

기초 Python programming

6장 조건문 (if 문)

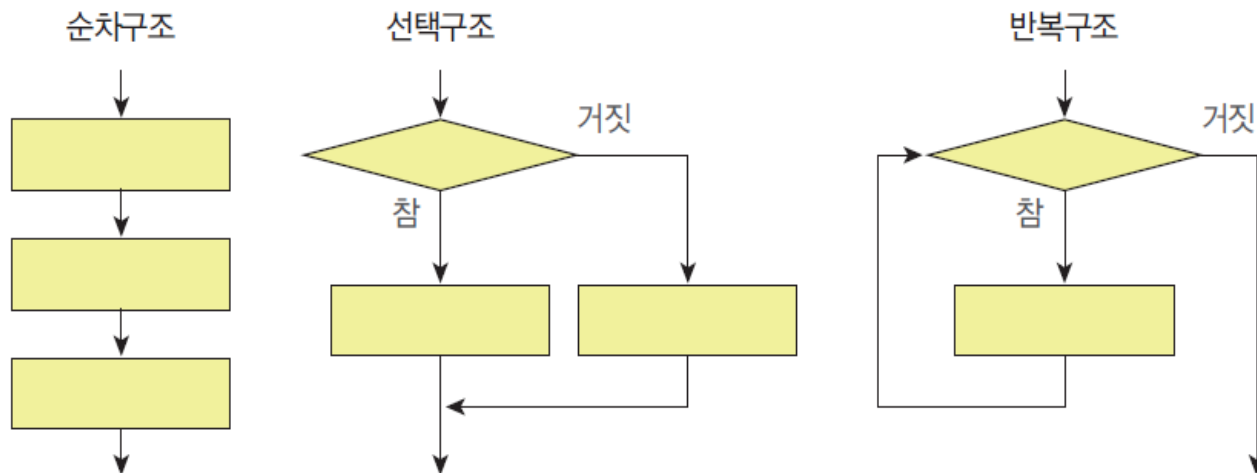


if 조건문

- 3가지의 기본 제어 구조

- ① 순차 구조(sequence) : 명령들을 순차적으로 실행하는 구조
- ② 선택 구조(selection) : 명령을 조건에 따라 선택하여 실행하는 구조
- ③ 반복 구조(iteration) : 동일한 명령들을 반복 실행하는 구조

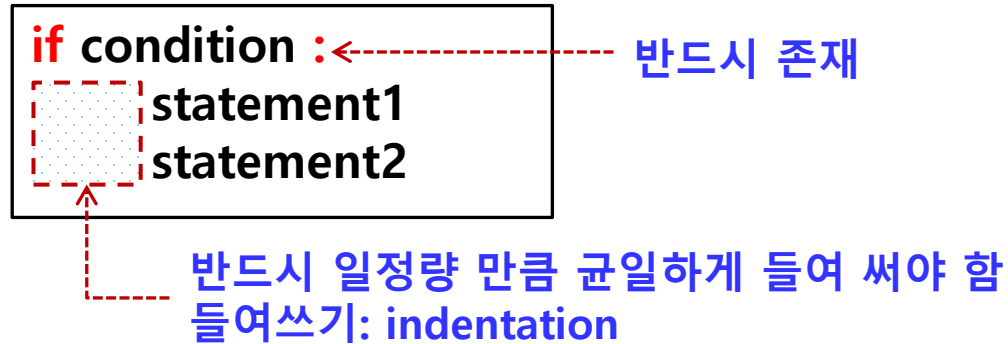
- 조건문(Conditional Statements)은 어떤 상황에 따라 실행해야 할 코드가 다를 때 사용(선택 구조에 해당)
- 선택 구조가 없다면, 프로그램은 어떤 조건, 데이터에서든지 동일한 결과를 산출하게 됨



