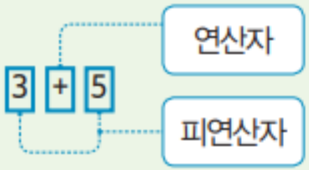
Chapter 04. 연산

연산자

02. 연산자

I. 연산자의 종류

- 연산자와 피연산자로 이루어지는 계산식을 수식이라고 부름



- 파이썬 프로그래밍에는 사칙연산을 비롯한 다양한 산술 연산자와 비교 연산자, 논리 연산자, 대입 연산자 등을 사용

표 4-3 다양한 연산자 종류

산술 연산자	+ − * / // % **
비교 연산자	〈 〉 〈= 〉= == !=
논리 연산자	and or not
대입 연산자	= += −= *= /= //= %=

01. 연산자

I. 연산자의 종류

▪ 산술 연산자

- 산술 연산자에는 덧셈(+), 뺄셈(-), 곱셈(*), 나눗셈(/)의 사칙연산 외에도 몫 (//)과 나머지(%)를 구하거나 제곱을 구하는 지수(**) 연산도 포함

표 4-4 산술 연산자의 종류

연산자	예시	의미	실행 결과
+	3 + 5	3과 5의 더하기	8
-	3 - 5	3에서 5를 빼기	-2
*	3 * 5	3과 5의 곱하기	15
/	3 / 5	3을 5로 나누기	0.6
//	3 // 5	3을 5로 나눈 몫	0
%	3 % 5	3을 5로 나눈 나머지	3
**	3 ** 5	3의 5제곱	243

✓ **TIP** 산술 연산자 사이에서는 지수(**), 곱셈 ·나눗셈 ·몫 ·나머지(*, /, //, %), 덧셈 ·뺄셈(+, -) 연산자의 순서로 우선순위가 정해짐

02. 연산자

I. 연산자의 종류

- 산술 연산자

실습 4-3 몫과 나머지 계산하기

① 나눠지는 수와 나누는 수를 정수형으로 입력받아 변수 x와 y에 각각 저장

```
>>> x = int(input("나눠지는 수 : "))
나눠지는 수 : 3000
>>> y = int(input("나누는 수 : "))
나누는 수 : 500
```

② 몫과 나머지 계산 수식을 실행한 결과를 화면에 출력

```
>>> print("%d를 %d로 나눈 몫 = %d" %(x, y, x//y))
3000를 500로 나눈 몫 = 6
>>> print("%d를 %d로 나눈 나머지 = %d" %(x, y, x%y))
3000를 500로 나눈 나머지 = 0
```

02. 연산자

I. 연산자의 종류

▪ 산술 연산자

실습 4-4 원의 면적 구하기

- 원의 면적은 s = π*r*r

① 원의 반지름을 실수형으로 입력받아 변수 radius에 저장

```
>>> radius = float(input("원의 반지름 : "))
원의 반지름 : 3.5
```

② 공식에 맞게 수식을 작성하고 계산한 원의 면적을 변수 area에 저장

```
>>> area = 3.14 * radius ** 2
```

③ 원의 면적 area를 실수형 포맷 코드를 사용하여 화면에 출력

```
>>> print("원의 면적 = %f" % area)
원의 면적 = 38.465000
```

02. 연산자

I. 연산자의 종류

▪ 비교 연산자

- 2개의 피연산자를 서로 비교하여 참이나 거짓으로 판별하는 수식(조건식)에 사용

표 4-5 비교 연산자의 종류

연산자	예시	의미	실행 결과
<	3 < 5	3은 5보다 작다	True
>	3 > 5	3은 5보다 크다	False
<=	3 <= 5	3은 5보다 작거나 같다	True
>=	3 >= 5	3은 5보다 크거나 같다	False
==	3 == 5	3은 5와 같다	False
!=	3 != 5	3은 5와 다르다	True

✓ **TIP** 비교 연산자는 관계 연산자라고도 부름

02. 연산자

I. 연산자의 종류

▪ 비교 연산자

실습 4-5 나이를 기준으로 영화 매표 여부 확인하기

- 영화관에서 17세 이상 관람 가능한 영화를 판매한다고 가정

① 나이를 입력받고 변수 age에 저장

```
>>> age = int(input("나이를 입력하세요 : "))
나이를 입력하세요 : 21
```

