

[Python]



Python으로 배우는 소프트웨어 원리

Chapter 05. 조건

목차

1. 제어 구조
2. 조건문

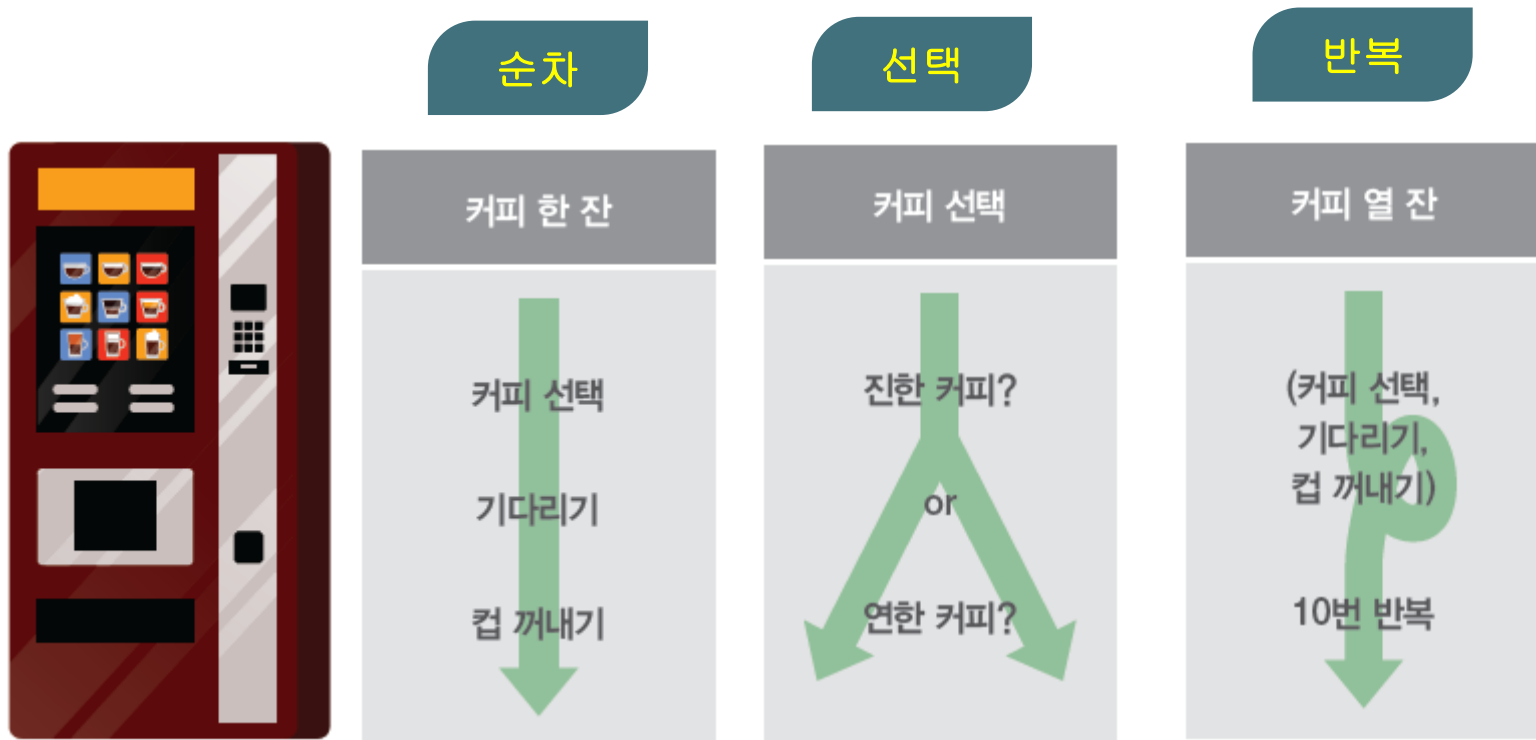
01

제어 구조

01. 제어 구조

[커피머신 사용법]

- 커피 종류를 선택하고 추출되면 컵을 꺼낸다.



01. 제어 구조

I. 제어 구조의 종류

- 프로그램에서 해야 할 동작의 흐름이나 순서를 처리하는 방법이 제어 구조
- 제어 구조는 순차 구조, 선택 구조, 반복 구조로 나뉨

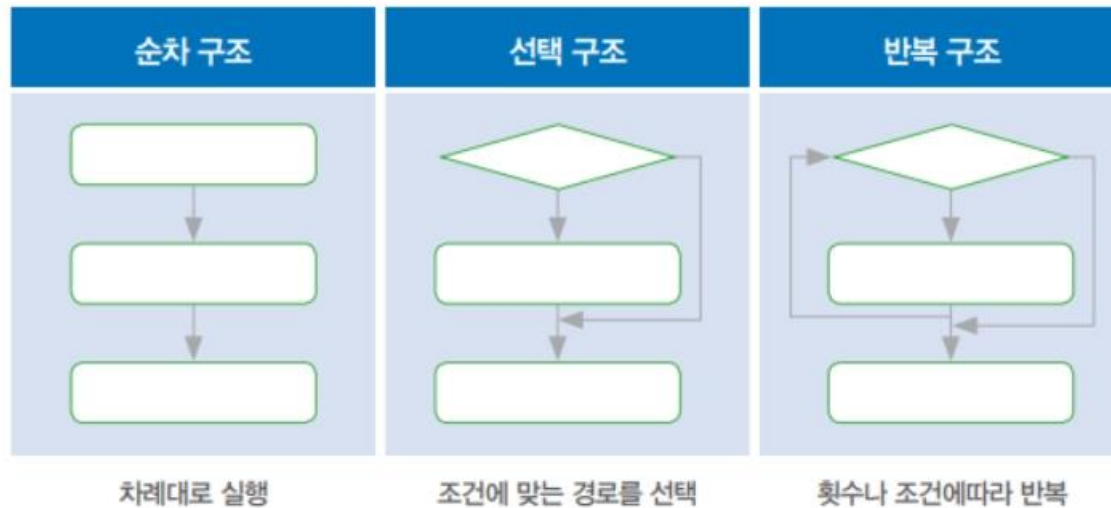


그림 5-2 프로그램의 3가지 제어 구조

01. 제어 구조

I. 제어 구조의 종류

■ 순차 구조

- 프로그램의 각 명령문이 순차적으로 실행되는 방식

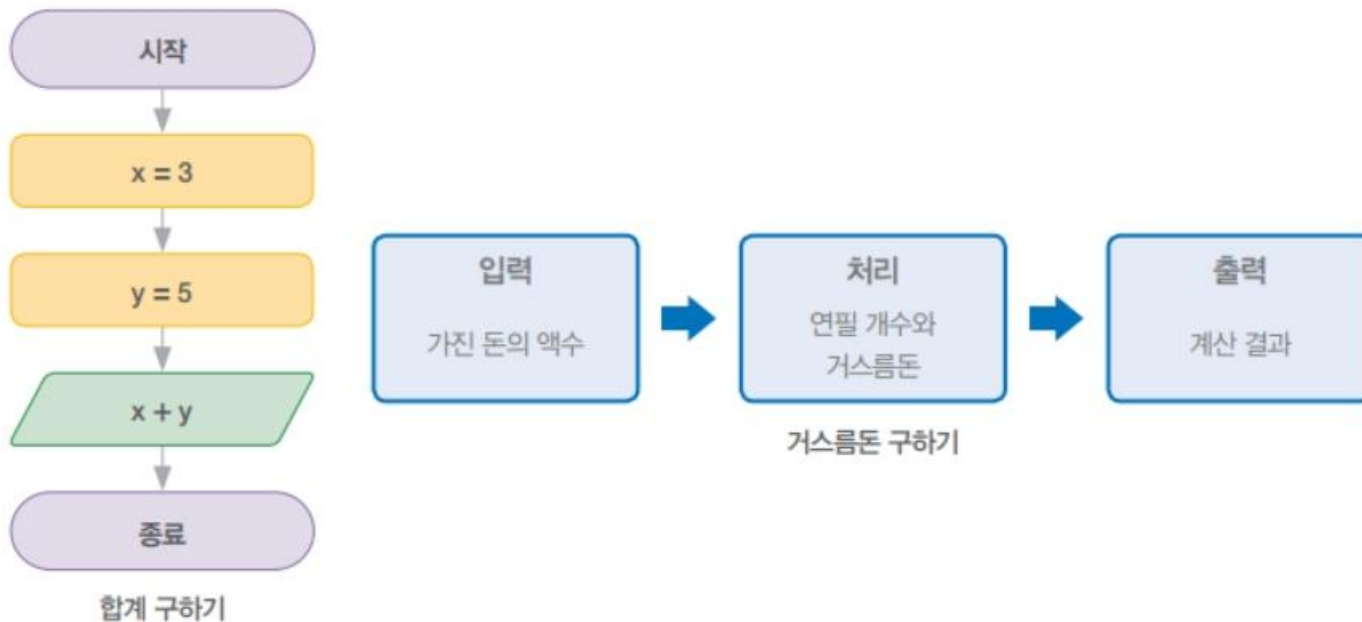


그림 5-3 순차 구조의 예

01. 제어 구조

I. 제어 구조의 종류

■ 순차 구조

실습 5-1

영화관 무인단말기로 팝콘 주문하기

code05-01.py

- 영화관 매점에 있는 무인단말기에서 주문할 수 있는 팝콘이 5천 원이라고 가정
- 수량을 입력하면 금액을 계산해서 출력하는 알고리즘을 순서도로 표현 & 코드 실습

① 알고리즘 생성



그림 5-4 팝콘 주문하기 알고리즘

01. 제어 구조

I. 제어 구조의 종류

- 순차 구조

실습 5-1

영화관 무인단말기로 팝콘 주문하기

code05-01.py

② 파이썬 코드로 작성

```
01 price = 5000
02 qty = int(input("팝콘 주문 수량 : "))
03 totalPrice = price * qty
04 print("총 주문금액 =", format(totalPrice, ','), "원")
```

