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

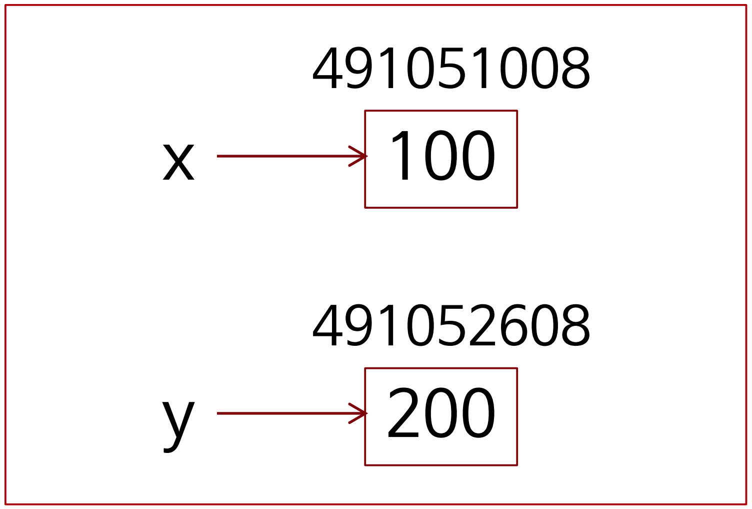
1. 정수 자료형과 연산

◆ 정수 (int) 표현하기

```
〉〉〉 x = 100
〉〉〉 y = 200
```

```
〉〉〉 x = 100 ; y = 200
```

```
〉〉〉 print(x, y)
100 200
〉〉〉 id(x)
491051008
〉〉〉 id(y)
491052608
```

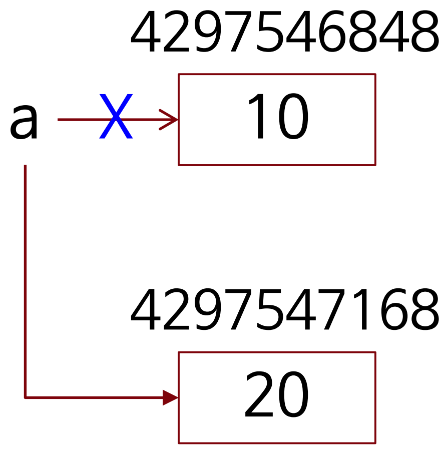


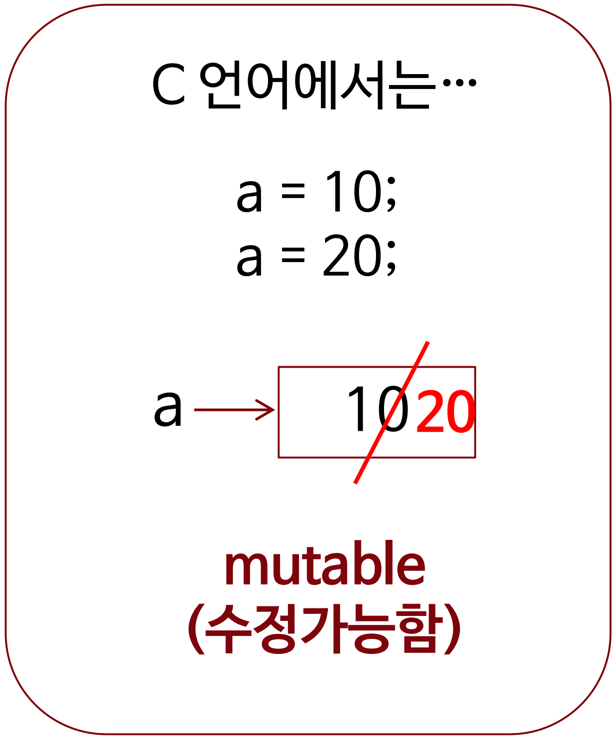
1. 정수 자료형과 연산

◆ 정수 객체는 **immutable** 하다.

immutable : 객체를 수정할 수 없음

```
>>> a = 10
>>> id(a)
4297546848
>>> a = 20
>>> id(a)
4297547168
```





1. 정수 자료형과 연산

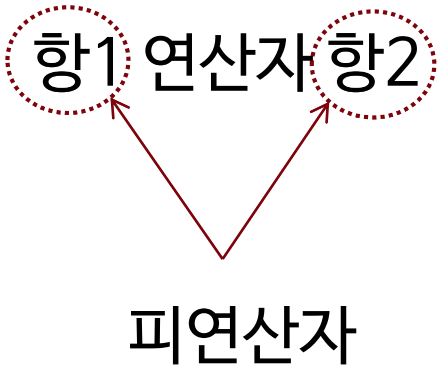
◆ 정수 자료형은 크기 제한이 없다.

