

14장

더 복잡한 의사 결정 내리기

14장 더 복잡한 의사 결정 내리기

14.1 여러 조건 조합하기

14.2 코드의 어떤 부분을 실행할지 선택하기

14.3 요약

14.1 여러 조건 조합하기



14.1 여러 조건 조합하기

» 두 문장으로 만들 수 있는 여러 문장의 참/거짓을 나타낸 진리표

문장 1 (예: 비가 온다)	연결에 쓴 단어 (〈그리고(and)〉, 〈또는(or)〉, 〈아님(not)〉)	문장 2 (예: 나는 배가 고프다)	결과 (예: 비가 온다. 〈연결단어(영어 단어)〉 나는 배가 고프다)
〈.〉True	〈그리고(and)〉	True	True
True	〈그리고(and)〉	False	False
False	〈그리고(and)〉	True	False
False	〈그리고(and)〉	False	False
True	〈또는(or)〉	True	True
True	〈또는(or)〉	False	True
False	〈또는(or)〉	True	True
False	〈또는(or)〉	False	False
(첫 문장 없음)	〈아님(not)〉	True	False
(첫 문장 없음)	〈아님(not)〉	False	True

표 14-1 두 문장을 ‘그리고’와 ‘또는’으로 연결하는 경우의 진리표



14.1 여러 조건 조합하기

- » 프로그래밍에서는 if 문 하나에 여러 조건식을 조합해서 쓸 수 있다.
- » 여러 조건식을 조합할 수 있으면 매 조건마다 if 조건문을 만들지 않아도 된다.
따라서 코드가 읽기 쉽고 이해하기 쉽게 깔끔하다.

» 셀프체크 14.1



14.1.1 조건은 참/거짓 값으로 계산되는 조건식

» 둘 이상의 식을 and나 or로 조합하면서 다음 코드와 같이 둘 이상의 식을 조합해 if 문을 만들 수 있다.

```
if num_a < 0 and num_b < 0:
    print("둘 다 음수")
```

코드 14-1 한 if 문 안에 여러 조건식 사용하기

14.1.2 연산자 우선순위 규칙

» 프로그래밍에서도 수학과 마찬가지로 우선순위가 존재한다.

표 14-2

연산의 순서. 위쪽에 있는 연산일수록 더 먼저 실행된다. 같은 우선순위에 있는 연산은 왼쪽 결합(left-associative)이다. 즉, 식 안에 우선순위가 같은 연산이 둘 이상 연속으로 쓰이면 왼쪽부터 오른쪽 순서로 계산한다

연산자	의미
()	괄호
**	제곱
*	곱셈
/	나눗셈
//	버림 나눗셈
%	나머지(모듈로)
+	덧셈
-	뺄셈
==	같다
!=	같지 않다
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다
is	두 객체가 같다(정체성 검사)
is not	두 객체가 같지 않다(정체성 검사)
in	객체가 다른 객체에 들어있다(멤버십 검사)
not in	객체가 다른 객체에 들어있지 않다(멤버십 검사)
not	논리 부정
and	논리 곱(AND)
or	논리 합(OR)



14.1.2 연산자 우선순위 규칙

» 셀프 체크 14.2



14.1.2 연산자 우선순위 규칙

» 의미를 착각하기 쉬운 코드 : `num_a < 0 and num_b < 0` 대신 `num_a and num_b < 0`라고 되어 있다.

```
if num_a and num_b < 0:
    print("둘 다 음수")
```

코드 14-2 의미를 착각하기 쉬운 코드

num_a	num_b	콘솔 출력
-1	-1	둘 다 음수
-1	1	
0	-1	
0	1	
1	-1	둘 다 음수
1	1	

표 14-3

코드 14-2를 실행하면서 num_a와 num_b에 여러 다른 값을 넣은 경우 콘솔 출력 결과

파이썬에서는 0이 아닌 모든 정수를 True로 취급하며, 0은 False로 취급한다.

