

IT 개론

5장. 조건문, 반복문

목차

1. 불리언 표현식
2. 조건 논리 – if 구문
3. 반복 논리 – while 구문
4. 반복 논리 – for 구문

1. 불리언 표현식

- ◆ True(참) 또는 False(거짓)의 값을 판단할 수 있는 문장을 불리언 표현식이라고 한다.
- ◆ Boolean 표현식의 예
 - 서울은 한국의 수도이다 → True
 - 사과는 과일이다 → True
 - 3은 5보다 크다 → False

조건식 기본 표현

if 조건문의 시작

실행 여부를 결정하는 조
건

조건식 끝에 붙이는 콜론 기
호

if people < apple / 5 :

print(' 신나는 사과 파티! 배 터지게 먹자! ')

들여쓰기(스페이스바 4번
혹은 TAB)

조건이 참(True)이면 실행할 코
드

조건식 기본 표현

코드 if.py

```
people = 3
```

```
apple = 20
```

```
if people < apple / 5:      True이므로  
    print('신나는 사과 파티! 배 터지게 먹자!') 실행
```

```
if apple % people > 0:      True이므로  
    print('사과 수가 맞지 않아! 몇 개는 쪼개 먹자!') 실행
```

```
if people > apple:          False이므로  
    print('사람이 너무 많아! 몇 명은 못 먹겠네... ') 실행하지 않고  
                                                         건너뛴
```

조건식 기본 표현

```
if ( 조건식 ) :
```

```
( 실행할 코드1 )
```

```
( 실행할 코드2 )
```

```
( 실행할 코드3 )
```

```
...
```

True / False 중 하나

조건식 기본 표현

불(Boolean) 연산

and 연산 : $p \text{ and } q$

-p와 q가 모두 참인지 판별

or 연산 : $p \text{ or } q$

-p와 q중 하나라도 참인지 판별

not 연산 : $\text{not } p$

-p의 조건을 뒤집음

조건식 기본 표현

불(Boolean) 연산

and	True	False
True	True	False
False	False	False

or	True	False
True	True	True
False	True	False

not	True	False
	False	True

조건식 기본 표현

코드 ifblock.py

```
if True:
```

```
    print('블록에 속한 코드')
```

← 블록

블록에 속한 코드

블록: 콜론(:) 다음에 들여 쓴 코드

조건식 기본 표현

코드

```
if True:
    !print('블록에 속한 코드')
    !print('한 줄 더')
    !print('또 한 줄 더')
    !
    !
```

블록을 여러 줄로 작성 할 때는 들여쓰기가 반드시
일정해야 한다.

조건식 기본 표현

코드

```
if True:
    print('블록에 속한 코드')
    print('한 줄 더')
    print('또 한 줄 더') # 두 칸 더 들여 썼으므로 오류가 발생합니다.
```

File "ifblock.py", line 4

```
    print('또 한 줄 더')
```

^

IndentationError: unindent does not match any outer indentation level

조건식 기본 표현

코드

```
if True:
    print('블록에 속한 코드')
    print('한 줄 더')
    print('또 한 줄 더')    # 두 칸 덜 들여 썼으므로 오류가 발생합니다.
```

File "ifblock.py", line 4

```
print('또 한 줄 더')
```

^

IndentationError: unexpected indent

조건식 기본 표현

코드

```
if True:
```

```
    print('블록에 속한 코드')
```

```
    print('한 줄 더')
```

```
    print('또 한 줄 더')
```

BLOCK

```
print('바깥에 있는 코드')    # 들여 쓴 만큼 다시 내어 쓰면 블록이 끝납니다.
```

들여 쓴 만큼 내어 쓰는 순간 앞에서 블록이 끝나고,
한번 끝난 블록은 다시 열 수 없다.

