

CHAPTER 04

연산자에 대해 알아보시다.

문제해결을 위한 파이썬 첫걸음

이미향 교수

smilequeen@gmail.com

학습 목표

- +, -, *, /, %, //, ** 등 산술 연산자를 이해합니다.
- >, <, >=, <=, ==, != 등 관계 연산자를 이해합니다.
- AND, OR, NOT에 대한 논리 연산자를 이해합니다.
- +=, -=, *=, /= 등 복합 대입 연산자에 대해 이해합니다.
- 산술 연산자, 관계 연산자, 논리 연산자 등 연산자들 간의 우선순위를 이해합니다.

이번 장에서 만들 프로그램

자동판매기 시뮬레이션 프로그램

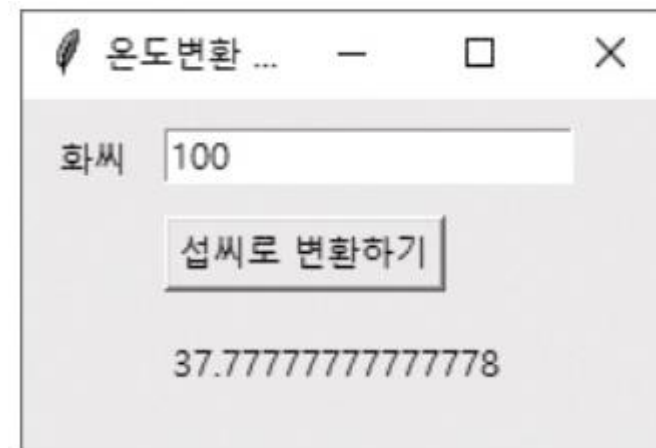
- 연산자 이용



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

온도 변환 윈도우 기반 프로그램 (화씨온도 -> 섭씨온도)

- tkinter() 모듈 이용



연산자

연산자(Operator)

- 연산자 (Operator)는 산술, 논리 연산 등을 수행하기 위한 특수 기호
- 연산자가 연산을 수행하는 값 또는 데이터를 피연산자 (operand)라고 함



4.1 산술 연산자

- 산술 연산자의 종류
- 더하기, 빼기, 곱하기, 나누기 등과 같은 수학 연산을 수행하는데 사용

산술 연산자	의미	사용 예	설명
+	덧셈	$x = 5 + 4$	9
-	뺄셈	$x = 5 - 4$	1
*	곱셈	$x = 5 * 4$	20
/	나눗셈	$x = 5 / 4$	1.5
//	나눗셈의 몫	$x = 5 // 4$	1
%	나눗셈의 나머지	$x = 5 \% 4$	1
**	제곱	$x = 5 ** 4$	625

4.1 산술 연산자 - 따라 해보기

- 기본적인 산술 연산자를 이용한 프로그램

```
>>> a = 7
>>> b = 5
>>> print(a + b, a - b, a * b, a / b)
12 2 35 1.4
```

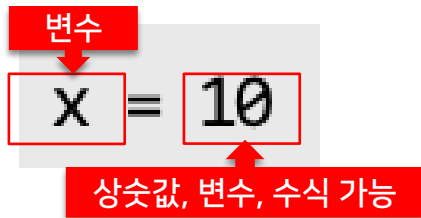
```
>>> print(a / b, a // b, a % b, a ** b)
1.4 1 2 16807
```

```
>>> data = [10, 20, 30, 40]
>>> sum = data[0] + data[1] + data[2] + data[3]
>>> print("합계 = ", sum)
합계 = 100
```

4.1 산술 연산자 - 수식과 대입

• 수식과 대입 연산자(=)

- 대입 연산자(=) 를 사용하여 계산 결과를 별도의 기억 장소(변수)에 저장
- 오른쪽에 위치한 특정 상숫값, 변수, 수식을 왼쪽에 지정한 변수에 대입



[가능 연산]

