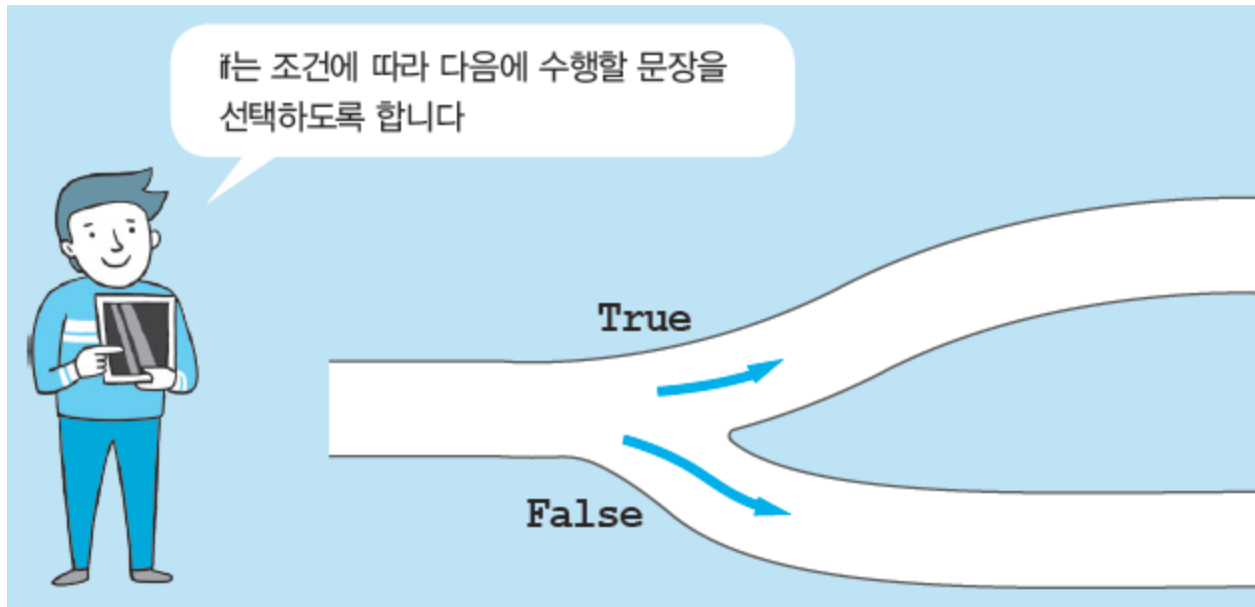


6. if 조건문

1. bool 자료형 이해하기
2. 관계 연산자
3. 논리 연산자
4. 산술연산자, 관계연산자, 논리연산자 간의 우선 순위
5. 조건 논리와 if 조건문
6. 중첩된 if 구문
7. 정리

1. bool 자료형 이해하기



1. bool 자료형 이해하기

◆ bool 자료형

- True 또는 False, 둘 중에 하나의 값을 가질 수 있는 자료형
- if 조건문에서는 어떤 문장이 True인지, False인지를 판단하는 것이 중요함.

```
>>> b1 = True; b2 = False
>>> type(b1) # b1은 bool 자료형
<class 'bool'>
>>> type(b2) # b2은 bool 자료형
<class 'bool'>
```

```
>>> b = 5 > 3 # 5>3의 결과를 b에 저장함.
>>> print(b)
True
>>> type(b)
<class 'bool'>
```

2. 관계 연산자

◆ 두 데이터 간의 대소를 비교하는 연산자

| 관계연산자 | 사용법 | 의미 |
|-------|----------|----------------|
| > | $a > b$ | a는 b보다 크다. |
| >= | $a >= b$ | a는 b보다 크거나 같다. |
| < | $a < b$ | a는 b보다 작다. |
| <= | $a <= b$ | a는 b보다 작거나 같다. |
| == | $a == b$ | a와 b는 같다. |
| != | $a != b$ | a와 b는 같지 않다. |

>=, <=, ==, != 는 반드시 두 기호를 붙여씁니다.

↑ ↑
 같다 같지 않다

같다'를 표현할 때 '='는 한 개만 쓰지 않도록 주의하세요.

2. 관계 연산자

```

a = int(input('Enter a : '))    # 정수 a 입력받기
b = int(input('Enter b : '))    # 정수 b 입력받기
x1 = a > b                       # a > b의 결과를 x1에 저장하기
x2 = a >= b                     # a >= b의 결과를 x1에 저장하기
x3 = a < b                      # a < b의 결과를 x1에 저장하기
x4 = a <= b                    # a <= b의 결과를 x1에 저장하기
x5 = a == b                    # a == b의 결과를 x1에 저장하기
x6 = a != b                    # a != b의 결과를 x1에 저장하기

```

```

print('{} > {} -> {}'.format(a, b, x1))
print('{} >= {} -> {}'.format(a, b, x2))
print('{} < {} -> {}'.format(a, b, x3))
print('{} <= {} -> {}'.format(a, b, x4))
print('{} == {} -> {}'.format(a, b, x5))
print('{} != {} -> {}'.format(a, b, x6))

```

[결과 1]

```

Enter a : 5
Enter b : 7
5 > 7 -> False
5 >= 7 -> False
5 < 7 -> True
5 <= 7 -> True
5 == 7 -> False
5 != 7 -> True

```

[결과 2]

```

Enter a : 7
Enter b : 7
7 > 7 -> False
7 >= 7 -> True
7 < 7 -> False
7 <= 7 -> True
7 == 7 -> True
7 != 7 -> False

```

3. 논리 연산자

◆ and / or / not

| 논리연산자 | 사용법 및 의미 |
|---------|---|
| A and B | A와 B가 모두 True인 경우에만 A and B는 True입니다. 나머지 경우는 모두 False입니다. |
| A or B | A와 B 둘 중에 하나라도 True이면 A or B는 True가 됩니다. 즉, A와 B가 모두 False인 경우에만 A or B는 False이고 나머지 경우는 모두 True입니다. |
| not A | A가 True이면 not A는 False입니다. A가 False이면 not A는 True입니다. 즉, not 뒤의 문장이 True인지 False인지를 판단한 후에 그 결과를 반대로 만들어 줍니다. (not 연산자가 나오면 뒷부분이 False인지를 판단하세요) |

| A and B | 결과 | A or B | 결과 | not A | 결과 |
|-----------------|-------|----------------|-------|-----------|-------|
| True and True | True | True or True | True | not True | False |
| True and False | False | True or False | True | not False | True |
| False and True | False | False or True | True | | |
| False and False | False | False or False | False | | |