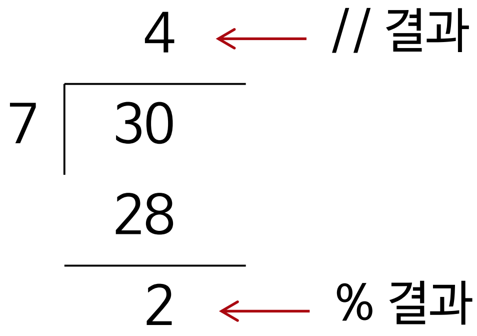
```
〉〉〉 2 ** 1024       # 2의 1024제곱
1797693134862315907729305190789024733617976978942306
5727343008115773267580550096313270847732240753602
1120113879871393357658789768814416622492847430639
47412437776789342486548527630221960124609411945308
29520850057688381506823424628814739131105408272371
63350510684586298239947245938479716304835356329624
224137216
```

1. 정수 자료형과 연산

◆ 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





1. 정수 자료형과 연산

◆ 산술 연산자 우선순위

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

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

1. 정수 자료형과 연산

◆ 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
```

1. 정수 자료형과 연산

◆ 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

```
>>> a = 10 ; b = 5 ; c = 22 ; d = 3
>>> a += c       # a가 32가 됨
>>> b -= d       # b가 2가 됨
>>> a //= 3      # a가 10이됨
>>> c %= 5       # c가  2가 됨

>>> a = 3; b = 2; c = 5; d = 10
>>> d += b + c - a ** b   # d가 8이 됨
```

2. 실수 자료형과 연산

◆ 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)

```
>>> 2.5e5       # 2.5 * 10^5
250000.0
>>> 3.25E-4   # 3.25 * 10^-4
0.000325
```

2. 실수 자료형과 연산

◆ 실수의 특징

- 실수 객체도 **immutable**하다.
- 실수는 저장할 때 약간의 문제를 일으킬 수도 있다.

```
>>> x = 1.5
>>> id(x)
57809984
>>> x = 2.0
>>> id(x)
61313536
```

```
>>> 0.1 + 0.1
0.2
>>> 2 * 0.2
0.4

>>> 0.2 + 0.1
0.30000000000000004
>>> 3 * 0.1
0.30000000000000004
```

2. 실수 자료형과 연산

◆ 실수의 연산

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

```
>>> 2.3 + 3.7
6.0
>>> 5.5 - 1.2
4.3
>>> 1.5 * 2.1
3.1500000000000004
>>> 20.2 / 10.1
2.0
>>> 1.5 ** 3.2
3.660092227792233
>>> 20.5 // 3.1
6.0
>>> 20.5 % 3.1
1.8999999999999995
```

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

3. 복소수 자료형과 연산

◆ 복소수 표현하기 (허수부에 j 또는 J를 붙인다)

```
>>> x = 2 + 5j
>>> y = 3.2 + 2.5J
>>> z = x + y
>>> print(z)
(5.2+7.5j)
>>> type(z)
<class 'complex'>
```

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

```
>>> a = int(5.6)      # 소수점 뒤를 무조건 버림
>>> b = int(-2.9)
>>> c = int('15')     # 문자 '15'를 정수로 변환함
>>> print(a,b,c)
5 -2 15
>>> score = '85'
>>> score + 5         # TypeError 발생
>>> int(score) + 5
90
```

4. 자료형 변환

- 정수, 문자 → 실수로 변환하기 : float() 내장함수

```
〉〉〉 x = float(3)
〉〉〉 y = float(100)
〉〉〉 z = float('15.7')
〉〉〉 print(x, y, z)
3.0 100.0 15.7
〉〉〉 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^y을 반환한다

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

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

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

6. math 모듈

◆ 모듈 - 연관된 함수들을 모아서 모듈로 관리한다

```
>>> import math
>>> math.fabs(-3)
3.0
>>> math.pow(2,5)
32.0
>>> 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
5
```