따라서 if -1은 if True처럼, if 0은 if False처럼 해석된다.

num_a = 1이고 num_b = -1인 경우 (`num_a and (num_b < 0)`)라는 식이 (`1 and (-1 < 0)`)으로 계산되고, 이 식은 결국 True and True와 같기 때문에 최종결과는 True다.

14.2 코드의 어떤 부분을 실행할지 선택하기



14.2.1 이 일을 하거나 저 일을 하거나

» if-elif-else 조건문

```
num = int(input("수를 하나 입력하세요: ")) ---- 사용자 입력
if num > 0: ---- 입력한 수가 0보다 큰지 조건 검사
    print("양수를 입력했습니다") ---- 메시지 출력
elif num < 0: ---- num > 0이 False면 이 조건에서 입력한 수가 0보다 작은지 검사
    print("음수를 입력했습니다") ---- 메시지 출력
else: ---- elif num < 0도 False면 else가 나머지 모든 경우를 한꺼번에 처리
    print("영을 입력했습니다") ---- 메시지 출력
```

코드 14-3 간단한 if-elif-else 조건문 예제

- 조건문은 if로 시작한다. if 문 뒤에는 elif나 else가 올 수 있다.
- 이런 구조는 여러 조건 중에서 첫 번째로 참이 되는 조건에 속한 코드 블록을 실행한다.

14.2.1 이 일을 하거나 저 일을 하거나

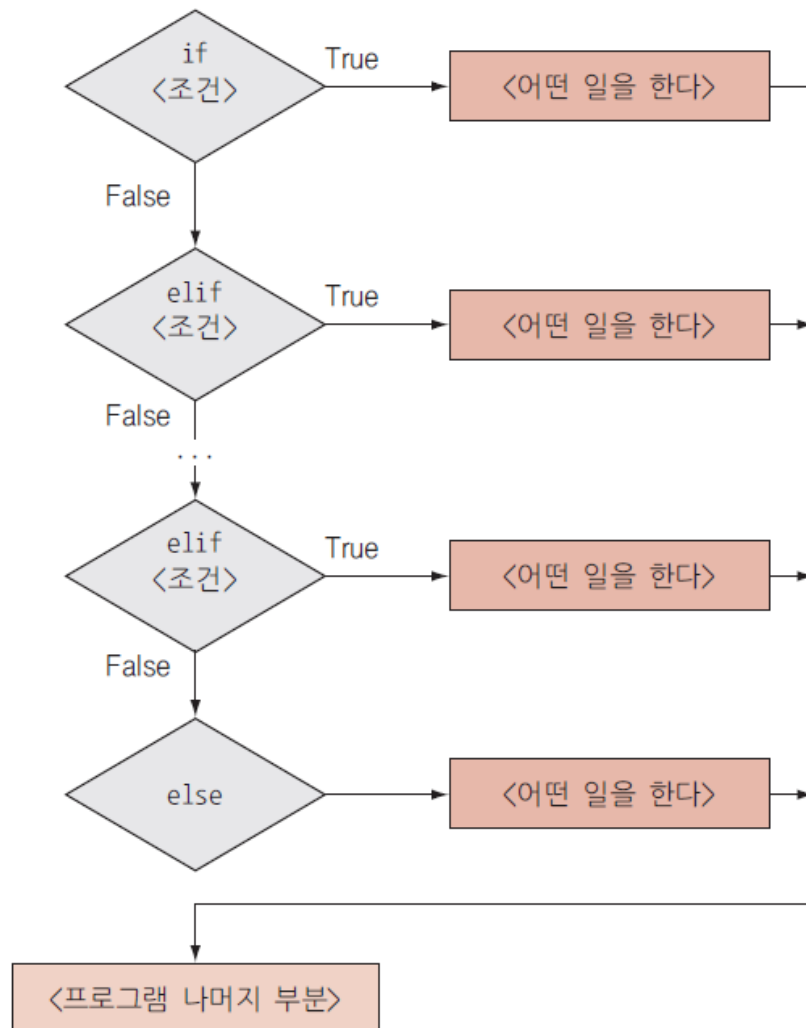


그림 14-1
일반적인 if-elif-else 코드 블록을 시각화한 그림.
프로그램의 주 흐름에서 최대 한 번만 <어떤 일을 한다> 부분으로
분기할 수 있다.
elif 블록은 없거나 1개만 있거나 여러 개 있을 수 있고,
else 블록은 있어도 되고 없어도 된다



14.2.1 이 일을 하거나 저 일을 하거나

» 셀프 체크 14.6

14.2.2 함께 사용하기

» 입력을 검사하는 간단한 프로그램이다.

