데이터프로그래밍 기초 2일차

2026-1 DS Bootcamp

부산대학교
데이터사이언스전문대학원
석사과정 박민서

PUSAN NATIONAL UNIVERSITY
Graduate School of Data Science

CONTENTS

Data Type & Operator

▪ 자료 (Data)

- 프로그램이 처리할 수 있는 모든 값
- 하나의 변수에 대한 관찰을 통해 얻어진 사실적인 정보

  Ex) 키, 몸무게, 나이, 성별, 혈액형, 이름, 학번, 합격 여부, 시험 결과, …

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 자료형 (Data Type)

- 자료의 기능과 역할에 따라 구분한 종류
- 숫자(int, float), 문자열(str), 논리형(bool) 등이 있음

- 숫자 (int, float, …)
- 문자열 (str)
- 논리형 (bool)
- …

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 자료형 (Data Type)

- **type( )**: 변수의 자료형을 확인하는 함수

```
my_int = 123                # 정수형 변수 선언
print(type(my_int))

my_float = 3.141592         # 실수형 변수 선언
print(type(my_float))

my_str = "DS 부트캠프"      # 문자열 변수 선언
print(type(my_str))
```

✓ 0.0s

```
<class 'int'>
<class 'float'>
<class 'str'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 숫자 자료형

- **int (Integer)**: 정수형 자료 (ex. 0, 1, -1, 2, 3, ⋯)
- **float (floating point,** 부동소수점**)**: 실수형 자료 (ex. 3.14, 123.45, 0.0, 1e-2, ⋯)

※ $52.275 = 5.2275 \times 10$
$= 0.52275 \times 10^2$
$= 0.052275 \times 10^3$

```
num1 = 1
num2 = 1.0
num3 = 1e-2

print(num1)
print(type(num1))

print(num2)
print(type(num2))

print(num3)
print(type(num3))
```

✓ 0.0s

```
1
<class 'int'>
1.0
<class 'float'>
0.01
<class 'float'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자 연산자

- 사칙 연산자 (+, -, *, /), 정수 몫 연산자 (//), 나머지 연산자 (%), 제곱 연산자 (**), …

```
num1 = 100
num2 = 4

print(num1 + num2)      # 더하기
print(num1 - num2)      # 빼기
print(num1 * num2)      # 곱하기
print(num1 / num2)      # 나누기 (결과는 실수형)
print(num1 // num2)     # 몫 연산 (결과는 정수)
print(num1 % num2)      # 나머지 연산
print(num1 ** num2)     # 거듭제곱 연산
```

✓ 0.0s

```
104
96
400
25.0
25
0
100000000
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 숫자 연산자

- int 변수와 float 변수의 연산 결과 ☞ float

```
my_int = 3
my_float = 1.5

print(my_int + my_float)
print(my_int - my_float)
print(my_int * my_float)
print(my_int / my_float)
```

✓ 0.0s

```
4.5
1.5
4.5
2.0
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 숫자 연산자

- 연산자의 우선순위


```
print(2 + 2 * 2)
✓ 0.0s
6
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 숫자 연산자

- 연산자의 우선순위 ☞ 괄호로 묶어 원하는 연산 우선 진행

```
print(2 + 2 - 2 * 2 / 2 * 2)
```
✓ 0.0s
```
0.0
```

```
print(2 + (2 - 2) * 2 / 2 * 2)
```
✓ 0.0s
```
2.0
```

순위	연산
1	(), [], {}, {:}
2	[], (), .
3	**
4	+, -, ~
5	*, /, //, %, @
6	+, -
7	<<, >>
8	&
9	^
10	|
11	in, not in, is, is not, <, <=, >, >=, !=, ==
12	not x
13	and
14	or
15	x if C else y
16	lambda
17	:=

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽
https://blog.naver.com/gmlrud2280/223417862543

Data Type & Operator

▪ 문자열 자료형

- **str (String)**: 따옴표(“ ”, ‘ ’)로 감싸 입력한 모든 자료

```
my_str = "Hello World!"
my_str2 = 'DS BootCamp'