② 비교 연산자를 이용하여 관람 가능한 나이인지 판단해서 출력

```
>>> print("매표 가능 여부 : ", age >= 17)
매표 가능 여부 :  True
```

실습 4-6 입력한 정수가 3의 배수인지 확인하기

- 정수를 하나 입력받고 그 수가 3의 배수이면 'True'를 출력하는 프로그램을 작성

02. 연산자

I. 연산자의 종류

- 비교 연산자

① 정수를 하나 입력받아 변수 x에 저장

```
>>> x = int(input("정수를 입력하세요 : "))
정수를 입력하세요 : 12345678
```

② 입력한 정수 x를 3으로 나누어 나머지를 변수 remain에 저장

```
>>> remain = x % 3
```

③ 변수 remain에 저장된 값이 0인지 비교 연산자를 이용해서 확인 후 출력

```
>>> print("x는 3의 배수입니까? : ", remain == 0)
x는 3의 배수입니까? :  True
```

✓ **TIP** '=='는 '같다'를 의미하는 비교 연산자이고, '='는 변수에 값을 저장하는 대입 연산

02. 연산자

I. 연산자의 종류

▪ 논리 연산자

- 2개 이상의 조건이 결합되어 전체 식의 True와 False를 판단하는 형태

표 4-6 논리 연산자의 종류

연산자	예시	예시의 조건	실행 결과
and	x and y	x와 y가 모두 True인 경우	True
		x와 y가 하나 이상 False인 경우	False
or	x or y	x와 y가 하나 이상 True인 경우	True
		x와 y가 모두 False인 경우	False
not	not x	x가 True인 경우	False
		x가 False인 경우	True

I. 연산자의 종류

▪ 논리 연산자

실습 4-7 학점 계산하기

- 점수가 60점 미만이거나 결석이 4회 이상이면 F 학점으로 판정하는 프로그램

① 점수와 결석 횟수를 입력받아 변수에 저장

```
>>> score = float(input("점수 입력 : "))
점수 입력 : 85.7
>>> absence = int(input("결석 횟수 입력 : "))
결석 횟수 입력 : 1
```

② 논리 연산자를 활용한 수식으로 F 학점 여부를 판단해서 출력

```
>>> print("F 학점 여부 = ", (score < 60) or (absence >= 4))
F 학점 여부 =  False
```

02. 연산자

I. 연산자의 종류

▪ 논리 연산자

실습 4-8 아이디와 비밀번호로 로그인하기

- 아이디와 비밀번호를 입력하고 로그인할 수 있는지 알려주는 프로그램

① 가입할 때 사용한 아이디와 비밀번호를 임의로 정해서 변수에 저장

```
>>> id = "hong"
>>> password = "1234"
```

② 로그인하려는 아이디와 비밀번호를 입력하고 변수에 각각 저장

```
>>> input_id = input("아이디를 입력하세요 : ")
아이디를 입력하세요 : hong
>>> input_pass = input("비밀번호를 입력하세요 : ")
비밀번호를 입력하세요 : 12345
```

③ 아이디와 비밀번호가 모두 일치하는지 판단해서 로그인 여부를 출력

```
>>> print("로그인 여부 : ", id == input_id and password == input_pass)
로그인 여부 :  False
```

02. 연산자

I. 연산자의 종류

▪ 대입 연산자

- 대입 연산자의 오른쪽에는 저장할 값이나 수식 혹은 다른 변수 이름을 사용
- 대입 연산자의 오른쪽에 수식이 있는 경우, 수식의 계산 결과가 변수에 저장
- 대입 연산자의 왼쪽에는 값이나 수식을 사용하면 안 되고 반드시 하나의 변수 이름만 사용

