

13장

프로그램에 의사 결정 도입하기

13장 프로그램에 의사 결정 도입하기

13.1 조건문으로 의사 결정 내리기

13.2 코드로 의사 결정 내리기

13.3 프로그램의 구조 잡기

13.4 요약

13.1 조건문으로 의사 결정 내리기



13.1.1 예/아니오를 묻는 질문과 참/거짓인 문장

» 불 값으로 계산되는 식들을 불 식(boolean expression)이라고 부르고, 불 식이 들어 있는 문장을 조건문(conditional statement)이라고 부른다.

» 셀프 체크 13.1

» 셀프 체크 13.2

13.2 코드로 의사 결정 내리기



13.2 코드로 의사 결정 내리기

» 파이썬에는 정해진 예약어(reserved word)나 키워드(keyword)가 있다.

- 키워드 : 파이썬 프로그래밍언어에서 특별한 의미를 가지는 단어
- 예약어 : 파이썬 프로그램 안에서 변수나 함수 이름으로 쓸 수 없는 단어
- 키워드는 거의 대부분 예약어이기도 하기 때문에 이 둘을 서로 혼용하곤 한다.

» 파이썬에서 의사 결정을 내릴 때 쓸 수 있는 조건문 중 가장 간단한 문장인 if 조건문을 만들 때 if 예약어/키워드를 사용한다.

13.2.1 의사 결정을 코드화하기: 예제

» 간단한 조건문 코드

```
num = int(input("Enter a number: "))
if num > 0:
    print("num is positive")
print("finished comparing num to 0")
```

사용자 입력을 기다렸다가 사용자가 입력한 값을
num이라는 변수에 저장

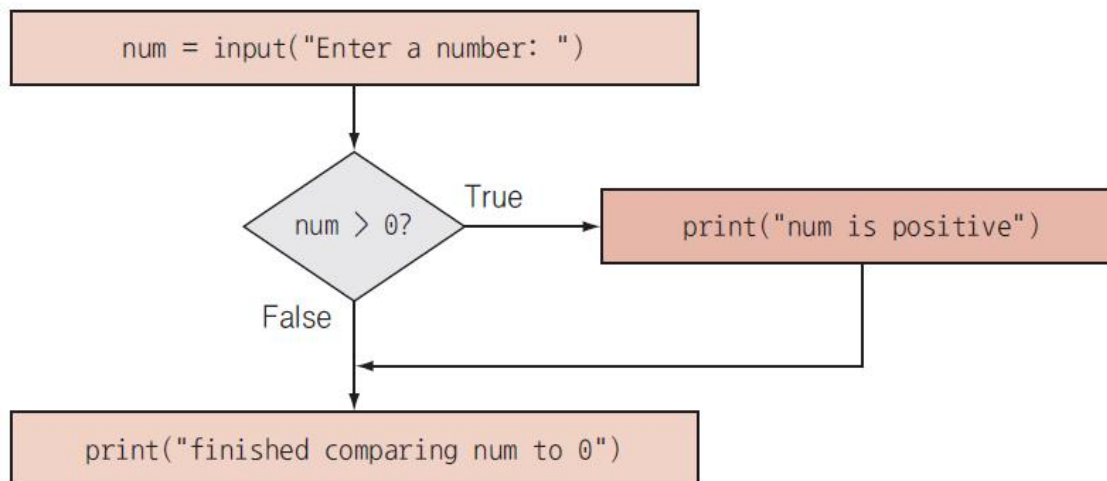
if num > 0: ---- num에 저장된 값이 0보다 큰지 검사하는 if 문

