



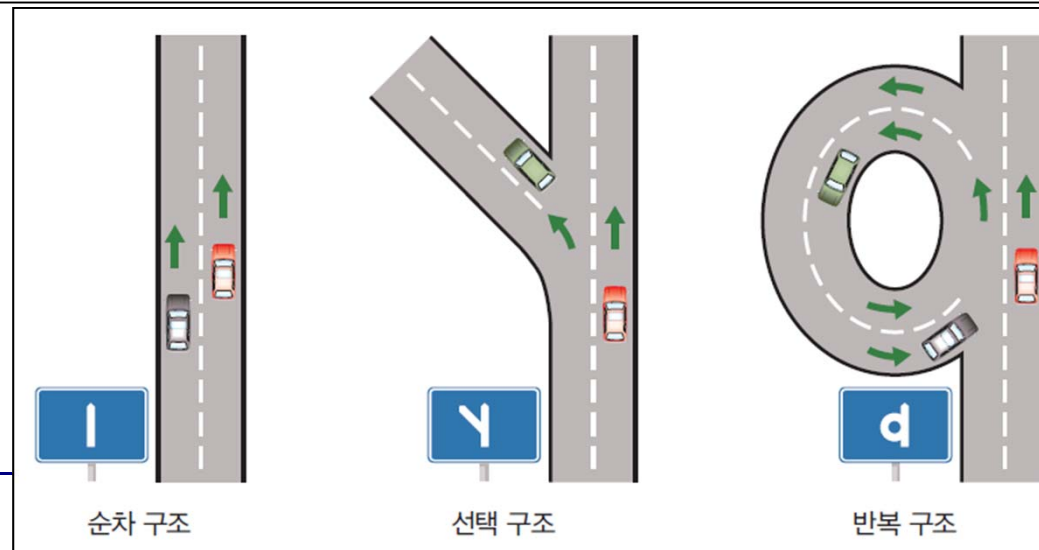
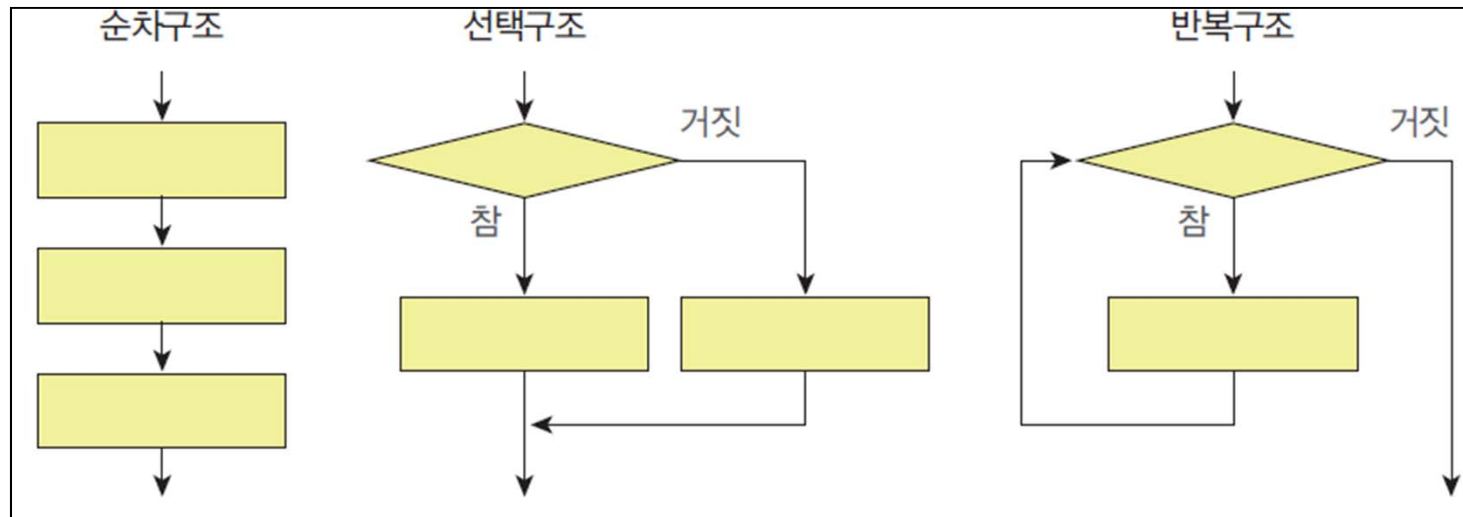
Ch 3.