③ 작성한 코드를 저장하고 실행

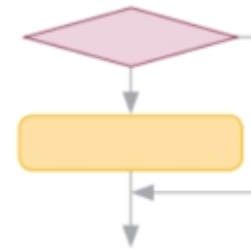
```
팝콘 주문 수량 : 3
총 주문금액 = 15,000 원
```


01. 제어 구조

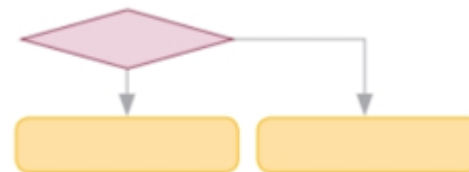
I. 제어 구조의 종류

■ 선택 구조

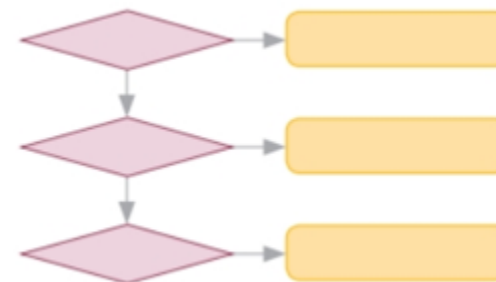
- 조건에 따라 실행할 동작을 선택
- 조건식을 포함하여 선택 구조를 기술하는 명령문을 조건문이라고 함
- 세가지 형태로 사용
 - if ~ : []
 - if ~ : []
else : []
 - if ~ : []
elif ~ : []
...
else : []



if 조건식 :
실행할 문장



if 조건식 :
실행할 문장 1
else :
실행할 문장 2



if 조건식 :
실행할 문장 1
elif :
실행할 문장 2
elif :
실행할 문장 3
else :
실행할 문장 4

01. 제어 구조

I. 제어 구조의 종류

■ 반복 구조

- 조건에 따라 특정 동작을 계속 반복할지 말지를 결정하거나, 혹은 정해진 횟수만큼 특정 동작을 반복하도록 작성

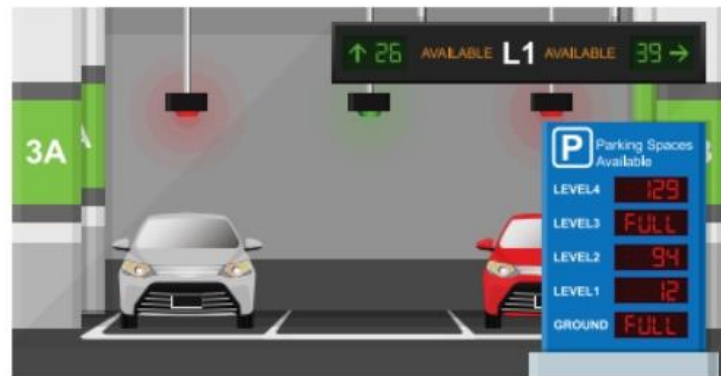
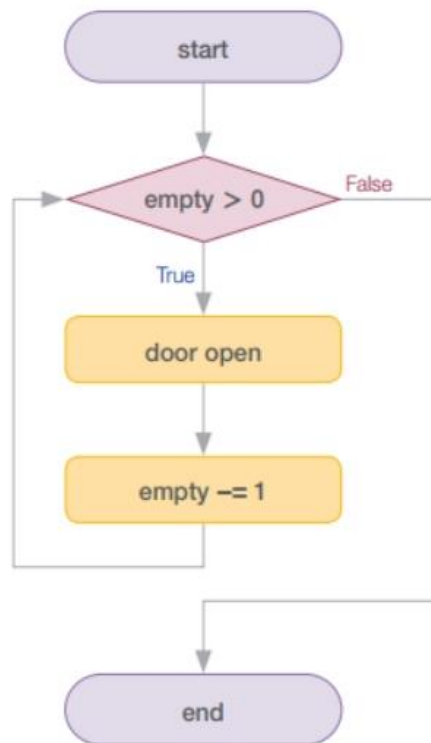


그림 5-7 주차장 출입문의 반복 동작을 보여주는 반복 구조의 예

02

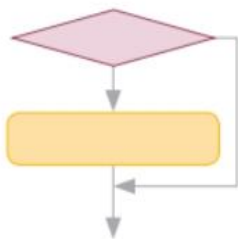
조건문

01. 제어 구조

I. 조건문의 종류

▪ if 문

- 가장 단순한 형태의 조건문으로, 조건식이 True인 경우 실행할 명령을 기술
 - if 뒤에 조건식을 기술한 다음, 반드시 콜론(:)을 입력
 - 조건식이 참인 경우 실행할 동작은 들여쓰기 후 입력
- ✓ **TIP** if 문의 콜론(:) 뒤에서 Enter 를 입력하면 자동으로 들여쓰기가 되며, Tab 이나 Space bar 로 직접 들여쓰기를 설정



if 조건식 :
실행할 문장

```
msg = input("인사말 입력: ")  
if len(msg) > 0:  
    print("> 반가워요!")
```

Ch05-hi.py

```
IDLE Shell 3.10.1  
File Edit Shell Debug Options Window Help  
= RESTART: D:/Documents/강의록/Python/source/05장/Ch05-hi.py  
인사말 입력: hi  
> 반가워요!  
>>>  
= RESTART: D:/Documents/강의록/Python/source/05장/Ch05-hi.py  
인사말 입력:  
>>> |  
Ln: 114 Col: 0
```

01. 제어 구조

I. 조건문의 종류

여기서 잠깐

파이썬의 들여쓰기

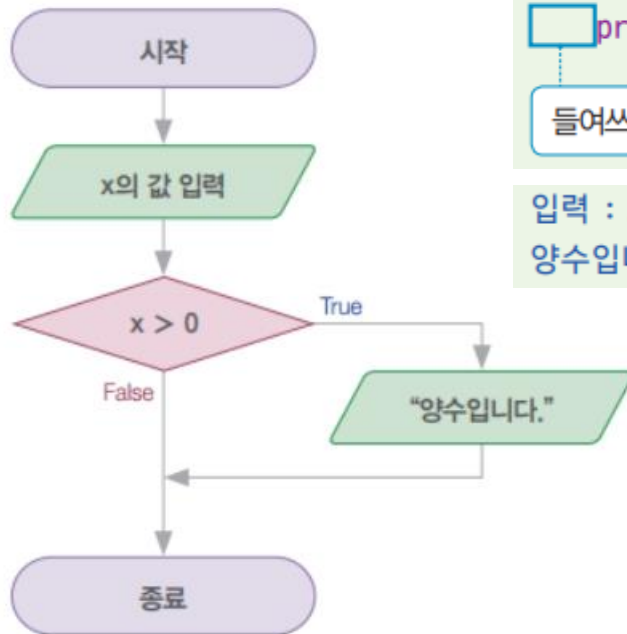
- 파이썬 언어에서는 들여쓰기가 중요
- 들여쓰기가 맞지 않으면 원하는 실행 결과가 나오지 않거나 오류로 처리
- 파이썬 셸이나 편집기에서는 if 문의 콜론(:) 뒤에서 Enter를 입력하면 자동으로 들여쓰기(Tab 이나 Space bar로는 직접 들여쓰기)
- 편집기에서는 Ctrl +] 와 Ctrl + [단축키로 들여쓰기와 내어쓰기를 실행할 수 있어서 여러 개 문장을 한 번에 처리하기 편리

01. 제어 구조

I. 조건문의 종류

▪ if 문

- 참인 조건에만 실행할 문장이 있는 경우



```
x = int(input("입력 : "))
if x > 0:
    print("양수입니다.")
```

