# 기초 PYTHON 프로그래밍

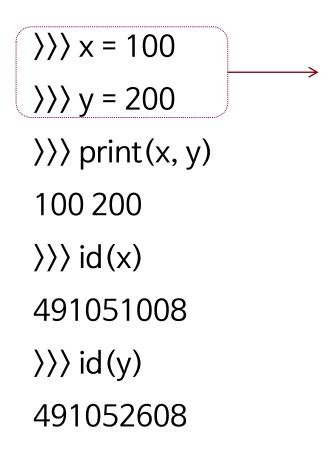
## 3. 수치 자료형과 연산자

- 1. 정수 자료형과 연산
- 2. 실수 자료형과 연산
- 3. 복소수 자료형과 연산
- 4. 자료형 변환
- 5. 수치 연산 함수들
- 6. math 모듈

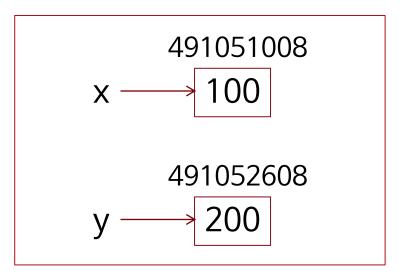




#### ◆ 정수 (int) 표현하기



$$\rangle\rangle\rangle$$
 x = 100; y = 200

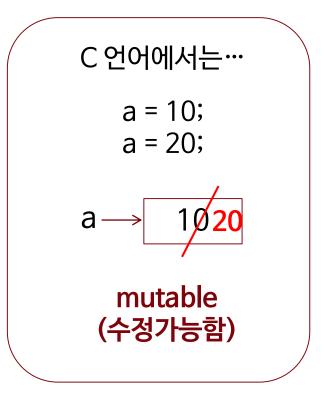




◆ 정수 객체는 immutable 하다.

#### immutable: 객체를 수정할 수 없음

$$\rangle\rangle\rangle$$
 a = 10  
 $\langle\rangle\rangle\rangle$  id (a)  
 $\langle\langle\rangle\rangle\rangle\rangle$  a = 20  
 $\langle\langle\rangle\rangle\rangle\rangle$  id (a)  
 $\langle\langle\rangle\rangle\rangle\rangle$  id (a)





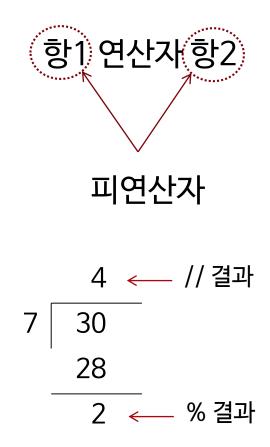
◆ 정수 자료형은 크기 제한이 없다.

〉〉〉 2 \*\* 1024 # 2의 1024제곱 



#### ◆ 산술 연산자

연산자	의미	예	결과
+	더하기	10 + 5	15
_	빼기	20 - 13	7
*	곱하기	3 * 10	30
/	나누기	100 / 8	12.5
**	지수계산	2 ** 5	32
//	몫	30 // 7	4
%	나머지	30 % 7	2





#### ◆ 산술 연산자 우선순위

연산자	설명	결합 순서
**	지수	<b>←</b>
* / % //	곱하기, 나누기, 나머지, 몫	<b>→</b>
+ -	더하기, 빼기	<b>→</b>

```
>>>> 2 + 3 * 5
17
>>>> 2 ** 3 ** 2 # 2 ** 9
512
>>>> (2 ** 3) ** 2 # 8 ** 2
64
>>>> 2 ** 3 * 4
32
>>>> 4 * 3 ** 2
36
```



◆ 할당 연산자와 산술 연산자 예제

```
〉〉〉〉 a = 10

〉〉〉 b = 20

〉〉〉 c = a + b  # a와 b를 더하여 c에 저장하시오

〉〉〉 a = a + 50  # a의 값을 50 증가하시오

〉〉〉 b = b + a  # b의 값을 a 만큼 증가하시오

〉〉〉 print(a, b, c)

60 80 30
```



#### ◆ 산술 연산자 간략히 쓰기

a = a + b	a += b
a = a - b	a -= b
a = a * b	a *= b
a = a / b	a /= b
a = a ** b	a **= b
a = a // b	a //= b
a = a % b	a %= b