$x = 10$	← 상수 대입
$y, z = 0, 0$	← 복합 상수 대입
$x = y = z = 1$	← 복합 상수 대입
$y = x$	← 변수 대입
$z = x + y * 5$	← 수식 대입
$x = x + y$	← 수식 대입

[불가능 연산]

$10 = x$
$x + 10 = y$
$x + z = z$

4.2 산술 연산자 - 따라 해보기

- 수식을 이용하여 원의 면적과 원의 둘레 구하기
 - 공식

원의 면적 = πr^2 , 원의 둘레 = $2\pi r$ (단, $\pi = 3.14$)

[소스코드] 4-1.py

#원의 면적과 둘레 구하기

radius = float(input("반지름을 입력: "))

area = 3.14 * radius * radius

원의 면적 구하기

around = 2 * 3.14 * radius

원의 둘레 구하기

print("원의 면적은 ", area, "이고, 원의 둘레는 ", around, "입니다.")

[실행결과]

반지름을 입력: 10.5

원의 면적은 346.185 이고, 원의 둘레는 65.94 입니다.

4.2 산술 연산자 - 따라 해보기

- 컴퓨터의 저장 단위 KB, MB, GB의 byte 수 계산하기
 - 계산 공식

$$1\text{KB} = 2^{10} \text{ byte}$$

$$1\text{MB} = 2^{20} \text{ byte}$$

$$1\text{GB} = 2^{30} \text{ byte}$$

[소스코드] 4-2.py

```
kb = 2 ** 10
mb = 2 ** 20
gb = 2 ** 30
print("1KB = " + str(kb) + "byte")
print("1MB = " + str(mb) + "byte")
print("1GB = " + str(gb) + "byte")
```

[실행결과]

```
1KB = 1024byte
1MB = 1048576byte
1GB = 1073741824byte
```

4.2 산술 연산자 - 따라 해보기

- 터틀 그래픽을 이용하여 계산하기
 - 가로 길이와 세로 길이를 입력 받은 후 사각형 그리기
 - 그려진 사각형의 면적을 구해서 출력하기

시작

import turtle

t = turtle.Turtle()

t.shape("turtle")

터틀 그래픽에서 문자열 입력받기

w = turtle.textinput("가로 입력", "가로 길이를 입력하세요")

h = turtle.textinput("세로 입력", "세로 길이를 입력하세요")

area = int(w) * int(h)

w = int(w)

h = int(h)

[소스코드] 4-3.py

제목표시줄

prompt
문자열

[입력 예시]

이어서

t.forward(w)

t.left(90)

t.forward(h)

t.left(90)

t.forward(w)

t.left(90)

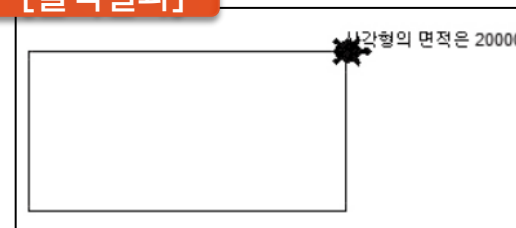
t.forward(h)

t.penup()

t.goto(w, h)

t.write("사각형의 면적은 " + str(area))

[출력결과]



4.2 관계 연산자

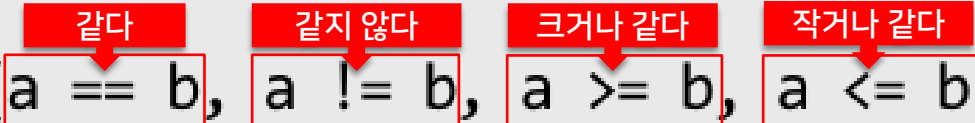
- 관계 연산자
 - 두 값을 비교하여 어떤 것이 큰지 또는 작은지, 같은지 등을 판단
 - 결과는 참(True) 또는 거짓(False)
 - 관계 연산자의 종류