조건식 기본 표현

코드

```
if True:
```

바깥 블록

```
    print('블록에 속한 코드')
```

안쪽 블록

```
    if False:
```

```
        print('한 줄 더')    # 이 문장은 실행되지 않습니다.
```

또 다른 안쪽 블록

```
    if True:
```

```
        print('또 한 줄 더')
```

또 다른 안쪽 블록

```
    print('블록의 끝 코드')
```

```
print('바깥에 있는 코드')
```

블록에 속한 코드

또 한 줄 더

블록의 끝 코드

바깥에 있는 코드

조건식 기본 표현

코드

```
if True:  
    pass    # 조건이 참일 때 실행할 코드  
  
else:  
    pass    # 조건이 거짓일 때 실행할 코드
```

if 조건문이 True가 아니라면, else에 있는 코드가 실행된다.

조건식 기본 표현



조건문의 세 가지 형식

조건문에는 다음과 같이 세 가지 형식이 있습니다.

if	if-else	if-elif-else
<pre>if <조건>: <코드 블록></pre>	<pre>if <조건>: <코드 블록> else: <코드 블록></pre>	<pre>if <조건>: <코드 블록> elif <조건>: <코드 블록> else: <코드 블록></pre>

1. 불리언 표현식

◆ 관계 연산자 – 두 데이터를 비교하는 연산자

연산자	의미
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다
==	같다
!=	같지 않다

```
>>> x = 100
>>> y = 200
>>> x > y
False
>>> x >= y
False
>>> x < y
True
>>> x <= y
True
>>> x == y
False
>>> x != y
True
```

1. 불리언 표현식

◆ 논리 연산자 – 불리언 연산자

- True 또는 False 값을 피연산자로 취해서 논리값을 계산한다.
- **and**, **or**, **not** 세 종류가 있다.

논리 연산자	설 명
not x	x가 False이면 True이고, True이면 False 이다.
x and y	x와 y가 모두 True이면, x and y는 True이다. 나머지 경우에 대해서는 x and y 는 False이다.
x or y	x와 y가 모두 False인 경우에만 False이다. 즉, x와 y 둘 중에 하나라도 True이면 x or y는 True이다.

1. 불리언 표현식

◆ 논리 연산자의 예

```
>>> a = 10
>>> b = 20
>>> a > 5 and b > 10
True
>>> a > 5 and b < 5
False
>>> a < 5 and b > 10
False
>>> a < 5 and b < 5
False
>>> 0 < a < b
True
```

```
>>> a > 5 or b > 10
True
>>> a > 5 or b < 5
True
>>> a < 5 or b > 10
True
>>> a < 5 or b < 5
False
>>> not a < 5
True
>>> not b > 10
False
```

2. 조건 논리의 기본 개념

◆ 조건 논리의 기본 개념

- 프로그램을 작성하다 보면 특정 조건을 만족하는지에 따라 명령을 선택적으로 수행해야 하는 경우가 발생한다.
- 순서도에서는 다음과 같이 판단 기호를 사용하여 조건 논리를 나타낸다.
- '조건'에는 반드시 **참/거짓을 판단할 수 있는 문장**이 와야 한다. 즉, 관계 연산자 또는 논리 연산자가 와야 한다.

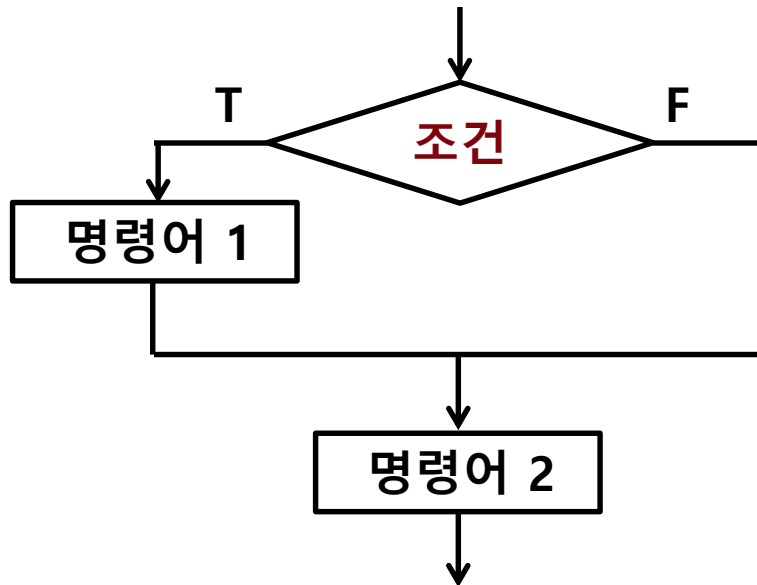


판단기호 - 기호 안에는 '조건'은 반드시 참/거짓을 판단할 수 있는 문장이어야 함.