콜론

들여쓰기

텍스트 입력 후 [Enter] 2번 누르면 실행

입력 : 3 양수입니다.	입력 : -5
------------------	---------

그림 5-8 참인 조건에만 문자열을 출력

01. 제어 구조

I. 조건문의 종류

▪ if 문

- 참인 조건에 실행할 문장이 여러 개인 경우

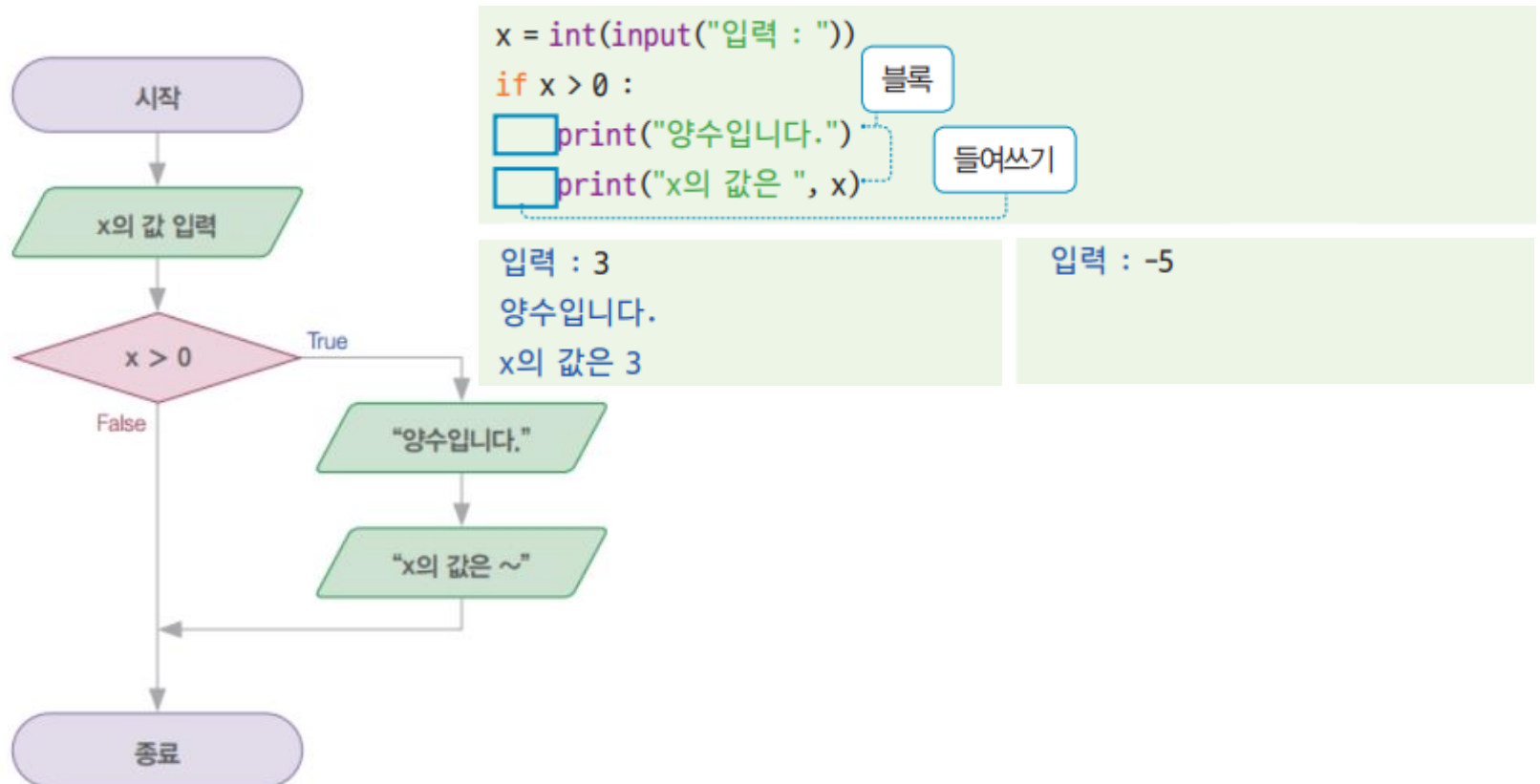


그림 5-9 참인 조건에만 여러 개의 출력문을 실행하는 if 문 사용 예

01. 제어 구조

I. 조건문의 종류

▪ if 문

- 조건과 상관없이 무조건 실행할 문장이 있는 경우

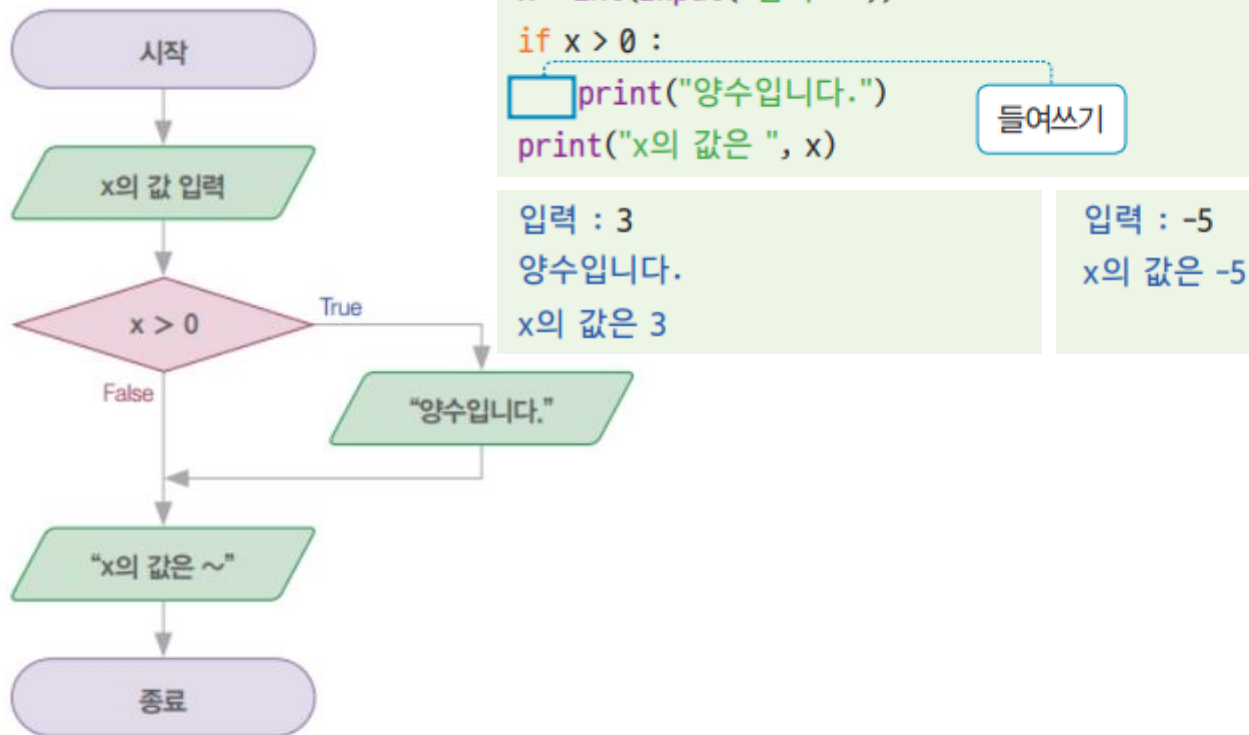


그림 5-10 조건 상관없이 x의 값을 출력

01. 제어 구조

I. 조건문의 종류

▪ if 문

- 비교, 논리 연산자를 활용하여 다양한 조건식을 표현하는 경우

```
x = 10
if x == 20 :
    print("x는 20입니다.")
```

비교 연산자(equal)

```
x = "user1"
if x != "user1" :
    print("로그인 실패")
```

비교 연산자(not equal)

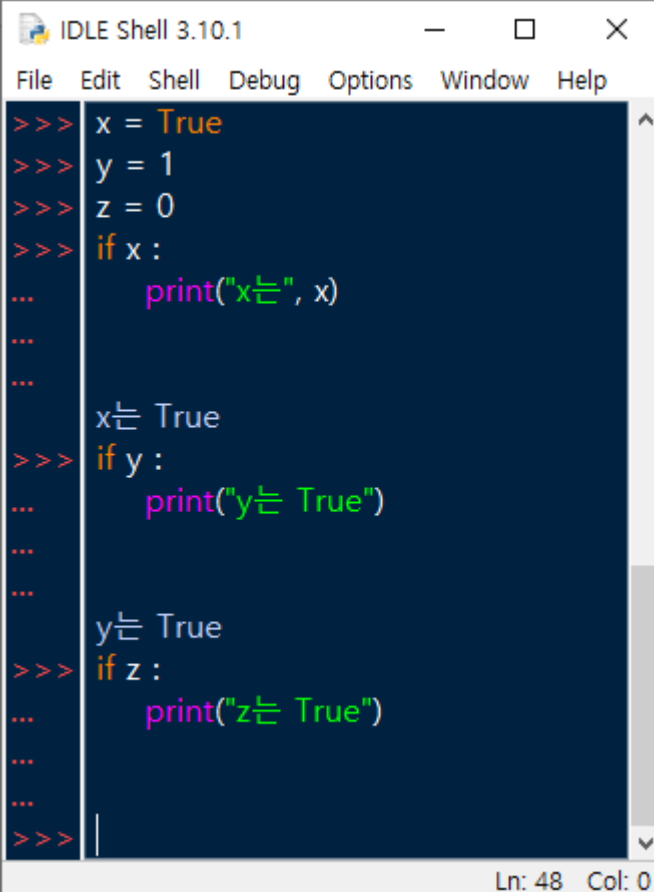
```
x = True
if x :
    print("x는 True입니다.")
```

x == True : 와 같은 의미

```
x = True ; y = False
if x or y :
    print("둘 중 하나는 True")
```

논리 연산자

❖ 조건식 결과 값이 0이면 False 아니면 True



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>>> x = True
>>> y = 1
>>> z = 0
>>> if x :
...     print("x는", x)
...
x는 True
>>> if y :
...     print("y는 True")
...
y는 True
>>> if z :
...     print("z는 True")
...
>>>
```

Ln: 48 Col: 0

01. 제어 구조

I. 조건문의 종류

▪ if 문

실습 5-2

미성년자 여부 확인하기

code05-02.py

- ① 이름과 나이를 입력하고 입장 가능한 나이인지 판단한 다음, 안내 메시지를 출력

```
01 name = input("이름을 입력하세요 : ")
02 age = int(input("나이를 입력하세요 : "))
03 if age >= 19 :
04     print(name, "님은 성년입니다.")
05     print("입장하세요~")
```

- ② 프로그램을 저장하고, 다양한 입력값을 사용해 실행

```
이름을 입력하세요 : 홍길동
나이를 입력하세요 : 25
홍길동 님은 성년입니다.
입장하세요~
```

```
이름을 입력하세요 : 김유신
나이를 입력하세요 : 17
```

01. 제어 구조

I. 조건문의 종류

▪ if 문

- if 문을 사용하여 F 학점인지 판정하는 프로그램
- 점수가 60점 미만이거나 결석이 4회 이상이면 F 학점으로 판정

실습 5-3

F 학점 여부 판정하기

code05-03.py

- ① 점수와 결석 횟수를 입력받아 변수에 저장하고, if 조건식으로 판단한 결과를 출력

```
01 score = float(input("점수 입력 : "))
02 absence = int(input("결석 횟수 입력 : "))
03 if score < 60 or absence >= 4:
04     print("F 학점입니다.")
```

- ② 프로그램을 저장하고, 다양한 입력값을 사용해 실행

점수 입력 : 76
결석 횟수 입력 : 4
F 학점입니다.

점수 입력 : 90.2
결석 횟수 입력 : 2

01. 제어 구조

[알고리즘-의사코드] 편의점 결제 과정

- (1) 청구금액 입력
- (2) 결제방법 입력
- (3.1) 만약 결제방법이 현금이면
 - (3.1.1) 받은금액 입력
 - (3.1.2) 거스름금액 = 받은금액 - 청구금액
 - (3.1.3) 만약 거스름금액 < 0 이면
 - (3.1.3.1) '금액 부족' 출력
 - (3.1.3.2) (2)로 이동
 - (3.1.4) 아니고, 거스름금액 >= 0 이면
 - (3.1.4.1) 거스름 금액 반환(출력)
 - (3.1.4.2) '감사합니다.' 출력
- (3.2) 만약 결제방법이 현금이 아니면(카드 결제이면)
 - (3.2.1) 카드상태 정상 여부 입력
 - (3.2.2) 만약 카드상태에 문제가 있다면
 - (3.2.2.1) (2)로 이동
 - (3.2.3) 아니면 '감사합니다.' 출력

01. 제어 구조

[프로그램 코드] 편의점 결제 코드

- (1) 청구금액 입력
- (2) 결제방법 입력
- (3.1) 만약 결제방법이 현금이면
 - (3.1.1) 받은금액 입력
 - (3.1.2) 거스름금액 = 받은금액 - 청구금액
 - (3.1.3) 만약 거스름금액 < 0 이면
 - (3.1.3.1) '금액 부족' 출력
 - (3.1.3.2) (2)로 이동
 - (3.1.4) 아니고, 거스름금액 >= 0 이면
 - (3.1.4.1) 거스름 금액 반환(출력)
 - (3.1.4.2) '감사합니다.' 출력
- (3.2) 만약 결제방법이 현금이 아니면(카드)
 - (3.2.1) 카드상태 정상 여부 입력
 - (3.2.2) 만약 카드상태에 문제가 있다면
 - (3.2.2.1) (2)로 이동
 - (3.2.3) 아니면 '감사합니다.' 출력

```
charge = 0           #청구 금액
inkum = 0            #받은 금액
outkum = 0           #거스름 금액
payment = 0          #결제방법(현금이면 1, 아니면 0)
cardstatus = 0       #카드상태(정상이면 1, 아니면 0)

charge = int(input("청구 금액(원): "))

while True:
    payment = int(input("결제방법(현금이면 1, 아니면 0)? : "))
    if payment:
        inkum = int(input("받은 금액(원): "))
        outkum = inkum - charge
        if outkum < 0:
            print("금액 부족!")
            continue
        if outkum >= 0:
            print(outkum, "원 반환")
            print("감사합니다.")
            break
    if not payment:
        cardstatus = int(input("카드상태(정상이면 1, 아니면 0)? : "))
        if not cardstatus:
            continue
        if cardstatus:
            print("감사합니다.")
            break
```

01. 제어 구조

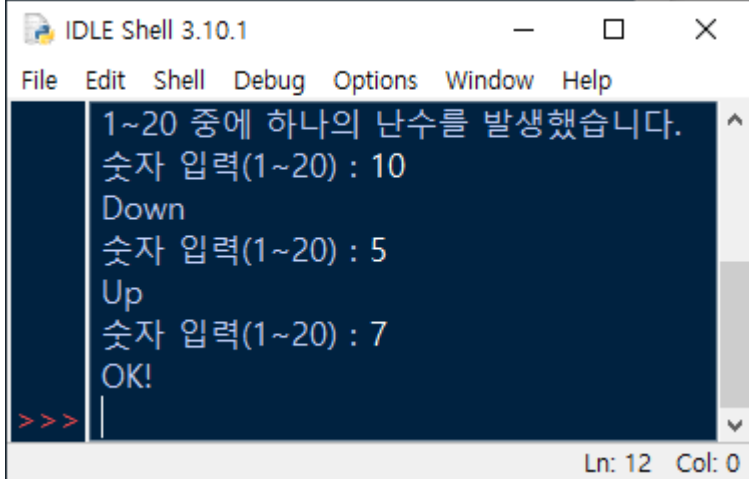
[실습 if] 자동 생성된 난수 값 맞추기

◆ 자동으로 생성한 난수를 맞추는 프로그램을 완성하시오. (if 조건문 사용)

- 난수는 1~20까지 중에 정수로 생성
- 맞추어 값을 키보드로 입력
- 난수 값과 입력 값을 비교하여 결과 출력
 - 'OK!' : 입력값 == 난수
 - 'Down' : 입력값 > 난수
 - 'Up' : 입력값 < 난수