- 사용자가 두 입력 중 어느 한쪽이라도 정수가 아닌 값을 넣으면 프로그램은 사용자에게 정해진 메시지를 표시하고, 들어쓰기되지 않은 다음 if 문으로 진행된다.

```
num_a = 5
num_b = 7
lucky_num = 7
if type(num_a) != int or type(num_b) != int:
    print("정수가 아닌 값을 입력했습니다")
else:
    if num_a > 0 and num_b > 0:
        print("두 정수 모두 양수입니다")
    elif num_a < 0 and num_b < 0:
        print("두 정수 모두 음수입니다")
    else:
        print("두 수의 부호가 서로 반대입니다")
if num_a == lucky_num or num_b == lucky_num:
    print("게다가 행운의 수를 맞추셨습니다!")
else:
    print("마음에 둔 수가 하나 있는데요...")
```

내포된 if-elif-else 그룹

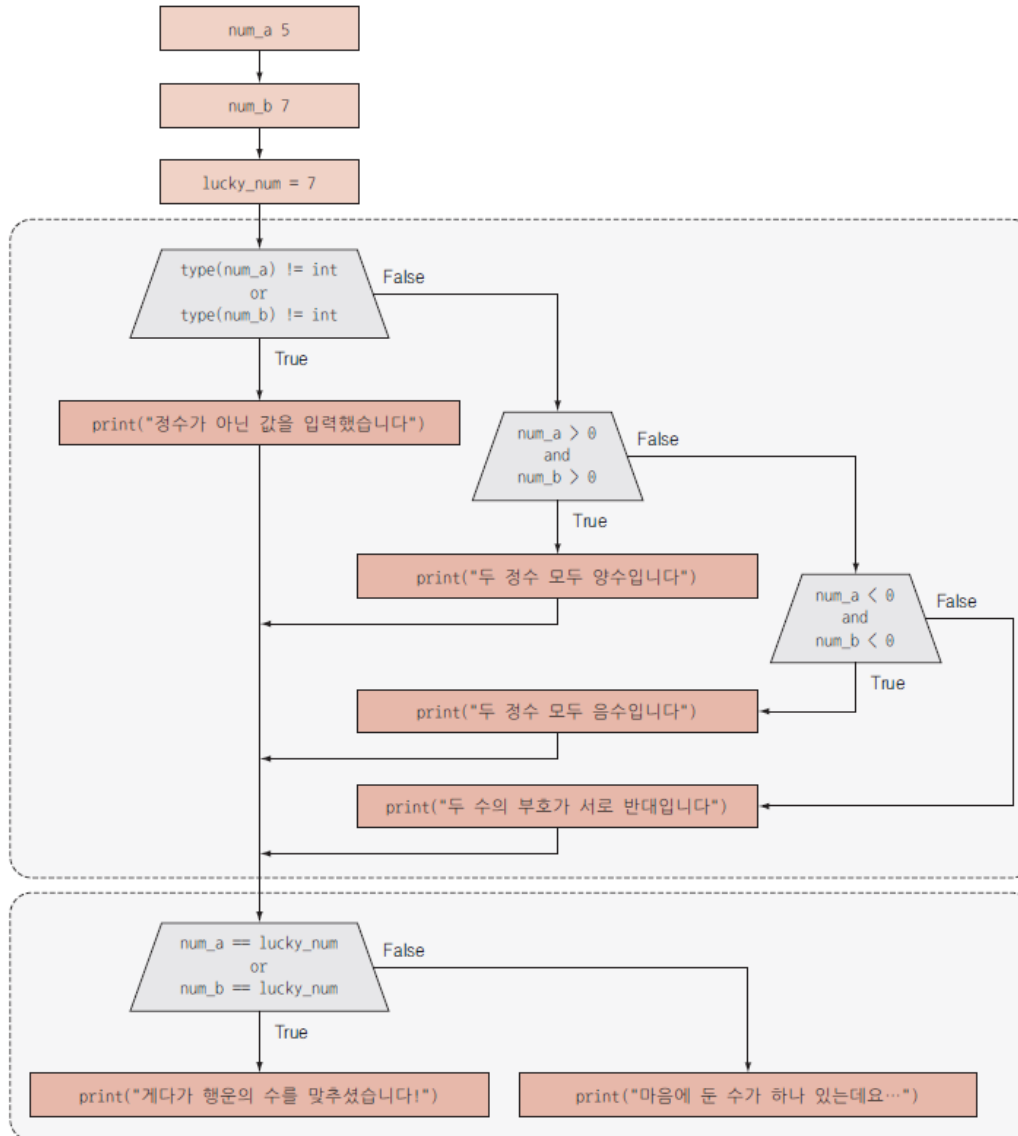
다른 if-else 그룹

내부에 if-elif-else 문을 포함하는 if-else 그룹

코드 14-5
if-elif-else 문의 예



14.2.2 함께 사용하기



코드 14-2
코드 14-5을 시각화해서 조건문 코드 블록을 보여
주는 그림

14.2.2 함께 사용하기

```
num_a = 5
num_b = 7
lucky_num = 7
if type(num_a) != int or type(num_b) != int:
    print("정수가 아닌 값을 입력했습니다")
elif num_a > 0 and num_b > 0:
    print("두 정수 모두 양수입니다")
elif num_a < 0 and num_b < 0:
    print("두 정수 모두 음수입니다")
else:
    print("두 수의 부호가 서로 반대입니다")
if num_a == lucky_num or num_b == lucky_num:
    print("게다가 행운의 수를 맞추셨습니다!")
else:
    print("마음에 둔 수가 하나 있는데요...")
```

코드 14-5의 else 블록을 여러 elif 블록으로 바꿨다

