

# 변수와 입력

# 목차

- 시작하기 전에
- 변수 만들기/사용하기
- 복합 대입 연산자
- 사용자 입력 : `input()`
- 문자열을 숫자로 바꾸기
- 숫자를 문자열로 바꾸기
- 키워드로 정리하는 핵심 포인트
- 확인문제

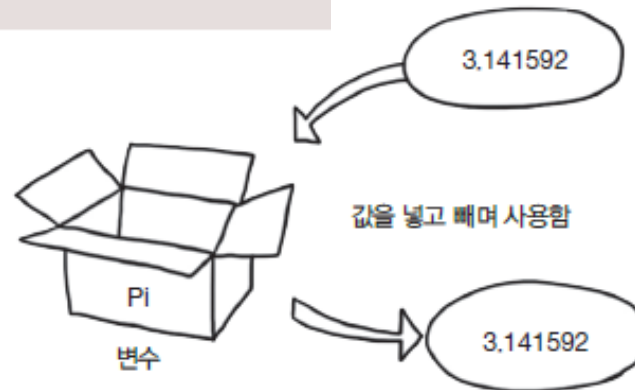
**[핵심 키워드]** 변수 선언, 변수 할당, 변수 참조, input(), int(), float(), str()

**[핵심 포인트]** 변수는 숫자뿐만이 아닌 모든 자료형을 의미하며, 파이썬에서 변수를 생성하는 것은 이를 사용하겠다고 선언하는 것이다. 변수에는 모든 자료형의 값을 저장할 수 있다.

## ● 변수

- 값을 저장할 때 사용하는 식별자
- 숫자뿐만 아니라 모든 자료형을 저장할 수 있음

```
>>> pi = 3.14159265  
>>> pi  
3.14159265
```



## ● 변수의 활용

- 변수를 선언하는 방법
  - 변수를 생성
- 변수에 값을 할당하는 방법
  - 변수에 값을 넣음
  - = 우변을 값을 좌변에 할당
- 변수를 참조하는 방법
  - 변수에서 값을 꺼냄
  - 변수 안에 있는 값을 사용

변수 = 값



값을 변수에 할당합니다.

- 변수를 참조

- 변수에 저장된 값을 출력

```
변수
```

- 변수에 저장된 값으로 연산

```
변수 + 변수
```

- 변수에 저장된 값을 출력

```
print(변수)
```

- 앞 예시에서 입력한 pi는 숫자 자료에 이름 붙인 것이기 때문에 숫자 연산 모두 수행할 수 있음

```
>>> pi = 3.14159265
>>> pi + 2
5.14159265
>>> pi - 2
1.1415926500000002
>>> pi * 2
6.2831853
>>> pi / 2
1.570796325
>>> pi % 2
1.1415926500000002
>>> pi * pi
9.869604378534024
```

- pi는 숫자 자료이므로 숫자와 문자열 연산은 불가능

```
pi + "문자열"
```

- 예시 - 원의 둘레와 넓이 구하기

```
01  # 변수 선언과 할당
02  pi = 3.14159265
03  r = 10
04
05  # 변수 참조
06  print("원주율 =", pi)
07  print("반지름 =", r)
08  print("원의 둘레 =", 2*pi*r)      # 원의 둘레
09  print("원의 넓이 =", pi*r*r)     # 원의 넓이
```

