



파이썬기초

3주차 - 1교시
파이썬의 연산자





학습내용

- 파이썬의 기본 연산
- 연산자의 활용



학습목표

- 기본 연산자에 대해 이해하고 활용할 수 있다.
- 연산자를 이용해서 여러 형태의 연산 수식을 만들 수 있다.



생각해 봅시다

프로그램 내부에서 데이터를 이용해서
새로운 데이터를 만들기 위한 방법들을
생각해봅시다.



01



파이썬의 기본 연산



1 | 연산과 연산자

01 연산

연산

하나 이상의 대상으로부터 새로운 것을 만드는 것

출처 | [네이버 백과사전]

1 | 연산과 연산자

02 연산자

연산자

연산을 하기 위한 기호, 즉 새로운 결과물을 만들기 위한 기호

a + b



피연산자

a + b



연산자

→ a와 b는 피연산자, + 기호는 연산자

1 | 연산과 연산자

03 연산자의 종류

| 연산자의 종류 | 설명 |
|---------|-------------------------|
| 산술연산자 | 숫자 타입의 데이터의 연산 |
| 관계연산자 | 데이터들의 비교연산을 수행 |
| 논리연산자 | 논리값의 연산 |
| 비트연산자 | 데이터의 2진수 연산 |
| 대입연산자 | 특정 값을 변수에 대입해주는 연산을 수행 |
| 멤버연산자 | 시퀀스 객체에 특정 요소의 포함여부를 확인 |
| 식별연산자 | 두 개체의 메모리 위치를 비교 |

1 | 연산과 연산자

04 산술연산자

| 연산기호 | 설명 | 사용 예($x=10, y=20, z=3$) |
|------|-----|---------------------------|
| + | 더하기 | $x + y \rightarrow 30$ |
| - | 빼기 | $x - y \rightarrow -10$ |
| * | 곱하기 | $x * y \rightarrow 200$ |
| / | 나누기 | $y / x \rightarrow 2.0$ |
| % | 나머지 | $y \% x \rightarrow 0$ |
| ** | 제곱 | $x ** z \rightarrow 1000$ |
| // | 몫 | $x // z \rightarrow 3$ |

1 | 연산과 연산자

04 산술연산자

```
In [1]:
```

```
x = 10
```

```
y = 20
```

```
z = 3
```

1 | 연산과 연산자

04 산술연산자

```
In [2]: print (x+y)
        print (x-y)
        print (x*y)
        print (y/x)
        print (y%x)
        print (x**z)
        print (x//z)
```

결과(Console)

```
30
-10
200
2.0
0
1000
3
```

1 | 연산과 연산자

05 관계연산자

- 대소 관계와 동등여부를 나타내는 연산자로 논리값을 결과값으로 반환
- “같다”의 기호는 "=="를 사용
- “같지 않다”라는 기호는 "!="를 사용

1 | 연산과 연산자

05 관계연산자

| 연산기호 | 설명 | 사용 예(x=10, y=20) |
|------|------------------------|-----------------------------------|
| == | 값이 같다 | $x == y \rightarrow \text{False}$ |
| != | 값이 같지 않다 | $x != y \rightarrow \text{True}$ |
| > | 왼쪽 값이 오른쪽 값보다 크다 | $x > y \rightarrow \text{False}$ |
| < | 왼쪽 값이 오른쪽 값보다 작다 | $x < y \rightarrow \text{True}$ |
| >= | 왼쪽 값이 오른쪽 값보다 크거나 동일하다 | $x >= y \rightarrow \text{False}$ |
| <= | 왼쪽 값이 오른쪽 값보다 작거나 동일하다 | $x <= y \rightarrow \text{True}$ |

1 | 연산과 연산자

05 관계연산자

```
In [2]: print (x==y)
        print (x!=y)
        print (x>y)
        print (x<y)
        print (x>=y)
        print (x<=y)
```

결과(Console)

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

1 | 연산과 연산자

06 논리연산자

- 여러 개의 관계연산자를 사용할 때 사용

| 연산기호 | 설명 | 사용 예(x = True, y= False) |
|------|---------------------------|--------------------------|
| and | 논리 AND 연산. 둘 다 참일 때만 참 | x and y = False |
| or | 논리 OR 연산. 둘 중 하나만 참 이어도 참 | x or y = True |
| not | 논리 NOT 연산. 논리 상태를 반전 | not (x and y) = True |

