아티니어(Artineer)

찍어먹는 Python 맛만보자!

-입문반-

1. 연산자

2. 산술 연산자

3. 관계 연산자

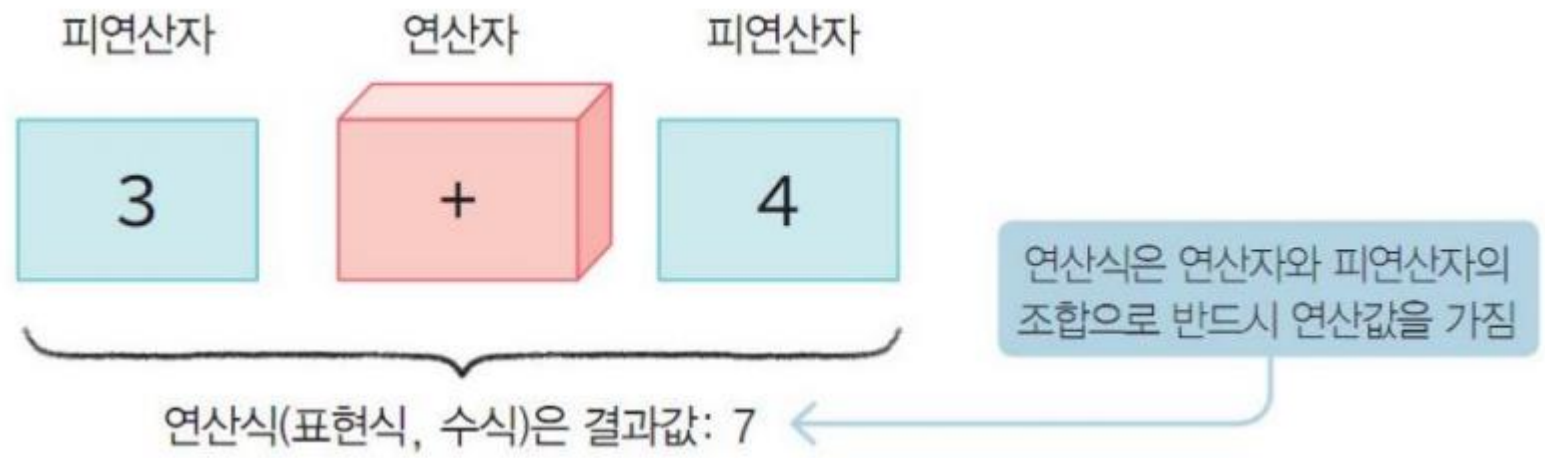
4. 논리 연산자

5. 비트 연산자

연산자

- 식 : 계산을 표현하는 기본적 수단으로 연산자, 피연산자, 괄호, 함수 호출 등으로 구성
- 연산자 : 연산을 수행하기 위한 기호나 키워드
- 피연산자 : 연산에 참여하는 변수나 값

예) '+'는 연산자, 3과 4는 피연산자



연산자 기능에 따른 연산자의 종류

연산자의 종류	연산자
산술 연산자	+, −, *, /, %
증감 연산자	++, −−
관계 연산자	>, <, >=, <=, ==, !=
논리 연산자	&&, ||, !
비트 연산자	&, |, ^, ~, <<, >>
대입 연산자	=, +=, −=, *=, /=, %=, &=, |=, ^=, >>=, <<=
조건 연산자	?:
그 밖의 연산자	,(콤마 연산자), sizeof, 형 변환 연산자

산술 연산자

수학적인 계산을 위해 사용하는 연산자

연산자	의미	사용 예	설명
+	더하기	a = 5 + 3	5와 3을 더한 값을 a에 대입
-	빼기	a = 5 - 3	5에서 3을 뺀 값을 a에 대입
*	곱하기	a = 5 * 3	5와 3을 곱한 값을 a에 대입
/	나누기	a = 5 / 3	5를 3으로 나눈 값을 a에 대입
//	나누기(몫)	a = 5 // 3	5를 3으로 나눈 후 소수점을 버리고 값을 a에 대입
%	나머지값	a = 5 % 3	5를 3으로 나눈 후 나머지값을 a에 대입
**	제곱	a = 5 ** 3	5의 3제곱을 a에 대입

산술 연산자 사용 예

```
a = 5
b = 3
print(a+b, a-b, a*b, a/b, a//b, a%b, a**b)
```

출력 결과는?

출력 결과

```
8 2 15 1.6666666666666667 1 2 125
```

산술 연산자 사용 예

```
a, b, c = 2, 3, 4
Print(a+b-c)
Print(a+b*c)
Print(a*b/c)
```

출력 결과는?