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 1

- 주어진 조건이 참인 경우에만 어떤 특정 명령을 실행한다.
- 조건의 판단 결과가 참(T)인 경우에 명령어 1을 실행하고 이어서 명령어 2를 실행한다. 조건의 판단 결과가 거짓(F)인 경우에는 명령어 2로 넘어가 실행한다.



파이썬 코드

```
if 조건 :  
    명령어 1  
명령어 2
```

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 1 예제

- 입력받은 성적(score)이 70점 이상이면 '축하합니다.'라고 출력하고 '수고하셨습니다'라고 출력한다.
- 70점 미만이면 '수고하셨습니다'라고만 출력한다.

```
score = int(input('성적을 입력하시오 : '))  
  
if score >= 70:  
    print('축하합니다')  
print('수고하셨습니다')
```

```
>>>  
성적을 입력하시오 : 90  
축하합니다  
수고하셨습니다  
-----  
>>>  
성적을 입력하시오 : 65  
수고하셨습니다
```

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 1 예제

- 파이썬 언어로 조건 논리를 작성하는 경우에 '블록(block)'을 분명히 구별하여야 한다.
- 만약에 나이가 20세 미만인 학생들만 등록할 수 있는 과정이라면, 다음과 같이 조건문이 작성된다.

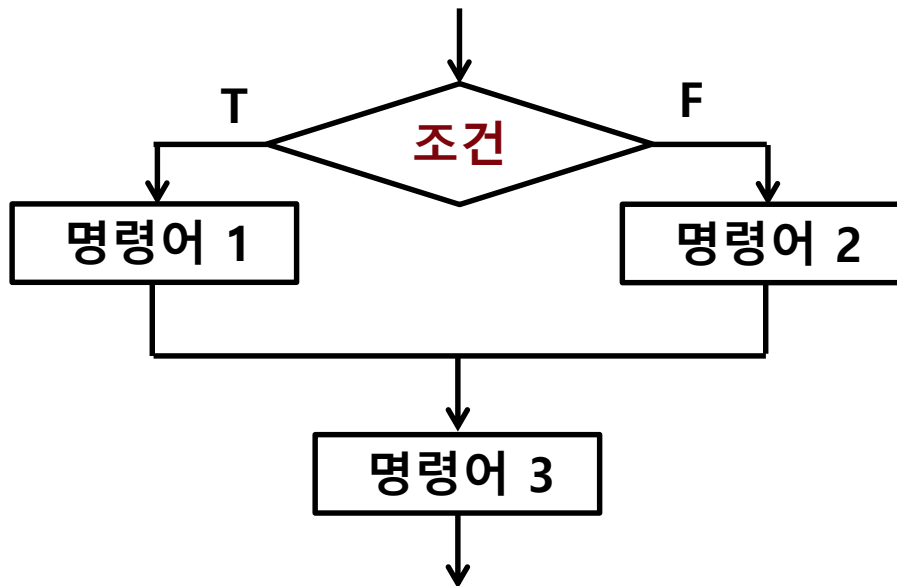
```
age = int(input('나이를 입력하시오 : '))  
if age >= 20:  
    print('너무 나이가 많습니다.')  
    print('이 반은 20세 미만의 학생들 반입니다.')  
    print('20세 이상 반을 찾아 가기 바랍니다.')
```

if 조건이 참일 때, 수행해야 하는 문장이 여러 줄이면 연관 구문들을 묶어야 한다. 이를 블록(block)이라 한다.

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 2

- 주어진 조건이 참인 경우와 거짓인 경우에 대한 명령어가 각각 존재한다.
- 조건 판단의 결과가 참(T)인 경우 명령어 1을 실행하고 이어서 명령어 3을 실행한다.
- 조건 판단의 결과가 거짓(F)인 경우에는 명령어 2를 실행하고 이어서 명령어 3을 실행한다.



