02

Variables

Python built-in objects

Object type	Example literals/creation
Numbers	1234, 3.1415, 0b111, 1_234, 3+4j, Decimal, Fraction
Strings	'code', "app's", b'a\x01c', 'h\u00c4ck', 'hÄck'
Lists	[1, [2, 'three'], 4.5], list(range(10))
Tuples	(1, 'app', 4, 'U'), tuple('hack'), namedtuple
Dictionaries	{'job': 'dev', 'years': 40}, dict(hours= 10)
Sets	set('abc'), {'a', 'b', 'c'}
Files	open('docs.txt'), open(r'C:\data.bin', 'wb')
Other core objects	Booleans, types, None
Program-unit objects	Functions, modules, classes
Implementation objects	Compiled code, stack tracebacks

Mark Lutz. (2025). Learning Python, 6th Edition. n.p.: O'Reilly Media, Inc.

Numeric types

Interpretation	Example literals
Integers (unlimited size)	1234, -24, 0, 9_999_999_999_999
Floating-point numbers	1.23, 1., 3.14e-10, 4E210, 4.0e+210
Complex number	3+4j, 3.0+4.0j, 3J (실수부 + 허수부)

Numeric types

```
>>> type(1234)
<class ‘int’>
>>> type(3.1415)
<class ‘float’>
>>> type(0b111)
<class ‘int’>
>>> type(1_234)
<class ‘int’>
>>> type(3+4j)
<class ‘complex’>
```

Type conversion

변환하고자 하는 자료형을 명시

이때 허수 → 실수 변환은 불가능

```
float(1234)
```

```
1234.0
```

```
int(3.1415)
```

```
3
```

```
complex(3)
```

```
(3+0j)
```

```
int(3+4j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 int(3+4j)

TypeError: int() argument must be a string, a bytes-like object or a real number, not 'complex'
```

Fix Code

Arithmetic Operators

산술 연산자(Arithmetic Operator)란?

- 사칙연산을 다루는 연산자
- 두 개의 피연산자(연산의 대상)를 가져야 하는 이항 연산자

  10 + 3: 10, 3은 피연산자, +는 연산자에 해당

- + : 더하기
- : 빼기
- * : 곱하기
- / : 나누기
- // : 정수 몫
- % : 나머지
- ** : 거듭제곱

예) 10 = 3 * 3 + 1

Arithmetic Operators

정수 ↔ 정수

```
10 + 3
```

```
13
```

```
10 - 3
```

```
7
```

```
10 * 3
```

```
30
```

```
10 / 3
```

```
3.3333333333333335
```

```
10 // 3
```

```
3
```

```
10 % 3
```

```
1
```

```
10 ** 3
```

```
1000
```

정수 ↔ 소수

```
10 + 3.0
```

```
13.0
```

```
10 - 3.0
```

```
7.0
```

```
10 * 3.0
```

```
30.0
```

```
10 / 3.0
```

```
3.3333333333333335
```

```
10 // 3.0
```

```
3.0
```

```
10 % 3.0
```

```
1.0
```

```
10 ** 3.0
```

```
1000.0
```

Arithmetic Operators

```
>>> 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1
1
>>> 0.1+0.2
0.3
>>> 0.2+0.4
0.6
```

In fact,

```
>>> 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1
0.9999999999999999
>>> 0.1+0.2
0.30000000000000004
>>> 0.2+0.4
0.6000000000000001
```

Floating Point

10진법(Decimal)

$29 = 20 * 10^{1} + 9 * 10^{0}$

16진법(Hexadecimal)

$29 = 1*16^{1} + 13*16^{0} = 0x1d$

8진법(Octal)

$29 = 3*8^{1} + 5*8^{0} = 0o35$

2진법(Binary) 2의 거듭제곱으로 표시

컴퓨터는 2진수로 숫자를 저장

$29 = 16 + 8 + 4 + 1$

$= 1*2^{4} + 1*2^{3} + 1*2^{2} + 0*2^{1} + 1*2^{0}$

$= 11101_{(2)} = 0b11101$

Floating Point

- 고정 소수점
  - 고정된 소수점 위치 표현 가능한 범위: -32,768 ~32,767 (정수 16bit, 소수 16bit 할당)
- 부동 소수점
  - 소수점 위치를 지수로 표현
  - 소수점 위치 유동적이므로 더 정밀하게 표현 가능 표현 가능한 범위: $\pm 1.18*10^{38}$ ~ $\pm 3.4*10^{38}$

이진수로 정확히 나타내지 못하는 수 존재

0.1 = 0.0001100110011001100...

0.2 = 0.0011001100110011001...

→ 근사값으로 저장된 두 수를 연산하면 누적 오차 발생

Memory of the objects

- Integer

  저장되는 숫자가 커질수록 더 많은 메모리 사용

- Float

  부동 소수점을 사용해 항상 같은 메모리 사용

*sys 모듈의 getsizeof() 함수를 사용하여 특정 객체가 차지하는 메모리의 양 확인 가능

```
>>> import sys
>>> sys.getsizeof(1234)
28
>>> sys.getsizeof(3.1415)
24
>>> sys.getsizeof(0b111)
28
>>> sys.getsizeof(1_234)
28
>>> sys.getsizeof(3+4j)
32
```

```
>>> sys.getsizeof(2 ** 50)
32
>>> sys.getsizeof(2 ** 100)
40
>>> sys.getsizeof(2 ** 200)
52
>>> sys.getsizeof(2 ** 400)
80
```

print()

print()

- 화면에 값을 출력할 때 사용
- 모든 자료형 출력 가능

```
>>> print(1, 1.0)
1, 1.0
>>> print("Hello World!")
Hello World!
```

Variables

변수(Variable): 데이터값을 저장하는 저장소

```
x = 10
print(x)
```

```
10
```

```
y = x
print(y)
```

```
10
```

Variables

- 변수 이름은 글자 또는 '_'로 시작해야 한다
- 변수 이름은 숫자로 시작해서는 안된다
- 변수 이름은 알파벳과 숫자, '_' 만 포함 가능하다
- 변수 이름은 대소문자 구분이 필요하다 (age, Age, 그리고 AGE는 각각 다른 변수)
- Python keywords에 속하는 단어들은 변수 이름으로 사용될 수 없다

  (and, if, for, etc.)

변수로 사용가능한 이름: myvar, my_var, _my_var, myVar, MYVAR, myvar2

사용할 수 없는 이름: 2myvar, my-var, my var

Variables

변수 활용 연산 예제

```
a = 10
b = 3
```

```
a + b
```

```
13
```

```
a - b
```

```
7
```

```
a / b
```

```
3.3333333333333335
```

```
a // b
```

```
3
```

```
a % b
```

```
1
```

Assignment Operators

Operator	Example	Same As
=	x = 5	x =5
+=	x += 3	x =x +3
-=	x -= 3	x =x -3
*=	x *= 3	x =x* 3
/=	x /= 3	x =x/ 3
%=	x %= 3	x =x%3
//=	x //= 3	x =x// 3
**=	x **= 3	x =x** 3

Variables

대입 연산자 활용 예제

```
c = a + b
print(c)
```

```
13
```

```
c += 10
print(c)
```

```
23
```

```
c *= a
print(c)
```

```
230
```

2주차 과제

- 02_Variables.ipynb