출력 결과

```
1 14 1.5
```

a+b-c

- 덧셈과 뺄셈은 연산자 우선순위가 동일하므로 어떤 것을 먼저 계산하든 동일
- 특별히 괄호가 없을 때는 왼쪽에서 오른쪽 방향으로 계산 -> (a+b)-c

a+b*c

- 덧셈(뺄셈)과 곱셈(나눗셈)이 같이 있으면 곱셈(나눗셈) 먼저 계산
- 괄호가 있다면 괄호 우선
- 방향은 왼쪽에서 오른쪽

산술 연산을 하는 문자열과 숫자의 상호 변환

```
s1, s2, s3 = "100", "100.123", "99999999999999999999999999"
print(int(s1) + 1, float(s2) + 1, int(s3) + 1)
```

출력 결과

```
101 101.123 100000000000000000000000000
```

숫자를 문자열로 변환

```
a = 100; b = 100.123
str(a) + '1'; str(b) + '1'
```

출력 결과

```
'1001'
'100.1231'
```

산술 연산자와 대입 연산자

- 대입 연산자는 연산자의 좌변(변수)에 우변의 값을 저장
- 좌변에는 반드시 변수만 사용
- 대입 연산자 = 외에도 산술 연산자와 함께 사용하는 대입 연산자 제공
- 파이썬에는 증가 연산자(++), 감소 연산자 (--)가 없음

연산자	사용 예	설명
+=	a += 3	a = a + 3과 동일
-=	a -= 3	a = a - 3과 동일
*=	a *= 3	a = a * 3과 동일
/=	a /= 3	a = a / 3과 동일
//=	a //= 3	a = a // 3과 동일
%=	a %= 3	a = a % 3과 동일
**=	a **= 3	a = a ** 3과 동일

```
1  ## 변수 선언 부분 ##
2  money, c500, c100, c50, c10 = 0, 0, 0, 0, 0
3
4  ## 메인 코드 부분 ##
5  money = int(input("교환할 돈은 얼마?"))
6
7  c500 = money // 500
8  money %= 500
9
10 c100 = money // 100
11 money %= 100
12
13 c50 = money // 50
14 money %= 50
15
16 c10 = money // 10
17 money %= 10
18
19 print("\n 500원짜리 ==> %d개" % c500)
20 print(" 100원짜리  ==> %d개" % c100)
21 print(" 50원짜리 ==> %d개" % c50)
22 print(" 10원짜리  ==> %d개" % c10)
23 print(" 바꾸지 못한 잔돈 ==> %d원 \n" % money)
```

2행 : 동전으로 교환할 돈(money)과 500원, 100원, 50원, 10원짜리 동전의 개수를 저장 할 변수 초기화

7행 : 500원짜리 동전의 개수를 구함

8행 : 다시 money를 500으로 나눈 후 나머지 값 저장

8행의 money%=500은 money=money%500과 동일

10~11행 : 100원짜리 동전을, 13~14행에서 50원짜리 동전을, 16~17행에서 10원짜리 동전을 구함

마지막 money에 저장된 값은 10 미만으로 바꿀 수 없는 나머지 돈

```
교환할 돈은 얼마?9876

500원짜리 ==> 19개
100원짜리 ==> 3개
50원짜리 ==> 1개
10원짜리 ==> 2개
바꾸지 못한 잔돈 ==> 6원
```

관계 연산자

- 어떤 것이 크거나 작거나 같은지 비교

! 참은 Ture, 거짓은 False로 표시

- 조건문이나 반복문에서 사용하며 단독으로는 거의 사용하지 않음

연산자	의미	설명
==	같다.	두 값이 동일하면 참
!=	같지 않다.	두 값이 다르면 참
〉	크다.	왼쪽이 크면 참
〈	작다.	왼쪽이 작으면 참
〉=	크거나 같다.	왼쪽이 크거나 같으면 참
〈=	작거나 같다.	왼쪽이 작거나 같으면 참

관계 연산자 사용 예

```
a, b = 100, 200
print(a == b , a != b,  a > b , a < b , a >= b , a <= b)
```

출력 결과

```
False True False True False True
```

비트 연산자

- 컴퓨터 내부적으로는 모든 숫자와 문자를 비트로 변환하여 저장
- 정수를 2진수로 변환한 후 각 자리의 비트끼리 연산 수행

비트 연산자의 종류

연산자	의미	설명
&	비트 논리곱(and)	둘 다 1이면 1
|	비트 논리합(or)	둘 중 하나만 1이면 1
^	비트 논리적 배타합(xor)	둘이 같으면 0, 다르면 1
~	비트 부정	1은 0으로, 0은 1로 변경
《	비트 이동(왼쪽)	비트를 왼쪽으로 시프트(Shift)
》	비트 이동(오른쪽)	비트를 오른쪽으로 시프트(Shift)