조건문



3가지의 기본 제어 구조

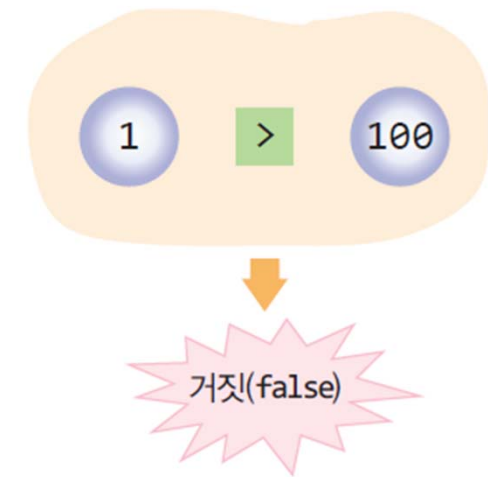
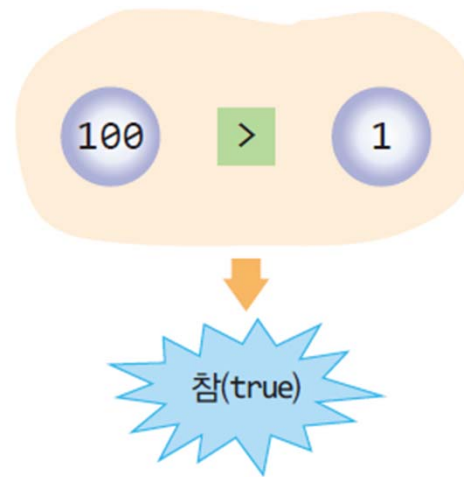
- 순차 구조(sequence) - 명령들이 순차적으로 실행되는 구조이다.
- 선택 구조(selection) - 둘 중의 하나의 명령을 선택하여 실행되는 구조이다.
- 반복 구조(iteration) - 동일한 명령이 반복되면서 실행되는 구조이다.



관계 연산자

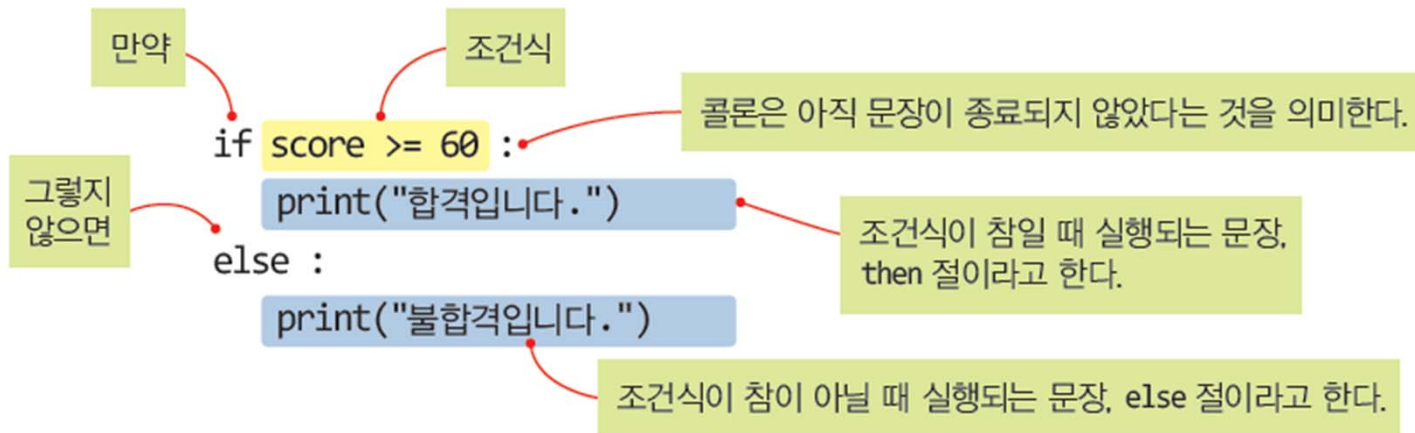
- 관계 연산자(relational operator)는 두 개의 피연산자를 비교하는 연산자