print(my_str)
print(type(my_str))
print(my_str2)
print(type(my_str2))
```

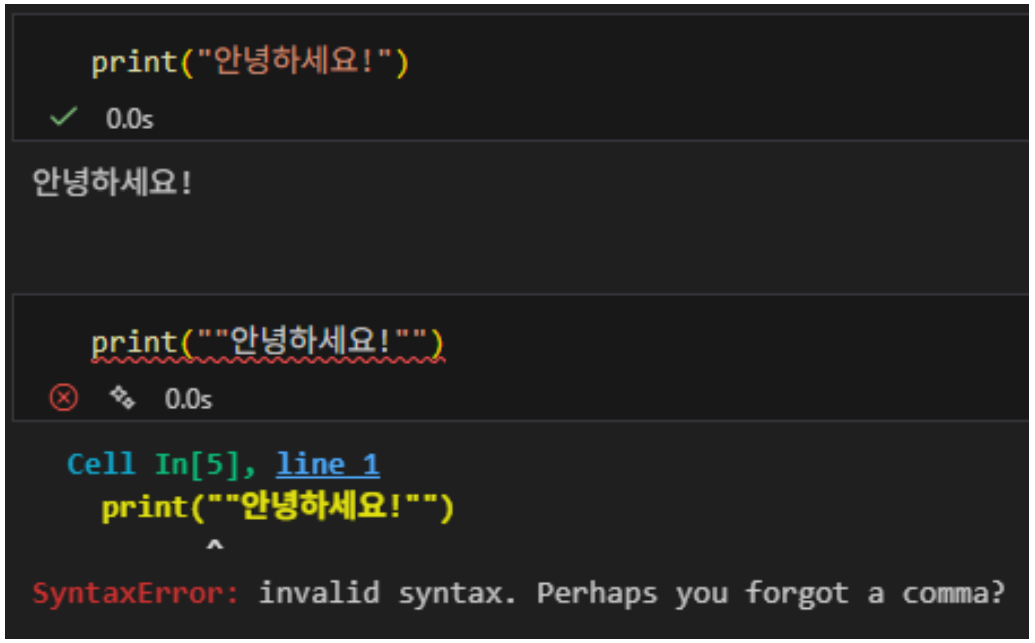
✓ 0.0s

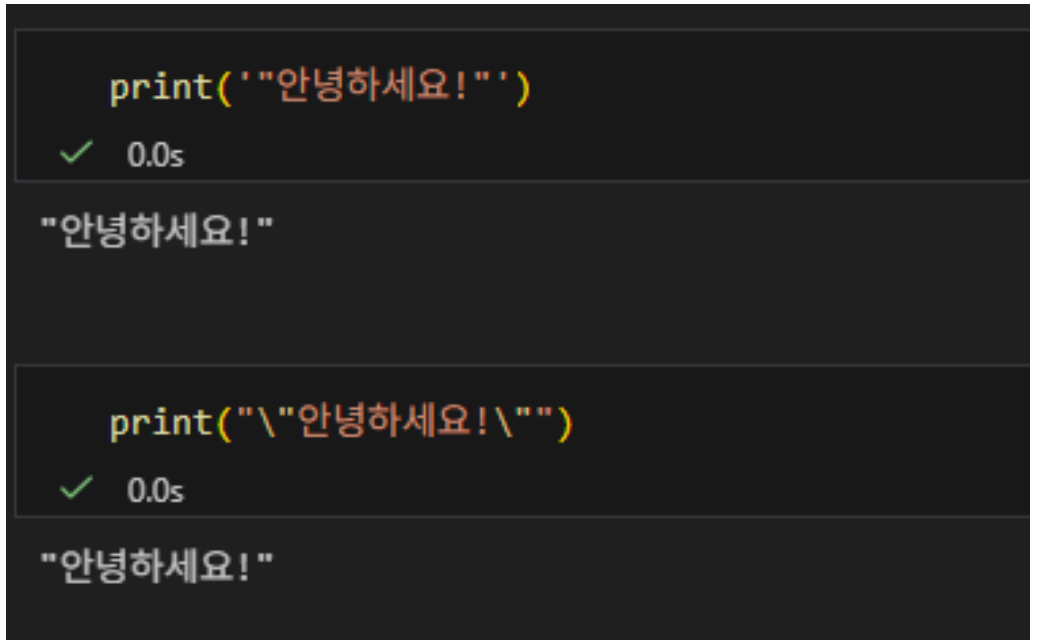
```
Hello World!
<class 'str'>
DS BootCamp
<class 'str'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 자료형

- 따옴표(“ ”, ‘ ’) 출력하는 방법: 작은 따옴표(‘ ’) or **이스케이프 문자**(＼, ₩) 활용





윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 자료형

- 이스케이프 문자 (Escape character): 기존에 정해진 규칙에서 벗어난 문자를 만들 때 사용

```
# 이스케이프 문자
# \n: 줄바꿈
# \t: 탭 한칸
# \b: 공백 한칸
# \": 큰따옴표 삽입
# \': 작은따옴표 삽입
# \\: 역슬래쉬(\) 삽입

print("Hello World!")

print("Hello\nWorld!")
print("Hello\tWorld!")
print("Hello\bWorld!")
print("Hello\\World!")
```

✓ 0.0s

```
Hello World!
Hello
World!
Hello   World!
HellWorld!
Hello\World!
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽
https://wikidocs.net/192095

▪ 문자열 연산자

- +: 문자열 연결 연산자 ☞ “문자열1” + “문자열2” = “문자열1문자열2”

```
s1 = "Hello"
s2 = "Everybody"

print(s1 + s2)
```

✓ 0.0s

```
HelloEverybody
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 연산자

- +: 문자열 연결 연산자 ☞ “문자열1” + 숫자 = 오류 발생

```
s1 = "Hello"
s2 = 123

print(s1 + s2)
```

⊗ 0.3s

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[12], line 4
      1 s1 = "Hello"
      2 s2 = 123
----> 4 print(s1 + s2)

TypeError: can only concatenate str (not "int") to str
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 연산자

- *: 문자열 반복 연산자 ☞ “문자열1” * 숫자 = 문자열 반복 출력

```
s1 = "Hello"
num = 3

print(s1 * num)
```

✓ 0.0s

```
HelloHelloHello
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 연산자

- [ ]: 문자열 인덱싱 ☞ 문자열 내부 문자 선택하는 연산자

문	자	열		연	산
[0]	[1]	[2]	[3]	[4]	[5]
[-6]	[-5]	[-4]	[-3]	[-2]	[-1]

```
s1 = "문자열 연산"

print(s1[0])
print(s1[1])
print(s1[2])
print(s1[3])
print(s1[4])
print(s1[5])
```

✓ 0.0s

```
문
자
열

연
산
```

```
s1 = "문자열 연산"

print(s1[-1])
print(s1[-2])
print(s1[-3])
print(s1[-4])
print(s1[-5])
print(s1[-6])
print(s1[-7])       # 범위 넘어선 index
```

⊗ 0.0s