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

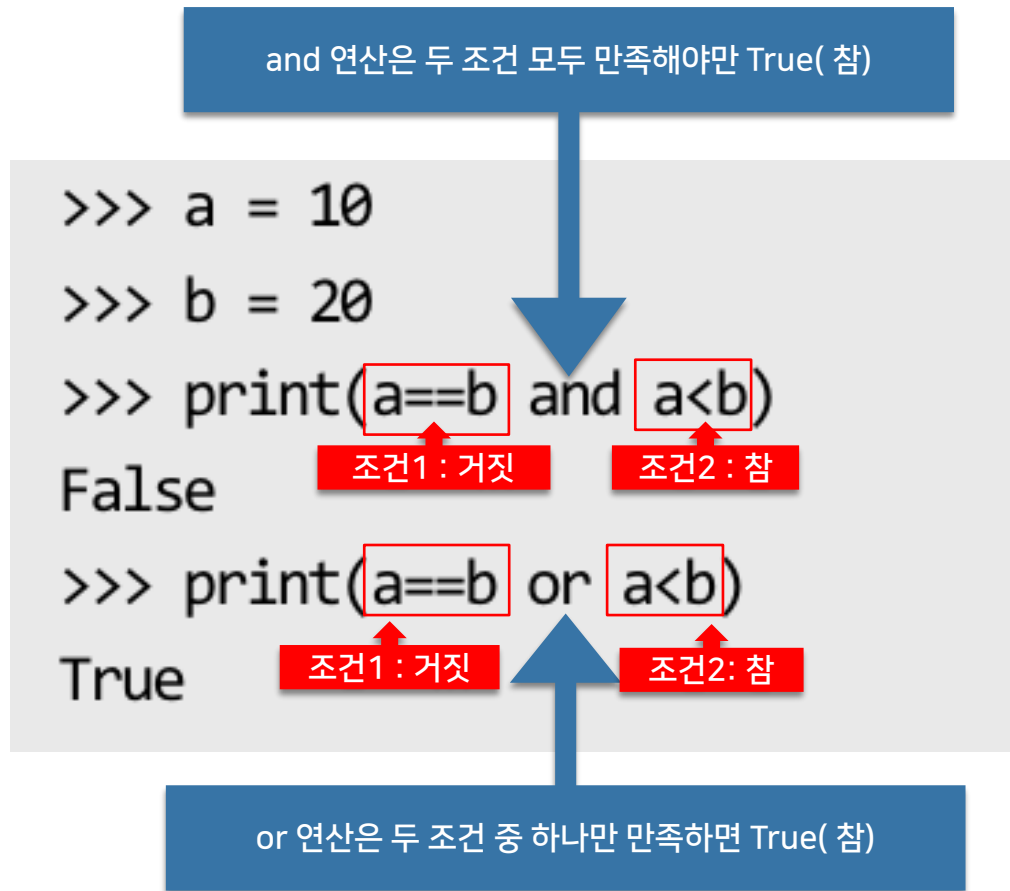
- 논리식에 대하여 그리고(and), 또는(or), 부정(not)에 대한 연산
 - and 연산자 : 나열된 조건이 모두 참인 경우에만 참(True)
 - or 연산자:
 - 나열된 조건 값 중 하나만 참이어도 참(True)
 - 나열된 조건 값이 모두 거짓일 경우에만 거짓(False)
 - not 연산자: 참은 거짓(False), 거짓은 참(True)으로 논리의 결과를 반대로 취함

- 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



4.4 복합 대입 연산자 - 따라 해보기

- 복합 대입 연산자를 이용한 프로그램

```
>>> a = 10; b = 20
>>> b += a
>>> print("a = ", a, "b = ", b)
a = 10 b = 30
```



Python

실습_코딩하기1

4.4 복합 대입 연산자 - 자동판매기

- 자동판매기 시뮬레이션 프로그램

- 자동판매기에서 잔돈이 거슬러지는 과정을 시뮬레이션하기
- 고객이 5,000원, 또는 1,000원의 지폐를 투입하고, 원하는 음료수를 선택했을 때, 음료수의 가격만큼 지불하고, 남은 금액을 잔돈으로 반환하기
- 잔돈은 1,000원, 500원, 100원 짜리로 반환 가능