연산	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x \geq y$	x가 y보다 크거나 같은가?
$x \leq y$	x가 y보다 작거나 같은가?



if-else 문

if-else 문



예제 #1

```
score = int(input("성적을 입력하시오: "))  
if score >= 60:  
    print("합격입니다.")  
else:  
    print("불합격입니다.")
```

성적을 입력하시오: 80
합격입니다.

예제 #2

```
num = int(input("정수를 입력하시오: "))  
if num % 2 == 0 :  
    print("짝수입니다.")  
else:  
    print("홀수입니다.")
```

정수를 입력하시오: 10
짝수입니다.

블록

- 만약 조건이 참인 경우에 여러 개의 문장이 실행되어야 한다면 어떻게 해야 하는가?

블록문

```
if score > 90 :
```

```
    print("합격입니다.")
```

```
    print("장학금도 받을 수 있습니다.")
```

블록: 여러 문장들을 묶은 것이다.

Lab: 영화 나이 제한 검사

나이를 입력하시오: 19
이 영화를 보실 수 있습니다.

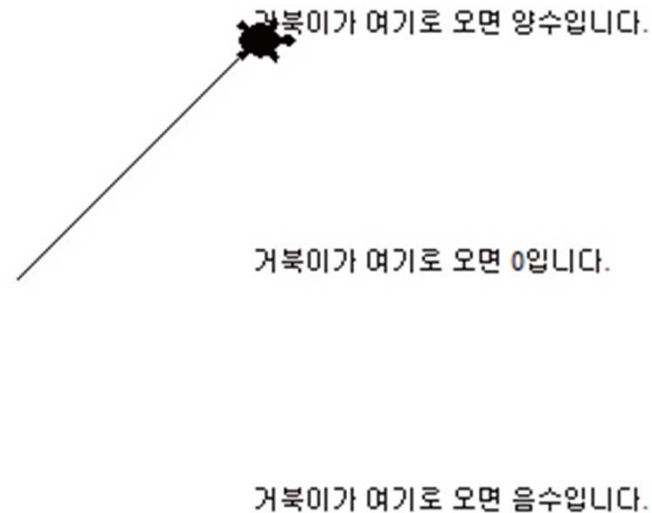
나이를 입력하시오: 14
이 영화를 보실 수 없습니다.

Solution

```
age = int(input("나이를 입력하시오: "))  
if age >= 15:  
    print("이 영화를 보실 수 있습니다.")  
else:  
    print("이 영화를 보실 수 없습니다.")
```