파이썬 코드

```
if 조건 :  
    명령어 1  
else :  
    명령어 2  
명령어 3
```


2. 조건 논리의 기본 개념

◆ 조건 논리 유형 2 예제

- 하나의 양의 정수를 입력받는다.
- 입력받은 수가 짝수이면 '짝수입니다'라고 출력하고 '프로그램 종료'라고 출력한다.
- 입력받은 수가 홀수이면 '홀수입니다'라고 출력하고 '프로그램 종료'라고 출력한다.

```
number = int(input('양의 정수를 입력하시오 : '))
```

```
if number%2 == 0:  
    print(number, '는 짝수입니다.')
```

```
else:  
    print(number, '는 홀수입니다.')
```

```
print('프로그램 종료')
```

```
>>>  
양의 정수를 입력하시오 : 20  
20 는 짝수입니다.  
프로그램 종료  
-----  
>>>  
양의 정수를 입력하시오 : 13  
13 는 홀수입니다.  
프로그램 종료
```

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 2 예제

```
a = int(input('Enter a : '))  
b = int(input('Enter b : '))
```

```
if a > b:  
    max = a  
else:  
    max = b
```

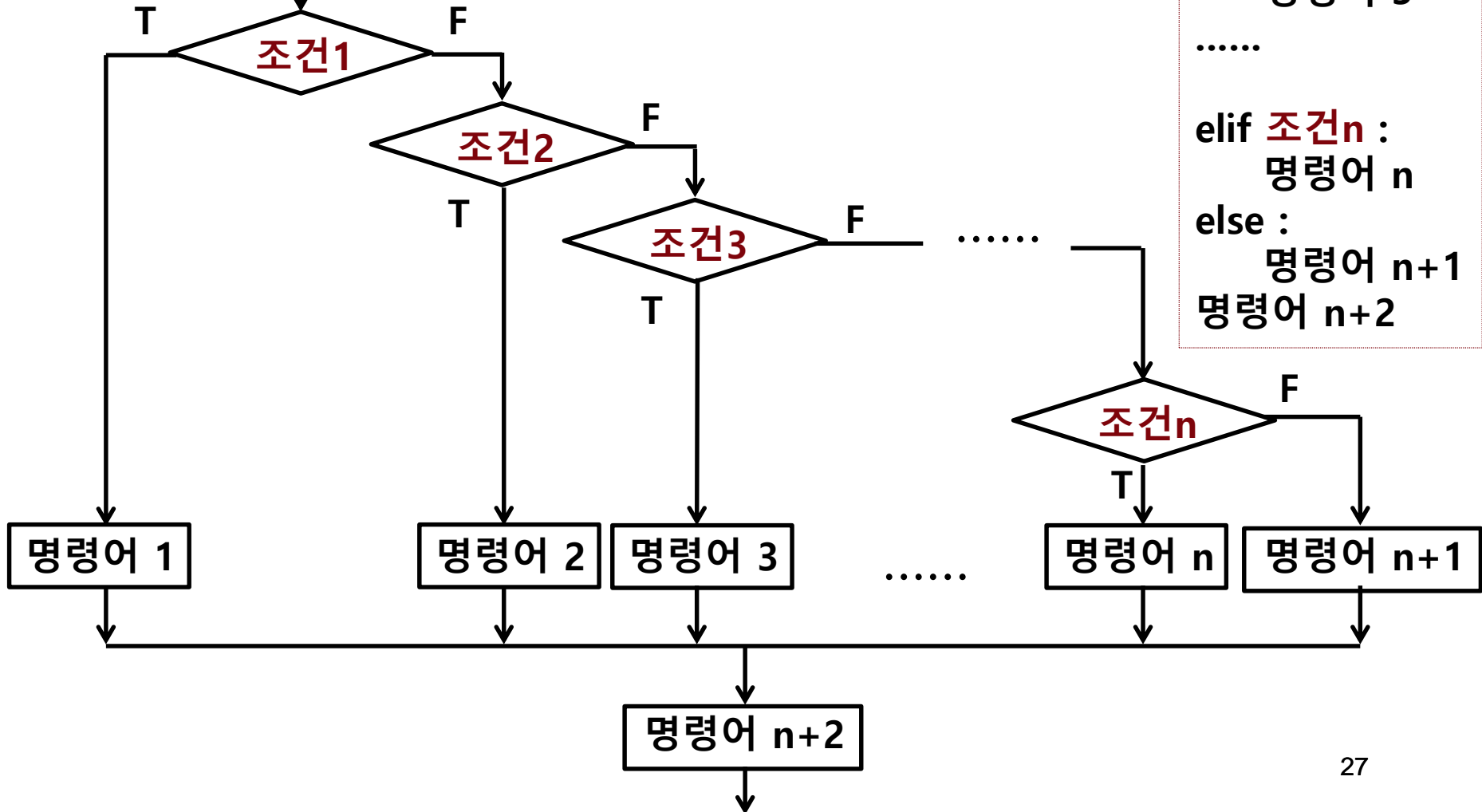


```
if a > b : max = a  
else : max = b
```

```
print(max)
```