```
# random 모듈 불러오기
import random as r
# 1에서 20 사이의 랜덤 정수 저장
rnum = r.randint(1, 20)
print("1~20 중에 하나의 난수를 발생했습니다.")
innum = 0

while innum != rnum:
    innum = int(input("숫자 입력(1~20) : "))
```



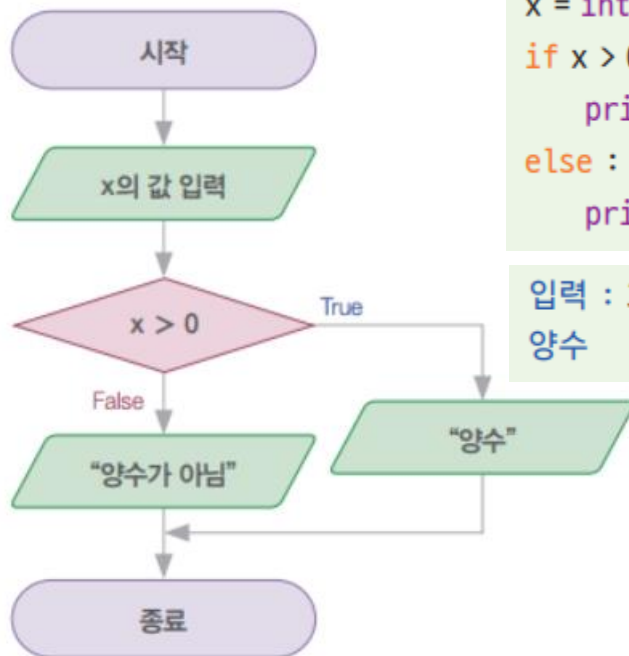
```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
1~20 중에 하나의 난수를 발생했습니다.
숫자 입력(1~20) : 10
Down
숫자 입력(1~20) : 5
Up
숫자 입력(1~20) : 7
OK!
>>> |
Ln: 12 Col: 0
```

01. 제어 구조

I. 조건문의 종류

▪ if~else 문

- 조건식 검사 결과가 참일 때와 거짓일 때, 각각 실행할 명령이 있는 경우에 사용



```
x = int(input("입력 : "))  
if x > 0 :  
    print("양수")  
else :  
    print("양수가 아님")
```

입력 : 3
양수

입력 : -5
양수가 아님

그림 5-11 if-else 문 사용 예

01. 제어 구조

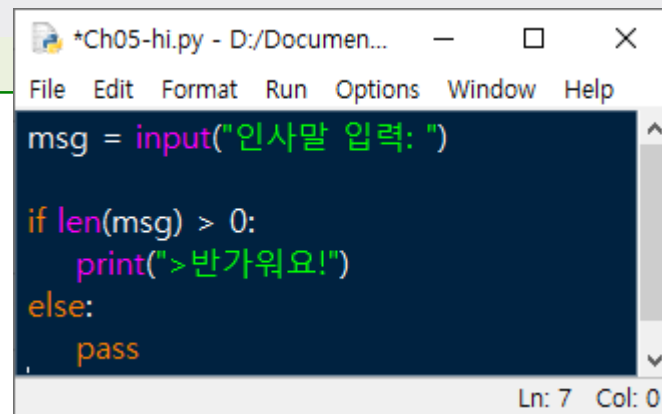
I. 조건문의 종류

여기서 잠깐

pass 실행

- 파이썬에서는 아무 것도 실행하지 않는 pass를 사용할 수 있음
- 다음 코드는 조건식이 True일 때 아무 동작 도 하지 않고, False일 때만 변수 x의 값을 출력

```
if x > 0 :  
    pass  
else :  
    print(x)
```



```
*Ch05-hi.py - D:/Documen...  
File Edit Format Run Options Window Help  
msg = input("인사말 입력: ")  
  
if len(msg) > 0:  
    print(">반가워요!")  
else:  
    pass  
  
Ln: 7 Col: 0
```


01. 제어 구조

I. 조건문의 종류

▪ if~else 문

실습 5-4

미성년자 확인 후 출입 안내하기

code05-04.py

① 이름과 나이를 입력하고 입장 가능한 나이인지 판단

```
01 name = input("이름을 입력하세요 : ")
02 age = int(input("나이를 입력하세요 : "))
03 if age >= 19 :
04     print(name, "님은 성년입니다.")
05     print("입장하세요~")
06 else :
07     print(name, "님은 미성년입니다.")
08     print("출입할 수 없습니다.")
```

② 프로그램을 저장하고, 다양한 입력값을 사용해 실행

이름을 입력하세요 : 홍길동
나이를 입력하세요 : 25
홍길동님은 성년입니다.
입장하세요~

이름을 입력하세요 : 김유신
나이를 입력하세요 : 17
김유신님은 미성년입니다.
출입할 수 없습니다.

01. 제어 구조

I. 조건문의 종류

▪ if~else 문

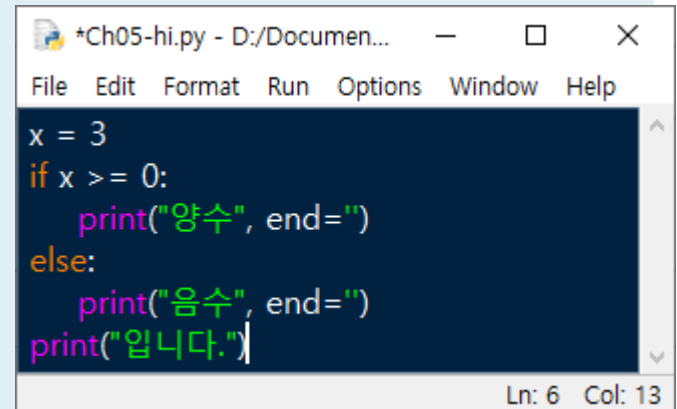
실습 5-5

AI 비서의 날씨 안내하기

code05-05.py

- ① 기온과 미세먼지 값을 입력받고, if~else 문으로 출력 문장의 내용을 다르게 생성

```
01 temp = int(input("기온 : "))
02 dust = int(input("미세먼지 : "))
03
04 if temp > 0 :
05     print("오늘 기온은 영상 %d 도입니다." % temp)
06 else :
07     print("오늘 기온은 영하 %d 도입니다." % temp)
08
09 print("미세먼지 농도는 %d 마이크로로그램으로," % dust, end=' ')
10
11 if dust > 80:
12     print("나쁨 수준입니다. 마스크를 착용하세요.")
13 else :
14     print("괜찮습니다.")
```



```
*Ch05-hi.py - D:/Documen...
File Edit Format Run Options Window Help
x = 3
if x >= 0:
    print("양수", end="")
else:
    print("음수", end="")
print("입니다.")
Ln: 6 Col: 13
```

2개의 문장을 한 줄로 이어서 출력

01. 제어 구조

I. 조건문의 종류

여기서 잠깐

print() 함수의 종료 문자 설정하기

- print() 함수를 사용하면 자동으로 줄 바꿈이 일어나서, 다음과 같이 사용하면 두 줄로 출력
- print() 함수는 기본적으로(**default**) 출력의 종료 문자(end)를 줄 바꿈('\n')으로 지정하기 때문
- 만약 2개의 문장을 한 줄로 이어서 출력하고 싶다면 print() 함수의 종료문자 인자(argument) end 값을 공백(' ')과 같은 다른 문자로 변경오하면 됨

```
09 print("미세먼지 농도는 %d 마이크로그램으로," % dust, end=' ')
    print("채적합니다.")
```

종료문자('\n')를 ' '로 바꿔 출력

01. 제어 구조

I. 조건문의 종류

▪ if~else 문

실습 5-5

AI 비서의 날씨 안내하기

code05-05.py

- ② 프로그램을 저장하고, 기온과 미세먼지의 값을 다르게 입력해서 실행 결과를 비교

기온 : -5
미세먼지 : 30
오늘 기온은 영하 -5 도입니다.
미세먼지 농도는 30 마이크로로그램으로, 쾌적합니다.

기온 : 25
미세먼지 : 90
오늘 기온은 영상 25 도입니다.
미세먼지 농도는 90 마이크로로그램으로, 나쁨 수준입니다. 마스크를 착용하세요.

```
09 print("미세먼지 농도는 %d 마이크로로그램으로," % dust, end=' ')
```

종료문자('\n')를 ' '로 바꿔 출력

```
print("쾌적합니다.")
```

01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문

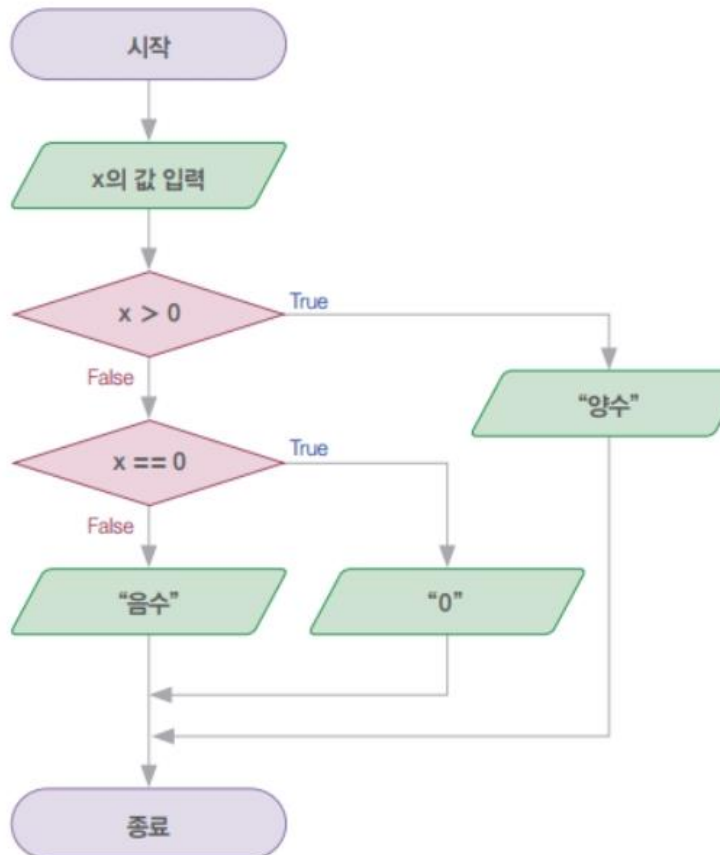


그림 5-12 다중 if 문 사용 예

```
x = int(input("입력 : "))
if x > 0 :
    print("양수")
else :
    if x == 0 :
        print("0")
    else :
        print("음수")
```

입력 : 3
양수

입력 : -5
음수

입력 : 0
0

01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문

실습 5-6

코디네이터가 기온에 따라 옷 골라주기

code05-06.py

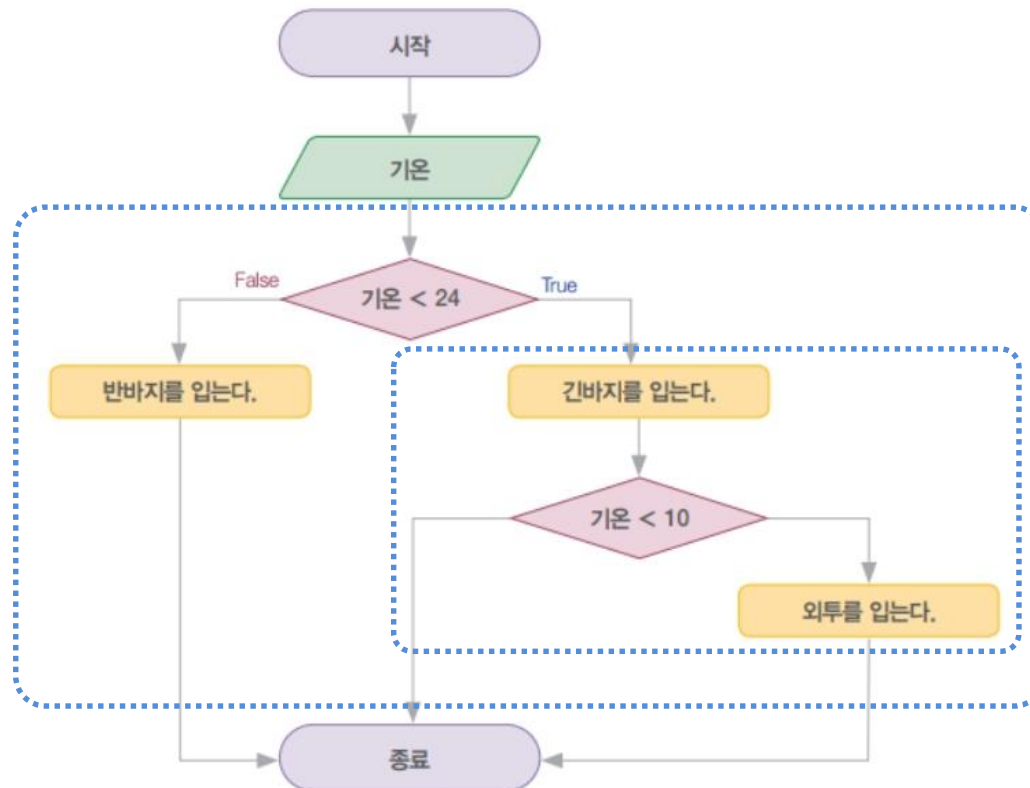


그림 5-13 옷 입기 순서도

01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문

실습 5-6

코디네이터가 기온에 따라 옷 골라주기

code05-06.py

- ① 기온을 입력받아 변수에 저장하고, 조건에 따라 다르게 옷 입기 메시지를 출력