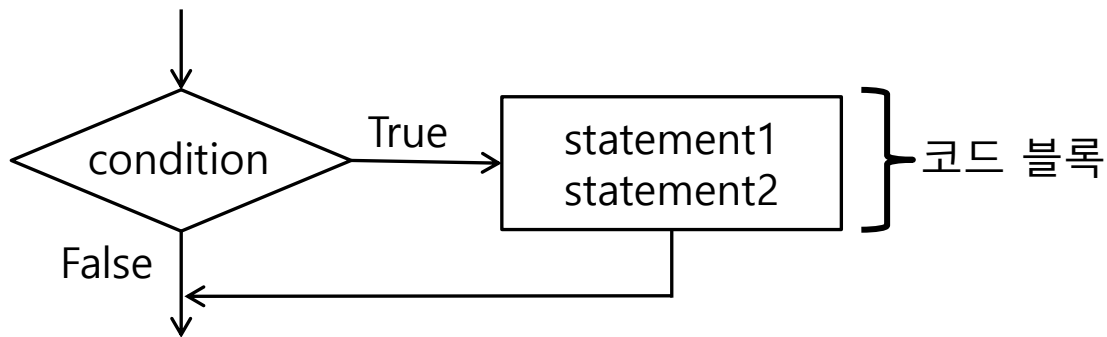
if 조건문

- if 조건문의 형식

- 조건식(condition)이 참이면 같은 크기로 들여쓰기 되어 있는 명령어들 (코드 블록)이 처리되고, 거짓이면 실행하지 않음



- if 조건문 순서도



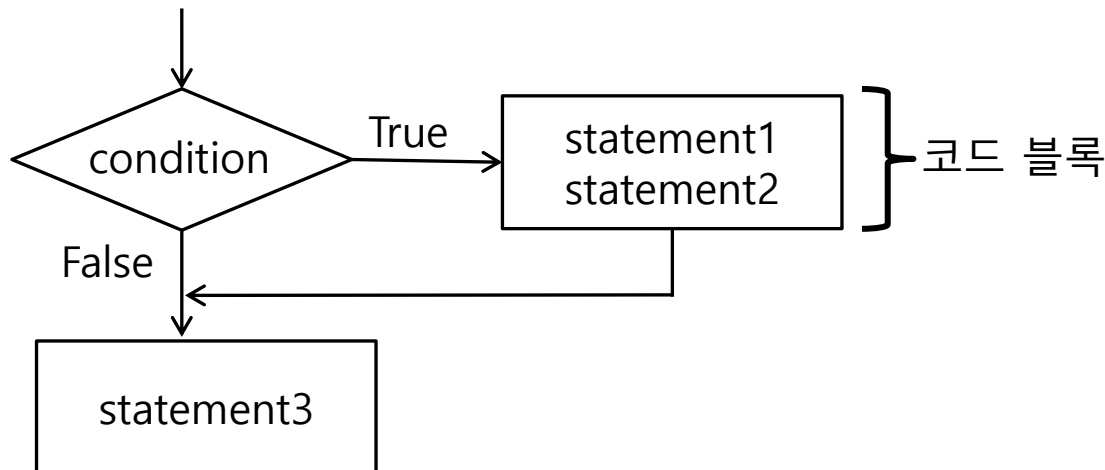
if 조건문

- if 조건문의 형식

- 조건(condition)이 True인 경우는 statement1와 statement2 실행 후, statement3 실행
- 조건(condition)이 False인 경우는 statement1와 statement2 실행 하지 않고, 들여쓰기 되어 있지 않는 statement3 실행

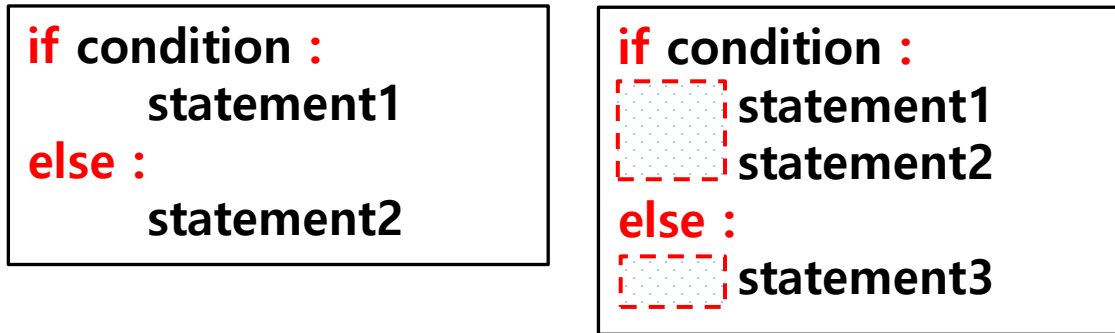
```
if condition :  
    statement1  
    statement2  
statement3
```