14.2.3 코드 블록 측면에서 생각하기

```
greeting = input("영어나 스페인어로 인사말을 입력하세요! ")
greet_en = ("hi", "Hi", "hello", "Hello")
greet_sp = ("hola", "Hola")
if greeting not in greet_en and greeting not in greet_sp: ---- if-elif로 구성된 코드 블록
    print("어떤 인사인지 이해할 수가 없네요.")
elif greeting in greet_en: ---- if-elif로 구성된 코드 블록
    num = int(input("1이나 2를 입력해주세요: "))
    print("영어를 하시는군요!")
    if num == 1: ---- if-elif 블록을 포함하는 내포된 블록
        print("one")

    elif num == 2: ---- if-elif 블록을 포함하는 내포된 블록
        print("two")
elif greeting in greet_sp: ---- if-elif로 구성된 코드 블록
    num = int(input("1이나 2를 입력해주세요: "))
    print("스페인어를 하시는군요!")
    if num == 1: ---- 다른 if-elif 블록을 포함하는 내포된 블록
        print("uno")
    elif num == 2:
        print("dos") ---- 다른 if-elif 블록을 포함하는 내포된 블록
```

코드 14-7

여러 if-elif-else 코드 블록이 조합된 예제



14.3 요약

- » (Q14.1) 사용자에게 두 수를 입력 받는 프로그램을 작성하라. 프로그램은 두 수의 관계를 출력한다. 즉, '두 수가 같습니다', '첫 번째 수가 두 번째 수보다 작습니다', '첫 번째 수가 두 번째 수보다 큼니다' 중 하나다.

- » (Q14.2) 사용자로부터 영어 문자열을 읽어 들이는 프로그램을 작성하라. 문자열에 영어 모음(a, e, i, o, u)이 하나 이상 들어 있다면 '모든 모음이 들어 있습니다!'라고 표시하라. 추가로 문자열이 a로 시작하고 z로 끝나면 '알파벳순입니다!'라고 출력하라.