3. 논리 연산자

◆ and / or / not

```
>>> a = 10; b = 5; c = 7
```

```
>>> x = a > b and b > c
```

```
>>> print(x)
```

```
False
```

```
>>> y = a > b or b > c
```

```
>>> print(y)
```

```
True
```

```
>>> a = 10; b = 5; c = 7
```

```
>>> m = not a > b # not 뒤 부분이 True임.
```

```
>>> print(m)
```

```
False
```

```
>>> n = not b > c # not 뒤 부분이 False임.
```

```
>>> print(n)
```

```
True
```

`a > b and b > c` 를 `a > b > c` 라고 써도 됩니다.

4. 산술연산자, 관계연산자, 논리연산자 간의 우선 순위

- ◆ 세 종류의 연산자가 섞여 있다면, 산술 > 관계 > 논리 연산자의 순서로 계산한다.

```
>>> a = 3; b = 5; c = 2
```

```
>>> print( a + b > b + c )
```

```
True
```

```
>>> print( a + b * c > a * b - c )
```

```
False
```

```
>>> print( a + b > c and a * b == 15 )
```

```
True
```

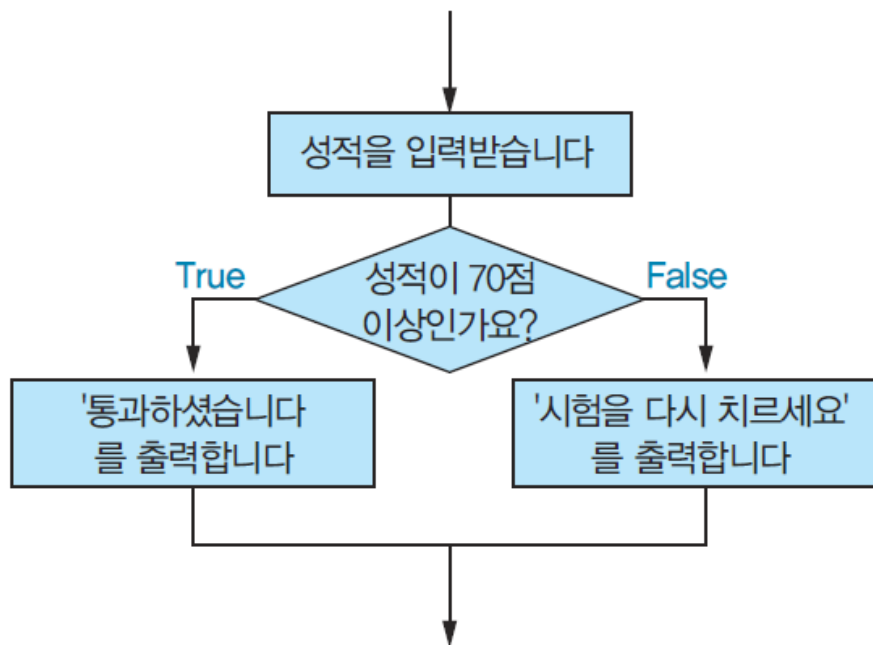
```
>>> print( a - b < 0 and not b == c )
```

```
True
```


5. 조건 논리와 if 조건문

◆ if 조건문의 세 가지 형태

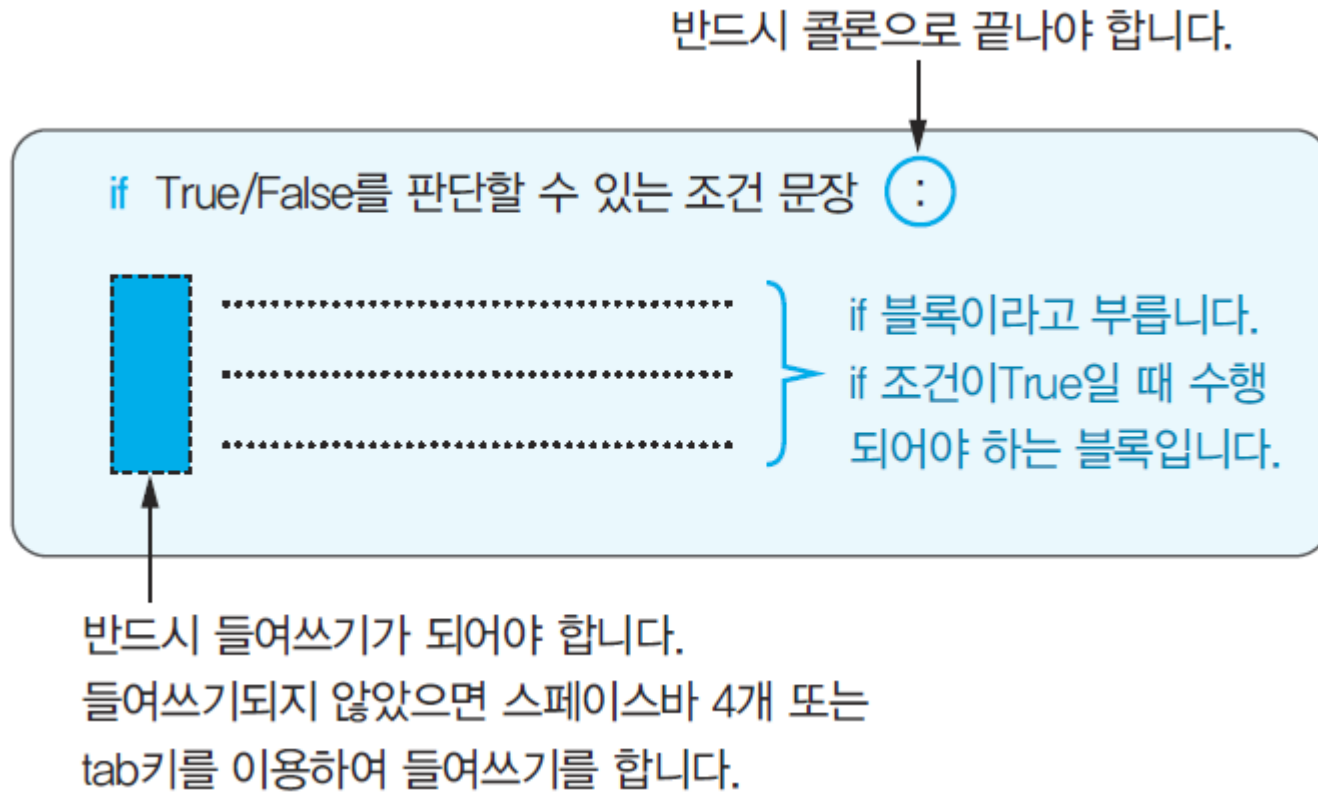
- If 만 있는 형태
- if ~ else 형태
- if ~ elif ~ else 형태



```
score = int(input('성적을 입력하세요 : '))  
if score >= 70 : # score가 70 이상이라면,  
    print('통과하셨습니다')  
else : # score가 70 이상이 아니라면,  
    print('시험을 다시 치르세요')
```

