

chapter 2

프로그램의 기본 재료

💖 아래와 같은 출력 결과가 나오도록 빈칸에 연산자를 넣어보세요.

```
>>> print(3   1   2)
2
>>> print(3   1   2)
4
>>> print(3   1   2)
1
```

💖 출력 결과가 다른 하나를 골라보세요.

1) `print(8 / 2)`

2) `print(2 * 2)`

3) `print(8.0 / 2)`

4) `print(8.5 // 2)`

💖 딸기 11개를 3명의 친구들에게 나눠주려고 합니다. 나눠주고 남은 딸기의 개수를 출력하는 코드로 올바른 것을 골라보세요.

1) `print(11 / 3)`

2) `print(11 % 3)`

3) `print(11 // 3)`

4) `print(11 ** 3)`

💖 다음 코드의 출력 결과를 적어보세요.

```
>>> print('2' + '4')
```

💖 다음 코드의 출력 결과로 올바른 것을 골라보세요.

```
>>> print('말랑' * 2)
```

- 1) 말랑2
- 2) 말랑 * 2
- 3) '말랑' * 2
- 4) 말랑말랑

💖 다음 코드의 출력 결과를 적어보세요.

```
>>> num1 = 9
>>> num2 = 2
>>> print(num1 // num2)

>>> print(num1 % num2)
```

❤ 다음 코드의 출력 결과를 적어보세요.

```
>>> count = 1
>>> count = count + 1
>>> count = 1
>>> print(count)
```


♥ 다음 코드의 출력 결과를 적어보세요.

```
>>> fruits = '자몽 레몬'  
>>> print(fruits + ' 멜론')
```

❤ 다음 중 변수의 이름으로 적합한 것을 모두 고르세요.

- 1) _fruits
- 2) 2nd_candy
- 3) candy bag
- 4) student_id

chapter 3

재료 모으기

♥ 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
>>>  
>>> print(nums)  
[1, 2, 3]
```

💖 다음 코드의 출력 결과를 적어보세요.

```
>>> fruits = []
>>> print(fruits)

>>> fruits.append('자몽')
>>> print(fruits)

>>> fruits.append('멜론')
>>> print(fruits)

>>> fruits.append('레몬')
>>> print(fruits)
```

♥ 다음 코드의 출력 결과를 적어보세요.

```
>>> fruits = ['자몽', '멜론', '레몬']  
>>> print(fruits[1])
```

💖 다음 코드의 실행 결과를 골라보세요.

```
>>> fruits = ['멜론', '레몬']  
>>> fruits[1] = '거봉'  
>>> print(fruits)
```

- 1) ['멜론', '레몬']
- 2) ['멜론', '거봉']
- 3) ['거봉', '레몬']
- 4) ['거봉', '멜론']

💖 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
>>> nums = [1, 3]
>>>
>>> nums.append(3)
>>>
>>> print(nums)
[1, 2, 3, 4]
```


💖 다음 코드의 출력 결과를 적어보세요.

```
>>> nums = [1, 2, 3, 4, 5]
>>> print(nums[2])

>>> print(nums[-3])
```

♥ 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
>>>  
>>> print(fruits)  
['자몽', '멜론', '레몬']  
>>>  
>>> print(fruits)  
['멜론', '레몬']
```

💖 다음 코드의 출력 결과를 적어보세요.

```
>>> nums = [1, 2, 3, 4, 5]
>>> print(nums[0:3])

>>> print(nums[2:5])
```

💖 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
>>>  
>>> print(fruits)  
['멜론', '거봉', '레몬']  
>>>  
>>> print(fruits)  
['거봉', '레몬', '멜론']
```

♥ 다음 코드의 출력 결과를 적어보세요.

```
>>> nums = [1, 1, 1, 2, 2, 3]
>>> print(nums.count(1))
```

chapter 4

횃수로 반복하기

♥ 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

01:

02:

`print(num)`

↳

3

1

2

💖 print()를 한번만 사용해서 아래 코드와 같은 결과가 나오게 하세요.

```
01: clovers = ['클로버1', '클로버2', '클로버3']  
02: print(clovers[0])  
03: print(clovers[1])  
04: print(clovers[2])
```

```
↳ 클로버1  
   클로버2  
   클로버3
```