실행결과

```
원주율 = 3.14159265
반지름 = 10
원의 둘레 = 62.831853
원의 넓이 = 314.159265
```



- 복합 대입 연산자

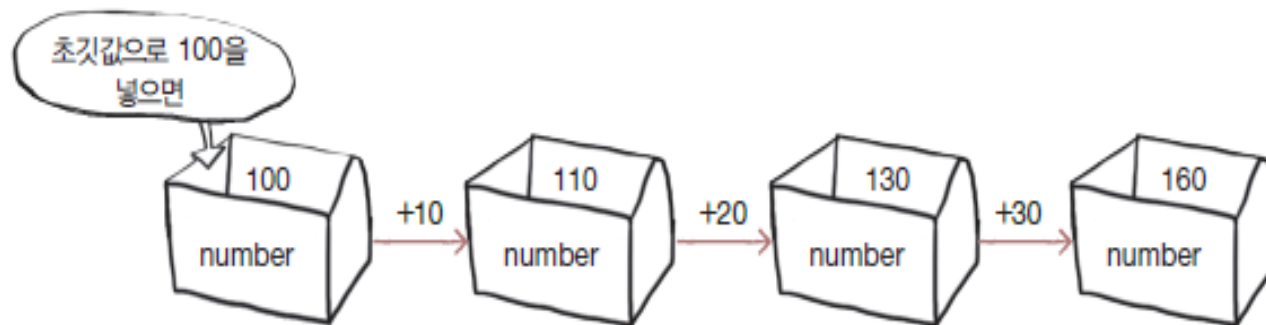
- 기본 연산자와 = 연산자 함께 사용해 구성

```
a += 10
```

연산자 이름	설명
+=	숫자 덧셈 후 대입
-=	숫자 뺄셈 후 대입
*=	숫자 곱셈 후 대입
/=	숫자 나눗셈 후 대입
%=	숫자의 나머지를 구한 후 대입
**=	숫자 제곱 후 대입

## 예시

```
>>> number = 100  
>>> number += 10  
>>> number += 20  
>>> number += 30  
>>> print("number:", number)  
number: 160
```

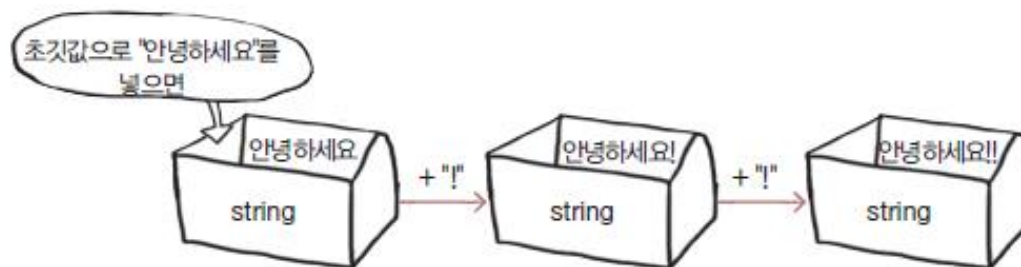


- 문자열 역시 복합 대입 연산자 사용 가능

연산자 이름	설명
<code>+=</code>	문자열 연결 후 대입
<code>*=</code>	문자열 반복 후 대입

- (

```
>>> string = "안녕하세요"
>>> string += "!"
>>> string += "!"
>>> print("string:", string)
string: 안녕하세요!!
```



# 사용자 입력 : input()

- input() 함수

- 명령 프롬프트에서 사용자로부터 데이터 입력받을 때 사용

- input() 함수로 사용자 입력받기

- 프롬프트 함수 : input 함수 괄호 안에 입력한 내용

```
>>> input("인사말을 입력하세요> ")
```

- 블록 (block) : 프로그램이 실행 중 잠시 멈추는 것

인사말을 입력하세요> | → 입력 대기를 알려주는 커서입니다. 커서는 프로그램에 따라 모양이 다를 수 있습니다.