print("num is positive") ---- num이 0보다 크다면 들여쓰기한 블록 안으로 들어가서 블록 안에 있는 문장을 실행

print("finished comparing num to 0") ----

num이 0과 같거나 0보다 작다면 들여쓰기한 블록 안에
들어가지 않고 이 줄을 바로 출력

코드 13-1 간단한 조건식 예



조건에 관계없이 항상 실행



13.2.1 의사 결정을 코드화하기

» 셀프 체크 13.3

13.2.2 의사 결정을 코드화하기:일반적인 방법

» 조건문 형식

〈if 조건문 앞의 코드〉

if 〈조건식〉: ---- 'if' 키워드는 조건문을 시작하며, 'if' 뒤에 조건식이 옴

〈어떤 일을 수행하는 코드〉 ---- 들여쓰기는 if 문의 조건이 참인

〈if 조건문 뒤의 코드〉

경우에만 실행되는 코드를 표현함

〈if 조건문 앞의 코드〉는 조건을 검사하기 전에 수행되고, 〈if 조건문 뒤의 코드〉는 조건문을 다 실행한 다음에 실행됨

코드 13-2

간단한 if 조건문을 작성하는 일반적인 방법

» 셀프 체크 13.4

13.3 프로그램의 구조 잡기



13.3 프로그램의 구조 잡기

- » 조건문은 프로그램 흐름을 둘로 나눈다. 조건문을 통해 프로그램에서 의사 결정을 내릴 수 있다.
- » 코드 중 일부를 들여 쓸 수 있다. 파이썬 인터프리터는 들여쓰기한 블록의 바로 위나 바로아래에 있는 문장과 들여쓰기 한 블록이 서로 특별한 연관 관계가 있음을 구분할 수 있다.
- » 지금까지는 프로그램의 모든 줄을 순서대로 실행했다. 이제부터는 어떤 조건을 만족하는 경우에만 특정 코드 블록을 실행하게 만들 수 있다.



13.3.1 조건문 여러 개 사용하기

» if 문을 여러 개 차례대로 사용하면 조건을 조합할 수 있다. 매 if 문을 만날 때마다 그 if 문에 속한 코드를 실행할지 결정한다.

```
num_a = int(input("Pick a number: ")) ---- 사용자 입력 받기
if num_a > 0: ---- 입력받은 수가 0보다 큰지 검사
    print("Your number is positive") ---- 앞의 조건이 참인 경우에만 이 문장을 실행
if num_a < 0: ---- 입력받은 수가 0보다 작은지 검사
    print("Your number is negative ") ---- 앞의 조건이 참인 경우에만 이 문장을 실행
if num_a == 0: ---- 입력받은 수가 0과 같은지 검사
    print("Your number is zero") ---- 앞의 조건이 참인 경우에만 이 문장을 실행
print("Finished!") ---- 조건과 관계 없이 이 문장을 실행
```

코드 13-3 여러 조건문이 순서대로 나열된 코드



13.3.2 의사 결정의 결과에 따라 새로운 의사 결정 내리기

» 조건문 내포(nesting)

<pre># 내포가 있는 코드 num_a = int(input("수를 하나 입력하세요: ")) num_b = int(input("수를 하나 입력하세요: ")) if num_a < 0: ----- 첫 번째 조건문 ----- print("num_a는 음수입니다") if num_b < 0: ----- 두 번째 조건문 ----- print("num_b는 음수입니다") ----- 실행할 문장 ----- print("끝")</pre>	<pre># 내포가 없는 코드 num_a = int(input("수를 하나 입력하세요: ")) num_b = int(input("수를 하나 입력하세요: ")) if num_a < 0: print("num_a는 음수입니다") if num_b < 0: print("num_b는 음수입니다") print("끝")</pre>
--	---

코드 13-4 내포 조건문과 연속적인 조건문의 차이점 비교

- **내포 조건문이 있는 코드**에서는 바깥쪽 조건문(`if num_a < 0`)이 참인 경우에만 내포된 조건문(`if num_b < 0`)이 실행된다. 더 나아가 내포된 조건문에 속한 코드 블록(`print("num_b는 음수입니다")`)은 두 조건문이 모두 참인 경우에만 실행된다.
- **내포가 없는 코드**에서는 프로그램이 실행될 때마다 두 번째 조건문(`if num_b < 0`)도 실행된다. 따라서 그 조건문에 속한 코드 블록(`print("num_b는 음수입니다")`)은 `num_a`의 값이 어떤 값이든 상관 없이 `num_b`가 0보다 작기만 하면 실행된다.