5. 조건 논리와 if 조건문

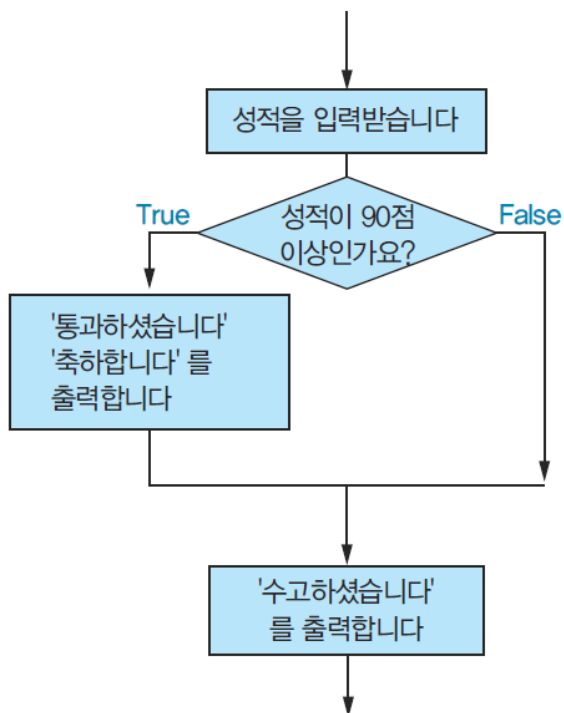
◆ if 만 있는 형태



5. 조건 논리와 if 조건문

◆ if 만 있는 형태

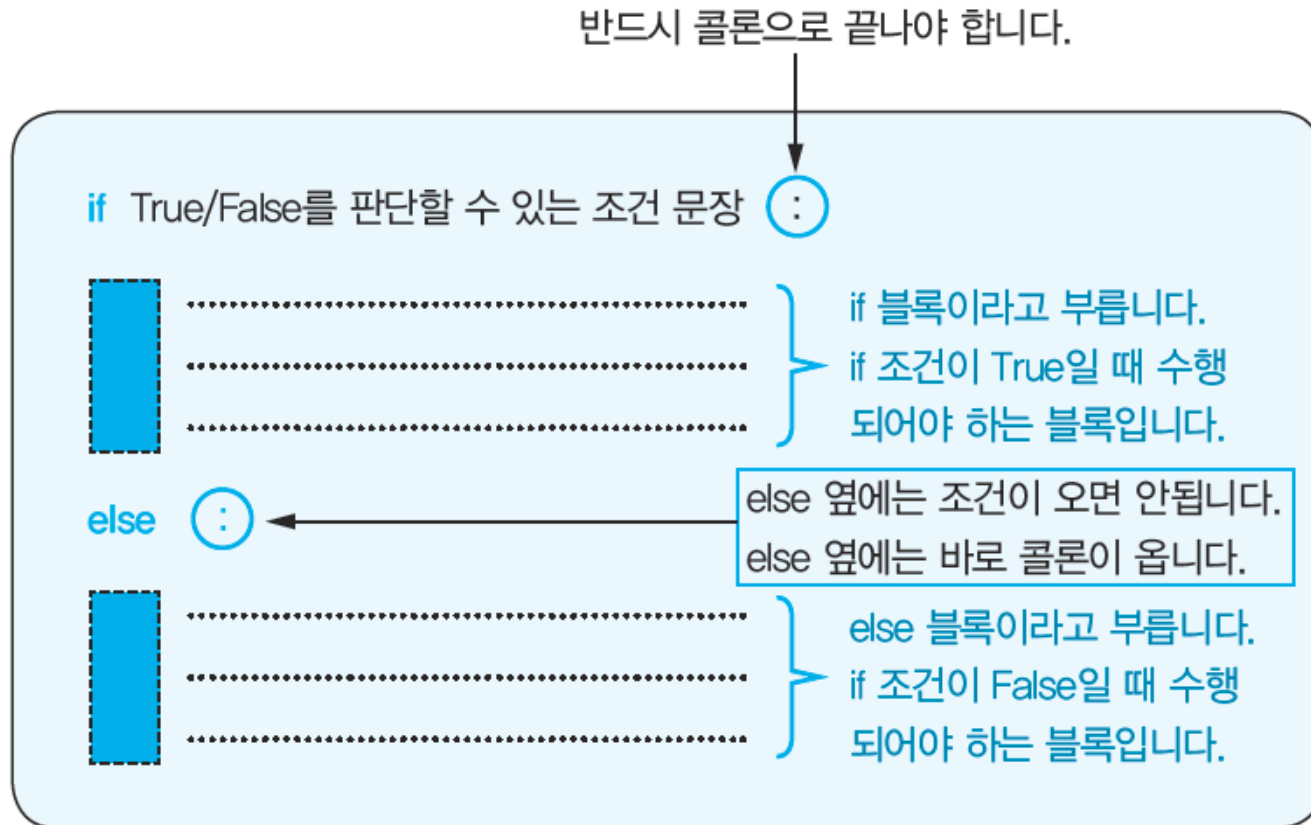
CODE 04 성적을 입력받아서 성적이 90점 이상인 경우에는 '통과하셨습니다. 축하합니다. 수고하셨습니다'를 출력하고, 90점 미만인 경우에는 '수고하셨습니다'만 출력하는 코드를 입력합니다.



```
score = int(input('성적을 입력하세요 : '))
if score >= 90:
    print('통과하셨습니다')
    print('축하합니다')
print('수고하셨습니다')
```