💖 리스트의 정수를 차례로 가져와 아래 그림과 같은 3층짜리 별을 찍어보세요.

```
01: stars = [2, 1, 3]
```

```
02:
```

```
03:
```

```
↳ **
```

```
  *
```

```
***
```

💖 리스트에 저장되어 있는 모든 카드 번호의 평균을 구해서 출력하세요.

```
01: total = 0
02: card_nums = [1, 3, 6, 7]
03:
04:
05:
```

```
↳ 4.25
```

♥ 다음 코드의 실행 결과를 골라보세요.

```
01: clovers = ['클로버1', '클로버2', '클로버3']  
02: for idx in [2, 1, 0]:  
03: print(clovers[idx])
```

↳ 클로버3
클로버2
클로버1

↳ 클로버1
클로버2
클로버3

↳ `SyntaxError: expected an indented block`

↳ `NameError: name 'clovers' is not defined`

💖 다음 코드의 출력 결과를 적어보세요.

```
01: for num in range(2):  
02:     print(num)
```

↳

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01:  clovers = ['클로버1', '클로버2', '클로버3']  
02:  
03:      print(clovers[idx])
```

```
↳  클로버1  
    클로버2  
    클로버3
```

♥️ for와 range()로 아래 그림과 같은 3층짜리 별을 찍어보세요.

01:

02:

↳

*

**

♥ 다음 중 출력 결과가 다른 하나를 골라보세요.

```
01: total = 0
02: for num in [0, 1, 2, 3]:
03:     total = total + num
04: print(total)
```

```
01: total = 0
02: for num in range(4):
03:     total = total + num
04: print(total)
```

```
01: total = 0
02: for num in range(0, 5):
03:     total = total + num
04: print(total)
```

```
01: total = 0
02: for num in range(1, 4):
03:     total = total + num
04: print(total)
```

♥ 다음 코드의 출력 결과를 적어보세요.

```
01: nums = [0, 1, 2, 3, 4, 5]
02: for num in nums[1:3]:
03:     print(num)
```

↳

chapter 5

조건 판단하기

💖 다음 코드의 출력 결과를 적어보세요.

```
01: num1 = 55
02: num2 = 13
03: print(num1 <= num2)
04: print(num1 != num2)
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01: MVP = '하트잭'  
02: print(MVP == '공작부인')
```

↳

💖 출력 결과가 다른 하나를 골라보세요.

1) `print(2 > 5)`

2) `print(2 != 5)`

3) `print(False)`

4) `print(2 == 5)`

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: odd_nums = []  
02: for num in range(10):  
03:  
04:  
05:
```

```
↳ [1, 3, 5, 7, 9]
```

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: switch = '켜짐'
02: if          :
03:     print('조명이 켜졌어요.')
04: else:
05:     print('조명이 꺼졌어요.')
```

↳ 조명이 켜졌어요.

♥ 다음 코드의 출력 결과를 적어보세요.

```
01: price = 10000
02: if price > 10000:
03:     print('너무 비싸요.')
04: elif price > 5000:
05:     print('괜찮은 가격이네요.')
06: else:
07:     print('정말 싸요.')
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01: input_number = -9
02: if input_number < 0:
03:     absolute_value = input_number * -1
04: else:
05:     absolute_value = input_number
06: print(absolute_value)
```

↳

♥ 다음 코드의 출력 결과를 적어보세요.

```
01: total_price = 0
02: choices = ['버섯스프', '당근주스', '벌꿀파이']
03: for choice in choices:
04:     if choice == '버섯스프':
05:         total_price = total_price + 8000
06:     elif choice == '당근주스':
07:         total_price = total_price + 4500
08:     elif choice == '벌꿀파이':
09:         total_price = total_price + 6000
10: print('총 주문금액은', total_price, '원입니다.')
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: print(clovers)
03: for clover in clovers:
04:     if clover == '클로버1':
05:         print(clover, '안녕!')
06:     elif clover == '클로버2':
07:         print(clover, '반가워!')
08:     elif clover == '클로버3':
09:         print(clover, '잘가!')
10:     else:
11:         print('넌 누구니?')
```

↳

💖 아래 조건을 참고해 윤년을 계산하는 코드를 작성하세요.

- 연도가 400으로 나누어 떨어지면 윤년입니다.
- 연수가 4로 나누어 떨어지면서 100으로 나누어 떨어지지 않으면 윤년입니다.