```
01 temp = int(input("오늘 기온을 입력하세요 : "))
02 print("오늘 기온은 %d도입니다." % temp)
03
04 if temp < 24 :
05     print("긴바지를 입으세요.")
06     if temp < 10 :
07         print("외투도 입는 게 좋겠네요.")
08 else :      #기온이 24도 이상인 경우
09     print("반바지를 입으세요.")
```

- ② 오늘 기온을 입력하세요 : 28
오늘 기온은 28도입니다.
반바지를 입으세요.

오늘 기온을 입력하세요 : 5
오늘 기온은 5도입니다.
긴바지를 입으세요.
외투도 입는 게 좋겠네요.

01. 제어 구조

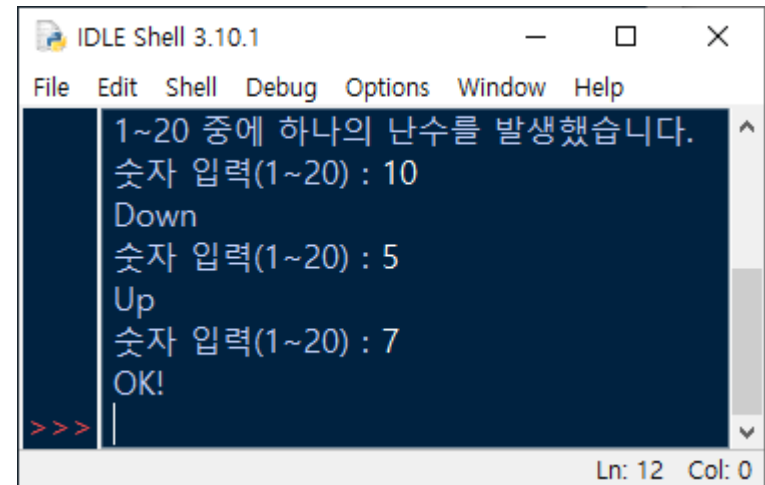
[실습 다중 if] 자동 생성된 난수 값 맞추기

◆ 자동으로 생성한 난수를 맞추는 프로그램을 완성하시오. (if~else 조건문 다중 사용)

- 난수 값과 입력 값을 비교하여 결과 출력
 - 'OK!' : 입력값 == 난수
 - 'Down' : 입력값 > 난수
 - 'Up' : 입력값 < 난수

```
# random 모듈 불러오기
import random as r
# 1에서 20 사이의 랜덤 정수 저장
rnum = r.randint(1, 20)
print("1~20 중에 하나의 난수를 발생했습니다.")
innum = 0

while innum != rnum:
    innum = int(input("숫자 입력(1~20) : "))
    if innum > rnum:
        print("Down")
    if innum < rnum:
        print("Up")
    if innum == rnum:
        print("OK!")
```



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
1~20 중에 하나의 난수를 발생했습니다.
숫자 입력(1~20) : 10
Down
숫자 입력(1~20) : 5
Up
숫자 입력(1~20) : 7
OK!
>>> |
Ln: 12 Col: 0
```


01. 제어 구조

[과제-1] 할인을 최대 적용

◆ 다음 요구사항에 맞는 프로그램을 완성하시오. (if~else 조건문 다중 사용)

[할인율-a] "지역주민"은 30% 할인

[할인율-b] "국가유공자/현역군인"은 50% 할인

[할인율-c] "70세 이상" 어르신은 100% 할인

➤ 할인율을 최대한 높게 결정해주어야 한다.

```
fee = 10000 #입장료
print(">입장료는 %d원입니다." %fee)
print(">신분이 어떻게 되는지 y나 기타 키로 답하세요.")

status = input(">>ooo 이신가요(y)? ").strip().lower()
if status == 'y':

    print(">>>%d%%가 할인되어 %d원입니다." % ( , ))
```

```
>입장료는 10000원입니다.
>신분이 어떻게 되는지 y나 기타 키로 답하세요.
>>70세 이상이신가요(y)?
>>국가유공자나현역군인 이신가요(y)?
>>지역주민이신가요(y)? y
>>>30%가 할인되어 7000원입니다.
```

종료 코드 0(으)로 완료된 프로세스

01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문

실습 5-7

체질량지수로 건강 상태 알아보기

code05-07.py

- ① 체질량지수(BMI)는 입력받은 몸무게를 키 제곱으로 나눔, 키를 미터(m) 단위로 계산



그림 5-14 BMI 계산과 프로그램 실행 순서

- ② 계산된 BMI 값에 따라 비만도 분류와 건강 위험도를 출력

표 5-1 BMI 계산 기준

체질량지수 범위	비만도 분류	건강 위험도
18.5 미만	저체중	높음
18.5 이상 25 미만	정상체중	낮음
25 이상 30 미만	과체중	낮음
30 이상	비만	높음

01. 제어 구조

I. 조건문의 종류

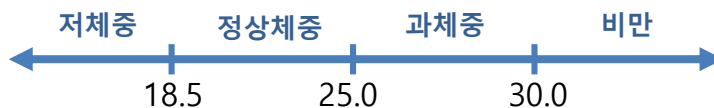
▪ 다중 if 문

- ② 계산된 BMI 값에 따라 비만도 분류와 건강 위험 도를 출력

표 5-1 BMI 계산 기준

체질량지수 범위	비만도 분류	건강 위험도
18.5 미만	저체중	높음
18.5 이상 25 미만	정상체중	낮음
25 이상 30 미만	과체중	낮음
30 이상	비만	높음

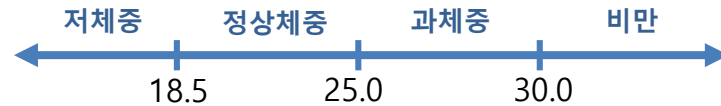
❖ 다중 if문 구성



01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문



실습 5-7

체질량지수로 건강 상태 알아보기

code05-07.py

```
③ 01 height = int(input("키 입력(cm) : "))
    02 weight = int(input("몸무게 입력(kg) : "))
    03
    04 bmi = weight / (height / 100) ** 2          # 키를 미터 단위로 변환한 후 계산
    05 print("*** 체질량지수 %.1f : " % bmi, end=' ') # 소수점 이하 1자리로 출력
    06
    07 if bmi < 25 :                                # 저체중이거나 정상 체중인 경우
    08     if bmi < 18.5 :
    09         print("저체중, 건강 위험도 높음 ***")
    10     else :
    11         print("정상체중, 건강 위험도 낮음 ***")
    12 else :                                        # 과체중이거나 비만인 경우
    13     if bmi < 30 :
    14         print("과체중, 건강 위험도 낮음 ***")
    15     else :
    16         print("비만, 건강 위험도 높음 ***")
```

01. 제어 구조

I. 조건문의 종류

▪ 다중 if 문

실습 5-7

체질량지수로 건강 상태 알아보기

code05-07.py

④

키 입력(cm) : 175

몸무게 입력(kg) : 73

*** 체질량지수 23.8 : 정상체중, 건강 위험도 낮음 ***

키 입력(cm) : 160

몸무게 입력(kg) : 47

*** 체질량지수 18.4 : 저체중, 건강 위험도 높음 ***

01. 제어 구조

I. 조건문의 종류

▪ if~elif 문

- if~elif 문은 여러 개의 조건을 필요한 만큼 계속 추가할 수 있어서 작성하기 용이
- 프로그램의 구조적인 면에서도 더 간단하고 이해하기 쉬움

다중 if 문

```
x = int(input("입력 : "))
if x > 0 :
    print("양수")
else :
    if x == 0 :
        print("0")
    else :
        print("음수")
```

if~elif 문

```
x = int(input("입력 : "))
if x > 0 :
    print("양수")
elif x == 0 :
    print("0")
else :
    print("음수")
```

=

01. 제어 구조

I. 조건문의 종류

▪ if~elif 문

실습 5-8

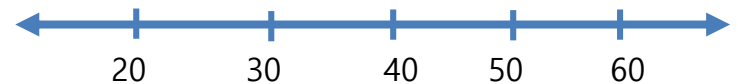
나이별 연령대 판단하기

code05-08.py

```
① 01 name = input("이름을 입력하세요 : ")
    02 age = int(input("나이를 입력하세요 : "))
    03
    04 if age < 20 :
    05     print(name, "님은 청소년입니다.")
    06 elif age < 30 :
    07     print(name, "님은 20대입니다.")
    08 elif age < 40 :
    09     print(name, "님은 30대입니다.")
    10 elif age < 50 :
    11     print(name, "님은 40대입니다.")
    12 elif age < 60 :
    13     print(name, "님은 50대입니다.")
    14 else :
    15     print(name, "님은 60대 이상입니다.")
```

```
② 이름을 입력하세요 : 홍길동
   나이를 입력하세요 : 36
   홍길동 님은 30대입니다.
```

```
이름을 입력하세요 : 김유신
나이를 입력하세요 : 17
김유신 님은 청소년입니다.
```



01. 제어 구조

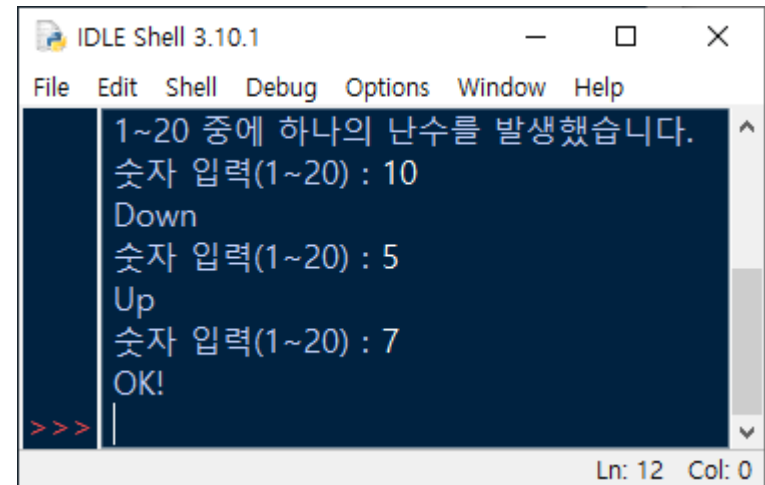
[실습 if~elif] 자동 생성된 난수 값 맞추기

◆ 자동으로 생성한 난수를 맞추는 프로그램을 완성하시오. (if~elif 조건문 사용)