5. 조건 논리와 if 조건문

◆ if ~ else 형태



반드시 들여쓰기 되어야 합니다.

들여쓰기가 되지 않았으면 스페이스바 4개 또는 tab키를 이용하여 들여쓰기를 합니다.

5. 조건 논리와 if 조건문

◆ if ~ else 형태

CODE 05 [CODE 04] 을 수정해서 만약에 90점 이상이면 그대로 '통과하셨습니다. 축하합니다. 수고하셨습니다'를 출력하고, 90점 미만이면 '90점 이상이 되어야 통과입니다. 수고하셨습니다'를 출력하도록 수정해 보겠습니다.

| | |
|--|--|
| <pre>score = int(input('성적을 입력하세요 : ')) if score >= 90: # 성적이 90점 이상인 경우 print('통과하셨습니다') print('축하합니다') else: # 성적이 90점 이상이 아닌 경우 print('90점 이상이 되어야 통과입니다') print('수고하셨습니다')</pre> | <div data-bbox="1116 675 1267 722">[결과 1]</div> <div data-bbox="1116 775 1522 975"> <p>성적을 입력하세요 : 95 통과하셨습니다 축하합니다 수고하셨습니다</p> </div> <div data-bbox="1116 1001 1267 1048">[결과 2]</div> <div data-bbox="1116 1100 1663 1246"> <p>성적을 입력하세요 : 85 90점 이상이 되어야 통과입니다 수고하셨습니다</p> </div> |
|--|--|

5. 조건 논리와 if 조건문

◆ if ~ else 형태

CODE 06 로그인 아이디를 표준 입력받아서 'admin'이 로그인했는지, 아니면 일반 사용자가 로그인했는지를 판단하는 프로그램을 작성하려고 해요. 만약에 로그인 아이디가 'admin'이면, 'Hi, admin! Please check log files.'라고 출력합니다. 그리고 일반 사용자라면 'Welcome! 아이디'를 출력합니다.

| | |
|---|---|
| <pre>id = input('Login : ') # 로그인 아이디를 입력받음. if id == 'admin': # id가 'admin'인 경우, print('Hi, admin!') print('Please check log files first.') else: # id가 'admin'이 아닌 경우, print('Welcome!', id)</pre> | <pre>Login : admin Hi, admin! Please check log files.</pre> |
| | <pre>Login : alice Welcome! alice</pre> |
| | <pre>Login : paul Welcome! paul</pre> |

5. 조건 논리와 if 조건문

◆ if ~ else 형태

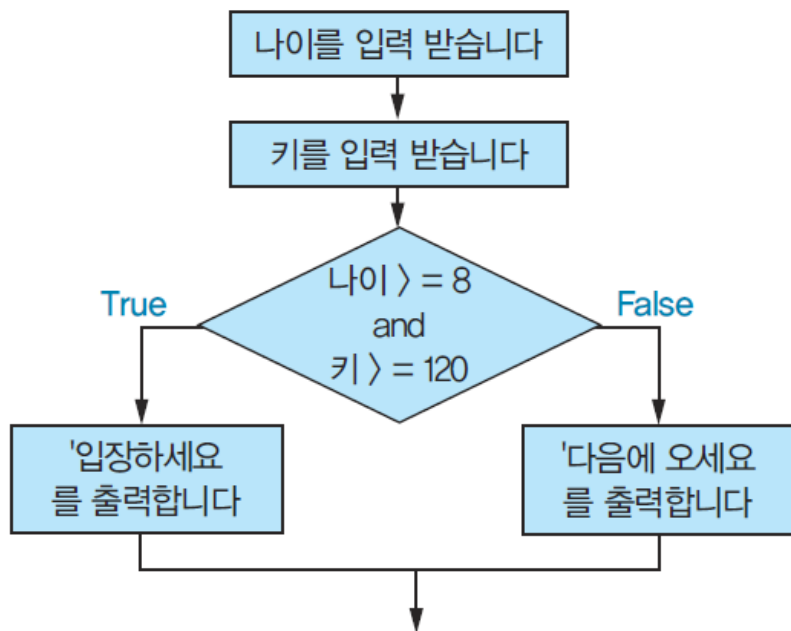
CODE 07 [CODE 06]을 조금 수정해 보겠습니다. 아이디를 표준 입력으로 받아서 admin이 로그인했는지 판단합니다. 이 때, admin을 모두 소문자, 모두 대문자, 또는 대소문자를 섞어서 입력하더라도 admin이 로그인했다고 판단해야 합니다. 만약에 로그인한 사용자가 admin이 아니면 'You are not admin'이라고 출력합니다.

| | | |
|--|--|--|
| <pre>id = input('Login : ') if id.lower() == 'admin': print('Hi, admin!') print('Please check log files.') else: print('You are not admin.')</pre> | <p>[결과 1]</p> <pre>Login : admin Hi, admin! Please check log files</pre> | <p>[결과 2]</p> <pre>Login : ADMIN Hi, admin! Please check log files</pre> |
| | <p>[결과 3]</p> <pre>Login : adMIn Hi, admin! Please check log files</pre> | <p>[결과 4]</p> <pre>Login : admim You are not admin</pre> |

5. 조건 논리와 if 조건문

◆ if ~ else 형태

CODE 08 놀이 공원에 있는 놀이 기구는 나이가 8세 이상이고, 키가 120cm 이상이어야 탈 수 있어요. 나이와 키를 입력받아서 놀이 기구를 탈 수 있으면 '입장하세요'라고 출력하고, 놀이 기구를 탈 수 없으면 '다음에 오세요'라고 출력하려고 해요. 조건을 어떻게 적어야 할까요? 이번에는 조건이 두 개 필요하겠죠? 이런 경우에 논리연산자를 사용합니다.

