

2. 사칙 연산

파이썬의 사칙 연산 규칙은 일반 수학 연산 규칙과 동일

$$30 - (10 + 20) * 3 + \frac{100}{2^2}$$

변수를 이용한 연산도 동일한 규칙을 따른다

+, - : 더하기, 빼기

/, // : 나누기 (소숫점 포함, 몫구하기)

% : 나머지

\$*\$: 곱하기

\$**\$: 거듭제곱

() 안의 계산을 먼저한다

In [1]:

```
1 30 - (10 + 20) * 3 + 100 / 2 ** 2
```

Out[1]:

-35.0

In [2]:

```
1 a = 30
2 b = 10
3 c = 20
4 d = 3
5 x = 100
6 k = 2
7
8 y = a - (b + c) * d + x / k ** 2
9
10 print(y)
```

-35.0

In [3]:

```
1 x = x + 1
2 print(x)
```

101

cascading assignment (계단식 할당)

exponential assignment (+= , -= ,

In [4]:

```
1 z = y = x = 1 + 2 + 3
2 print(x, y, z)
```

6 6 6

In [5]:

```
1 print (a, b)
```

30 10

value swapping

In [6]:

```
1 a , b = b, a
2 print(a, b)
```

10 30

In [7]:

```
1 2 ** 3
```

Out[7]:

8

exponential

In [8]:

```
1 2 ** 3 ** 4
```

Out[8]:

2417851639229258349412352

In [9]:

```
1 (2 ** 3) ** 4
```

Out[9]:

4096

In [10]:

```
1 3 ** 400
```

Out[10]:

```
705507910865533257124642715759347962165079496127873157628712232092
620855515829341565792985294471341581549523348253559118669297930718
24566694145084454535257027960285323760313192443283334088001
```

In [11]:

```
1 3.0 ** 400
```

Out[11]:

```
7.055079108655333e+190
```

In [12]:

```
1 time = 50
2 day = time / 24
3 day
```

Out[12]:

```
2.0833333333333335
```

나눗셈

floor division

In [13]:

```
1 day = time // 24
2 day
```

Out[13]:

```
2
```

modulo (나머지)

In [24]:

```
1 remainder = 30 % 24
2 remainder
```

Out[24]:

```
6
```

+=, -=, *=, /=, %= : augmented assignment

In [25]:

```
1 x = 1
```

In [26]:

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

Out[26]:

2

In [27]:

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

Out[27]:

3

In [28]:

```
1 y = 10
```

In [29]:

```
1 y = y - 1
2 y
```

Out[29]:

9

In [30]:

```
1 y -= 1
2 y
```

Out[30]:

8

In [21]:

```
1 z = 5
```

In [22]:

```
1 z = z * 2
2 z
```

Out[22]:

10

In [23]:

```
1 z *= 2
2 z
```

Out[23]:

20

print 함수 내의 연산

In [36]:

```
1 integer1 = 11
2 integer2 = 5
3 print(integer1 * integer2)
```

55

In [37]:

```
1 print(integer1 // integer2)
```

2

In [38]:

```
1 print(integer1 % integer2)
```

1

In [39]:

```
1 print(integer1 / integer2)
```

2.2