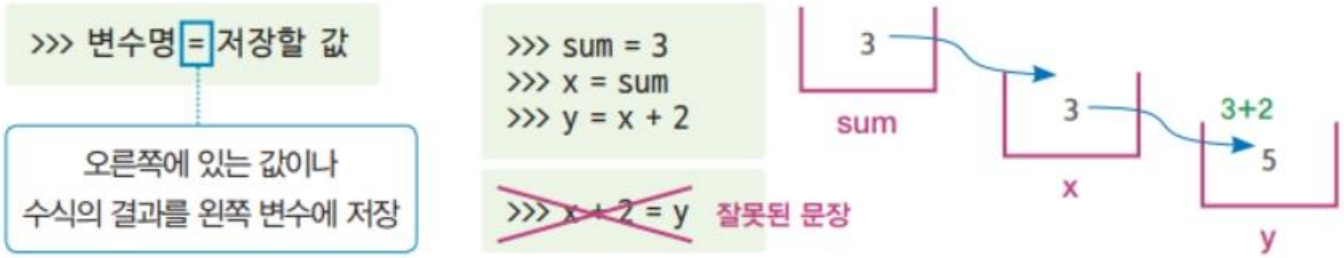


그림 4-12 대입 연산자의 규칙

02. 연산자

I. 연산자의 종류

▪ 대입 연산자

- 대입 연산자의 오른쪽에 다른 변수명이 오면 값이 복사되는데, 그림과 같이 하나의 수식에 대입 연산자를 여러 번 사용 가능
- 다음 수식에서는 변수 c에 10을 먼저 저장하고, 그 값이 다른 변수 b와 a에 차례대로 복사

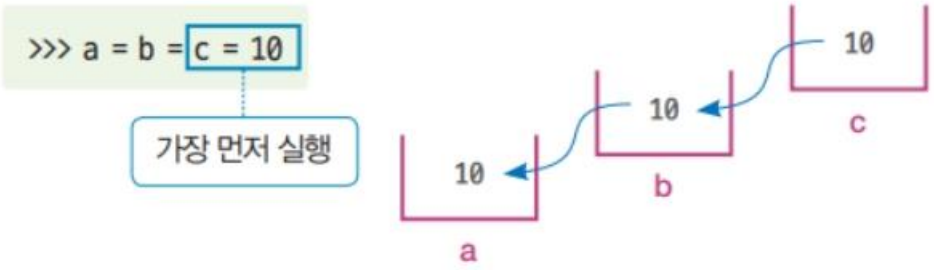


그림 4-13 대입 연산자의 실행 순서

02. 연산자

I. 연산자의 종류

▪ 대입 연산자

- 대입 연산자와 산술 연산자를 함께 표기한 연산자들이 복합 대입 연산자

표 4-7 복합 대입 연산자의 종류

연산자	예시	의미	실행 결과
+=	x += 5	x = x + 5	x의 값이 5 증가
-=	x -= 5	x = x - 5	x의 값이 5 감소
*=	x *= 5	x = x * 5	x는 5배수 값으로 변경
/=	x /= 5	x = x / 5	x를 5로 나눈 값으로 변경
//=	x //= 5	x = x // 5	x를 5로 나눈 몫으로 변경
%=	x %= 5	x = x % 5	x를 5로 나눈 나머지로 변경

I. 연산자의 종류

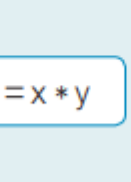
▪ 대입 연산자

실습 4-9 복합 대입 연산자 사용하기

- 변수 x와 y에 다음과 같이 복합 대입 연산자를 활용한 수식을 만들어 실행

① 변수 x와 y에 각각 정수 10과 5를 저장하고 수식을 만들어 실행

```
>>> x = 10
>>> y = 5
>>> x += y
>>> x *= y
>>> x %= y
```



② 변수 x와 y를 출력하여 변경된 값을 확인

```
>>> print("x = %d\ny = %d" %(x, y))
x = 0
y = 5
```

02. 연산자

I. 연산자의 종류

하나 더 알기 연산자의 우선순위

연산자의 우선순위

앞서 살펴본 여러 가지 연산자들이 수식에 같이 사용될 때는 연산자 우선순위에 따라 실행 순서가 결정된다. 산술 > 비교 > 논리 > 대입 연산자의 순으로 우선순위가 높다. 하지만 어떤 상황에서도 가장 먼저 실행해야 할 부분은 괄호( ) 안에 있는 수식이다.