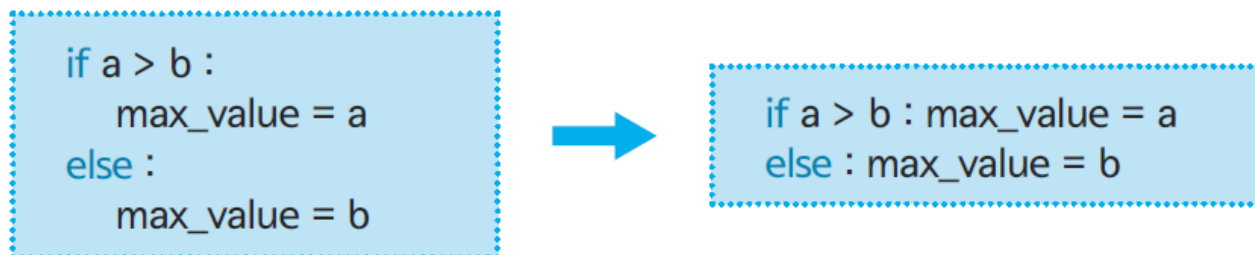


```
age = int(input('나이를 입력하세요 : '))
height = int(input('키를 입력하세요 : '))
if age >= 8 and height >= 120:
    print('입장하세요')
else:
    print('다음에 오세요')
```


5. 조건 논리와 if 조건문

◆ if ~ else 형태

- 블록 안이 간단할 때는 콜론 옆에 바로 코드를 적어도 됨



- if ~ else 형태에서는 if 블록이 끝나면 바로 else 블록을 적어야 함

```
a = 10; b = 15; c = 20
```

- ① `if a < b:`
- ② `print('a is smaller than b')`
- ③ `print('between if and else')` # 에러 발생 라인, if와 else의 연결을 끊습니다.
- ④ `else:` # if~else만 있는 경우에는 else는 반드시 if 블록이 끝나면 바로 나와야 합니다.
- ⑤ `print('a is larger than b')`

5. 조건 논리와 if 조건문

◆ if ~ elif ~ else 형태



5. 조건 논리와 if 조건문

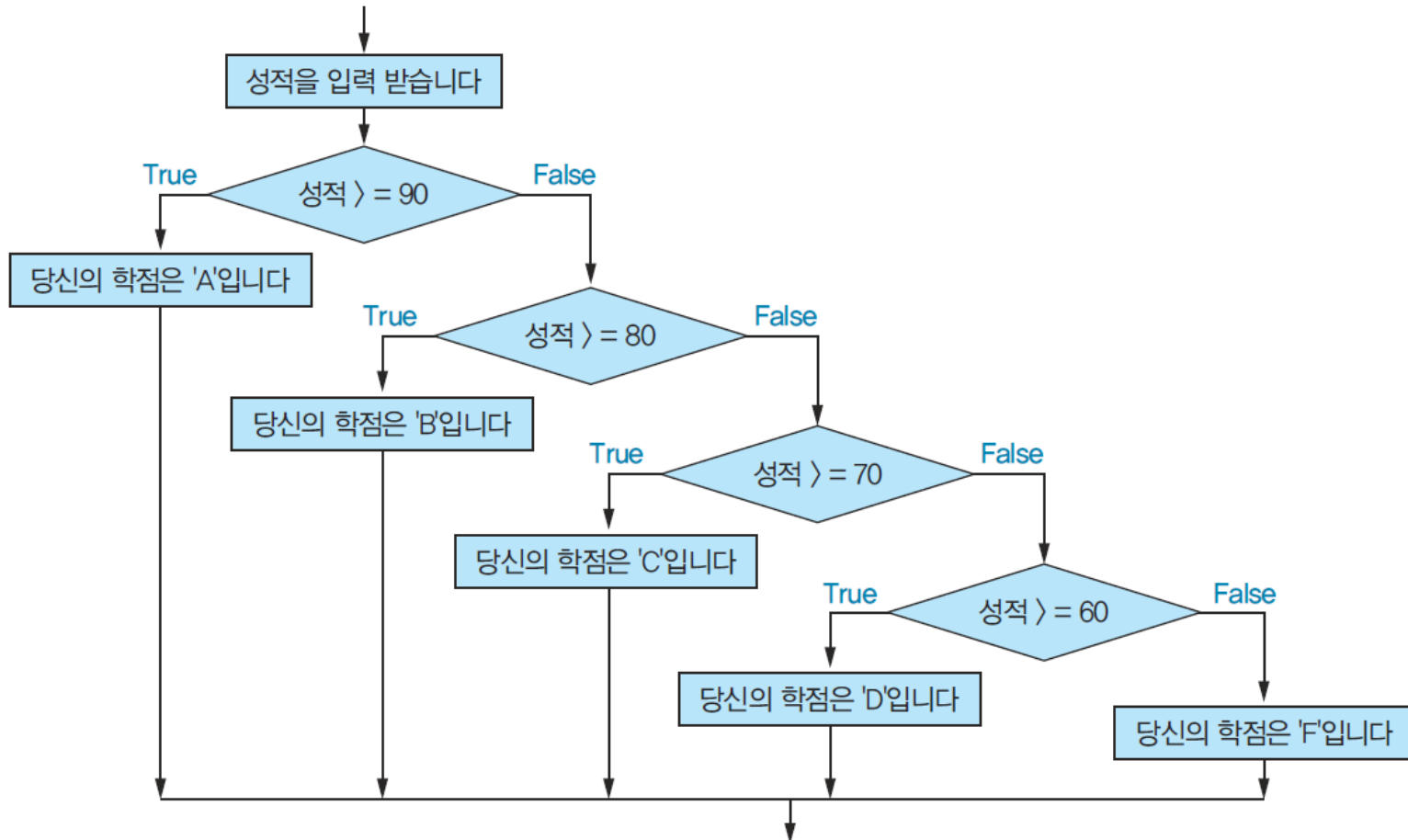
◆ if ~ elif ~ else 형태

CODE 09 성적을 입력받아서 학점을 출력하는 코드를 작성하려고 합니다. 다음의 조건에 따라 학점을 A부터 F까지 매기려고 합니다.