Lab: 부호에 따라 거북이를 움직이자

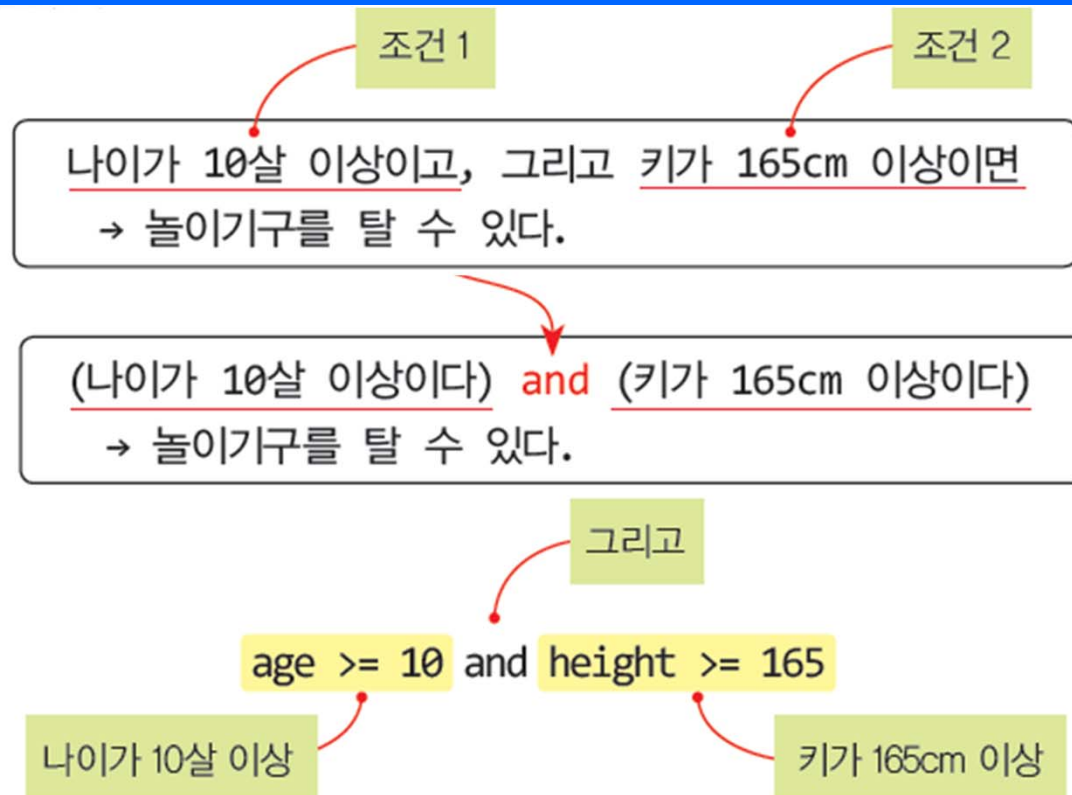
- 사용자로부터 정수를 받아서 정수의 부호에 따라서 거북이를 (100, 100), (100, 0), (100,-100)으로 움직이는 프로그램을 작성해보자.



Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.penup() # 펜을 올려서 그림이 그려지지 않게 한다.
t.goto(100, 100) # 거북이를 (100, 100)으로 이동시킨다.
t.write("거북이가 여기로 오면 양수입니다.")
t.goto(100, 0)
t.write("거북이가 여기로 오면 0입니다.")
t.goto(100, -100)
t.write("거북이가 여기로 오면 음수입니다.")
t.goto(0, 0) # (0, 0) 위치로 거북이를 이동시킨다.
t.pendown() # 펜을 내려서 그림이 그려지게 한다.
s = turtle.textinput("", "숫자를 입력하시오: ")
n=int(s)
if( n > 0 ):
    t.goto(100, 100)
if( n == 0 ):
    t.goto(100, 0)
if( n < 0 ):
    t.goto(100, -100)
```

논리 연산자



연산	의미
x and y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
x or y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
not x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

Lab: 동전 던지기 게임

- 동전을 던지기 게임을 작성해 보자.
- import random한 후에 random.randrange(2) 하면 0이나 1을 랜덤하게 생성할 수 있다.

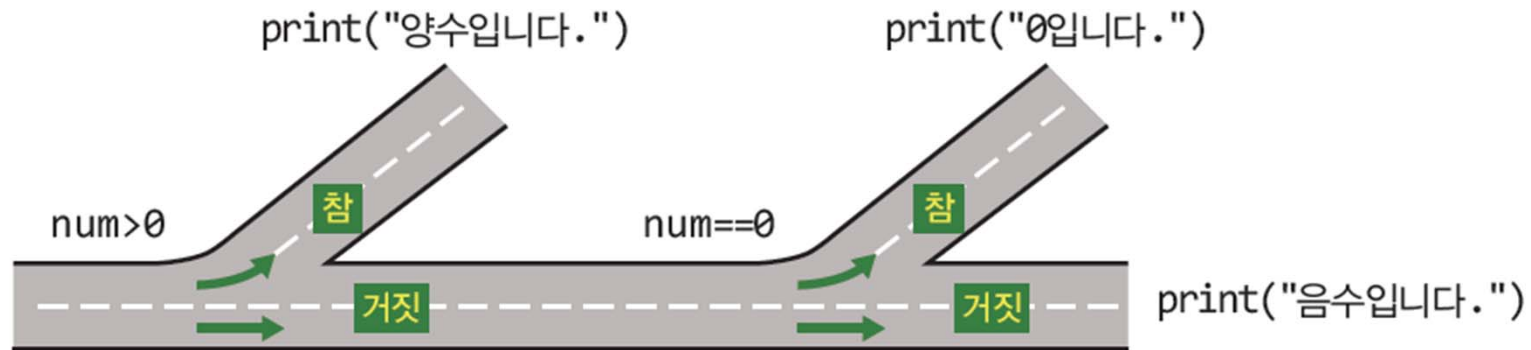
동전 던지기 게임을 시작합니다.
뒷면입니다.
게임이 종료되었습니다.