```
산
연

열
자
문

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[17], line 9
      7 print(s1[-5])
      8 print(s1[-6])
----> 9 print(s1[-7])

IndexError: string index out of range
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 문자열 연산자

- [:]: 문자열 범위 선택 연산 ☞ 문자열의 특정 범위에 속한 문자 선택하는 연산자

  Ex. s1[0:3] ☞ s1 문자열 속 0번째 문자부터 3번째 문자 전까지 선택

```
s1 = "문자열 연산"

print(s1[0:3])      # 문자열
print(s1[1:5])      # 자열 연
print(s1[3:-1])     #  연

print(s1[1:])       # 자열 연산
print(s1[:4])       # 문자열(공백)
```

✓ 0.0s

```
문자열
자열 연
 연
자열 연산
문자열
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 복합 대입 연산자

- (연산자) + “=” 조합의 연산자

  Ex. ‘a += 10’ ☞ ‘a = a + 10’

```
num = 10
num += 10       # num = num + 10 = 20
print(num)      # 20
num -= 10       # num = num - 10 = 10
print(num)      # 10
num *= 10       # num = num * 10 = 100
print(num)      # 100
num //= 5       # num = num // 5 = 20
print(num)      # 20
num %= 3        # num = num % 3 = 2
print(num)      # 2
```

✓ 0.0s

```
20
10
100
20
2
```

```
s = "Hello"
s += "!"        # s = s + "!" = "Hello!"
print(s)

s *= 3          # s = s * 3 = "Hello!Hello!Hello!"
print(s)
```

✓ 0.0s

```
Hello!
Hello!Hello!Hello!
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- 숫자와 관련된 유용한 함수

```
print(abs(-123)) # 123
print(divmod(23, 6)) # 3, 5
print(pow(2, 10)) # 1024
print(pow(2, 10, 6)) # 1024 % 6 = 4
print(round(3.141592, 3)) # 3.142
```

✓ 0.0s

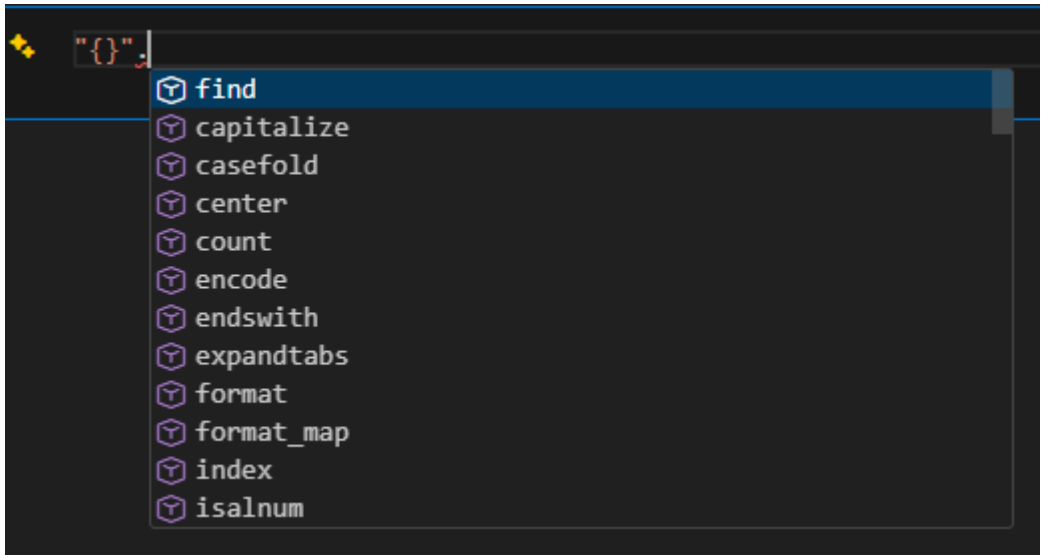
```
123
(3, 5)
1024
4
3.142
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Data Type & Operator

▪ 숫자형, 문자열의 다양한 기능

- format( ): 문자열형 내장함수, 중괄호 '{ }'에 매개변수를 대입하는 구조




```
# 중괄호와 매개변수의 수 일치해야함
print("{}".format(10))
print("{}, {}".format(10, 20))
print("제 이름은 {}이고, 나이는 {}살 입니다.".format("박민서", 28))
# 일치하지 않는 경우
print("{}, {}".format(10, 20, 30, 40, 50)) # 10, 20까지만 출력
print("{}, {}".format(10))      # 오류 발생
```

0.0s

```
10
10, 20
제 이름은 박민서이고, 나이는 28살 입니다.
10, 20

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[40], line 7
      5 # 일치하지 않는 경우
      6 print("{}, {}".format(10, 20, 30, 40, 50))
----> 7 print("{}, {}".format(10))

IndexError: Replacement index 1 out of range for positional args tuple
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- format( ): 문자열형 내장함수, 중괄호 '{ }'에 매개변수를 대입하는 구조

```
# 정수형 자료 + format()

print("{:d}".format(123))       # 기본 정수 출력
print("{:5d}".format(123))      # 5칸 확보 후 정수 출력 (오른쪽에)
print("{:05d}".format(123))     # 5칸 확보 후 정수 출력 + 빈칸은 0으로 채움
print("{:05d}".format(-123))    # 5칸 확보 후 정수 출력 + 빈칸은 0으로 채움
print("{:+d}".format(123))      # + 기호 출력
print("{:+d}".format(-123))     # - 기호 출력
print("{: d}".format(123))      # 부호 칸 확보 (양수)
print("{: d}".format(-123))     # 부호 칸 확보 (음수)
print("{:+5d}".format(123))     # 5칸 확보, + 기호 출력
print("{:+5d}".format(-123))    # 5칸 확보, - 기호 출력
print("{:=+5d}".format(123))    # 5칸 확보, 맨 앞에 + 기호 출력
print("{:=+5d}".format(-123))   # 5칸 확보, 맨 앞에 - 기호 출력
```