1 | 연산과 연산자

06 논리연산자

• 논리연산 값

| a | b | a and b | a or b |
|-------|-------|---------|--------|
| true | true | true | true |
| true | false | false | true |
| false | true | false | true |
| false | false | false | false |

1 | 연산과 연산자

06 논리연산자

```
In [2]: x = True  
        y = False
```

```
In [2]: print(x and y)  
        print(x or y)  
        print(not (x and y))
```

```
False  
True  
True
```


1 | 연산과 연산자

07 대입연산자

| 연산기호 | 설명 | 사용 예($x = 10, y = 20$) |
|------|--------------------------------|-----------------------------------|
| = | 왼쪽 변수에 오른쪽 값을 할당 | $z = x + y \rightarrow z = x + y$ |
| += | 왼쪽 변수에 오른쪽 값을 더하고 결과를 왼쪽변수에 할당 | $z += x \rightarrow z = z + x$ |
| -= | 왼쪽 변수에서 오른쪽 값을 빼고 결과를 왼쪽변수에 할당 | $z -= x \rightarrow z = z - x$ |
| *= | 왼쪽 변수에 오른쪽 값을 곱하고 결과를 왼쪽변수에 할당 | $z *= x \rightarrow z = z * x$ |

1 | 연산과 연산자

07 대입연산자

| 연산기호 | 설명 | 사용 예(x = 10, y= 20) |
|------|-------------------------------------|-----------------------------------|
| /= | 왼쪽 변수에서 오른쪽 값을 나누고 결과를 왼쪽변수에 할당 | $z /= x \rightarrow z = z / x$ |
| %= | 왼쪽 변수에서 오른쪽 값을 나눈 나머지의 결과를 왼쪽변수에 할당 | $z \% = x \rightarrow z = z \% x$ |
| **= | 왼쪽 변수에 오른쪽 값만큼 제곱을 하고 결과를 왼쪽변수에 할당 | $z ** = x \rightarrow z = z ** x$ |
| //= | 왼쪽 변수에서 오른쪽 값을 나눈 몫의 결과를 왼쪽변수에 할당 | $z //= x \rightarrow z = z // x$ |

1 | 연산과 연산자

07 대입연산자

```
In [8]: x = 10  
        y = 20  
        z = 0
```

1 | 연산과 연산자

07 대입연산자

```
In [9]: z = x + y  
print(z)  
z += x # z = z+x  
print(z)  
z -= x # z = z-x  
print(z)  
z *= x # z = z * x  
print(z)
```

결과(Console)

```
30  
40  
30  
300
```

1 | 연산과 연산자

07 대입연산자

```
In [9]: z %= 8 # z = z % 8  
print(z)  
z **= x # z = z ** x  
print(z)  
z //= x # z = z // x  
print(z)
```

결과(Console)

```
4  
1048576  
104857
```

1 | 연산과 연산자

08 비트연산자

• $x = 10$ (0000 1010), $y = 20$ (0001 0100)

bin() 함수

2진수로 변환



1 | 연산과 연산자

08 비트연산자

| 연산기호 | 설명 | 사용 예 |
|------|-------------------------|--|
| & | AND 연산. 둘 다 참일 때만 만족 | $(x \& y) = 0 \rightarrow 0000\ 0000$ |
| | OR 연산. 둘 중 하나만 참 이여도 만족 | $(x y) = 30 \rightarrow 0001\ 1110$ |
| ^ | XOR 연산. 둘 중 하나만 참일 때 만족 | $(x \wedge y) = 30 \rightarrow 0001\ 1110$ |
| ~ | 보수 연산. | $(\sim x) = -11 \rightarrow 1111\ 0101$ |

1 | 연산과 연산자

08 비트연산자

| 연산기호 | 설명 | 사용 예 |
|------|---|---------------------------------------|
| << | 왼쪽 시프트 연산자. 변수의 값을 왼쪽으로 지정된 비트 수 만큼 이동 | $x \ll 2 = 40 \rightarrow 0010\ 1000$ |
| >> | 오른쪽 시프트 연산자. 변수의 값을 오른쪽으로 지정된 비트 수 만큼 이동 | $x \gg 2 = 2 \rightarrow 0000\ 0010$ |

1 | 연산과 연산자

08 비트연산자

```
In [6]: x = 10  
        y = 20  
        bin(10) #00001010  
        bin(20) #00010100  
        bin(30) #00011110  
        '0b11110'
```

