

파이썬 프로그래밍

수치형 자료형, 문자열 자료형



한국기술교육대학교
온라인평생교육원

■ 수치 자료형

1. 정수형 상수

```
a = 23 # 10진 정수
b = 023 # 8진 정수
c = 0x23 # 16진 정수
print type(a), type(b), type(c)
print a, b, c
```

```
<type 'int'> <type 'int'> <type 'int'>
23 19 35
```

- 정수형 상수: 소수 영역이 없는 수
- a=23 → a에 23이라는 정수형 상수 할당(23은 10진 정수)
- 숫자 앞에 숫자 0을 붙이면 8진 정수
- 숫자 앞에 0x(숫자 0, 문자 x)를 붙이면 16진 정수
- a, b, c의 타입 → int(정수형 상수)
- print a, b, c → a, b, c에 할당된 정수가 출력됨
- 10진 정수는 그대로 출력
- 8진·16진 정수는 10진 정수로 값이 바뀌어 출력

```
import sys
print sys.maxint # 최대 정수 값 확인
```

```
9223372036854775807
```

- 그렇다면 정수형 상수로 나타낼 수 있는 최대 정수는?
- sys 모듈의 maxint를 확인하면 알 수 있음

■ 수치 자료형

2. 실수형 상수

```
a = 1.2
b = 3.5e3
c = -0.2e-4
print type(a), type(b), type(c)
print a, b, c
```

```
<type 'float'> <type 'float'> <type 'float'>
1.2 3500.0 -2e-05
```

- 실수형 상수: 소수 영역이 있는 수
- $3.5e3 = 3.5 \times 10^3 = 3500$ ($e3 \rightarrow 10^3 = 1000$)
- 실수형 상수는 3500이 아닌 3500.0
- $e-4 \rightarrow 10^{-4} = 1/10000$
- $-0.2e-4 = -0.2 \times (1/10000) = -2 \times (1/100000) = -2e-5$
- a, b, c의 타입 \rightarrow float(실수형 상수)
- print a, b, c \rightarrow 각 표현 방식에 따라 출력됨
- $3.5e3 \rightarrow 3500.0$
- $-0.2e-4 \rightarrow -2e-5$
- print로 인한 변화는 str(repr) 내장 함수에 의한 것

■ 수치 자료형

3. 롱형 상수

```
h1 = 123456789012345678901234567890L
# 마지막에 L을 붙여서 명시적으로 long 형이라고 알려도 되고
print type(h1)
print h1 * h1
print
```

```
<type 'long'>
15241578753238836750495351562536198787501905199875019052100
```

- 롱형: 정수형 상수의 최대 범위를 벗어난 수
- 롱형은 메모리가 허용하는 한 유효자리 수가 무한
- 롱형을 명시하는 방법 → 숫자 끝에 L을 붙여줌
- `print h1*h1` → 메모리가 허용하는 한에서 출력됨

```
h2 = 123456789012345678901234567890
# L을 붙이지 않아도 int형이 담을 수 있는 수치를 초과하면 자동으로 long형이 된다.
print type(h2)
print h2 * h2
print
```

```
<type 'long'>
15241578753238836750495351562536198787501905199875019052100
```

- 숫자 끝에 L을 붙이지 않아도 `maxint`를 넘으면 롱형
- `print h2*h2` → 메모리가 허용하는 한에서 출력됨

■ 수치 자료형

3. 롱형 상수

```
h3 = 123L  
print type(h3)  
print
```

```
h4 = 123  
print type(h4)
```

```
<type 'long'>
```

```
<type 'int'>
```

- 작은 수여도 숫자 끝에 L을 붙이면 롱형
- 숫자 끝에 L을 안 붙이면 정수형 상수(int)로 인식
- 숫자 끝에 L을 붙이지 않아도 maxint를 넘으면 롱형
- 자동으로 L이 나타나는 것은 ipython 노트북의 기능
- 일반적인 코딩 환경에서는 나타나지 않음