```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2)
if coin == 0 :
    print("앞면입니다.")
else :
    print("뒷면입니다.")
print("게임이 종료되었습니다.")
```

조건을 연속하여 검사

- 다음과 같이 진행하는 코드를 작성하려면?



연속적인 if-else 문

```
num = int(input("정수를 입력하시오: "))  
  
if num > 0:  
    print("양수입니다.")  
elif num == 0:  
    print("0입니다.")  
else:  
    print("음수입니다.")
```

정수를 입력하시오: 10
양수입니다.

Lab: 종달새가 노래할까?

- 동물원에 있는 종달새가 다음과 같은 **2**가지 조건이 충족될 때 노래를 한다고 하자.
 - 오전 6시부터 오전 9시 사이
 - 날씨가 화창하다.

난수 이용

- 현재 시각을 난수로 생성하고 날씨도 [True, False] 중에서 랜덤하게 선택하자. 종달새가 노래를 부를 것인지, 조용히 있을 것인지를 판단해보자.

```
import random  
time = random.randint(1, 24)  
sunny = random.choice([True, False])
```

좋은 아침입니다. 지금 시각은 1시 입니다.
현재 날씨가 화창하지 않습니다.
종달새가 노래를 하지 않는다.

Solution

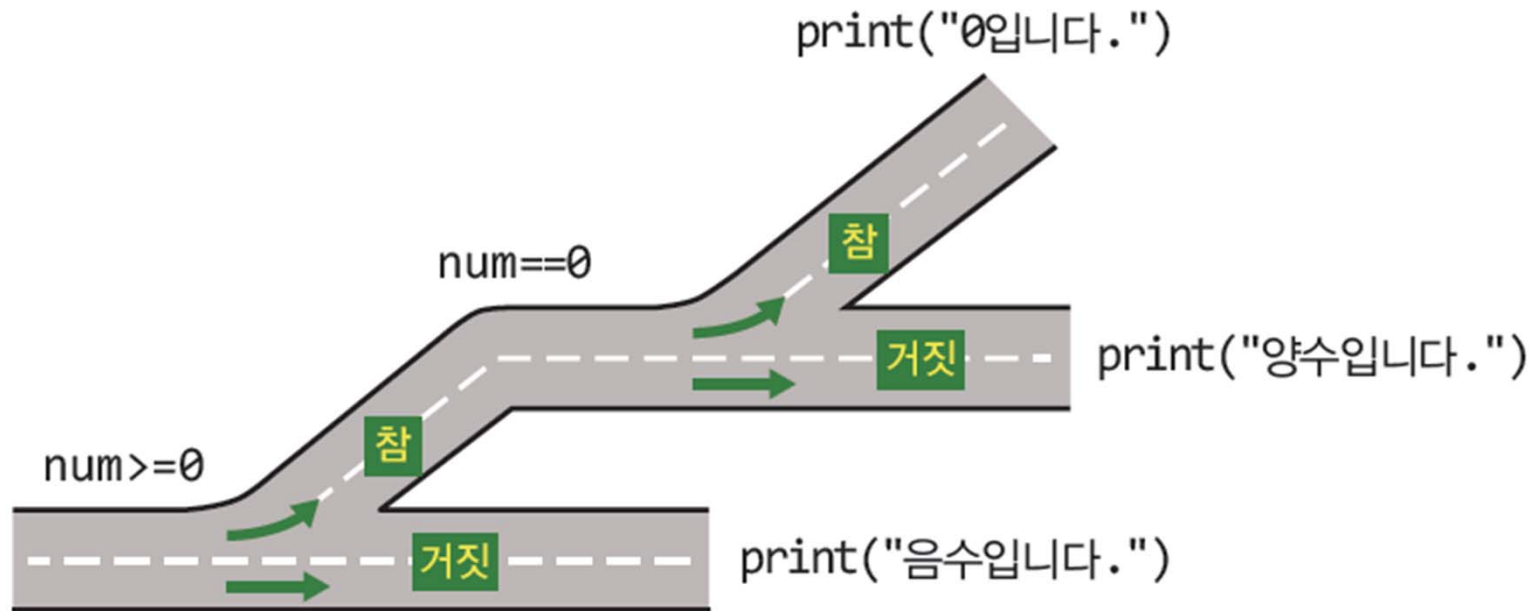
```
import random
time = random.randint(1, 24)
print("좋은 아침입니다. 지금 시각은 " + str(time) + "시 입니다.")