비트 연산자 사용 예

```
10 & 7
123 & 456
0xFFFF & 0x0000
```

출력 결과

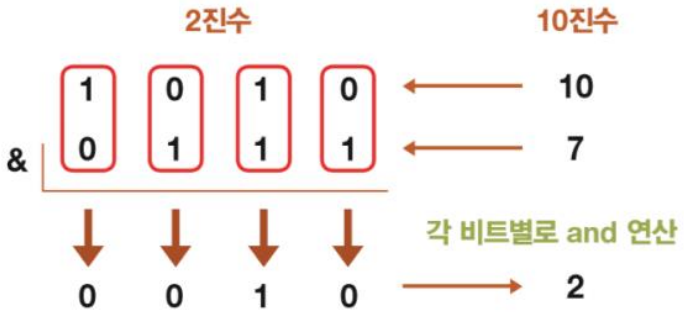
```
2 72 0
```

- 123&456은 123의 2진수인 1111011, 456의 2진수인 111001000의 & 결과인 1001000가 되므로 10진수 72 출력(자릿수가 다를 때 빈 자리에 0으로 채운 후 연산)
- 0과 비트 논리곱 수행하면 어떤 숫자든 무조건 0 출력

비트 논리곱

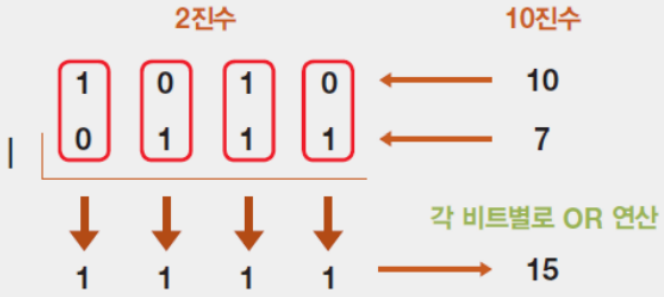
- &는 비트 논리곱을 수행한 결과가 나옴
- 0는 False, 1은 True

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1



비트 논리합

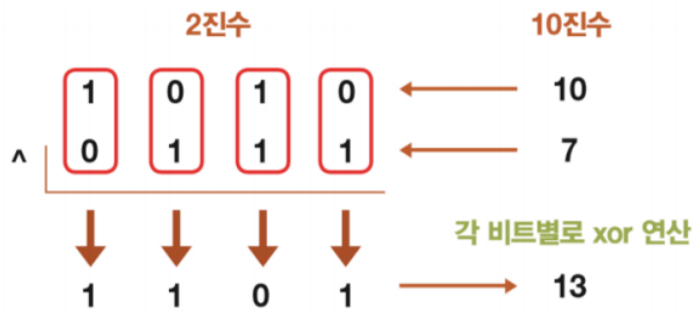
A	B	A|B
0	0	0
0	1	1
1	0	1
1	1	1



비트 배타적 논리합

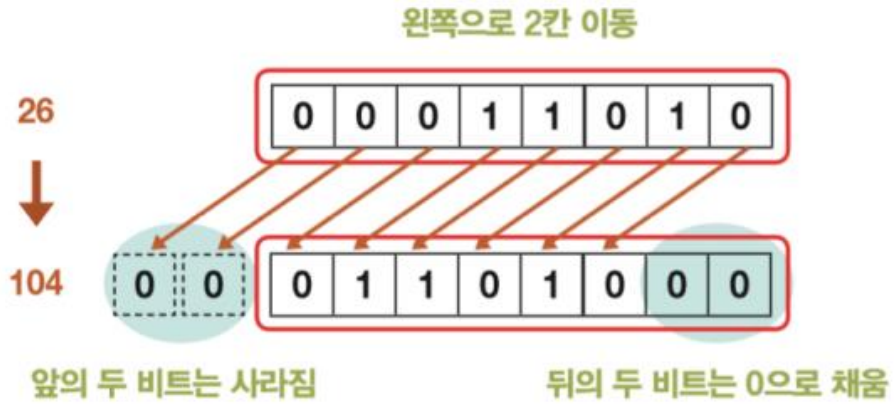
- 두 값이 다르면 1, 같으면 0

A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0



시프트 연산자

- 비트로 나열된 값으로 오른쪽이나 왼쪽으로 이동
- 왼쪽 시프트 연산자 : 왼쪽으로 시프트 할 때 마다 2^n을 곱한 효과




```
a = 10
a << 1; a << 2; a << 3; a << 4
```

출력 결과

```
20 40 80 160
```

- 오른쪽 시프트 연산자


```
a = 10
a >> 1; a >> 2; a >> 3; a >> 4
```

출력 결과

```
5 2 1 0
```

연산자 우선순위

- 여러 개의 연산자가 있을 경우 정해진 순서

우선순위	연산자	의미
1	() [] {}	괄호, 리스트, 딕셔너리, 세트 등
2	**	지수
3	+ - ~	단항 연산자
4	* / % //	산술 연산자
5	+ -	
6	<< >>	비트 시프트 연산자
7	&	비트 논리곱
8	^	비트 배타적 논리합
9	|	비트 논리합
10	< > >= <=	관계 연산자
11	== !=	동등 연산자
12	= %= /= //= -= += *= **=	대입 연산자
13	not	논리 연산자
14	and	
15	or	
16	if~else	비교식

끝!