■ 수치 자료형

4. 복소수형 상수

```
a = 10 + 20j  
print a
```

```
b = 10 + 5j  
print a + b
```

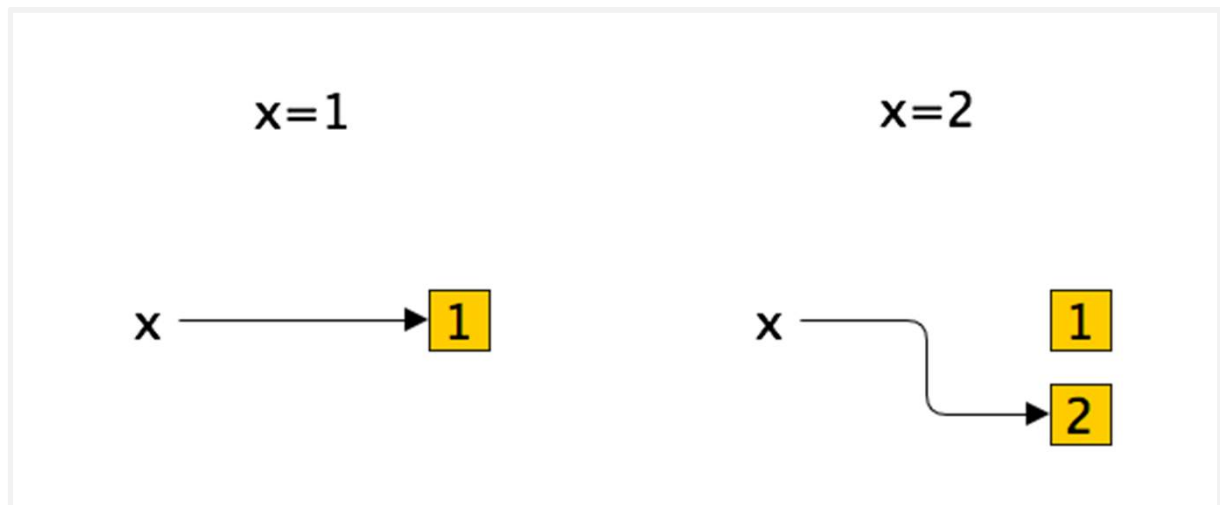
```
(10+20j)  
(20+25j)
```

- 복소수형 상수: 실수부(ex. 10)와 허수부(ex. 20j)로 구성
- 복소수형 상수는 실수부와 허수부가 각각 계산됨

■ 수치 자료형

5. 수치 자료형의 치환

```
x = 1  
x = 2
```



- x=1로 입력한 후 x=2로 치환할 때
- x=1 입력 → x가 1을 가리키도록 하는 주소(레퍼런스) 생김
- 그 후 x=2 입력 → x가 2를 가리키도록 하는 주소(레퍼런스) 생김
- 그리고 기존에 1을 가리키던 주소는 없어짐
- 이 때 1은 쓰레기(garbage)가 되어 수집 가능

▣ 수치 자료형

6. 수치 연산과 관련된 내장 함수

```
print abs(-3)
print int(3.141592)
print int(-3.1415)
print long(3)
print float(5)
print complex(3.4, 5)
print complex(6)
```

```
3
3
-3
3
5.0
(3.4+5j)
(6+0j)
```

- `print abs(숫자)` → 숫자의 절대값이 출력됨
- `print int(숫자)` → 숫자의 정수형이 출력됨
- -3.9999가 -4에 가깝지만 정수형은 -3
- `print long(숫자)` → 숫자의 롱형이 출력됨(L은 나타나지 않음)
- `print float(숫자)` → 숫자의 실수형이 출력됨
- `print complex(a, b)` → $(a+bj)$ 와 같이 복소수형이 출력됨
- `print complex(a)` → $(a+0j)$ 와 같이 실수부에만 숫자 나타남

■ 수치 자료형

6. 수치 연산과 관련된 내장 함수

```
print divmod(5, 2)
print pow(2, 3)
print pow(2.3, 3.5)
```

```
(2, 1)
8
18.4521691056
```

- `print divmod(a, b)` → (a를 b로 나눈 몫, a를 b로 나눈 나머지)
- `print pow(a, b)` → a의 b승 값이 출력됨
- `pow` 함수는 실수(소수 부분이 있는 수)도 입력이 가능함

■ 수치 자료형

7. math 모듈의 수치 연산 함수

```
import math
print math.pi
print math.e
print math.sin(1.0) # 1.0 라디안에 대한 사인 값
print math.sqrt(2) # 제곱근
```

```
3.14159265359
2.71828182846
0.841470984808
1.41421356237
```

- math 모듈은 수학적으로 정의된 변수, 함수 지원됨
- math.pi → 파이 값 / math.e → 지수 값
- math.sin → 사인 값 / math.sqrt → 제곱근 값

■ 수치 자료형

7. math 모듈의 수치 연산 함수

```
r = 5.0 # 반지름
a = math.pi * r * r # 면적

degree = 60.0
rad = math.pi * degree / 180.0 # 각도를 라디안으로 변환
print math.sin(rad), math.cos(rad), math.tan(rad) #sin, cos, tan
```

```
0.866025403784 0.5 1.73205080757
```

- 반지름이 5인 원 a의 면적을 구하는 공식
- 각도가 60도일 때 라디안 값을 구하는 공식
- 그 후 라디안 값을 싸인, 코싸인, 탄젠트 값으로 출력
- math 모듈로 여러 가지 수치 연산 활용 가능