부터 시작.
- range() 함수의 step은 얼마씩 증가시킨 값을 갖게 할 것인지 결정
- range(0, 10, 3)으로 수정하면 출력되는 결과는 [0, 3, 6, 9]

중첩 반복문

❖ 2차원 데이터에서 추출한 1개의 행

❖ 2중 반복문일 경우 2차원 데이터

```
for row in rows :
```

바깥 반복문을 처리할 문장

```
for data in row :
```

안쪽 반복문을 처리할 문장

❖ 안쪽 반복문의 들여쓰기

❖ 바깥 반복문의 들여쓰기

break & continue

- 탈출문 break
- Break를 만나면 break를 포함하는 반복문을 즉시 탈출합니다. (반복중단)
- Continue는 반복문 내에서 continue 이후의 한 회차의 반복문을 건너뛰고 반복문은 계속 진행됩니다.

```
num = 0
while num < 10:
    num += 1
    if num == 5:
        break
    print(num, end=" ")
```

1 2 3 4

```
num = 0
while num < 10:
    num += 1
    if num == 5:
        continue
    print(num, end=" ")
```

1 2 3 4 6 7 8 9 10



Chapter 4

수고하셨습니다