- 난수 값과 입력 값을 비교하여 결과 출력
 - 'OK!' : 입력값 = 난수
 - 'Down' : 입력값 > 난수
 - 'Up' : 입력값 < 난수

```
# random 모듈 불러오기
import random as r
# 1에서 20 사이의 랜덤 정수 저장
rnum = r.randint(1, 20)
print("1~20 중에 하나의 난수를
발생했습니다.")
innum = 0

while innum != rnum:
    innum = int(input("숫자 입력(1~20) : "))
    if innum > rnum:
        print("Down")
    if innum < rnum:
        print("Up")
    if innum == rnum:
        print("OK!")
```



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
1~20 중에 하나의 난수를 발생했습니다.
숫자 입력(1~20) : 10
Down
숫자 입력(1~20) : 5
Up
숫자 입력(1~20) : 7
OK!
>>> |
```

Ln: 12 Col: 0

01. 제어 구조

I. 조건문의 종류

▪ if~elif 문

- ② 계산된 BMI 값에 따라 비만도 분류와 건강 위험 도를 출력

표 5-1 BMI 계산 기준

체질량지수 범위	비만도 분류	건강 위험도
18.5 미만	저체중	높음
18.5 이상 25 미만	정상체중	낮음
25 이상 30 미만	과체중	낮음
30 이상	비만	높음

❖ if~elif 문 구성



01. 제어 구조

I. 조건문의 종류

▪ if~elif 문

실습 5-9

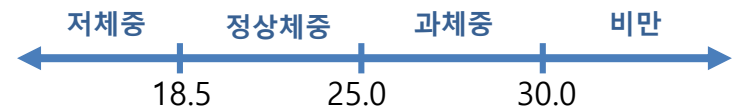
if~elif 문을 사용하여 체질량지수로 건강 상태 알아보기

code05-09.py

```
① 01 height = int(input("키 입력(cm) : "))
    02 weight = int(input("몸무게 입력(kg) : "))
    03
    04 bmi = weight / (height / 100) ** 2
    05 print("*** 체질량지수 %.1f : " % bmi, end=' ')
    06
    07 if bmi < 18.5 :
    08     print("저체중, 건강 위험도 높음 ***")
    09 elif bmi < 25 :
    10     print("정상체중, 건강 위험도 낮음 ***")
    11 elif bmi < 30 :
    12     print("과체중, 건강 위험도 낮음 ***")
    13 else :
    14     print("비만, 건강 위험도 높음 ***")
```

```
② 키 입력(cm) : 170
    몸무게 입력(kg) : 65
    *** 체질량지수 22.5 : 정상체중, 건강 위험도 낮음 ***
```

```
키 입력(cm) : 175
    몸무게 입력(kg) : 95
    *** 체질량지수 31.0 : 비만, 건강 위험도 높음 ***
```



01. 제어 구조

I. 조건문의 종류

▪ 조건문의 복잡도 추정

- 복잡도complexity는 각 처리 경우에 대한 전체 평균 연산 횟수로 판단

```
01 height = int(input("키 입력(cm) : "))
02 weight = int(input("몸무게 입력(kg) : "))
03
04 bmi = weight / (height / 100) ** 2
05 print("*** 체질량지수 %.1f : " % bmi, end=' ')
06
07 if bmi < 25 :
08     if bmi < 18.5 :
09         print("저체중, 건강 위험도 높음 ***")
10     else :
11         print("정상체중, 건강 위험도 낮음 ***")
12 else :
13     if bmi < 30 :
14         print("과체중, 건강 위험도 낮음 ***")
15     else :
16         print("비만, 건강 위험도 높음 ***")
```

```
01 height = int(input("키 입력(cm) : "))
02 weight = int(input("몸무게 입력(kg) : "))
03
04 bmi = weight / (height / 100) ** 2
05 print("*** 체질량지수 %.1f : " % bmi, end=' ')
06
07 if bmi < 18.5 :
08     print("저체중, 건강 위험도 높음 ***")
09 elif bmi < 25 :
10     print("정상체중, 건강 위험도 낮음 ***")
11 elif bmi < 30 :
12     print("과체중, 건강 위험도 낮음 ***")
13 else :
14     print("비만, 건강 위험도 높음 ***")
```

01. 제어 구조

I. 조건문의 종류

■ 조건문의 복잡도, 완전성 검증

실습 5-9

if~elif 문을 사용하여 체질량지수로 건강 상태 알아보기

code05-09.py

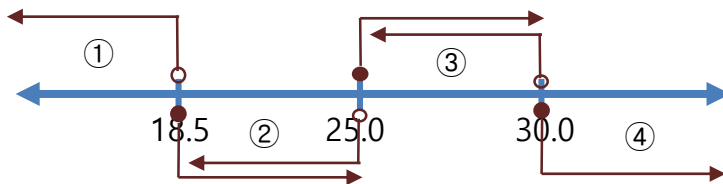
A. 복잡도 : 8.0 (평균 연산 횟수로 추정, print()는 제외)

- 저체중 : $1 + (3 + 3 + 1) = 8$ 회
- 정상체중 : $(1) + 3 + (3 + 1) = 8$ 회
- 과체중 : $(1 + 3) + 3 + (1) = 8$ 회
- 비만 : $(1 + 3 + 3) + 1 = 8$ 회
- 평균 복잡도 = $(8 + 8 + 8 + 8) / 4$
 $= 32 / 4 = 8.0$

#[A] 복잡도 : $32/4=8.0$

```
if bmi < 18.5 :  
    print("저체중, 건강 위험도 높음 ***")  
if bmi >= 18.5 and bmi < 25 :  
    print("정상체중, 건강 위험도 낮음 ***")  
if bmi >= 25.0 and bmi < 30 :  
    print("과체중, 건강 위험도 낮음 ***")  
if bmi >= 30 :  
    print("비만, 건강 위험도 높음 ***")
```

❖ 완전성



❖ 검증 Test : 경계점 중심으로 선택

18.4, 18.5, 18.6, 25.0, 25.1, 30.0, 30.1

01. 제어 구조

I. 조건문의 종류

■ 조건문의 복잡도, 완전성 검증

실습 5-9

if~elif 문을 사용하여 체질량지수로 건강 상태 알아보기

code05-09.py

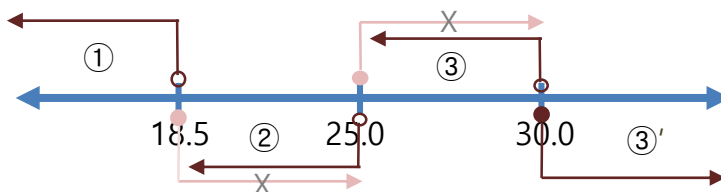
B. 복잡도 : 2.25

- 저체중 : 1회
 - 정상체중 : 1 + 1 = 2회
 - 과체중 : 2 + 1 = 3회
 - 비만 : 3회
- 평균 복잡도 = (1 + 2 + 3 + 3) / 4
= 9 / 4 = 2.25

#[B] 복잡도 : 9/4=2.25

```
if bmi < 18.5 :  
    print("저체중, 건강 위험도 높음 ***)  
elif bmi < 25 :  
    print("정상체중, 건강 위험도 낮음 ***)  
elif bmi < 30 :  
    print("과체중, 건강 위험도 낮음 ***)  
else :  
    print("비만, 건강 위험도 높음 ***)
```

❖ 완전성



❖ 검증 Test : 경계점 중심으로 선택

18.4, 18.5, 18.6, 25.0, 25.1, 30.0, 30.1

01. 제어 구조

I. 조건문의 종류

■ 조건문의 복잡도, 완전성 검증

실습 5-9

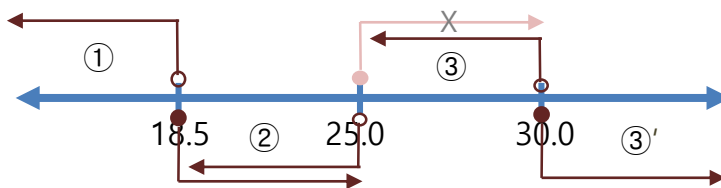
if~elif 문을 사용하여 체질량지수로 건강 상태 알아보기

code05-09.py

C. 복잡도 : 2.25

- 저체중 : 1회
 - 정상체중 : $1 + 1 = 2$ 회
 - 과체중 : $2 + 1 = 3$ 회
 - 비만 : 3회
- 평균 복잡도 = $(1 + 2 + 3 + 3) / 4$
 $= 9 / 4 = 2.25$

❖ 완전성



#[C] 복잡도 : $9/4=2.25$

```
if bmi >= 18.5 : #정상체중 이상
    if bmi < 25 :
        print("정상체중, 건강 위험도 낮음 ****")
    else :      #과체중 이상
        if bmi < 30 :
            print("과체중, 건강 위험도 낮음 ****")
        else :
            print("비만, 건강 위험도 높음 ****")
else :
    print("저체중, 건강 위험도 높음 ****")
```

❖ 검증 Test : 경계점 중심으로 선택

18.4, 18.5, 18.6, 25.0, 25.1, 30.0, 30.1

01. 제어 구조

I. 조건문의 종류

■ 조건문의 복잡도, 완전성 검증

실습 5-9

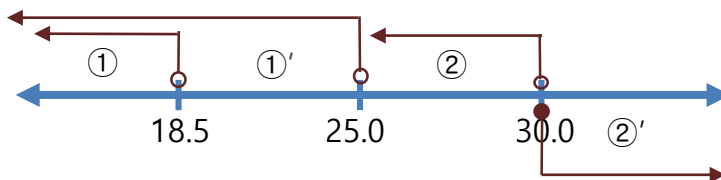
if~elif 문을 사용하여 체질량지수로 건강 상태 알아보기

code05-09.py

D. 복잡도 : 2.0

- 저체중 : 1 + 1 = 2회
 - 정상체중 : 1 + 1 = 2회
 - 과체중 : 1 + 1 = 2회
 - 비만 : 1 + 1 = 2회
- 평균 복잡도 = (2 + 2 + 2 + 2) / 4
= 8 / 4 = 2.0

❖ 완전성



#[D] 복잡도 : 8/4=2.0

```
if bmi < 25 : # 저체중이거나 정상체중
    if bmi < 18.5 :
        print("저체중, 건강 위험도 높음 ****")
    else :
        print("정상체중, 건강 위험도 낮음 ****")
else : # 과체중이거나 비만
    if bmi < 30 :
        print("과체중, 건강 위험도 낮음 ****")
    else :
        print("비만, 건강 위험도 높음 ****")
```

❖ 검증 Test : 경계점 중심으로 선택

18.4, 18.5, 18.6, 25.0, 25.1, 30.0, 30.1

01. 제어 구조

[실습] 복잡도 비교

◆ 아래 세개 코드의 복잡도를 구하여 비교해보세요.

- [A] 복잡도 :
- [B] 복잡도 :
- [C] 복잡도 :

[A]

```
if innum > rnum:
    print("Down")
if innum < rnum:
    print("Up")
if innum == rnum:
    print("OK!")
```

[B]

```
if innum > rnum:
    print("Down")
else:
    if innum < rnum:
        print("Up")
    else:
        if innum == rnum:
            print("OK!")
```

[C]

```
if innum > rnum:
    print("Down")
elif innum < rnum:
    print("Up")
else:
    print("OK!")
```


01. 제어 구조

II. 조건문의 활용

실습 5-10

단위 변환기 만들기

code05-10.py



그림 5-15 온도, 무게, 길이의 단위 변환

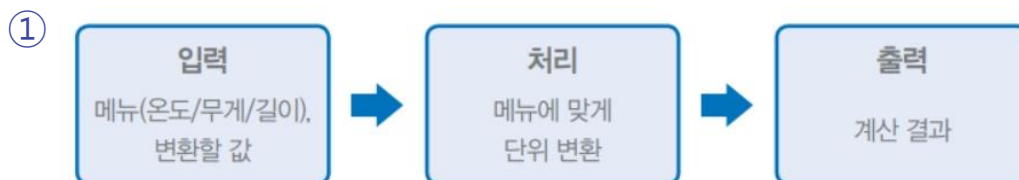


그림 5-16 단위 변환기 실행 순서

②

화씨(°F) → 섭씨(°C)	$(\text{화씨온도} - 32) / 1.8$
파운드(lb) → 킬로그램(kg)	$\text{파운드무게} * 0.453592$
인치(in) → 센티미터(cm)	$\text{인치길이} * 2.54$

표 5-2 단위 변환 계산

01. 제어 구조

II. 조건문의 활용

실습 5-10

단위 변환기 만들기

code05-10.py