```
01: year = 2016
```

```
02:
```

```
03:
```

```
04:
```

```
05:
```

```
06:
```

```
07:
```

```
↳ 2016 년은 윤년입니다.
```

chapter 6

조건으로 반복하기

💖 다음 코드의 출력 결과를 적어보세요.

```
01: count = 0
02: while count < 3:
03:     print(count)
04:     count = count + 1
```

↳

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: count = 1
02:
03:     count = count + 1
04:     print(count)
```

```
↳ 2
   3
   4
```

♥ 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: count = 0
02: while count <= 5:
03:
04:
05:
```

```
↳ 1
   3
   5
```

💖 다음은 1부터 5까지의 총합을 구하는 코드입니다. 빈칸을 채워보세요.

```
01: sum = 0
02: count = 1
03: while count <= 5:
04:
05:
06:
```

↳ 총합은 15

💖 다음은 3부터 1까지 거꾸로 세는 코드입니다. 빈칸을 채워보세요.

```
01: count = 3
```

```
02:
```

```
03:
```

```
04:
```

```
↳ 3
```

```
2
```

```
1
```

↳ 클로버1
클로버2
클로버3

♥ 그림과 같은 실행화면을 출력하는 코드를 고르세요.

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: cnt = 0
03: while cnt < 3:
04:     print(clovers[cnt])
05:     cnt = cnt + 1
```

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: cnt = 1
03: while cnt < 3:
04:     print(clovers[cnt])
05:     cnt = cnt + 1
```

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: for cnt in range(0, 2):
03:     print(clovers[cnt])
```

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: for cnt in range(1, 3):
03:     print(clovers[cnt])
```

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: price = 0
02: while price != -1:
03:     price = int(input('가격을 입력하세요 (종료:-1): '))
04:
05:
06:
07:
08:
09:
```

```
↳   가격을 입력하세요 (종료:-1): 3000
      정말 싸요.
      가격을 입력하세요 (종료:-1): 15000
      너무 비싸요
      가격을 입력하세요 (종료:-1): 7500
      괜찮은 가격이에요.
      가격을 입력하세요 (종료:-1): -1
```

💖 다음 중 출력 결과가 다른 코드를 골라보세요.

```
01: for i in range(3):
02:     print('안녕 거북이', i)
```

```
01: for j in range(0, 3):
02:     print('안녕 거북이', j)
```

```
01: k = 0
02: while k <= 3:
03:     print('안녕 거북이', k)
04:     k = k + 1
```

```
01: l = 0
02: while True:
03:     print('안녕 거북이', l)
04:     l = l + 1
05:     if l == 3:
06:         break
```

💖 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요.

```
01: num = 1
02: while True:
03:
04:
05:     print(num)
06:     num = num + 1
```

```
↳ 1
   2
   3
```

💖 소수를 판정하는 코드를 작성하세요.

- 소수(prime number): 1과 자기 자신 외에 나누어 떨어지지 않는 수입니다.

```
01: while True:
02:     number = int(input('2 이상의 정수를 입력하세요 (종료:-1): '))
03:
04:
05:
06:
07:
08:
09:
10:
11:
12:
13:
14:
15:
```

```
↳ 2 이상의 정수를 입력하세요 (종료:-1): 13
   13 은(는) 소수입니다.
   2 이상의 정수를 입력하세요 (종료:-1): -1
```

chapter 7

자료 모으기. 둘

♥ 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
>>>  
>>> print(nums)  
(1, 2, 3)
```


💖 다음 중 출력 결과가 다른 코드를 고르세요.