1 | 연산과 연산자

08 비트연산자

```
In [6]: print (x&y)
        print (x|y)
        print (x^y)
        print (~x)
        print (x<<2)
        print (x>>2)
```

결과(Console)

```
0
30
30
-11
40
2
```

1 | 연산과 연산자

09 멤버연산자

• $x = 10, y = 20, \text{list} = [10, 20, 30, 40, 50]$

| 연산기호 | 설명 | 사용 예 |
|--------|-----------------------|--|
| in | list 내에 포함되어 있으면 참 | $(x \text{ in list}) = \text{True}$ |
| not in | list 내에 포함되어 있지 않으면 참 | $(y \text{ not in list}) = \text{False}$ |

1 | 연산과 연산자

09 멤버연산자

```
In [10]: x = 10  
         y = 20  
         list = [10, 20, 30, 40, 50]
```

```
In [11]: print(x in list)  
         print(x not in list)
```

True
False

1 | 연산과 연산자

10 식별연산자

- 두 개체의 **메모리 위치**를 비교함
- $x = 10, y = 10$

| 연산기호 | 설명 | 사용 예 |
|--------|-----------------------|--|
| is | 개체메모리 위치나 값이 같다면 참 | $(x \text{ is } y) = \text{True}$ |
| is not | 개체메모리 위치나 값이 같지 않다면 참 | $(x \text{ is not } y) = \text{False}$ |

1 | 연산과 연산자

10 식별연산자

In [12]:

```
x = 10
```

```
y = 10
```

```
z = 20
```

In [13]:

```
print(x is y)
```

```
print(x is z)
```

```
print(x is not y)
```

True

False

False

1 | 연산과 연산자

11 연산자의 우선순위

- 가장 높은 우선 순위부터 모든 연산자를 보여 줌

| 연산자 | 설명 |
|----------|------------------|
| ** | 지수 (전원으로 인상) |
| + - | 단항 플러스와 마이너스 |
| * / % // | 곱하기, 나누기, 나머지, 몫 |
| + - | 덧셈과 뺄셈 |
| >> << | 좌우 비트 시프트 |

1 | 연산과 연산자

11 연산자의 우선순위

- 가장 높은 우선 순위부터 모든 연산자를 보여 줌

| 연산자 | 설명 |
|--------------------------|-----------------------|
| & | 비트 'AND' |
| ^ | 비트 전용 'OR'와 정기적인 'OR' |
| <= < > >= | 비교 연산자 |
| < > == != | 평등 연산자 |
| = %= /= //= -= += *= **= | 할당 연산자 |

1 | 연산과 연산자

11 연산자의 우선순위

- 가장 높은 우선 순위부터 모든 연산자를 보여 줌

| 연산자 | 설명 |
|------------|--------|
| is is not | 식별 연산자 |
| in not in | 멤버 연산자 |
| not or and | 논리 연산자 |

02



연산자의 활용



1 | 코딩해보기

01 과목별 점수의 총합과 평균 및 총점과 평균 출력

- 국어, 영어, 수학 점수의 총합과 평균을 구하고 각 과목의 점수와 총점, 평균을 출력해보자.

```
In [12]: kor_score = 100
          eng_score = 70
          math_score = 80
          sum_score = kor_score + eng_score + math_score
          avg_score = sum_score / 3
```

1 | 코딩해보기

01 과목별 점수의 총합과 평균 및 총점과 평균 출력

- 국어, 영어, 수학 점수의 총합과 평균을 구하고 각 과목의 점수와 총점, 평균을 출력해보자.

```
In [12]: print('국어 점수', kor_score)
          print('영어 점수', eng_score)
          print('수학 점수', eng_score)
          print('총점', sum_score)
          print('평균 점수', avg_score)
```

결과(Console)

```
국어 점수 100
영어 점수 70
수학 점수 70
총점 250
평균 점수
83.33333333333333
```

1 | 코딩해보기

02 성별이 남성이고 18세 이상의 성인 여부를 판단하는 조건식

- 성별이 남성이고 18세 이상의 성인 여부를 판단하는 조건식을 만들어 봅시다.



성별 판별

gender = male | female



나이 판별

age = 숫자

1 | 코딩해보기

02 성별이 남성이고 18세 이상의 성인 여부를 판단하는 조건식

- 성별이 남성이고 18세 이상의 성인 여부를 판단하는 조건식을 만들어 봅시다.

```
In [18]: gender = 'male'
age = 20
check = gender == 'male' and age >= 18
print(check)
```

True



파이썬기초

NEXT
조건문