sunny = random.choice([True, False])
if sunny:
    print ("현재 날씨가 화창합니다. ")
else:
    print ("현재 날씨가 화창하지 않습니다. ")

# 종달새가 노래를 할 것인지를 판단해보자.
if time >= 6 and time < 9 and sunny:
    print ("종달새가 노래를 한다.")
else
    print ("종달새가 노래를 하지 않는다.")
```

중첩 if-else문

- if 문 안에 다른 if 문이 들어갈 수도 있다. 이것을 중첩 if 문이라고 한다.



예제

```
num = int(input("정수를 입력하시오: "))
if num >= 0:
    if num == 0:
        print("0입니다.")
    else:
        print("양수입니다.")
else:
    print("음수입니다.")
```

정수를 입력하시오: 10
양수입니다.

Lab: 로그인 프로그램

- 사용자로부터 아이디를 받아서 프로그램에 저장된 아이디와 일치하는지 여부를 출력하는 프로그램을 작성해보자.

아이디를 입력하시오: ilovepython
환영합니다.

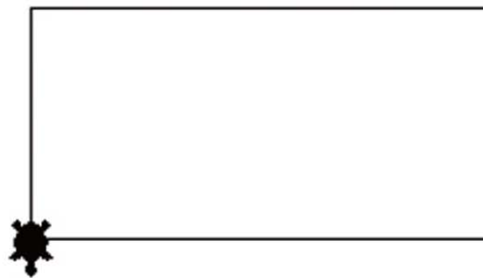
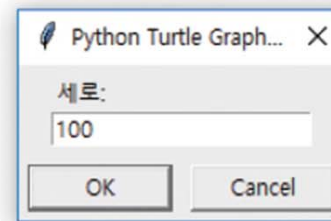
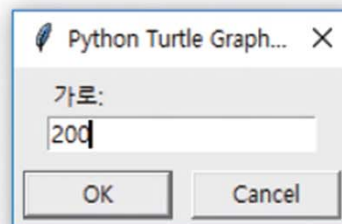
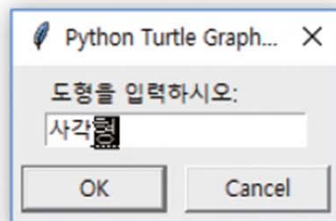
아이디를 입력하시오: iloveruby
아이디를 찾을 수 없습니다.

Solution