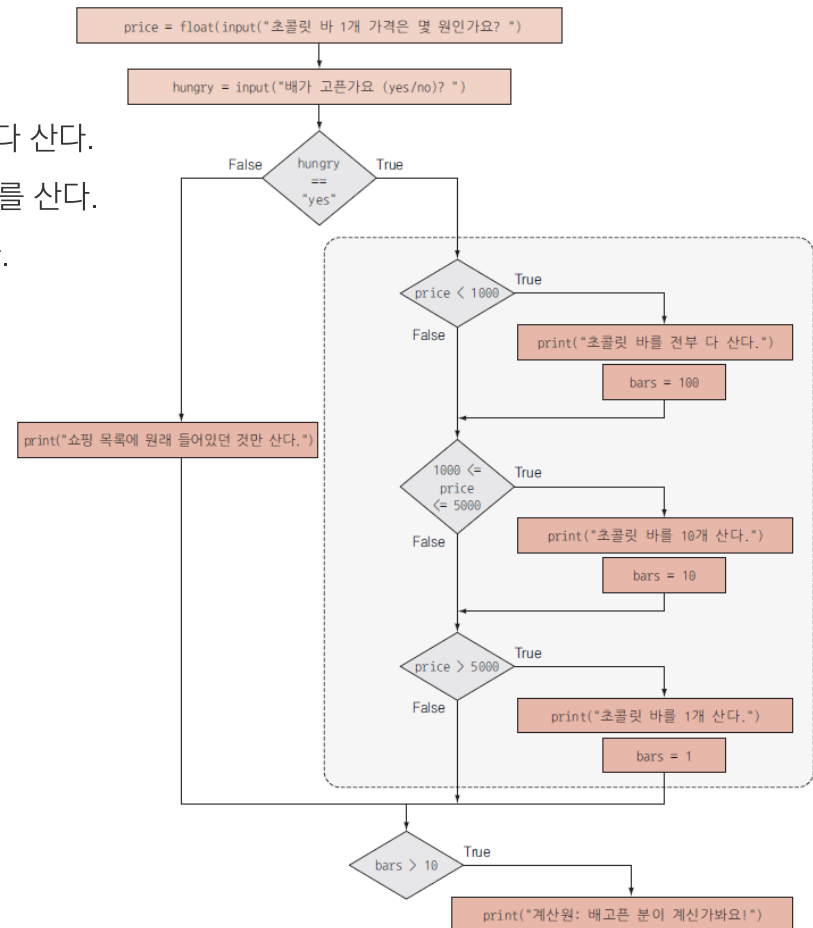
13.3.2 의사 결정의 결과에 따라 새로운 의사 결정 내리기

» 셀프 체크 13.6

13.3.3 내포 조건문을 사용하는 더 복잡한 예제

- » 초콜릿 바를 몇 개 사면 될지 결정할 수 있게 도와주는 프로그램을 작성하기
- » 프로그램은 흐름도에 있는 단계대로 여러분을 도울 것이다.

- 사용자가 배가 고프는지 물어본다.
- 초콜릿 바 하나의 가격이 얼마인지 물어본다.
- 배가 고프고, 초콜릿 바 하나가 1,000원 미만이면 초콜릿 바를 전부 다 산다.
- 배가 고프고, 초콜릿 바 하나 가격이 1,000원~5,000원 사이면 10개를 산다.
- 배가 고프고, 초콜릿 바 하나 가격이 5,000원을 초과하면 1개만 산다.
- 배가 고프지 않다면 초콜릿 바를 사지 않는다.
- 사용자가 사는 초콜릿 바 개수에 따라서 계산원이 한 마디 한다



13.3.3 내포 조건문을 사용하는 더 복잡한 예제

```
price = float(input("초콜릿 바 1개 가격은 몇 원인가요? "))
hungry = input("배가 고픈가요 (yes/no)? ")

bars = 0

if hungry == "yes":
    if price < 1000:
        print("초콜릿 바를 전부 다 산다.")
        bars = 100
    if 1000 <= price <= 5000:
        print("초콜릿 바를 전부 다 산다.")
        bars = 10
    if price > 5000:
        print("초콜릿 바를 1개 산다.")
        bars = 1

if hungry == "no":
    print("쇼핑 목록에 원래 들어 있던 것만 산다.")

if bars > 10:
    print("계산원: 배고픈 분이 계신가요!")
```

사용자 입력 받기

배고픈지 결정하기 위한 조건문

초콜릿 바 가격이 1,000원 미만인지 결정하기 위한 조건문

초콜릿 바가 1,000원 미만일 때 수행할 동작

초콜릿 바 가격이 1,000원 이상 5,000원 이하인지 결정하기 위한 조건문. 초콜릿 바가 1,000원 이상 5,000원 이하일 때 수행할 동작

초콜릿 바 가격이 5,000원 초과인지 결정하기 위한 조건문. 초콜릿 바가 5,000원 초과일 때 수행할 동작

사용자가 배고프지 않아서 'no'라고 답했을 때 적절한 문장을 출력하기 위한 조건문

초콜릿 바를 10개 초과해서 산 경우에만 메시지를 출력하기 위한 조건문

코드 13-5
초콜릿 바를 몇 개 살지
결정하는 조건문



13.4 요약

- » (Q13.3) 사용자에게 문자열을 입력 받아서 그 문자열에 공백이 하나라도 있으면 ‘입력한 문자열에 공백이 있습니다’라고 출력하라.

- » (Q13.4) ‘내가 생각한 정수를 맞춰보세요!’라고 출력하고, 생각한 정수(비밀 값)는 변수에 저장하라. 사용자가 변수에 있는 정수보다 더 작은 수를 입력하면 ‘너무 작아요’라고 출력하라. 사용자가 추측한 값이 비밀 값보다 더 크다면 ‘너무 커요’라고 출력하라. 마지막으로 사용자가 비밀 값과 같은 값의 정수를 입력하면 ‘정답입니다!’라고 출력하라.