$\rangle\rangle$ a = 10; b	0 = 5; $c = 22$ ; $d = 3$		
$\rangle\rangle\rangle$ a += c	# a가 32가 됨		
$\rangle\rangle\rangle$ b -= d	# b가 2가 됨		
⟩⟩⟩ a //= 3	# a가 10이됨		
⟩⟩⟩ c %= 5	# c가 2가 됨		
$\rangle\rangle\rangle$ a = 3; b = 2; c = 5; d = 10			
$\rangle\rangle\rangle$ d += b + 6	c-a**b #d가8이됨		



- ◆ 실수 (float) 표현하기
  - 소수점으로 표현하기

```
>>> a = 10.5
>>> b = 11.
>>> c = .5
>>> print(a,b,c)
10.5 11.0 0.5
```

과학적 표기 방법 (scientific notation)

```
\rangle\rangle\rangle 2.5e5 # 2.5 * 10<sup>5</sup> 250000.0 \rangle\rangle\rangle 3.25E-4 # 3.25 * 10<sup>-4</sup> 0.000325
```



#### ◆ 실수의 특징

- 실수 객체도 immutable하다.
- 실수는 저장할 때 약간의 문제를 일으킬 수도 있다.



#### ◆ 실수의 연산

- 정수 자료형에서 사용하는 연산자를 모두 사용할 수 있다.
- 정수와 실수 자료형을 같이 연산하면 결과는 실수가 된다.

```
\rangle\rangle\rangle 2.3 + 3.7
      6.0
  \text{\chi}\text{\sigma} 5.5 - 1.2
  4.3
\rangle\rangle\rangle 1.5 * 2.1
    3.1500000000000004
  \langle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\r
      2.0
    \langle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\r
    3.660092227792233
    6.0
      >>> 20.5 % 3.1
        1.89999999999999
```

```
⟩⟩⟩ a = 10 # 정수
>>> b = 20.0 # 실수
\rangle\rangle c = a + b
\rangle\rangle print(a, b, c)
10 20.0 30.0
\rangle\rangle type(c)
(class 'float')
```

## 3. 복소수 자료형과 연산



◆ 복소수 표현하기 (허수부에 j 또는 J를 붙인다)

$$\rangle\rangle\rangle x = 2 + 5j$$
  
 $\rangle\rangle\rangle y = 3.2 + 2.5J$   
 $\rangle\rangle\rangle z = x + y$   
 $\rangle\rangle\rangle print(z)$   
 $(5.2+7.5j)$   
 $\rangle\rangle\rangle type(z)$   
 $\langle class 'complex' \rangle$ 

## 3. 복소수 자료형과 연산



#### ◆ 복소수 연산하기

```
    〉〉〉 x = 5 + 10j
    〉〉〉 x.real # 실수부를 알려준다
    5.0
    〉〉〉 x.imag # 허수부를 알려준다
    10.0
    〉〉〉 x.conjugate() # 켤레복소수 반환한다
    (5-10j)
```

## 4. 자료형 변환



#### 실수, 문자 → 정수로 변환하기 : int() 내장함수



## 4. 자료형 변환



◆ 정수, 문자 → 실수로 변환하기: float() 내장함수

```
\rangle\rangle\rangle x = float(3)
\rangle\rangle\rangle y = float(100)
\rangle\rangle\rangle z = float('15.7')
\rangle\rangle print(x, y, z)
3.0 100.0 15.7
\rangle\rangle type(x); type(y); type(z)
(class 'float')
(class 'float')
(class 'float')
```

## 5. 수치 연산 함수들



#### ◆ 수치 연산 관련 내장함수

함수	설명
abs(x)	x의 절대값을 반환한다
divmod(x,y)	(x//y, x%y) 쌍을 반환한다
pow(x,y)	x <sup>y</sup> 을 반환한다

```
>>> abs(-3)
3

>>> divmod(17,4)
(4, 1)

>>> pow(2,5)
32
```

#### 6. math 모듈

5



◆ <mark>모듈 - 연관된 함수들을 모아서 모듈로 관리한다</mark>

```
>>> import math
\rangle\rangle math.fabs(-3)
3.0
\rangle\rangle math.pow(2,5)
32.0
\rangle\rangle math.sqrt(16)
4.0
〉〉〉 math.floor(4.5) # 4.5 이하의 정수 중에서 가장 큰 정수
4
〉〉〉 math.ceil(4.5) # 4.5 이상의 정수 중에서 가장 작은 정수
```