```
③ 01 menu = int(input("번호 입력(1:온도, 2:무게, 3:길이) : "))
    02 value = float(input("변환할 값 입력 : "))
    03
    04 if menu == 1 :          # 온도 단위 변환
    05     print("변환 결과 = %.2f ℃" % ((value - 32) / 1.8))      # 화씨 → 섭씨
    06 elif menu == 2 :       # 무게 단위 변환
    07     print("변환 결과 = %.2f kg" % (value * 0.453592))        # 파운드 → kg
    08 elif menu == 3 :       # 길이 단위 변환
    09     print("변환 결과 = %.2f cm" % (value * 2.54))            # 인치 → cm
    10 else :                 # 잘못된 메뉴 선택
    11     print("메뉴 선택 오류!")
```

```
④ 번호 입력(1:온도, 2:무게, 3:길이) : 1
    변환할 값 입력 : 62
    변환 결과 = 16.67 ℃
```

```
번호 입력(1:온도, 2:무게, 3:길이) : 2
    변환할 값 입력 : 2.5
    변환 결과 = 1.13 kg
```

✓ **TIP** 각종 단위 기호는 한글 자음 'ㄹ'을 입력하고 한자 키를 눌러서 기호로 변환

01. 제어 구조

II. 조건문의 활용

실습 5-11

주민등록번호로 성별 판단하기

code05-11.py

회원 정보

회원 이름 : 홍길동

주민번호 : 1 2 3 4 5 6 - X x x x x x x

성별(여자 : 2, 4 · 남자 : 1, 3)

그림 5-17 성별 판단하는 기준

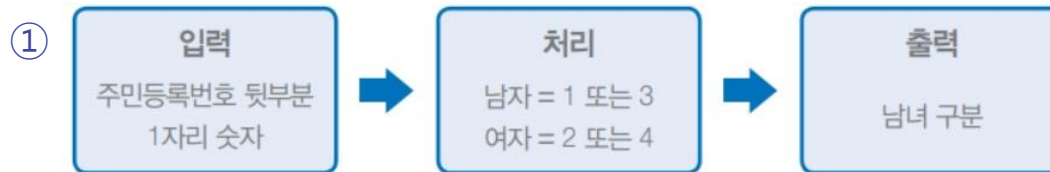


그림 5-18 성별 판단 순서

01. 제어 구조

II. 조건문의 활용

실습 5-11

주민등록번호로 성별 판단하기

code05-11.py

```
② 01 number = int(input("주민등록번호 뒷부분 한 자리 숫자를 입력하세요 : "))
    02
    03 if 1 <= number <= 4 :
    04     if number % 2 == 1 :      # 2로 나눈 나머지가 1인 경우(=1 또는 3)
    05         print("회원은 남자입니다.")
    06     else :
    07         print("회원은 여자입니다.")
    08 else :                        # 잘못된 입력인 경우
    09     print("올바른 값이 아닙니다!")
```

③ 주민등록번호 뒷부분 한 자리 숫자를 입력하세요 : 2
회원은 여자입니다.

주민등록번호 뒷부분 한 자리 숫자를 입력하세요 : 3
회원은 남자입니다.

01. 제어 구조

II. 조건문의 활용

실습 5-12

연령별 영화 추천하기

code05-12.py

```
① 01 age = int(input("나이를 입력하세요 : "))
    02
    03 print("** 추천 영화 목록 **")
    04 if age < 20:
    05     print("어벤저스")
    06 if age < 40:
    07     print("물란")
    08 if 20 <= age < 60:
    09     print("테넷")
    10 if 20 <= age < 40 or age >= 60:
    11     print("오!문희")
```

영화 \ 연령대	10대 이하	20~30대	40~50대	60대 이상
어벤저스				
물란				
테넷				
오!문희				

그림 5-19 연령별 영화 추천하기

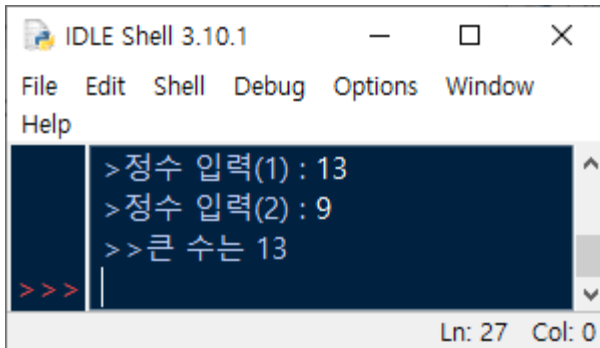
```
② 나이를 입력하세요 : 43
   ** 추천 영화 목록 **
   테넷
```

01. 제어 구조

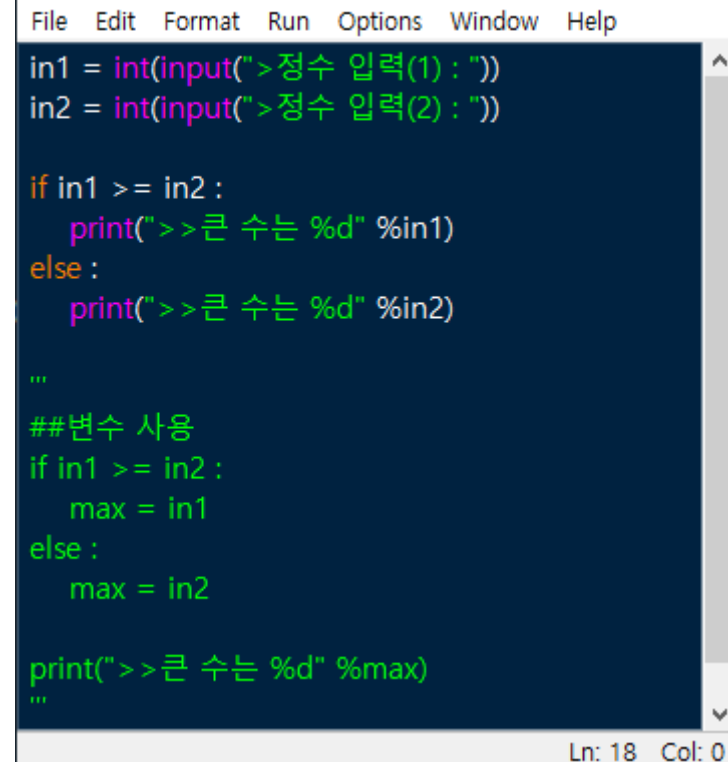
[실습] Ch05-Ex01-01 >> 두 수 중에 큰 수 출력

◆ 두 개의 정수를 입력 받아 둘 중에 큰 수를 출력하시오. (조건문 사용)

- 큰 수를 기억할 변수(max)를 사용하면
 - 큰 수에 대한 **재사용** 가능
 - 프로그램에 대한 **확장성** 용이 (여러 개 입력에 대한 확장)
 - 프로그램의 **가독성** 증가



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>정수 입력(1) : 13
>정수 입력(2) : 9
>>큰 수는 13
>>>
Ln: 27 Col: 0
```



```
File Edit Format Run Options Window Help
in1 = int(input(">정수 입력(1) : "))
in2 = int(input(">정수 입력(2) : "))

if in1 >= in2:
    print(">>큰 수는 %d" %in1)
else:
    print(">>큰 수는 %d" %in2)

'''
##변수 사용
if in1 >= in2:
    max = in1
else:
    max = in2

print(">>큰 수는 %d" %max)
'''
Ln: 18 Col: 0
```

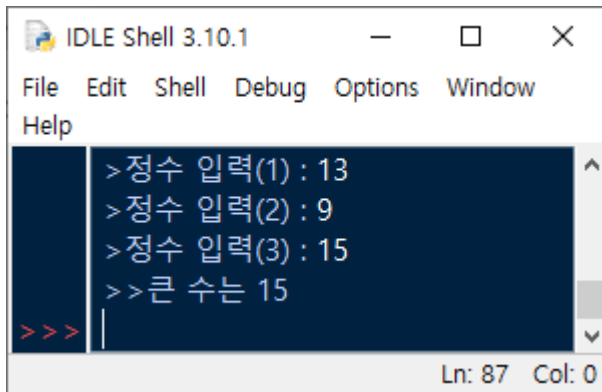
01. 제어 구조

[실습] Ch05-Ex01-02 >> 세 수 중에 큰 수 출력

◆ 세 개의 정수를 입력 받아 둘 중에 큰 수를 출력하시오. (조건문 사용)

➤ 큰 수를 기억할 변수(max)를 사용하면

- 큰 수에 대한 **재사용** 가능
- 프로그램에 대한 **확장성** 용이 (여러 개 입력에 대한 확장)
- 프로그램의 **가독성** 증가



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>정수 입력(1) : 13
>정수 입력(2) : 9
>정수 입력(3) : 15
>>큰 수는 15
>>>
Ln: 87 Col: 0
```

```
if in1 >= in2 :
    if in1 >= in3 :
        print(">>큰 수는 %d" %in1)
    else :
        print(">>큰 수는 %d" %in3)
else :
    if in2 >= in3 :
        print(">>큰 수는 %d" %in2)
    else :
        print(">>큰 수는 %d" %in3)
...
##변수 사용
max = in1
if max < in2 :
    max = in2

