



CHAPTER 6

커피 자동 주문기



학습 목표

기본 학습 목표

- `input()` 함수를 적절하게 사용할 수 있다.
- `print()` 함수를 적절하게 사용할 수 있다.
- 축약 치환 연산자인 `+=`를 사용할 수 있다.

심화 학습 목표

- 리스트를 이용하여 산술 연산을 수행할 수 있다.



핵심 학습 요소

핵심 학습 요소

- 정수형 변수, if 문, input() 함수, int() 함수, print() 함수

문제 상황



문제 상황

CT

- 친구들과 현금 커피 자동 주문기 앞에서 아메리카노 2잔과 카페라테 2잔, 카푸치노 1잔을 주문하였다. 아메리카노는 한잔에 2500원이고, 카페라테는 한잔에 3000원, 카푸치노는 한잔에 3000원이다. 커피 종류와 잔 수를 결정한 후, 현금 15000원을 투입하고 잔돈 1000원을 받았다. 커피 자동 주문기가 어떻게 커피 총 금액을 계산하고 잔돈을 돌려주는지 궁금하다.



문제 분석

✧ 컴퓨터의 역할

컴퓨터로 커피 자동 주문기를 만들기 위해 다음과 같은 항목을 생각할 수 있다.

- 사용자로부터 커피의 종류와 잔 수를 주문받기
- 선택한 커피 정보를 이용하여 지불해야 할 총 금액을 계산하기
- 사용자로부터 돈을 입금 받으면, 잔돈 계산하고 결과 출력해 주기