- 명령 프롬프트에서 글자 입력 후 [Enter] 클릭

```
인사말을 입력하세요> 안녕하세요 [Enter]  
'안녕하세요'
```

- input 함수의 결과로 산출 (리턴값)
  - 다른 변수에 대입하여 사용 가능

```
>>> string = input("인사말을 입력하세요> ")
인사말을 입력하세요> 안녕하세요 [Enter]
>>> print(string)
안녕하세요
```

- input() 함수의 입력 자료형
  - type() 함수로 자료형 알아보기

```
>>> print(type(string))  
<class 'str'>
```

```
>>> number = input("숫자를 입력하세요> ")  
숫자를 입력하세요> 12345   
>>> print(number)  
12345
```

```
>>> print(type(number))  
<class 'str'>
```

- input() 함수는 사용자가 무엇을 입력해도 결과는 무조건 문자열 자료형

## – 예시 – 입력 자료형 확인하기

```
01  # 입력을 받습니다.  
02  string = input("입력> ")  
03  
04  # 출력합니다.  
05  print("자료:", string)  
06  print("자료형:", type(string))
```

실행결과 1

입력> 52273

자료: 52273

자료형: <class 'str'>

실행결과 2

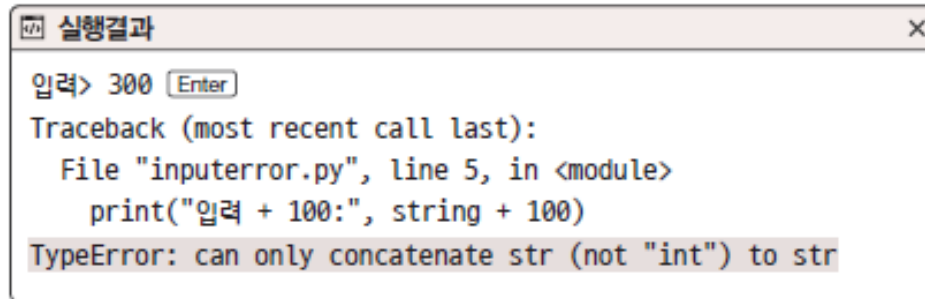
입력> True

자료: True

자료형: <class 'str'>

## – 예시 – 입력받고 더하기

```
01  # 입력을 받습니다.  
02  string = input("입력> ")  
03  
04  # 출력합니다.  
05  print("입력 + 100:", string + 100)
```



The screenshot shows a terminal window with the title '실행결과'. The user has entered '300' at the '입력>' prompt. The program then raises a `TypeError` because it attempts to concatenate a string with an integer. The error message is: `TypeError: can only concatenate str (not "int") to str`.

```
입력> 300 [Enter]  
Traceback (most recent call last):  
  File "inputerror.py", line 5, in <module>  
    print("입력 + 100:", string + 100)  
TypeError: can only concatenate str (not "int") to str
```



- **캐스트** (cast)

- input() 함수의 입력 자료형은 항상 문자열이므로 입력받은 문자열을 숫자 연산에 활용하기 위해 숫자로 변환
- **int() 함수**
  - 문자열을 int 자료형으로 변환.
- **float() 함수**
  - 문자열을 float 자료형으로 변환

## – 예시 – int() 함수 활용하기

```
01 string_a = input("입력A> ")
02 int_a = int(string_a)
03
04 string_b = input("입력B> ")
05 int_b = int(string_b)
06
07 print("문자열 자료:", string_a + string_b)
08 print("숫자 자료:", int_a + int_b)
```



실행결과

입력A> 273 Enter

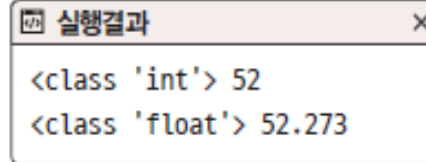
입력B> 52 Enter

문자열 자료: 27352

숫자 자료: 325

- 예시 - int() 함수와 float() 함수 활용하기

```
01 output_a = int("52")
02 output_b = float("52.273")
03
04 print(type(output_a), output_a)
05 print(type(output_b), output_b)
```



```
<class 'int'> 52
<class 'float'> 52.273
```

- 예시 - int() 함수와 float 함수 조합하기

```
01 input_a = float(input("첫 번째 숫자> "))
02 input_b = float(input("두 번째 숫자> "))
03
04 print("덧셈 결과:", input_a + input_b)
05 print("뺄셈 결과:", input_a - input_b)
06 print("곱셈 결과:", input_a * input_b)
07 print("나눗셈 결과:", input_a / input_b)
```



```
첫 번째 숫자> 273 Enter
두 번째 숫자> 52 Enter
덧셈 결과: 325.0
뺄셈 결과: 221.0
곱셈 결과: 14196.0
나눗셈 결과: 5.25
```