✓ 0.0s

```
123
  123
00123
-0123
+123
-123
 123
-123
 +123
 -123
+ 123
- 123
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- format( ): 문자열형 내장함수, 중괄호 '{ }'에 매개변수를 대입하는 구조

```
# 실수형 자료 + format()

print("{:f}".format(3.1415))        # 기본 실수 출력 (소숫점 6자리까지)
print("{:15f}".format(3.1415))      # 15칸 확보 후 실수 출력 (오른쪽에)
print("{:015f}".format(3.1415))     # 15칸 확보 + 빈칸은 0으로 채움
print("{:15.3f}".format(3.1415))    # 15칸 확보 + 소숫점 아래 3자리까지 출력 (반올림)
print("{:15.2f}".format(3.1415))    # 15칸 확보 + 소숫점 아래 2자리까지 출력 (반올림)
print("{:15.1f}".format(3.1415))    # 15칸 확보 + 소숫점 아래 1자리까지 출력 (반올림)
```

✓ 0.0s

```
3.141500
       3.141500
00000003.141500
          3.142
           3.14
            3.1
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- 문자열과 관련된 유용한 함수

```
# 문자열 함수
my_str = " DS BootCamp " # 양 옆 공백
print("기본 문자열:", my_str)
# upper(), lower(): 문자열을 모두 대문자(소문자)로 변환
print("upper() 적용 후:", my_str.upper())
print("lower() 적용 후:", my_str.lower())
# strip(), lstrip(), rstrip(): 양 옆 [왼쪽, 오른쪽] 공백(이스케이프 문자까지) 제거
print("strip() 적용 후:", my_str.strip())
print("lstrip() 적용 후:", my_str.lstrip())
print("rstrip() 적용 후:", my_str.rstrip())
```

✓ 0.0s

```
기본 문자열:  DS BootCamp
upper() 적용 후:  DS BOOTCAMP
lower() 적용 후:  ds bootcamp
strip() 적용 후: DS BootCamp
lstrip() 적용 후: DS BootCamp
rstrip() 적용 후:  DS BootCamp
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- 문자열과 관련된 유용한 함수

```
print("기본 문자열:", my_str)
# len(): 문자열 길이 구하는 함수
print("len() 적용 후:", len(my_str))
# find(), rfind(): 문자열 내 특정 문자 찾는 함수 (문자의 인덱스 반환)
print("find() 함수 적용 후:", my_str.find("o"))
print("rfind() 함수 적용 후:", my_str.rfind("o")) # 오른쪽부터 찾았을 때,
print("find() 함수 적용 후:", my_str.find("b")) # 없는 문자를 찾을 땐 -1 반환
```

✓ 0.0s

```
기본 문자열:  DS BootCamp
len() 적용 후: 13
find() 함수 적용 후: 5
rfind() 함수 적용 후: 6
find() 함수 적용 후: -1
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- 문자열과 관련된 유용한 함수

```
print("기본 문자열:", my_str)
# in 연산자: 문자열 내 특정 문자(문자열)이 있는지 없는지 반환
print("안녕" in "안녕하세요")
print("안녕" in my_str)
```

✓ 0.0s

```
True
False
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 숫자형, 문자열의 다양한 기능

- 문자열과 관련된 유용한 함수

```
my_str = " DS BootCamp\t "
print("기본 문자열:", my_str)
# split(): 문자열을 특정한 문자로 자르는 함수 -> list가 반환
print("split() 함수 적용 후:", my_str.split()) # 이스케이프 문자까지 다 자름
print("split(\" \") 함수 적용 후:", my_str.split(" "))
print("split(\"a\") 함수 적용 후:", my_str.split("a"))

my_list = ['DS', 'BootCamp']
# join(): 문자열을 가진 리스트를 연결해주는 함수
print("\"\".join(my_list) 함수 적용 후:", "".join(my_list))
print("\" \".join(my_list) 함수 적용 후:", " ".join(my_list))
print("\"_\".join(my_list) 함수 적용 후:", "_".join(my_list))
```

✓ 0.0s

```
기본 문자열:  DS BootCamp
split() 함수 적용 후: ['DS', 'BootCamp']
split(" ") 함수 적용 후: ['', 'DS', 'BootCamp\t', '']
split("a") 함수 적용 후: [' DS BootC', 'mp\t ']
"".join(my_list) 함수 적용 후: DSBootCamp
" ".join(my_list) 함수 적용 후: DS BootCamp
"_".join(my_list) 함수 적용 후: DS_BootCamp
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ f-string (f-문자열)

- 문자열 내부에 다른 expression을 삽입하려 할 때 쓰는 연산자 (Python 3.6 이상)

  ☞ f“문자열{원하는 변수 or 표현식}문자열…” 형식

```
# format
print("제 이름은 {}이고, 나이는 {}살 입니다.".format("박민서", 28))
# f-string
name = "박민서"
age = 28
print(f"제 이름은 {name}이고, 나이는 {age}살 입니다.")
```