✧ 사용자

- 사용자들이 주문을 할 수 있는 방법
- 커피 목록과 금액을 제시하는 방법
- 최종적으로 처리 결과를 제시하는 방법



문제 분석

✧ 입력

- 각 커피 종류에 대한 잔의 수
- 사용자가 지불할 금액

✧ 출력

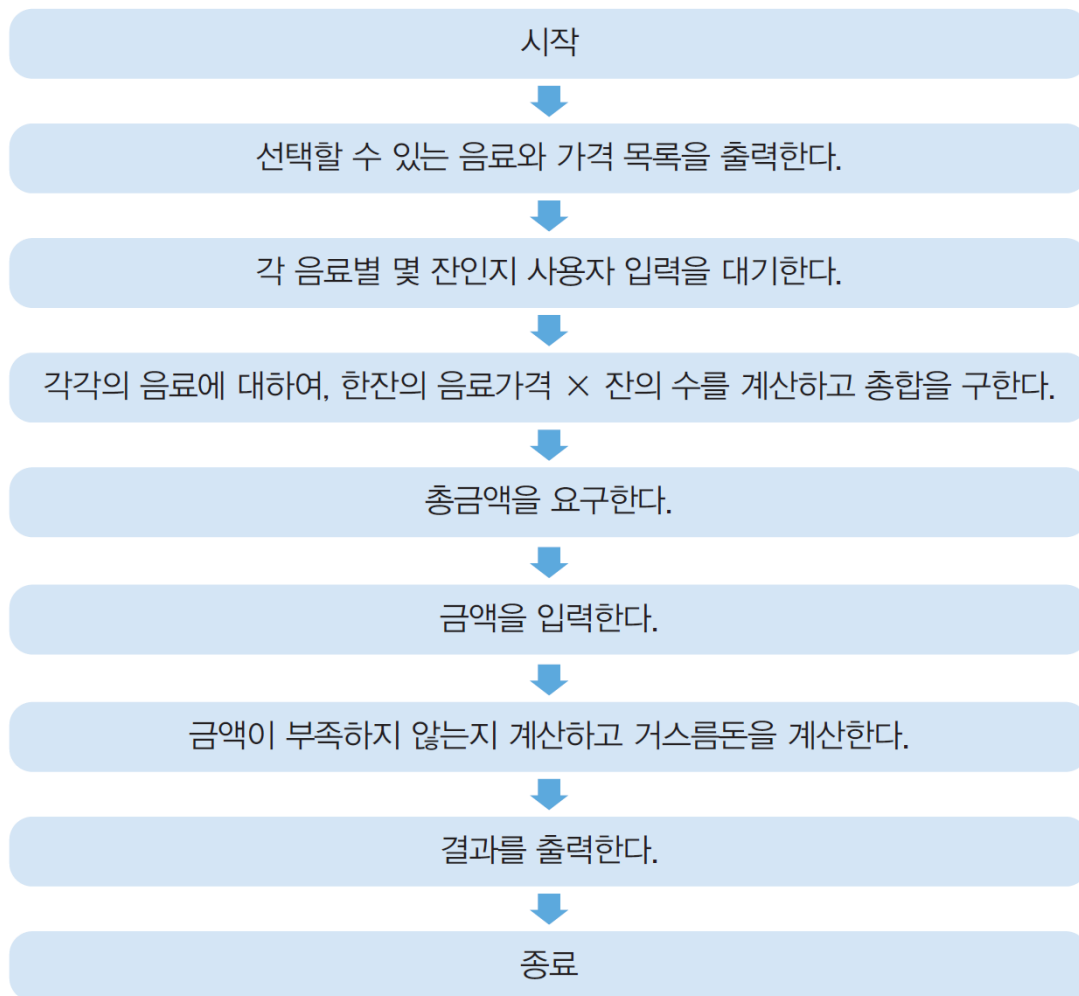
- 커피의 종류와 한잔 금액
- 주문한 총금액
- 거스름돈

✧ 데이터

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

알고리즘

☒ 전체 알고리즘 뼈대





알고리즘

❖ 알고리즘 효율성

- 시공간의 효율성을 위해서 문제 해결에서 반복문을 사용하지 않고 정해진 3개의 커 피 메뉴에 대한 기본적인 알고리즘을 작성한다.
- 반복문을 사용하면 알고리즘의 효율성이 무엇에 영향을 받을지 생각해보자.



프로그래밍

❖ 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 입력값이 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) : "))
```

- 축약치환연산자 `+=`

예제 | $a = a + b \Rightarrow a += b$



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)
```

✧ 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



문고 답하기

Q 문고 답하기

문 $a = 2$ 일 때, $a * = 3 + 4$ 는 얼마일까? $2 * 3 + 4$ 이므로 $(2 * 3) + 4 = 10$ 일까? 아니면, $2 * (3 + 4) = 14$ 일까?



프로그래밍

■ 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

- a가 10이고 b가 20.5일 때, a와 b를 출력하는 방식

```
a = 10, b = 20.5
```

- 코마(,)의 사용

```
print("a = ", a, ", b = ", b)
```

- % 기호의 사용

```
print("a = %d, b = %d" % (a, b))
```

- format() 함수의 사용

```
'a = {}, b = {}'.format(a, b)
```



문고 답하기

Q 문고 답하기

문 a = 2, b = 3일 때, print(a)를 수행한 후, print(b)를 했을 때, 2, 3으로 출력하려면, 어떻게 하면 좋을까? 즉, 3을 다음 줄에 출력하지 않고, 2 바로 옆으로 출력한다.



프로그래밍 – 전체 프로그램

```
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")
```




실습해보기

실습해보기 6-1

위 프로그램 02 ~ 04 줄까지, 입력값으로 숫자가 아닌 문자가 들어오면 제대로 처리하지 못한다. 변수 `noA`, `noL`, `noC`가 모두 정수값을 가질 때만 프로그램을 수행하도록 프로그램의 일부를 수정해보자. 이를 체크하려면 `isdigit()` 이라는 함수를 사용하면 된다. 예를 들어, `a = '20'`을 수행한 후, `a.isdigit()` 함수를 호출하면 `True`를 결괏값으로 출력한다.

테스트와 디버깅

입력값	출력 결과	확인 및 유의사항
<p>주문할 음료를 말씀하세요. 아메리카노 개수 (잔) : 2 카페라테 개수 (잔) : 2 카푸치노 개수 (잔) : 1</p>	<p>총 금액은 : 14000 원</p>	<ul style="list-style-type: none"> ■ 예상된 테스트 결과인지 확인한다. ■ 총 금액이 14000원이 아니면 변수 sum의 계산 수식이 제대로 되어 있는지 print() 함수 등을 이용해 디버깅한다.
<p>지불하실 금액을 입력하세요. » 10000</p>	<p>금액이 부족합니다.</p>	<ul style="list-style-type: none"> ■ 금액을 부족하게 입력하였는데 잔돈을 계산하면 잘못된 프로그램이므로 잔돈 계산하는 부분을 디버깅한다.
<p>지불하실 금액을 입력하세요. » 20000</p>	<p>거스름돈은 6000 원 입니다.</p>	<ul style="list-style-type: none"> ■ 충분한 금액이 입력되면, 잔돈을 맞게 계산하는지 확인하여 잘못된 계산이면 잔돈 계산하는 부분을 디버깅한다.

심화 활동



심화 활동

CT

다음과 같이 더 다양한 음료와 가격이 두 개의 리스트에 저장되어 있다고 할 때, 두 개의 리스트와 반복문을 이용하여 문제를 해결해보자.

```
blist = ['아메리카노', '카페라테', '카푸치노', '오렌지주스', '콜라', '자몽주스']  
plist = [2500, 3000, 3000, 4000, 1500, 4000]
```