- 만약에 성적이 90점 이상이면 '당신의 학점은 A입니다'를 출력합니다.
- 만약에 성적이 80점 이상이고 90점 미만이면 '당신의 학점은 B입니다'를 출력합니다.
- 만약에 성적이 70점 이상이고 80점 미만이면 '당신의 학점은 C입니다'를 출력합니다.
- 만약에 성적이 60점 이상이고 70점 미만이면 '당신의 학점은 D입니다'를 출력합니다.
- 만약에 성적이 60점 미만이면 '당신의 학점은 F입니다'를 출력합니다.

5. 조건 논리와 if 조건문

◆ if ~ elif ~ else 형태



5. 조건 논리와 if 조건문

◆ if ~ elif ~ else 형태

```
score = int(input('성적을 입력하세요 : '))    # 성적을 입력받습니다.  
if score >= 90:                                # 성적이 90점 이상인 경우  
    print("당신의 학점은 'A'입니다")  
elif score >= 80:                              # score < 90 and score >= 80  
    print("당신의 학점은 'B'입니다")  
elif score >= 70:                              # score < 80 and score >= 70  
    print("당신의 학점은 'C'입니다")  
elif score >= 60:                              # score < 70 and score >= 60  
    print("당신의 학점은 'D'입니다")  
else:                                           # score < 60  
    print("당신의 학점은 'F'입니다")
```

5. 조건 논리와 if 조건문

◆ if ~ elif ~ else 형태

CODE 10 어느 놀이 공원에서는 아래 기준으로 입장료를 결정합니다. 나이를 입력받아서 아래 기준에 따라 입장료를 정하여 출력하는 코드를 작성해 보았어요

- 만약 나이가 65세 이상이면 입장료는 8,000원입니다.
- 만약 나이가 20세 이상 65세 미만인 경우에는 입장료가 10,000원입니다.
- 만약 나이가 7세 이상 20세 이하이면 입장료가 7,000원입니다.
- 만약 나이가 7세 미만이면 무료입니다.

5. 조건 논리와 if 조건문

◆ if ~ elif ~ else 형태

```
age = int(input('나이를 입력하세요 : '))
```

```
if age >= 65:           # 나이가 65세 이상인 경우
```

```
    print('입장료는 8000원입니다')
```

```
elif 20 <= age < 65:    # 나이가 20세 이상이고 65세 미만인 경우
```

```
    print('입장료는 10000원입니다')
```

```
elif 7 <= age < 20:     # 나이가 7세 이상이고 20세 미만인 경우
```

```
    print('입장료는 7000원입니다')
```

```
else:                  # 나이가 7세 미만인 경우
```

```
    print('무료 입장입니다')
```

```
print('프로그램을 끝냅니다')
```

[결과 1]

나이를 입력하세요 : 70
입장료는 8000원입니다
프로그램을 끝냅니다

[결과 2]

나이를 입력하세요 : 30
입장료는 10000원입니다
프로그램을 끝냅니다

[결과 3]

나이를 입력하세요 : 15
입장료는 7000원입니다
프로그램을 끝냅니다

[결과 4]

나이를 입력하세요 : 5
무료 입장입니다
프로그램을 끝냅니다

5. 조건 논리와 if 조건문

◆ 조건문에 in / not in을 사용하는 예제

CODE 11 다음은 영문자 하나를 입력받아서 그 문자가 자음인지 모음인지를 판단하는 코드입니다. 이런 경우에 in 연산자가 유용합니다.

```
vowels = 'aeiouAEIOU' # 모음을 모두 모아서 문자열로 만듦.
v = input('영문자를 하나 입력하세요 : ')
if v in vowels: # 입력받은 문자가 vowels에 있는 문자인 경우
    print(v, '는 모음입니다')
else:          # 입력받은 문자가 vowels에 없는 문자인 경우
    print(v, '는 모음이 아닙니다')
```

[결과 1]

영문자를 하나 입력하세요 : I
I 는 모음입니다

[결과 2]

영문자를 하나 입력하세요 : M
M 는 모음이 아닙니다

5. 조건 논리와 if 조건문

◆ 조건문에 in / not in을 사용하는 예제

CODE 12 리스트 fruits에는 내가 좋아하는 과일들이 저장되어 있습니다. 과일을 입력받아서 내가 좋아하는 과일인지 판단하는 프로그램을 작성해 볼게요. 대괄호 []를 이용해서 데이터를 묶어 놓는 자료형이에요. 문자열과 똑같이 in 연산자를 이용해서 리스트에 있는 데이터인지 아닌지를 판단해요.

```
fruits = ['melon', 'strawberry', 'orange'] # 데이터 3개 있는 리스트
f = input('Enter fruit : ') # 과일 이름 입력받기
if f in fruits:                # 입력받은 과일이 리스트 fruits에 있는 경우
    print('I like', f)
else:                          # 입력받은 과일이 리스트 fruits에 없는 경우
    print('I don\'t like', f)
```

[결과 1]
Enter fruit : orange
I like orange

[결과 2]
Enter fruit : apple
I don't like apple

[결과 3]
Enter fruit : apple
I don't like apple

5. 조건 논리와 if 조건문

◆ 조건문에 in / not in을 사용하는 예제

CODE 13 튜플에는 내가 좋아하는 색 목록이 들어 있습니다. 친구도 나와 같은 색을 좋아하는지 알고 싶습니다. 튜플에는 모두 소문자로 저장되어 있고, 친구는 대소문자 상관없이 좋아하는 색을 입력합니다.

```
colors = ('blue', 'green', 'purple')      # 데이터가 3개 있는 튜플
color = input('What is your favorite color? ') # 색을 입력받기
if color.lower() in colors:                # 입력받은 색이 colors에 있는 경우
    print('Oh!', color.title(), 'is also my favorite color.')
else:                                     # 입력받은 색이 colors에 없는 경우
    print(color.title(), 'is not my favorite color.')
```

[결과 1]
What is your favorite color? BLUE
Oh! Blue is also my favorite color.