✓ 0.0s

```
제 이름은 박민서이고, 나이는 28살 입니다.
제 이름은 박민서이고, 나이는 28살 입니다.
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ f-string (f-문자열)

- 문자열 내부에 다른 expression을 삽입하려 할 때 쓰는 연산자 (Python 3.6 이상)

  ☞ f“문자열{원하는 변수 or 표현식}문자열…” 형식

```
# format
print("제 이름은 {}이고, 나이는 {}살 입니다.".format("박민서", 28))
# f-string
name = "박민서"
age = 28
print(f"제 이름은 {name}이고, 나이는 {age}살 입니다.")
```

✓ 0.0s

```
제 이름은 박민서이고, 나이는 28살 입니다.
제 이름은 박민서이고, 나이는 28살 입니다.
```

```
# format()
print("{:05d}".format(123))

# f-string
num = 123
print(f"{num:05d}")
```

✓ 0.0s

```
00123
00123
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 자료형 변환 함수

- int ( ) : int형으로 변환　　float ( ) : float형으로 변환　　str ( ) : str형으로 변환

```
my_int = 123
my_float = float(my_int)    # my_int를 실수형으로 변환한 값 저장
print(my_float)             # 123.0 출력
my_str = str(my_int)        # my_int를 문자열형으로 변환한 값 저장
print(my_str)               # "123" 출력

print(my_int + my_int)      # 123 + 123 = 246   (int)
print(my_float + my_float)  # 123.0 + 123.0 = 246.0 (float)
print(my_str + my_str)      # "123" + "123" = "123123" (str)

s = "DS BootCamp"
my_int2 = int(s)            # "DS BootCamp"를 int형으로 변환 => 오류
print(my_int2)
```

0.0s

```
123.0
123
246
246.0
123123

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[28], line 12
      9 print(my_str + my_str)      # "123" + "123" = "123123" (str)
     11 s = "DS BootCamp"
---> 12 my_int2 = int(s)            # "DS BootCamp"를 int형으로 변환 => 오류
     13 print(my_int2)

ValueError: invalid literal for int() with base 10: 'DS BootCamp'
```

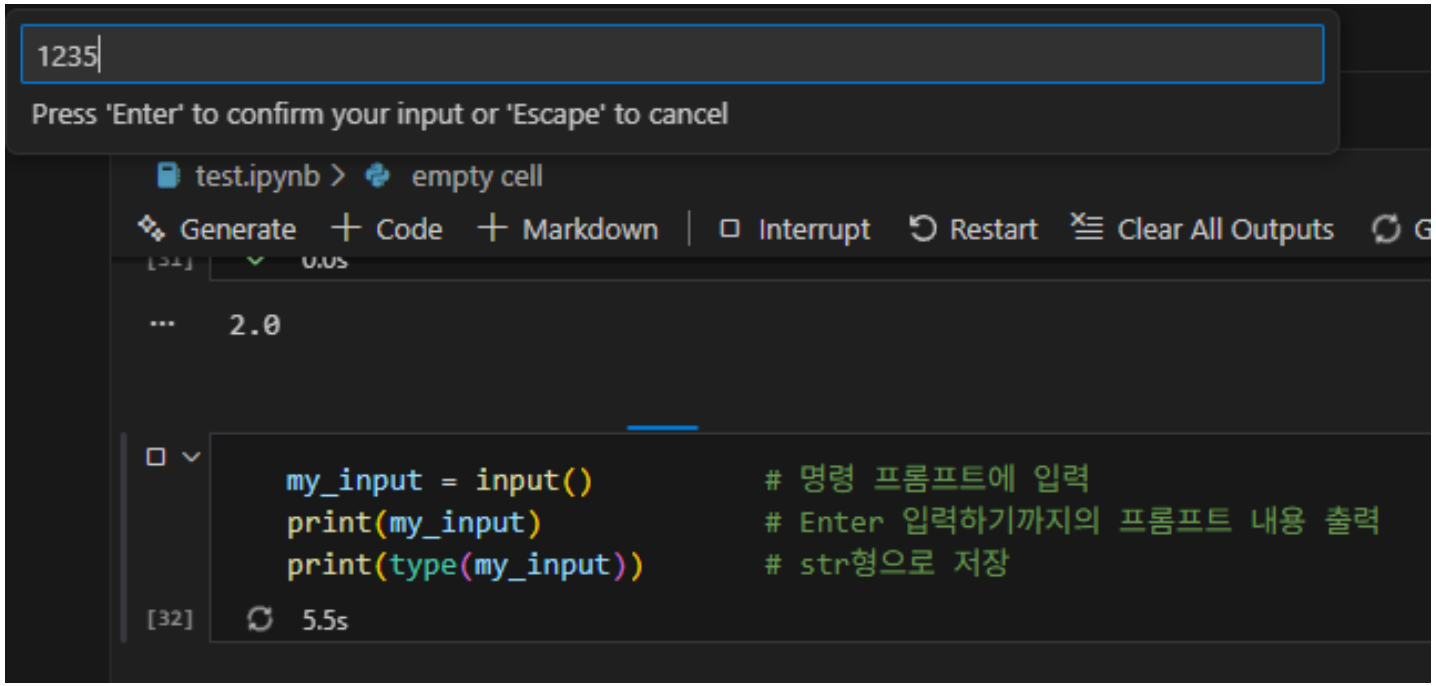
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Q&A