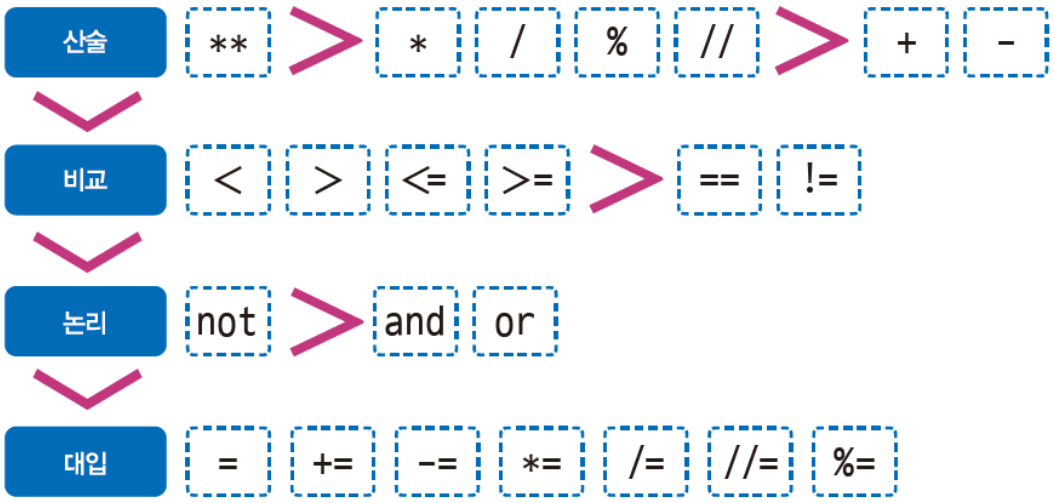


그림 4-14 연산자의 우선순위

II. 연산자의 활용

실습 4-10 정다각형의 내각의 합과 한 내각의 크기 계산하기 code04-10.py

- 정수 n(3<=n<=6)을 입력하면 정n각형의 내각의 합과 한 내각의 크기를 계산해서 출력하는 프로그램

①

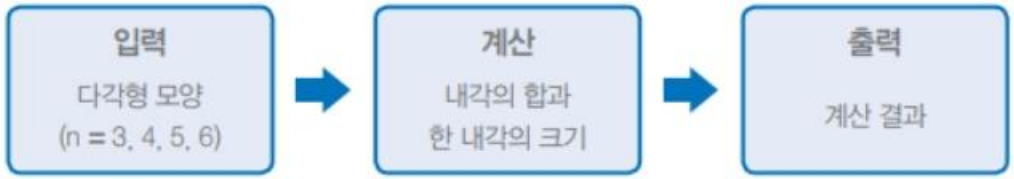


그림 4-15 내각의 합과 한 내각의 크기 계산 순서

②

표 4-8 내각의 합과 크기 계산

정n각형	내각의 합	한 내각의 크기
3 <= n <= 6	180/(n-2)	180*(n-2)/n

II. 연산자의 활용

실습 4-10 정다각형의 내각의 합과 한 내각의 크기 계산하기 code04-10.py

③ 코드 편집기를 열고, 실행 순서와 정해진 계산식에 맞게 필요한 동작을 기술

```
01  n = int(input("3에서 6 사이의 정수를 입력하세요 : "))   # 다각형 모양 선택
02  sumAngle = 180 * (n - 2)                              # 내각의 합 계산
03  intAngle = 180 * (n - 2) / n                          # 한 내각의 크기
04  print("정%d각형의 내각의 합 = %d도" % (n, sumAngle))
05  print("정%d각형의 한 내각의 크기 = %d도" % (n, intAngle))
```

④ 파일을 저장하고 코드 편집기 [Run] 메뉴의 [Run Module]을 실행

```
3에서 6 사이의 정수를 입력하세요 : 6
정6각형의 내각의 합 = 720도
정6각형의 한 내각의 크기 = 120도
```