[결과 2]
What is your favorite color? REd
Red is not my favorite color.

5. 조건 논리와 if 조건문

◆ 조건문에 in / not in을 사용하는 예제

CODE 14 집합 subjects에는 A 영어 학원에서 가르치는 과목들이 저장되어 있습니다. 내 동생은 A학원에 자기가 듣고 싶은 과목이 있는지 알고 싶습니다. A학원에서 가르치는 과목들은 'toefl', 'toeic', 'gre', 'lsat'입니다.

```
subjects = {'toefl', 'toeic', 'gre', 'lsat'} # 데이터 4개인 집합
sub = input('Enter subject you want to register : ')
if sub in subjects: # 입력받은 과목이 집합 subjects에 있는 경우
    print('You can register', sub, 'class.')
else: # 입력받은 과목이 집합 subjects에 없는 경우
    print('There is no', sub, 'class.')
```

[결과 1]
Enter subject you want
to register : toeic
You can register toeic
class.

[결과 2]
Enter subject you want
to register : sat
There is no sat class.

5. 조건 논리와 if 조건문

- ◆ 빈 문자열, 빈 리스트, 빈 튜플, 빈 집합, 빈 사전은 False입니다
 - 빈 문자열이 True 또는 False를 판단하는 자리에 있으면 False로 판단함

| | | |
|--|---|--|
| <pre> c = '' # c는 빈 문자열 if c: # 빈 문자열은 False print('if block') else: print('else block') </pre> | <pre> c = 'A' # len(c) = 1 if c: print('if block') else: print('else block') </pre> | <pre> c = ' ' # 스페이스 있음 if c: print('if block') else: print('else block') </pre> |
| else block | if block | if block |

5. 조건 논리와 if 조건문

◆ 빈 문자열, 빈 리스트, 빈 튜플, 빈 집합, 빈 사전은 False입니다

- 빈 리스트가 True 또는 False를 판단하는 자리에 있으면 False로 판단함

| | |
|---|-----------------------|
| <pre>L = [] # 빈 리스트, len(L) = 0 if L: # if False: 로 해석합니다. print('if block') else: print('else block')</pre> | <pre>else block</pre> |
|---|-----------------------|

- 빈 튜플이 True 또는 False를 판단하는 자리에 있으면 False로 판단함

| | |
|--|-----------------------|
| <pre>T = () # 빈 튜플, len(T) = 0 if T: # if False: 로 해석합니다. print('if block') else: print('else block')</pre> | <pre>else block</pre> |
|--|-----------------------|

5. 조건 논리와 if 조건문

◆ 빈 문자열, 빈 리스트, 빈 튜플, 빈 집합, 빈 사전은 False입니다

- 빈 집합이 True 또는 False를 판단하는 자리에 있으면 False로 판단함

| | |
|--|------------|
| <pre> S = set() # 빈 집합은 set()으로 만듭니다, len(S)=0 if S: # if False: 로 해석합니다. print('if block') else: print('else block') </pre> | else block |
|--|------------|

- 빈 사전이 True 또는 False를 판단하는 자리에 있으면 False로 판단함

| | |
|---|------------|
| <pre> D = {} # {}는 빈 집합입니다, len(D)=0 if D: print('if block') else: print('else block') </pre> | else block |
|---|------------|

5. 조건 논리와 if 조건문

◆ 빈 문자열, 빈 리스트, 빈 튜플, 빈 집합, 빈 사전은 False입니다

컨테이너 자료형은 비어 있을 수 있어요.



빈 문자열 → "
빈 리스트 → []
빈 튜플 → ()
빈 집합 → set()
빈 사전 → { }

이렇게 빈 객체가 True 또는 False를 판단해야 하는 자리에 있으면 False라고 판단합니다. 그리고 비어 있지 않은 경우에는 True라고 판단합니다.

6. 중첩된 if 구문

◆ If 구문 안에 또 다시 if 구문이 오는 경우

CODE 15 놀이 동산에서 놀이 기구 타는 문제를 다시 보겠습니다. 나이가 8세 이상, 키가 120cm 이상인 경우만 놀이 기구를 탈 수 있습니다. 이번에는 다음과 같은 형태로 if 구문을 작성하려고 합니다.

- 나이를 입력받습니다.
- 나이가 8세 미만이면, '8세 이상이어야 입장 가능합니다'라고 출력합니다.
- 나이가 8세 이상이면, 키를 입력받습니다.
 - 키가 120cm 이상이면 '입장 가능합니다'를 출력합니다.
 - 키가 120cm 미만이면 '키가 120cm 이상이어야 입장가능합니다'를 출력합니다.

6. 중첩된 if 구문

◆ If 구문 안에 또 다시 if 구문이 오는 경우

| | |
|---|---|
| <pre>age = int(input('나이를 입력하세요 : ')) if age >= 8: # 입력받은 나이가 8세 이상인 경우, 키를 입력받음 height = int(input('키를 입력하세요 : '))</pre> | <p>[결과 1] 나이를 입력하세요 : 10 키를 입력하세요 : 115 키가 120cm 이상이어야 입장 가능합니다</p> |
| <pre> if height >= 120: # 나이가 8세 이상, 키가 120 이상인 경우 print('입장 가능합니다') else: # 나이가 8세 이상, 키가 120 미만인 경우</pre> | <p>[결과 2] 나이를 입력하세요 : 9 키를 입력하세요 : 121 입장 가능합니다</p> |
| <pre> print('키가 120cm 이상이어야 입장 가능합니다') else: # 입력받은 나이가 8세 미만인 경우 print('8세 이상이어야 입장 가능합니다')</pre> | <p>[결과 3] 나이를 입력하세요 : 7 8세 이상이어야 입장 가능합니다</p> |

6. 중첩된 if 구문

◆ If 구문 안에 또 다시 if 구문이 오는 경우

CODE 16 현재 admin의 패스워드는 1234로 설정되어 있어요. 아이디를 입력받아서 대소문자 관계없이 admin이 맞다면, 패스워드를 입력하라는 메시지가 나오도록 할 거예요. 이때, 패스워드를 1234로 올바르게 입력하면 ‘Welcome, admin’이라는 메시지가 출력되도록 하고, 만약에 패스워드가 틀리면 ‘Wrong password’라는 메시지가 출력되도록 합니다. 만약에 처음부터 아이디가 admin이 아니라면, ‘You are not admin’이라고 출력되도록 합니다.