- ValueError 예외

- 변환할 수 없는 것을 변환하려 할 경우
- 숫자가 아닌 것을 숫자로 변환하려 할 경우

```
int("안녕하세요")  
float("안녕하세요")
```

## ❗ 오류

```
Traceback (most recent call last):  
  File "intconvert.py", line 2, in <module>  
    int_a = int(string_a)  
ValueError: invalid literal for int() with base 10: '안녕하세요'
```

- 소수점이 있는 숫자 형식의 문자열을 int() 함수로 변환하려 할 때

```
int("52.273")
```

**오류**

```
Traceback (most recent call last):
```

```
File "intconvert.py", line 2, in <module>
```

```
    int_a = int(string_a)
```

```
ValueError: invalid literal for int() with base 10: '52.273'
```

- `str()` 함수
  - 숫자를 문자열로 변환

`str(다른 자료형)`

```
01 output_a = str(52)
02 output_b = str(52.273)
03 print(type(output_a), output_a)
04 print(type(output_b), output_b)
```

실행결과

```
<class 'str'> 52
<class 'str'> 52.273
```

- **변수 선언** : 변수를 생성하는 것을 의미
- **변수 할당** : 변수에 값을 넣는 것을 의미
- **변수 참조** : 변수에서 값을 꺼내는 것
- **input() 함수** : 명령 프롬프트에서 사용자로부터 데이터 입력 받음
- **int() 함수** : 문자열을 int 자료형으로 변환
- **float 함수** : 문자열을 float 자료형으로 변환
- **str() 함수** : 숫자를 문자열로 변환

- 변수에 값을 할당하기 위한 구문입니다. 빈칸에 알맞은 기호를 쓰세요.

변수 이름  값

- 숫자에 적용할 수 있는 복합 대입 연산자입니다. 왼쪽 연산자 항목에 알맞은 기호를 써 보세요.

연산자	내용
<input type="text"/>	숫자 덧셈 후 대입
<input type="text"/>	숫자 뺄셈 후 대입
<input type="text"/>	숫자 곱셈 후 대입
<input type="text"/>	숫자 나눗셈 후 대입
<input type="text"/>	숫자 나머지 구한 후 대입
<input type="text"/>	숫자 제곱 후 대입



- 다음 코드는 inch 단위의 자료를 입력 받아 cm를 구하는 예제입니다. 빈칸에 알맞은 내용을 넣어 코드를 완성해 주세요. (1inch = 2.54cm)

```
str_input =  ("숫자 입력> ")  
num_input =  (str_input)  
  
print()  
print(num_input, "inch")  
print((num_input * 2.54), "cm")
```

실행결과 1

숫자 입력> 1

1.0 inch

2.54 cm

실행결과 2

숫자 입력> 26

26.0 inch

66.04 cm

- 원의 반지름을 입력 받아 원의 둘레와 넓이를 구하는 코드입니다. 빈칸에 알맞은 내용을 넣어 코드를 완성해 주세요.
  - 둘레 :  $2 * \text{원주율} * \text{반지름}$
  - 넓이 :  $\text{원주율} * \text{반지름} * \text{반지름}$

```
str_input =  ("원의 반지름 입력> ")
num_input =  (str_input)
print()
print("반지름: ", num_input)
print("둘레: ", 2 * 3.14 * )
print("넓이: ", 3.14 *  ** 2)
```

실행결과 1

원의 반지름 입력> 2

반지름: 2.0  
둘레: 12.56  
넓이: 12.56

실행결과 2

원의 반지름 입력> 5

반지름: 5.0  
둘레: 31.400000000000002  
넓이: 78.5