II. 연산자의 활용

하나 더 알기 **주석의 이용**

- 프로그램을 작성할 때 참고할 사항이나 필요한 설명을 자유롭게 기술
- 주석은 '# ' 기호부터 그 줄의 마지막 내용까지에 해당하고, 프로그램 실행에서 제외되는 부분
- 주석을 여러 행에 나누어 작성하려면 큰따옴표나 작은따옴표 3개(' '', " """)를 이용
- 편집기에서는 단축키를 이용해서 주석 기호(##)를 자동으로 붙일 수 있는데, 주석 처리할 문장을 선택하고 Alt + 3 / 제거하는 단축키는 Alt + 4

```
01  '''=======================
02    다각형의 내각 계산 프로그램
03    ======================='''
04  n = int(input("3에서 6 사이의 정수를 입력하세요 : "))          # 다각형 모양 선택
05  sumAngle = 180 / (n - 2)                                    # 내각의 합 계산
```

II. 연산자의 활용

실습 4-11 거스름돈 계산하기 code04-11.py

- 문구점에서 연필을 사려고 한다. 연필 한 자루가 400원이라고 한다면, 몇 자루의 연필을 살 수 있고 거스름돈은 얼마인지 계산하는 프로그램

① 프로그램이 실행되는 순서를 먼저 생각

그림 4-16 연필 개수와 거스름돈 계산 순서

② 가진 돈이 3천 원이라고 가정하고 IDLE 쉘에서 간단하게 먼저 계산, 나눗셈 연산으로 계산하면 살 수 있는 연필의 개수는 7.5 자루

```
>>> 3000 / 400
7.5
```

II. 연산자의 활용

실습 4-11 거스름돈 계산하기 code04-11.py

③ 연필은 자루 단위로만 살 수 있기 때문에 다른 계산 방법을 사용해야 함, 몫과 나머지를 계산하는 산술 연산자(//, %)를 사용해서 다시 계산

```
>>> 3000 // 400
7
>>> 3000 % 400
200
```

④ 몫과 나머지 계산 방법과 실행 순서에 맞게 전체 프로그램을 작성

```
01  money = int(input("원 단위로 액수를 입력하세요. : "))  # 가진 돈의 액수
02  pencil = 400                                          # 연필 가격
03  print("연필 개수: %d 자루" % (money // pencil))        # 연필 개수 계산 후 출력
04  print("거스름돈 : %d 원" % (money % pencil))           # 거스름돈 계산 후 출력
```

③ 실행 결과 확인

II. 연산자의 활용

실습 4-12 박테리아 개체 수 계산하기 code04-12.py

- 만약 시간당 두 배로 분열하는 박테리아 100개가 있다면, 한 시간 후 200개, 두 시간 후에는 400개로 증식
- 하루가 지나고 나면 박테리아 개체 수는 얼마로 늘어나 있을지 계산

①

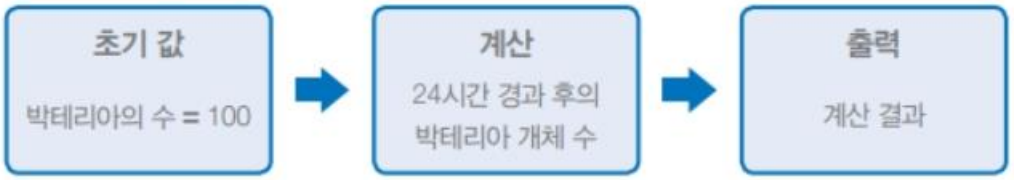


그림 4-20 박테리아 개체 수 계산 순서

II. 연산자의 활용

실습 4-12 박테리아 개체 수 계산하기 code04-12.py

② 시간(x)이 경과함에 따라 증식하는 개체 수에 대해 지수 함수식 생성

```
>>> 100                          # 시작 개체 수
100
>>> 100 * 2                      # 1시간 후 = 100*2
200
>>> (100 * 2) * 2                # 2시간 후 = 100*2^2
400
>>> (100 * 2 * 2) * 2            # 3시간 후 = 100*2^3
800                              # 100 * (2 ** x)이므로 x 시간 후 = 100*2^x
```

③ 실행 순서에 맞게 전체 프로그램을 만들고 저장

```
01  number = 100                 # 시작 개체 수
02  x = 24                       # 경과 시간
03  result = number * (2 ** x)   # x 시간 후 박테리아의 개체 수 계산
04  print("%d개의 박테리아의 %d시간 후 개체 수 = %d" % (number, x, result))
```