```
id = "ilovepython"
s = input("아이디를 입력하시오: ")
if s == id:
    print("환영합니다.")
else:
    print("아이디를 찾을 수 없습니다.")
```

Lab: 도형그리기

- 터틀 그래픽을 이용하여 사용자가 선택하는 도형을 화면에 그리는 프로그램 작성해보자.
- 도형은 “사각형”, “삼각형”, “원” 중의 하나이다. 각 도형의 치수는 사용자에게 물어보도록 하자.



Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

s = turtle.textinput("", "도형을 입력하시오: ")
if s == "사각형" :
    s = turtle.textinput("", "가로: ")
    w=int(s)
    s = turtle.textinput("", "세로: ")
    h=int(s)
    t.forward(w)
    t.left(90)
    t.forward(h)
    t.left(90)
    t.forward(w)
    t.left(90)
    t.forward(h)
```


문제

- `random.randint(1,10)`를 이용하여 임의의 숫자를 만들고 그 숫자를 찾아가는 프로그램을 작성하시오.

1~10:7

입력 값보다 작습니다.

답은 4 입니다.

- 먼저 아래를 실습하시오.

```
- import random  
- val=random.randint(1,10)  
- print(val)
```

Ch 3.

조건문

컴퓨팅 사고와 함께하는 파이썬 기초부터 인공지능 응용까지
(연두에디션)

3장 정리문제

■ 아래와 같은 커피 자동 주문 판매기를 만들어 보자

- Input() 함수 사용
- Print() 함수 사용
- 복합연산자 사용

- 커피 주문 판매기에서 아메리카노 2잔, 카페라테 2잔, 카푸치노 1잔을 주문하였다.
- 아메리카노 2500원, 카페라테 3000원, 카푸치노 3000원
- 현금 15000원을 투입
- 잔돈을 받았다.

