

문제해결을 위한 코딩 첫걸음

⋮ 4장 연산자에 대해 알아보시다(2)

서울세계도서관정보대학도서관방문기관으로 선정

한성대학교 노은희 교수

학습 목표

- $+$, $-$, $*$, $/$, $\%$, $//$, $**$ 등 산술 연산자를 이해합니다.
- $>$, $<$, $>=$, $<=$, $==$, $!=$ 등 관계 연산자를 이해합니다.
- `and`, `or`, `not`에 대한 논리 연산자를 이해합니다.
- $+=$, $-=$, $*=$, $/=$ 등 복합 대입 연산자에 대해 이해합니다.
- 산술 연산자, 관계 연산자, 논리 연산자 등 연산자들 간의 우선순위를 이해합니다.



4.2 관계 연산자

- 관계 연산자
 - 두 값을 비교하여 어떤 것이 큰지 또는 작은지, 같은지 등을 판단
 - 결과는 참(True) 또는 거짓(False)
 - 관계 연산자의 종류

관계 연산자	의미	사용 예	설명
==	같다	5 == 4	False
!=	같지 않다	5 != 4	True
>	크다	5 > 4	True
<	작다	5 < 4	False
>=	크거나 같다	5 >= 4	True
<=	작거나 같다	5 <= 4	False



4.2 관계 연산자 - 따라 해보기

- 기본적인 관계 연산자를 이용한 프로그램

```
>>> a = 7
```

```
>>> b = 5
```

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

False True True False

비교연산자의 결과는 True(참) 또는 False(거짓)



4.3 논리 연산자

- 논리 연산자
 - 논리식에 대하여 그리고(and), 또는(or), 부정(not)에 대한 연산
 - and 연산자 : 나열된 조건이 모두 참인 경우에만 참(True)
 - or 연산자:
 - 나열된 조건 값 중 하나만 참이어도 참(True)
 - 나열된 조건 값이 모두 거짓일 경우에만 거짓(False)
 - not 연산자: 참은 거짓(False), 거짓은 참(True)으로 논리의 결과를 반대로 취함
 - 논리 연산자의 종류

논리 연산자	의미	사용 예	설명
not	~아니다. 부정	not(5 > 3)	False
and	~이고, 그리고	5 > 3 and 5 < 7	True
or	~이거나, 또는	5 < 3 or 5 > 7	False



4.3 논리 연산자 - 따라 해보기

- 논리 연산자를 이용한 프로그램

and 연산은 두 조건 모두 만족해야만 True(참)

```
>>> a = 10
```

```
>>> b = 20
```

```
>>> print(a==b and a<b)
```

```
False
```

조건1 : 거짓

조건2 : 참

```
>>> print(a==b or a<b)
```

```
True
```

조건1 : 거짓

조건2 : 참

or 연산은 두 조건 중 하나만 만족하면 True(참)



4.4 복합 대입 연산자

- 대입연산자와 복합 대입 연산자
 - 복합 대입 연산자
 - 대입 연산자를 필요에 따라 간결하게 사용한 것

sum = sum + 10

대입 연산자

==

sum += 10

복합 대입 연산자

- 복합 대입 연산자의 종류

복합 대입 연산자	사용 예	설명
+=	x += 1	x = x + 1
-=	x -= 1	x = x - 1
*=	x *= y	x = x * y
/=	x /= 1	x = x / 1
//=	x //= 1	x = x // 1
%=	x %= 1	x = x % 1
**=	x **= y	x = x ** y



4.4 복합 대입 연산자 - 따라 해보기

- 복합 대입 연산자를 이용한 프로그램

```
>>> a = 10
```

```
>>> b = 20
```

```
>>> b += a
```

```
>>> b
```

```
30
```

```
>>> b -= a
```

```
>>> b
```

```
20
```

동일

$b = b + a$

동일

$b = b - a$



4.4 복합 대입 연산자 - 자동판매기

- 자동판매기 시뮬레이션 프로그램
 - 자동판매기에서 잔돈이 거슬러 지는 과정을 시뮬레이션 하기
 - 고객이 5,000원, 또는 1,000원의 지폐를 투입하고, 원하는 음료수를 선택했을 때, 음료수의 가격만큼 지불하고, 남은 금액을 잔돈으로 반환하기
 - 잔돈은 1,000원, 500원, 100원짜리로 반환 가능