④ 실행 결과 확인

II. 연산자의 활용

하나 더 알기 **format() 함수**

- 숫자 데이터의 형식을 지정할 때 이용

표 4-9 format() 함수 사용법

예시	의미	실행 결과
format(123456789, ',')	세자리마다 쉼표(,) 넣기	'1,234,567'
format(123456789, '15')	자릿수를 15로 하기	' 123456789'
format(123456789, '<15')	자릿수를 15로 하고 왼쪽 정렬하기	'123456789 '
format(123456789, '015')	자릿수를 15로 하고 빈 자리를 0으로 채우기	'000000123456789'
format(123456789, 'x')	16진수로 표시하기	'75bcd15'
format(123456789, 'e')	지수 표기법으로 표시하기	'1.234568e+08'
format(12345.6789, '.2f')	소수점 이하 2자리로 실수 표시하기	'12345.68'

II. 연산자의 활용

실습 4-13 친구들과 피자 나눠 먹기 code04-13.py

- 세 명이 모여 피자를 주문, 피자를 2판 주문한다면 한 사람당 몇 조각씩?

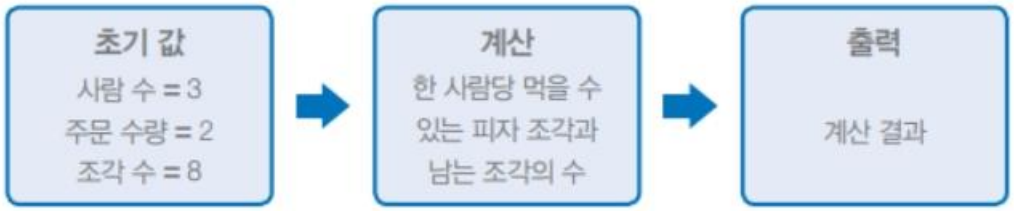
① 

그림 4-22 피자 나눠먹기 계산 순서

② 실행 순서에 맞게 전체 프로그램을 만들고 저장

```
01  person = 3                          # 사람 수
02  order = 2                           # 피자 주문 수량
03  piece = 8                           # 피자당 조각 수
04  result = (order * piece) // person  # 한 사람당 먹을 수 있는 조각 수
05  remain = (order * piece) % person   # 남는 조각 수
06
07  print("%d 조각씩 먹을 수 있고, %d 조각이 남습니다." % (result, remain))
```

II. 연산자의 활용

실습 4-14 장학금 대상인지 확인하기 code04-14.py

- 내년도 장학금을 타려면 올해 학점 평균은 3.0 이상, 봉사시간은 10시간 이상 필요

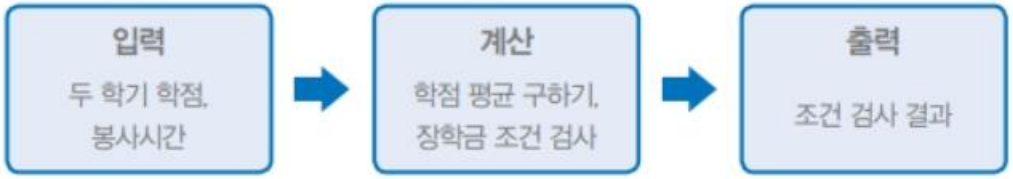
① 

그림 4-23 장학금 조건 구하기 계산 순서

② 장학금을 받는 조건은 학점 평균과 봉사시간이 모두 정해진 기준 이상이어야 하므로, 비교 연산식 2개를 논리 연산자 and로 결합

그림 4-24 장학금 대상 기준

II. 연산자의 활용

실습 4-14 장학금 대상인지 확인하기 code04-14.py

③ 실행 순서와 문제의 조건식에 맞게 프로그램을 만들고 저장

```
01  grade1 = float(input("1학기 학점 입력 : "))
02  grade2 = float(input("2학기 학점 입력 : "))
03  time = int(input("봉사시간 입력 : "))
04
05  average = (grade1 + grade2) / 2              # 학점 평균 구하기
06  result = (average >= 3.0) and (time >= 10)   # 장학금 대상인지 검사하기
07  print("장학금 대상 여부 =", result)           # 결과 출력하기
```

④ 실행 결과 확인

```
1학기 학점 입력 : 3.34
2학기 학점 입력 : 2.85
봉사시간 입력 : 12
장학금 대상 여부 = True
```