```
>>> my_var = 1,  
>>> print(type(my_var))
```

```
>>> my_var = (1,)   
>>> print(type(my_var))
```

```
>>> my_var = (1)   
>>> print(type(my_var))
```

```
>>> my_var = 1, 2  
>>> print(type(my_var))
```

♥ 다음 코드의 출력 결과를 적어보세요.

```
01: my_tuple = (3.14, 2.71)
02: print(my_tuple)
03: print(my_tuple[1])
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01:  nums = 1, 2, 3
02:  for idx in range(3):
03:      print(nums[idx])
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01: diamonds = 1, 5, 6, 7
02: ace, king, queen, jack = diamonds
03: print(queen)
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01:  alice = {}  
02:  alice['성별'] = '여'  
03:  alice['나이'] = 13  
04:  alice['혈액형'] = 'AB'  
05:  print(alice)
```

↳

💖 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  
03:  print(alice['전화번호'])
```

```
↳  010-1234-5678
```

💖 앨리스의 혈액형을 출력하려고 합니다. 다음 중 올바른 코드를 고르세요.

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  print(alice['혈액형'])
```

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  print(alice[혈액형])
```

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  print(alice['AB'])
```

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  print(alice[AB])
```

💖 앨리스의 나이를 14로 바꾸는 코드를 작성하세요.

```
01:  alice = {'성별': '여', '나이': 13, '혈액형': 'AB'}  
02:  
03:  print(alice['나이'])
```

```
↳ 14
```


💖 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
01:  alice = {'성별': '여', '나이': 14, '혈액형': 'AB', '전화번호': '010-1234-5678'}
02:
03:  print(alice)
```

```
↳    {'성별': '여', '나이': 13, '혈액형': 'AB'}
```

chapter 8

모아서 다시 쓰기

♥ 다음 코드의 출력 결과를 적어보세요.

```
01: def welcome():  
02:     print('이상한 나라에 오신 것을 환영합니다.')  
03:  
04: welcome()
```

↳

💖 다음 코드의 출력 결과를 적어보세요.

```
01: def welcome(name):  
02:     print(name, '님 이상한 나라에 오신 것을 환영합니다.')  
03:  
04: welcome('앨리스')  
05: welcome('도도새')
```

↳

♥ 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
01:
02:
03:
04: draw_stars(3)
05: draw_stars(2)
06: draw_stars(1)
```

```
↳ ***
   **
   *
```

♥ 아래와 같은 출력 결과가 나오도록 빈칸을 채워보세요.

```
01:
02:
03:
04: print(concat('빨주노초', '파남보'))
```

↳ 빨주노초파남보

♥ 정수 두 개를 입력 받아 두 정수의 뺄 값과 나눈 값을 돌려주는 함수 `sub_div()`를 만들어 보세요.

```
01:  
02:  
03:  
04: print(sub_div(6, 3))
```

```
↳ (3, 2.0)
```

♥ 카드의 이름과 수를 입력으로 받아 판결을 내리는 함수를 만들어 보세요.

```
01:  
02:  
03:  
04:  
05: judge_cards('하트', 2)  
06: judge_cards('클로버', 3)
```

```
↳ 하트 1 유죄!  
   하트 2 유죄!  
   클로버 1 유죄!  
   클로버 2 유죄!  
   클로버 3 유죄!
```


💖 다음은 무작위로 한 장의 카드를 뽑는 코드입니다. 빈칸을 채워보세요.

```
01:  
02: clovers = ['클로버1', '클로버2', '클로버3']  
03:
```

↳ 클로버2

💖 중복되지 않는 카드 두 장을 뽑도록 빈칸을 채워보세요.

```
01:  
02:  clovers = ['클로버1', '클로버2', '클로버3']  
03:
```

```
↳   클로버2, 클로버3
```

❤ 다음 중 출력 결과가 다른 하나를 고르세요.

```
01: import random
02: print(random.choice([1, 2, 3]))
```

```
01: import random
02: print(random.choice(range(1, 4)))
```

```
01: import random
02: print(random.randint(1, 3))
```

```
01: import random
02: print(random.randint(1, 4))
```