6. 중첩된 if 구문

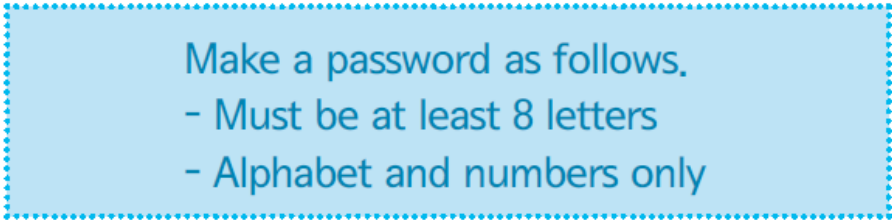
◆ If 구문 안에 또 다시 if 구문이 오는 경우

| | |
|---|--|
| <pre>id = input('Login : ') # 아이디 입력받기 if id.lower() == 'admin': # id를 소문자로 바꾸었을 때, 'admin'인 경우 pwd = input('Password : ') # 패스워드 입력받기 if pwd == '1234': # 입력한 패스워드가 '1234'인 경우 print('Welcome, admin') else: # id는 'admin'이지만, 패스워드가 '1234'가 아닌 경우 print('Wrong password') else: # id가 소문자로 바꾸었을 때, 'admin'이 아닌 경우 print('You are not admin')</pre> | <p>[결과 1] Login : admin Password : 1234 Welcome, admin</p> |
| | <p>[결과 2] Login : admin Password : 1111 Wrong password</p> |
| | <p>[결과 3] Login : admmin You are not admin</p> |

6. 중첩된 if 구문

◆ If 구문 안에 또 다시 if 구문이 오는 경우

CODE 17 이번 문제는 새로운 회원 가입 코드예요. 우선 `input()` 함수를 이용해서 영문 이름을 first name, last name 순으로 받아서 first name의 첫 문자와 last name을 합하여 아이디를 만들어 줍니다. 이 때, 아이디는 모두 소문자로 만듭니다. 아이디를 만든 후에 출력하고, 다음과 같은 문장이 화면에 나오도록 합니다.



Make a password as follows.
- Must be at least 8 letters
- Alphabet and numbers only

그리고 패스워드를 입력받습니다. 패스워드는 길이가 8자 이상이어야 하고, 영문자와 숫자로만 구성되어야 해요(영문자로만 또는 숫자로만 구성되어도 되고, 영문자와 숫자가 섞여 있어도 됩니다). 만약에 이 조건에 맞는 패스워드를 입력하면 ‘Nice Password’라고 출력하고, 길이가 8자가 안 되는 패스워드를 입력하면, ‘Must be at least 8 letters’라는 에러 메시지가 나오도록 합니다. 그리고 만약에 특수 문자가 포함되어 있으면 ‘Alphabet and numbers only’라고 출력되도록 합니다.

6. 중첩된 if 구문

```
first_name = input('Enter your first name : ')    # first name 입력받기
last_name = input('Enter your last name : ')     # last name 입력받기
login_id = first_name[0].lower() + last_name.lower()    # id 만들기
print('Welcome! your id is "', login_id, '"', sep='')
print()
print('=' * 32)
print('| Make a password as follows. |')           # 따옴표 세 개짜리로 출력할 수도 있음.
print('| - Must be at least 8 letters |')
print('| - Alphabet and numbers only |')
print('=' * 32)
print()
password = input('Input new password : ')        # 패스워드를 입력받음.

if len(password) >= 8:                            # 패스워드 길이가 8 이상인 경우
    if password.isalnum():                        # isalnum()은 문자와 숫자로만 구성되었는지 판단함.
        print('Nice password')
    else:
        print('Alphabet and numbers only')
else:                                             # 패스워드 길이가 8 미만인 경우
    print('Must be at least 8 letters')
```

6. 중첩된 if 구문

[결과 1]

Enter your first name : Alice

Enter your last name : Lee

Welcome! your id is "alee"

=====

I Make a password as follows I

I - Must be at least 8 letters I

I - Alphabet and numbers only I

=====

Input new password : alice12345

Nice password

[결과 2]

Enter your first name : Tom

Enter your last name : Kim

Welcome! your id is "tkim"

=====

I Make a password as follows I

I - Must be at least 8 letters I

I - Alphabet and numbers only I

=====

Input new password : 123hh123

Nice password

[결과 3]

Enter your first name : Paul

Enter your last name : Park

Welcome! your id is "ppark"

=====

I Make a password as follows I

I - Must be at least 8 letters I

I - Alphabet and numbers only I

=====

Input new password : paul*1233

Alphabet and numbers only

[결과 4]

Enter your first name : Jenny

Enter your last name : Chang

Welcome! your id is "jchang"

=====

I Make a password as follows I

I - Must be at least 8 letters I

I - Alphabet and numbers only I

=====

Input new password : jj1234

Must be at least 8 letters

7. 정리

- ◆ if 조건문에 대해서 학습하였음.
- ◆ if만 있는 형태의 조건문.
- ◆ if ~ else 형태의 조건문.
- ◆ if ~ elif ~ elif ~ else 형태의 조건문.
- ◆ 중첩된 조건문도 많이 사용함.