Input & Print

▪ 입출력 함수

- input( ): 명령 프롬프트에서 사용자로부터 데이터를 입력 받을 때 사용 ☞ str형으로 저장



입력

```
1235
<class 'str'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Input & Print

▪ 입출력 함수

- input( ): 명령 프롬프트에서 사용자로부터 데이터를 입력 받을 때 사용 ☞ str형으로 저장

```
# 명령 프롬프트에 정수형 입력 (문자 입력하면 오류 발생)
my_input = int(input())
print(my_input)
print(type(my_input))
```

✓ 3.2s

```
12345
<class 'int'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

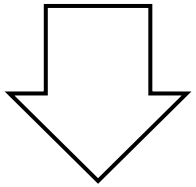
Input & Print

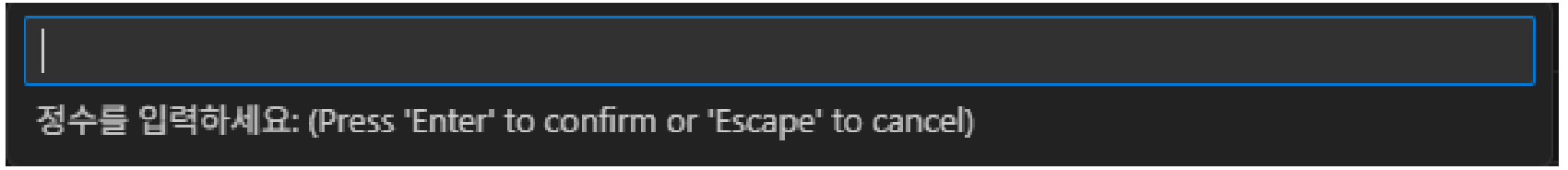
▪ 입출력 함수

- input( ): 명령 프롬프트에서 사용자로부터 데이터를 입력 받을 때 사용 ☞ str형으로 저장

```
# 명령 프롬프트에 표기할 내용
my_input = int(input("정수를 입력하세요: "))
print(my_input)
```

34.0s





윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Input & Print

▪ 입출력 함수

- print( ): 괄호 안에 출력하고 싶은 것 입력

```
# 문자열 하나만 출력
print("Hello BootCamp!")
# 여러개 출력 (항목마다 띄워쓰기 적용)
print("Hello", "BootCamp", 3)
# 줄바꿈
print()
# end: 출력물 마지막에 추가할 문자열 입력 (기본값: \n)
print("Hello", "BootCamp", 3, end="!")
```

✓ 0.0s

```
Hello BootCamp!
Hello BootCamp 3

Hello BootCamp 3!
```

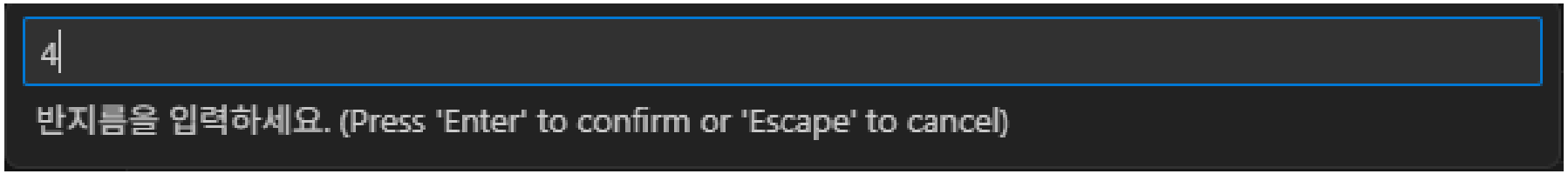
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Practice

▪ 실습 1

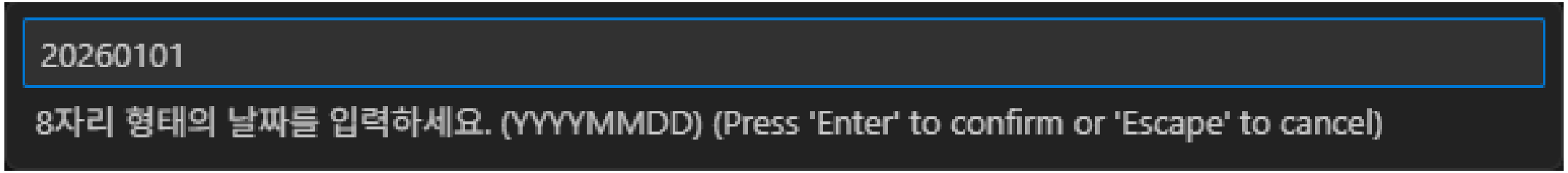
- 아래와 같이 작동하는 프로그램의 코드를 작성하세요.
  - 입력 프롬프트: " 원의 반지름을 입력하세요.”
  - 출력: 원의 둘레는 ~ cm 입니다. 원의 넓이는 ~ cm^2 입니다.

  (원주율은 3.14로 계산)



▪ 실습 2

- 아래와 같이 작동하는 프로그램의 코드를 작성하세요.
  - 입력 프롬프트: "8자리 형태의 날짜를 입력하세요. (YYYYMMDD)“
  - 입력값 예시: 20260101
  - 출력: 2026년 01월 01일
- 입력과 출력을 바꾼 프로그램도 작성해 보세요.



윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽
https://wikidocs.net/192095

Conditional Statement

▪ Boolean (불, bool)

- True / False 값만 가지는 자료형
- 독립적으로 쓰는 일은 잘 없음 ☞ 어떤 명제(Statement)의 결과로 쓰임

```
my_bool = True
print(my_bool)
print(type(my_bool))
```

✓ 0.0s

```
True
<class 'bool'>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 비교 연산자

- bool형 값을 만들 수 있는 연산자

```
print(10 == 100)    # '같다'
print(10 != 100)    # '다르다'
print(10 < 100)     # '작다'
print(10 > 100)     # '크다'
print(10 <= 100)    # '작거나 같다'
print(10 >= 100)    # '크거나 같다'
```

✓ 0.0s

```
False
True
True
False
True
False
```

```
print("apple" == "banana")
print("apple" != "banana")
print("apple" > "banana")   # 사전 순서로 비교 가능
print("apple" < "banana")
```

✓ 0.0s

```
False
True
False
True
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 논리 연산자

- bool형끼리 사용할 수 있는 연산자

```
print(not True)
print(not False)
```

✓ 0.0s

```
False
True
```

```
condition1 = 10 > 100           # False
condition2 = "apple" < "banana" # True

print(condition1 and condition2) # False and True = False
print(condition1 or condition2)  # False or True = True
```

✓ 0.0s

```
False
True
```

Conditional Statement

▪ 조건문 (if 구문)

- if (bool 값이 나오는 조건식):
  조건이 참이면 실행할 구문
  ...

```
score = 90
if score >= 70:          # score가 70점 이상이면
    print("합격입니다")    # 해당 구문 실행
```

✓ 0.0s

```
합격입니다
```

```
score = 60
if score >= 70:          # score가 70점 이상이면
    print("합격입니다")    # 해당 구문 실행
```

✓ 0.0s

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Conditional Statement

▪ 조건문 (if 구문)

- if (bool 값이 나오는 조건식):
  조건이 참이면 실행할 구문
  ...

```
score = 75
if score >= 90:
    print("90점 이상입니다.")    # 실행 X
if score >= 80:
    print("80점 이상입니다.")    # 실행 X
if score >= 70:
    print("70점 이상입니다.")    # 실행 O
if score >= 60:
    print("60점 이상입니다.")    # 실행 O
```

✓ 0.0s

```
70점 이상입니다.
60점 이상입니다.
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Conditional Statement

▪ 조건문 (if ~ else구문)

- if (bool 값이 나오는 조건식):
  조건이 참이면 실행할 구문

  else:
  조건이 거짓이면 실행할 구문

```
score = 75
if score >= 90:
    print("합격입니다")
else:
    print("불합격입니다")
```

✓ 0.0s

```
불합격입니다
```

```
my_num = 13

if my_num % 2 == 0:     # my_num을 2로 나눈 나머지가 0 => 짝수
    print("짝수입니다.")
else:                   # 아니라면 => 홀수
    print("홀수입니다.")
```

✓ 0.0s

```
홀수입니다.
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Conditional Statement

▪ 조건문 (elif 구문)

- if (조건1):
  조건10이 참이면 실행할 구문
  elif (조건2):
  조건2가 참이면 실행할 구문
  elif (조건3):
  조건30이 참이면 실행할 구문
  else:
  모든 조건이 거짓이면 실행할 구문

```
my_grade = 3.5

if my_grade == 4.5:
    print("A+")
elif my_grade == 4.0:
    print("A0")
elif my_grade == 3.5:
    print("B+")
else:
    print("B0 이하")
```

✓ 0.0s

```
B+
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Conditional Statement

▪ 조건문 (elif 구문)

- if (조건1):
  조건10이 참이면 실행할 구문
  elif (조건2):
  조건2가 참이면 실행할 구문
  elif (조건3):
  조건30이 참이면 실행할 구문
  else:
  모든 조건이 거짓이면 실행할 구문

```
my_grade = 3.5

if my_grade == 4.5:
    print("A+")
elif my_grade == 4.0:
    print("A0")
elif my_grade == 3.5:
    print("B+")
else:
    print("B0 이하")
```

✓ 0.0s

```
B+
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Loop Statement

▪ 반복문 (for 구문)

- **for 반복할 변수 in 반복할 수 있는 자료:**

  반복할 코드

※ '반복할 수 있는 자료' = list(리스트), dict(딕셔너리), string(문자열), range(범위)

```
my_str = "DS BootCamp"

for s in my_str:        # 문자열을 처음부터 탐색하고, s라는 변수로 나타냄
    print(s)            # 문자 출력
```

✓ 0.0s

```
D
S

B
o
o
t
C
a
m
p
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Loop Statement

▪ range( ): 범위

- range(시작 값 [선택, 값 포함], 끝 값 [필수, 값 포함 X], 증가 단위 [선택, 기본=1])
- 입력 매개변수로 무조건 정수 사용

```
for i in range(1, 10, 2):
    print(i)
```
✓ 0.0s

```
1
3
5
7
9
```

```
for i in range(0, 101, 10):
    print(i)
```
✓ 0.0s

```
0
10
20
30
40
50
60
70
80
90
100
```

```
for i in range(10, 0, -1):
    print(i)
```
✓ 0.0s

```
10
9
8
7
6
5
4
3
2
1
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 중첩 반복문

- 반복문을 중첩해서 사용하는 구조

```
for i in range(0, 10):
    for j in range(0, 10):
        print(10 * i + j + 1, end = " ")
    print()
```

✓ 0.0s

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 실습 3

- 아래와 같이 작동하는 프로그램의 코드를 작성하세요.
  - 입력 프롬프트: “층수를 입력하세요 (정수)”
  - 입력값 예시: 5
  - 출력:

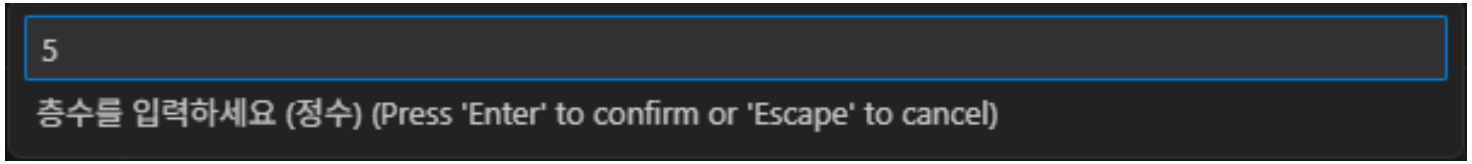
*

**

***

****

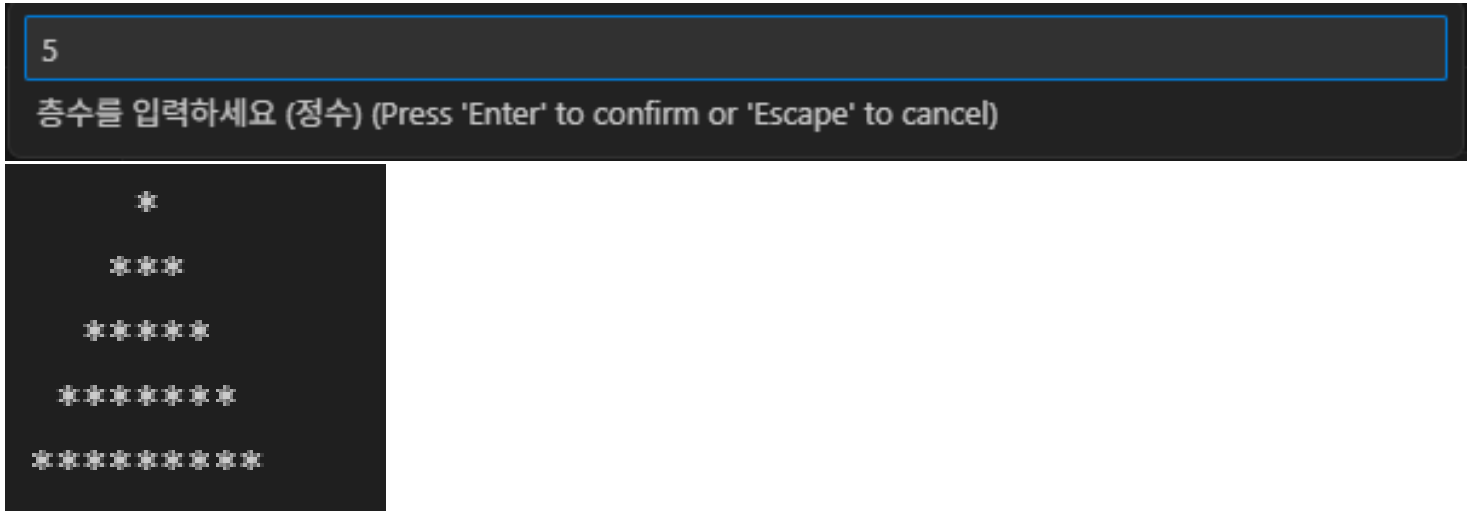
*****



▪ 실습 3 (심화)

- 아래와 같이 작동하는 프로그램의 코드를 작성하세요.
  - 입력 프롬프트: “층수를 입력하세요 (정수)”
  - 입력값 예시: 5
  - 출력:

```
    *
   ***
  *****
 *******
*********
```



윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽
https://wikidocs.net/192095

Loop Statement

▪ while 반복문

- **while (조건식):**

  조건식이 참일 때 실행할 코드

※ 조건식을 통해 while 구문을 끝낼 수 있는 구조일 때 사용 ☞ 무한 반복 방지

```
count = 1

while count <= 3:
    print(f"{count}회 실행되었습니다.")
    count += 1

print("코드 종료")
```

✓ 0.0s

```
1회 실행되었습니다.
2회 실행되었습니다.
3회 실행되었습니다.
코드 종료
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

Loop Statement

▪ break / continue

- 반복문 내부에서 사용하는 키워드

```python
my_int = 5

while True: # 계속 실행 (무한 반복)
    # 정수 입력 받고
    check = int(input("정수를 입력하세요."))
    # 입력받은 정수가 my_int와 같다면
    if check == my_int:
        print(f"check = {check}, my_int를 찾았습니다.")
        break # 반복문 탈출
    else: # 같지 않다면
        print(f"check = {check}, 입력한 정수와 my_int는 다릅니다")
        continue # 반복문 처음부터 다시 실행
```

✓ 4.8s

```
check = 1, 입력한 정수와 my_int는 다릅니다
check = 2, 입력한 정수와 my_int는 다릅니다
check = 3, 입력한 정수와 my_int는 다릅니다
check = 4, 입력한 정수와 my_int는 다릅니다
check = 6, 입력한 정수와 my_int는 다릅니다
check = 5, my_int를 찾았습니다.
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 84-271쪽

▪ 2일차 과제

- GitHub 사이트에서 “2일차_과제.ipynb” 다운로드
- 코드 작성 후, “**본인이름**_2일차_과제.ipynb”로 저장
- 저장한 과제 파일 전송 (이메일 주소: minsuh99@pusan.ac.kr)

  기한: ~ 2/3 PM 23:59:59

Q&A