문제 분석

■ 입력

- 각 커피 종류에 대한 잔의 수
- 사용자가 지불할 금액

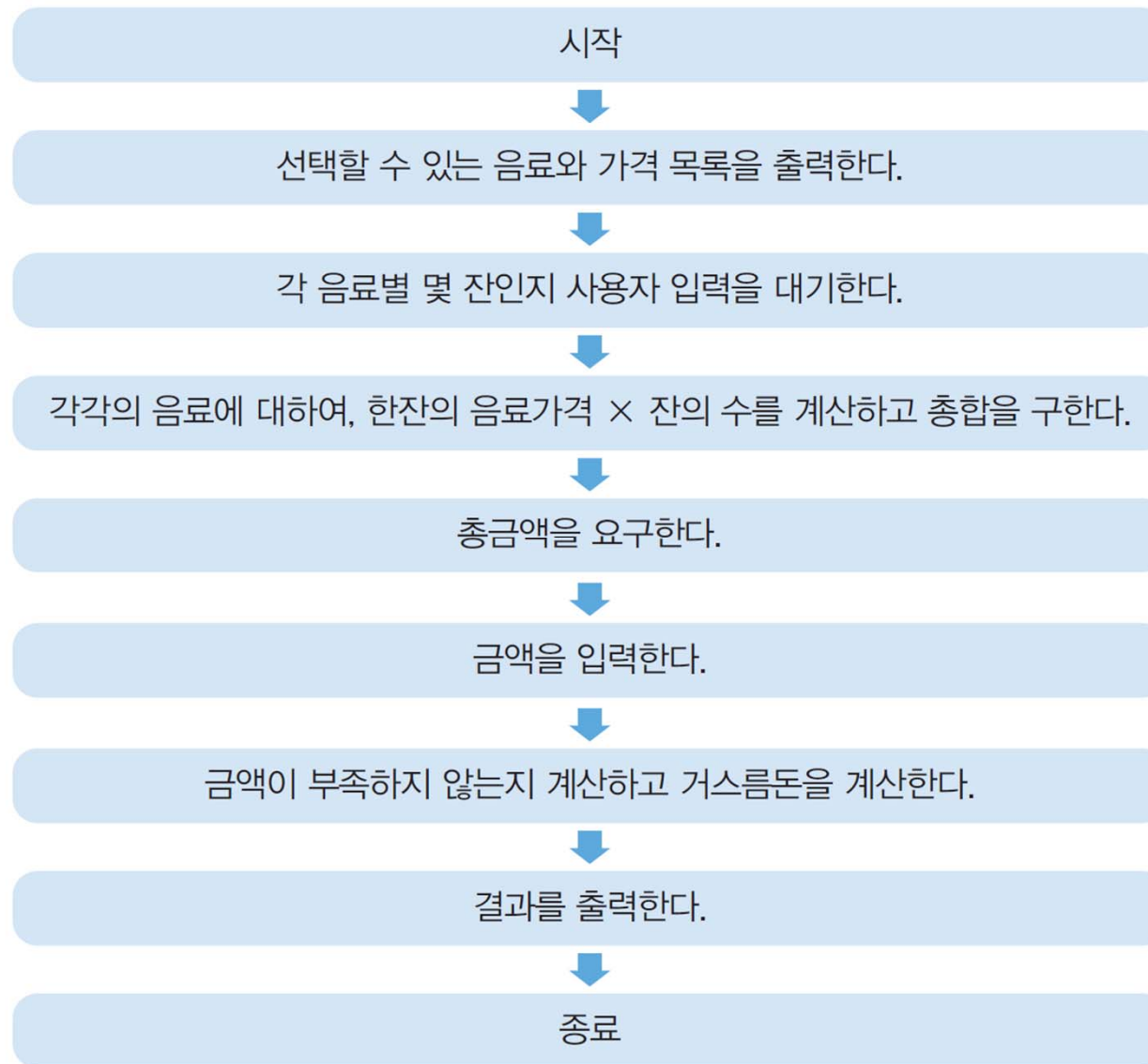
■ 출력

- 커피의 종류와 한잔 금액
- 주문한 총금액
- 거스름돈

■ 데이터

- 각 커피의 한잔 가격을 나타내는 변수
- 각 잔의 수를 나타내는 변수
- 총 금액을 나타내는 변수

생각하기



변수정하기

■ 변수

- noA=주문한 아메리카노에 대한 잔의 수
- noL=주문한 카페라테에 대한 잔의 수
- noC=주문한 카푸치노에 대한 잔의 수
- Sum=지불해야 할 총 금액
- Money=사용자가 지불한 금액

프로그래밍

- 함수 `input()`을 이용하여 사용자 입력을 받고, 받아들이는 입력 값이 문자열이므로 정수형 연산을 하기 위해 `int()` 함수를 적용하여 정수형 변수로 변환한다.

```
noA = int(input("아메리카노 개수 (잔) : "))  
noL = int(input("카페라테 개수 (잔) : "))  
noC = int(input("카푸치노 개수 (잔) : "))
```

프로그래밍

■ 총 주문 금액의 계산

```
sum = 0
sum += noA * 2500          # sum = sum + (noA * 2500)
sum += noL * 3000          # sum = sum + (noL * 3000)
sum += noC * 3000          # sum = sum + (noC * 3000)
```

■ 거스름돈 계산하기

```
print("총 금액은", sum, "원")
money = int(input("해당 금액을 입력하시오.>> "))
if money < sum:
    print("입력한 금액이 부족합니다.")
else:
    print("거스름돈은", money - sum, "원입니다")
```


프로그래밍

■ 완성

```
01  print("주문할 음료를 말씀하세요.")
02  noA = int(input("아메리카노 개수 (잔) : "))
03  noL = int(input("카페라테 개수 (잔) : "))
04  noC = int(input("카푸치노 개수 (잔) : "))
05
06  sum = 0
07  sum += noA * 2500
08  sum += noL * 3000
09  sum += noC * 3000
10
11  print("총 금액은 :", sum, "원")
12
13  money = int(input("지불하실 금액을 입력하세요. >> "))
14  if money < sum:
15      print("금액이 부족합니다.")
16  else:
17      print("거스름돈은", money - sum, "원 입니다")
```