- if 조건문 순서도

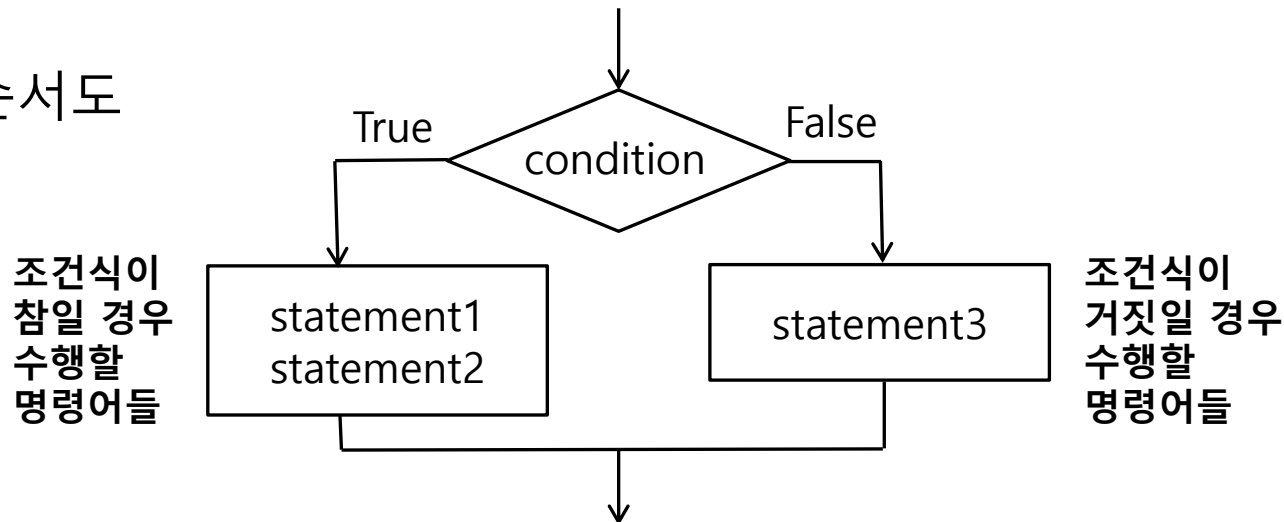


if - else 조건문

- 조건 (condition)이 False일 때, 수행할 명령어들이 있다면 else를 사용



- 순서도



조건(condition)

- if 조건문에서 "조건"이란 참과 거짓을 판단하는 명령어를 말함
- 조건을 판단하기 위해 사용되는 명령어
 - 자료형의 참과 거짓
 - in 연산자, not in 연산자
 - 관계연산자
 - 논리연산자
 - 산술연산자

```
n = -10
if n < 0 :
    n = 0
    m = 10
```

```
print(n)    # print 0
```

```
print(m)    # print 10, 만약 n 값이 >= 0 이면?
```

조건 $n < 0$ 이 True 이면 실행되는 코드 블록
균일하게 들여 써야 함

조건(condition)

- 자료형의 값으로 참과 거짓을 결정
- Python에서는 **0**, **0.0**, **""**(빈 문자열) 등은 **False**로 간주하고 나머지 값들은 모두 **True**로 간주

자료형	참	거짓
숫자	0이 아닌 숫자	0
문자열	"abc"	""
리스트	[1,2,3]	[]
튜플	(1,2,3)	()
딕셔너리	{"a":"b"}	{}

```
n = int(input("Enter a number : "))
```

```
s = "even"
```

```
if n % 2 :    # n이 홀수일 때 True
```

```
    s = "odd"
```

```
print("{} is {}".format(n,s))
```

이 명령어가 없고, 입력
데이터가 짝수인 경우,
예상 결과는?

- ① n이 홀수인 경우 2로 나눈 나머지가 1이 되어 if 문의 조건이 참이 되어 변수 s의 값이 "odd"로 변경
- ② n이 짝수인 경우는 if 문의 조건이 거짓이므로 if 블록이 실행되지 않음
- ③ print() 함수 실행

조건(condition)

```
x = float(input("Enter a number : "))  
if not x :      # x가 0일 때 조건이 True  
    print("The x's value is zero")  
else :  
    print("The x's value is not zero")
```

```
name = input("Name? : ")  
if name != "":  # enter key만 입력되었는지 체크  
    print("{} is your name".format(name))
```



```
if name :
```

```
if len(name) != 0 :
```


if - else 조건문 예제

- 입력된 연도가 윤년인지 아닌지를 판단하는 프로그램

```
year = int(input("연도를 입력하시오: "))  
if ( (year % 4 == 0 and year % 100 != 0) or year % 400 == 0):  
    print(year, "년은 윤년입니다.")  
else :  
    print(year, "년은 윤년이 아닙니다.")
```