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 3



```
if 조건1 :  
    명령어 1  
elif 조건2 :  
    명령어 2  
elif 조건3 :  
    명령어 3  
.....  
elif 조건n :  
    명령어 n  
else :  
    명령어 n+1  
    명령어 n+2
```

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 3 예제

- 성적을 입력받는다.
- 성적이 90점 이상이면 '당신의 학점은 A입니다'라고 출력한다.
- 성적이 80점 이상이고 90점 미만이면 '당신의 학점은 B입니다'라고 출력한다.
- 성적이 70점 이상이고 80점 미만이면 '당신의 학점은 C입니다'라고 출력한다.
- 성적이 70점 미만이면 '당신의 학점은 D입니다'라고 출력한다.

2. 조건 논리의 기본 개념

◆ 조건 논리 유형 3 예제

```
score = int(input('성적을 입력하시오 : '))

if score >= 90 :
    grade = 'A'
elif 80 <= score < 90 :    # elif 80 <= score :
    grade = 'B'
elif 70 <= score < 80 :    # elif 70 <= score :
    grade = 'C'
else:
    grade = 'D'

print('당신의 학점은', grade, '입니다.')
```

```
>>>
성적을 입력하시오 : 95
당신의 학점은 A 입니다.
>>> =====
>>>
성적을 입력하시오 : 88
당신의 학점은 B 입니다.
>>> =====
>>>
성적을 입력하시오 : 70
당신의 학점은 C 입니다.
>>> =====
>>>
성적을 입력하시오 : 60
당신의 학점은 D 입니다.
```

2. 조건 논리의 기본 개념



조건문의 세 가지 형식

조건문에는 다음과 같이 세 가지 형식이 있습니다.

if	if-else	if-elif-else
<pre>if <조건>: <코드 블록></pre>	<pre>if <조건>: <코드 블록> else: <코드 블록></pre>	<pre>if <조건>: <코드 블록> elif <조건>: <코드 블록> else: <코드 블록></pre>

2. 조건 논리의 기본 개념 _ Test1

X값을 입력받은 후

x가 0보다 작으면 x는 음수입니다.

x가 0이면 x는 0입니다.

x가 0보다 크면 x는 양수입니다. 를 출력하라

```
x = int(input("Please enter an integer: "))
```

```
if x < 0:
    print('{}는 음수입니다'.format(x))
elif x == 0:
    print('{}는 영입니다'.format(x))
else:
    print('{}는 양수입니다'.format(x))
```

2. 조건 논리의 기본 개념 _ Test2

가위, 바위, 보 중 두 개의 값을 받은 후 승패를 출력하도록 하라.

X, Y

Scissor/ Rock/ Paper

X = input(가위, 바위, 보를 입력해주세요.)

2. 조건 논리의 기본 개념

정수를 입력받은 후 짝수인지 홀수인지 판단하여 짝수인지 홀수인지 출력하세요.