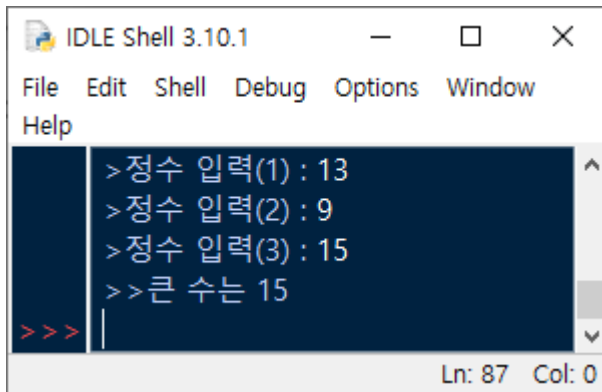
if max < in3 :
    max = in3
print(">>큰 수는 %d" %max)
```

01. 제어 구조

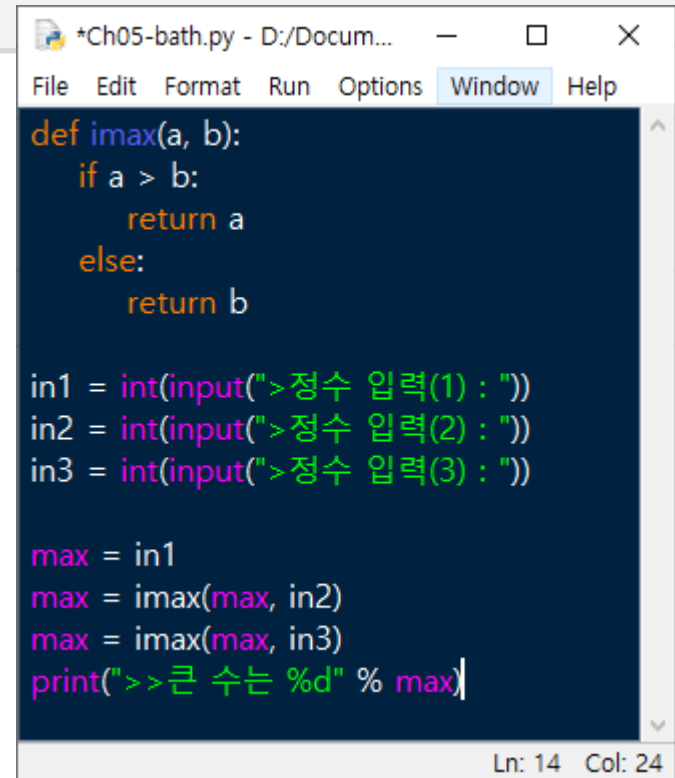
[실습] Ch05-Ex01-03 >> 세 수 중에 큰 수 출력

◆ 세 개의 정수를 입력 받아 둘 중에 큰 수를 출력하시오. (함수 호출 사용)

- 큰 수를 기억할 변수(max)를 사용하면
 - 큰 수에 대한 **재사용** 가능
 - 프로그램에 대한 **확장성** 용이 (여러 개 입력에 대한 확장)
 - 프로그램의 **가독성** 증가



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>정수 입력(1) : 13
>정수 입력(2) : 9
>정수 입력(3) : 15
>>큰 수는 15
>>>
Ln: 87 Col: 0
```



```
*Ch05-bath.py - D:/Docum...
File Edit Format Run Options Window Help
def imax(a, b):
    if a > b:
        return a
    else:
        return b

in1 = int(input(">정수 입력(1) : "))
in2 = int(input(">정수 입력(2) : "))
in3 = int(input(">정수 입력(3) : "))

max = in1
max = imax(max, in2)
max = imax(max, in3)
print(">>큰 수는 %d" % max)
Ln: 14 Col: 24
```


01. 제어 구조

[실습] Code05Ex01-03 >> 세 수 중에 큰 수 출력 (**3항연산자** 사용)

◆ 세 개의 정수를 입력 받아 둘 중에 큰 수를 출력하시오. (조건문 사용)

- 3항연산자 사용
- 변수 = {true 시 반환 값} if 조건식 else {false 시 반환 값}

##3항연산자 사용

```
max = in1 if (in1 >= in2) else in2  
print(">>큰 수는 %d" %max)
```

*Ch05-Ex02.py - D:\Documents\강의록\Python\컴퓨터사고와파이썬-김지연\...

File Edit Format Run Options Window Help

##3항연산 사용(1)

```
max = (in1 if in1 >= in2 else in2) if ((in1 if in1 >= in2 else in2) > in3) else in3  
print(">>큰 수는 %d" %max)
```

##3항연산 사용(2)

```
max = in1  
max = (in2 if in2 > max else max) if ((in2 if in2 > max else max) > in3) else in3  
print(">>큰 수는 %d" %max)
```

Ln: 33

```
"""  
in1 = int(input("1. 정수 입력 : "))  
in2 = int(input("2. 정수 입력 : "))  
in3 = int(input("3. 정수 입력 : "))  
"""  
if in1 >= in2 :  
    if in1 >= in3 :  
        print(">>큰 수는 %d" %in1)  
    else :  
        print(">>큰 수는 %d" %in3)  
else :  
    if in2 >= in3 :  
        print(">>큰 수는 %d" %in2)  
    else :  
        print(">>큰 수는 %d" %in3)  
"""
```

##변수 사용

```
max = in1  
if max < in2 :  
    max = in2  
  
if max < in3 :  
    max = in3  
print(">>큰 수는 %d" %max)
```

Ln: 16

01. 제어 구조

II. 조건문의 활용

실습 5-13

성적 등급 결정하기

code05-13.py

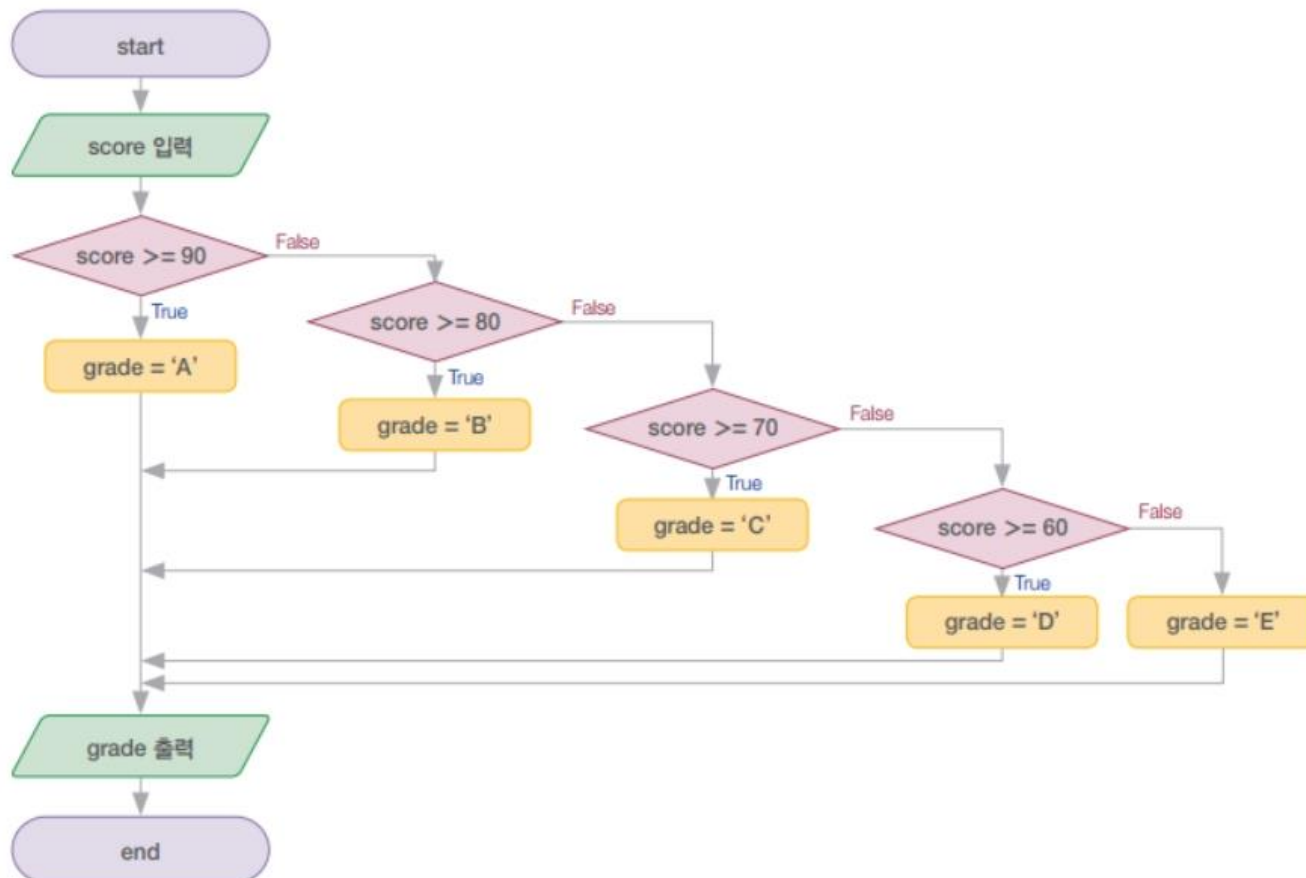


그림 5-20 성적 등급 결정하기 알고리즘 순서도

01. 제어 구조

II. 조건문의 활용

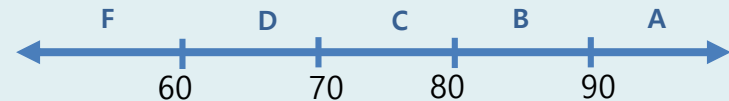
실습 5-13

성적 등급 결정하기

code05-13.py

①

```
01 score = int(input("점수를 입력하세요 : "))
02 grade = ''
03 ##복잡도 : (1+2+3+4+4)/5 = 2.8
04 if score >= 90:
05     grade = 'A'
06 elif score >= 80:
07     grade = 'B'
08 elif score >= 70:
09     grade = 'C'
10 elif score >= 60:
11     grade = 'D'
12 else:
13     grade = 'F'
14
15 print("등급은", grade)
```



##복잡도 : (1+4+7+10+10)/5 = 6.4

```
if score >= 90:
    grade = 'A'
elif score < 90 and score >= 80:
    grade = 'B'
elif score < 80 and score >= 70:
    grade = 'C'
elif score < 70 and score >= 60:
    grade = 'D'
else:
    grade = 'F'
```

② 점수를 입력하세요 : 94
등급은 A

점수를 입력하세요 : 70
등급은 C

01. 제어 구조

[과제-2] 점수에 대한 등급 판단 (A, A+, B, B+, C, C+, D, D+, F) 등급

◆ 한개의 점수를 입력 받아 아래 등급으로 환산하여 출력하시오. (조건문 사용)

- 점수 범위 : 0 ~ 100
- 95 이상은 'A+', 90 이상 95 미만은 'A',
85 이상 90 미만은 'B+', 80 이상 85 미만은 'B',
75 이상 80 미만은 'C+', 70 이상 75 미만은 'C',
65 이상 70 미만은 'D+', 60 이상 65 미만은 'D',
0 이상 60 미만은 'F'등급
- 점수 범위를 초과하면 "값이 범위(0~100)를 초과함!" 메시지 출력
- ✓ 총 명령어 스텝 수는 앞의 (A, B, C, D, F) 등급보다 몇 스텝이 추가되었는가?

```
>점수 입력 : 100
>>100의 등급은 A
>점수 입력 : 97
>>97의 등급은 A+
>점수 입력 : 90
>>90의 등급은 A
>점수 입력 : 88
>>88의 등급은 B+
>점수 입력 : 59
>>59의 등급은 F+
>점수 입력 : 0
>>0의 등급은 F
>점수 입력 : 101
>>101는 범위를 초과한 값입니다.
>점수 입력 : -10
>>-10는 범위를 초과한 값입니다.
>점수 입력 :
```

01. 제어 구조

II. 조건문의 활용

실습 5-14

환율 계산하기

code05-14.py

표 5-3 각국의 환율

①

국가	환율(원)	단위
미국	1188.50	달러
영국	1570.13	파운드
러시아	15.01	루블
중국	173.93	위안

②

```
01 money = int(input("환전하려는 금액(원) : "))
02 nation = input("국가 선택(미국/영국/러시아/중국) : ")
03
04 exchange = 0; currency = " "
05
06 if nation == "미국":
07     exchange = money / 1188.50
08     currency = "달러"
09 elif nation == "영국":
10     exchange = money / 1570.13
```

01. 제어 구조

II. 조건문의 활용

실습 5-14

환율 계산하기

code05-14.py

```
③ 11     currency = "파운드"
    12     elif nation == "러시아":
    13         exchange = money / 15.01
    14         currency = "루블"
    15     elif nation == "중국":
    16         exchange = money / 173.93
    17         currency = "위안"
    18     else:
    19         print("국가를 잘못 입력했습니다.")
    20
    21     print("%.2f" % exchange, currency)           # 소수점 이하 2자리로 단위도 함께 출력
```

④ 환전하려는 금액(원) : 500000
국가 선택(미국/영국/러시아/중국) : 미국
420.70 달러

환전하려는 금액(원) : 500000
국가 선택(미국/영국/러시아/중국) : 러시아
33311.13 루블

Thank You !

[Python]