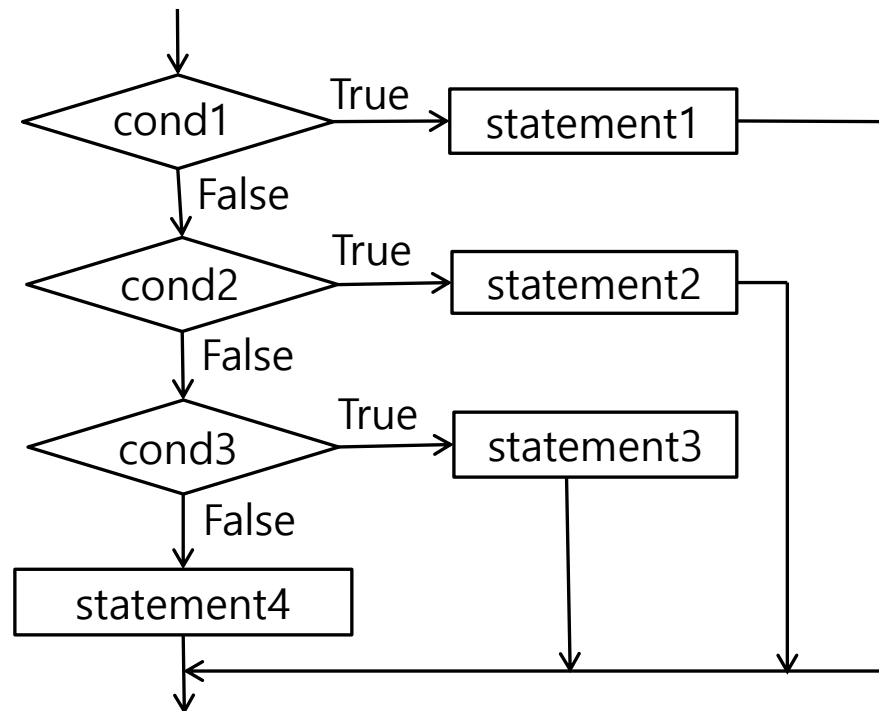
출력

```
연도를 입력하시오: 2012  
2012 년은 윤년입니다.
```

if – elif - else 조건문

- 다양한 조건을 판단하기 위해 사용
 - elif는 앞 조건문이 거짓일 때 다시 조건을 검사하는 if문
 - 마지막 else는 불필요하면 생략 가능

```
if cond1 :  
    statement1  
elif cond2 :  
    statement2  
elif cond3 :  
    statement3  
else :  
    statement4
```



if – elif - else 조건문 예제

- 입력 받은 점수에 grade를 부여하는 코드

```
score = int(input("점수 입력 : "))
if score >= 90:
    print("수")
elif 80 <= score < 90:    # elif score >= 80 :
    print("우")
elif 70 <= score < 80:
    print("미")
elif 60 <= score < 70:
    print("양")
else:
    print("가")
```

if – elif - else 조건문 예제

- 2 또는 3으로 나누어지는지 판정하는 코드

```
n = int(input("Enter a number : "))

if (n % 2 == 0) and (n % 3 == 0) :
    print("%d is divided by both 2 and 3." %n)

elif (n % 2 == 0) and (not n % 3 == 0) :
    print("%d is divided by 2 but not by 3." %n)

elif (not n % 2 == 0) and (n % 3 == 0) :
    print("%d is divided by 3 but not by 2." %n)

else :
    print("%d is neither divided by 2 nor by 3." %n)
```

중첩(Nested) if 문

- 조건을 확인 후 또 다른 조건을 검사해야 하는 경우, 중첩된 if-else 구조를 사용
 - if 문의 코드 블록 안에 또 다른 if 문을 사용(들여쓰기 유의)

```
if cond1 :  
    statements  
    . . . . .  
    if cond1_1 :  
        statements  
        . . . . .  
    statements  
elif cond2 :  
    statements  
else :  
    statements
```

여기 코드 블록에도 중첩 if 문
가능

중첩(Nested) if 문 예제

- 조건 검사 단계가 많아질수록 한 쪽으로 치우치는 코드로 작성될 수 있음

```
score = int(input("점수 입력 : "))
if score >= 90 :
    print("수")
else:
    if score >= 80 :
        print("우")
    else:
        if score >= 70 :
            print("미")
        else:
            if score >= 60 :
                print("양")
            else:
                print("가")
```

출력 예시들

```
점수 입력 : 91
수
프로그램 종료
```

```
점수 입력 : 59
가
프로그램 종료
```

중첩(Nested) if 문 예제

- 2 또는 3으로 나누어지는지 판정하는 코드(앞에서 if – elif 문으로 작성했던 코드)

```
n = int(input("Enter a number : "))

if n % 2 == 0 :
    if n % 3 == 0 :
        print("%d is divided by both 2 and 3." %n)
    else :
        print("%d is divided by 2 but not by 3." %n)

else :
    if n % 3 == 0 :
        print("%d is divided by 3 but not by 2." %n)
    else :
        print("%d is neither divided by 2 nor by 3." %n)
```