지불한 돈을 입력: 5000
구입할 음료수 가격 입력: 2100
거스름돈은 2900원 입니다
1000원 지폐의 수 => 2
500원 동전의 수 => 1
100원 동전의 수 => 4

[실행결과]





4.4 복합 대입 연산자 - 자동판매기

- 자동판매기 시뮬레이션 프로그램

[소스코드] 4-4.py

시작

```
price, money, change = 0, 0, 0
c1000s, c500s, c100s = 0, 0, 0
```

#돈과 음료수 가격 입력

```
money = int(input("지불한 돈을 입력: "))
price = int(input("구입할 음료수 가격 입력: "))
```

#거스름돈 계산

```
change = money - price
```

이어서

#잔돈 계산

```
c1000s = change // 1000
change %= 1000
c500s = change // 500
change %= 500
c100s = change // 100
```

```
print("거스름돈은 %d원 입니다" %(money-price))
print("1000원 지폐의 수 => ", c1000s)
print("500원 동전의 수 => ", c500s)
print("100원 동전의 수 => ", c100s)
```



4.5 연산자 우선 순위 - 연산의 모호성

- 연산자 우선순위
 - 연산의 모호성을 해결하기 위해 연산자 연산 순서를 결정
 - 여러 연산자가 섞여 있을 때 어떤 연산자가 먼저 수행되는가에 따라 수식의 결과가 달라질 수 있음.

'+' 연산 후 '*'를 하는 경우

$$a = 3 + 4 * 5$$

Diagram illustrating the evaluation of the expression $a = 3 + 4 * 5$ when addition is performed first. A bracket under $3 + 4$ indicates the addition step, resulting in 7. Then, a larger bracket under $7 * 5$ indicates the multiplication step, resulting in 35.

'*' 연산 후 '+'를 하는 경우

$$a = 3 + 4 * 5$$

Diagram illustrating the evaluation of the expression $a = 3 + 4 * 5$ when multiplication is performed first. A bracket under $4 * 5$ indicates the multiplication step, resulting in 20. Then, a larger bracket under $3 + 20$ indicates the addition step, resulting in 23.



4.5 연산자 우선 순위

- 파이썬에서의 연산자 우선순위

우선순위	연산자	설명
1	()	괄호
2	**	지수승
3	*, /, //, %	곱셈, 나눗셈, 몫, 나머지
4	+, -	덧셈, 뺄셈
5	== (왼쪽값과 오른쪽 값이 같냐?) != (왼쪽값과 오른쪽 값이 다르냐?) <, <=, >, >=	비교연산자 (관계연산자)
6	not	논리연산자
7	and	
8	or	
9	= %=	대입연산자 복합대입연산자



생각해보기

- 국어, 영어 점수를 입력 받아 평균을 구한다. 다음의 경우 오류는 무엇인가?

[소스코드] 4-5.py

```
kor = int(input("국어: "))  
eng = int(input("영어: "))  
ave = kor + eng / 2  
print("평균: ", ave)
```

[실행결과]

```
국어: 96  
영어: 84      ← 잘못된 결과임  
평균: 138.0
```




따라 해보기 - BMI 계산 프로그램

- 체질량지수 BMI(Body Mass Index) 구하기
 - 체중(kg)을 키(m)의 제곱 값으로 나눈 값을 구해 비만도를 판정
 - 변환 공식
 -

BMI Chart

BMI	분류
18.5미만	저체중
18.5 ~ 24.99	정상
25~29.99	과체중
30~34.9	경도비만
35~39.9	중증도 미만
40이상	고도비만

$$BMI = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

Weight : 체중
Height : 키(키는 m단위)

[소스코드]4-7.py

```
w = float(input("몸무게를 입력(kg): "))  
h = float(input("키를 입력(m): "))  
bmi = w / ( h * h )  
print("BMI = : ", bmi )
```

[실행결과]

```
몸무게를 입력(kg): 58  
키를 입력(m): 1.67  
BMI = : 20.796729893506402
```



마무리

- 이번 시간에는 관계연산자/논리연산자/복합대입연산자 / 연산자 우선순위에 대해 알아보았습니다.