[실행결과]

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

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)
```

4.4 복합 대입 연산자 - 자동판매기

- 자동판매기 시뮬레이션 프로그램
 - 자동판매기에서 잔돈이 거슬러지는 과정을 시뮬레이션하기
 - 고객이 20,000원의 지폐를 투입하고, 원하는 음료수를 선택했을 때, 음료수의 가격(12,900원)만큼 지불하고, 남은 금액을 잔돈으로 반환하기
 - 잔돈은 5,000원, 1,000원, 500원, 100원 짜리로 반환 가능

[실행결과]

```
지불한 돈을 입력=>20000
구입한 음료수 가격 입력=>12900
거스름돈은 7100원 입니다
5000의 개수 => 1
1000의 개수 => 2
500의 개수 => 0
100의 개수 => 1
```

4.4 복합 대입 연산자 - 자동판매기

```
1 #변수 선언 및 값 초기화
2 price, money, change = 0, 0, 0
3 c5000s, c1000s, c500s, c100s = 0, 0, 0, 0
4
5 #돈과 음료수 가격 입력 받기
6 money = int(input("지불한 돈을 입력=>"))
7 price = int(input("구입한 음료수 가격 입력=>"))
8
9 #거스름돈 계산
10 change = money - price
11
12 #잔돈 계산
13 c5000s = change // 5000
14 change = change % 5000
15 c1000s = change // 1000
16 change = change % 1000
17 c500s = change // 500
18 change = change % 500
19 c100s = change // 100
20
21 #단위별 거스름돈 갯수 출력하기
22 print(f"거스름돈은 {money - price}원 입니다")
23 print("5000의 개수 => ", c5000s)
24 print("1000의 개수 => ", c1000s)
25 print("500의 개수 => ", c500s)
26 print("100의 개수 => ", 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

$$a = \underbrace{3 + 4}_{7} * 5$$
$$\underbrace{ * 5}_{35}$$

'*' 연산 후 '+'를 하는 경우

$$a = 3 + \underbrace{4 * 5}_{20}$$
$$\underbrace{3 + }_{23}$$

4.5 연산자 우선 순위

- 파이썬에서의 연산자 우선순위

우선순위	연산자	설명
1	(), [], { }	괄호, 리스트, 딕셔너리, 세트 등
2	**	지수
3	+, -, ~	단항 연산자
4	*, /, %, //	산술 연산자
5	+, -	산술 연산자
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	논리 연산자
15	or	논리 연산자
16	if~else	비교식

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 - 온도 변환 프로그램

- 화씨온도를 입력 받아 섭씨온도로 바꾸는 프로그램
 - 변환 공식

$$C = \frac{5}{9}(F - 32)$$

C: 섭씨온도, F: 화씨온도

[실행결과]

```
화씨 온도 입력: 100  
섭씨 온도: 37.77777777777778
```

[소스코드] 4-6.py

```
#섭씨온도 구하기  
f = float(input("화씨온도 입력: "))  
c = float(5 * ( f - 32 ) / 9)  
print("섭씨온도: ", c )
```

따라 해보기 - BMI 계산 프로그램

- BMI(Body Mass Index) 구하기

- 체중(kg)을 키(m)의 제곱 값으로 나눈 값을 구해 비만도를 판정
- 변환 공식

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

Weight : 체중
Height : 키

[실행결과]

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

[소스코드]45-7.py

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

따라 해보기 - 태어난 년도의 띠 구하기

• 태어난 년도의 띠 구하기

• 적용 공식

```
year = (birthYear + 8) % 12
```

year : 띠를 찾을 수 있는 인덱스(index) 번호

- 띠에 대한 연도의 시작이 '쥐'로부터 시작되지는 않았기 때문에 생년월일 birthyear에 8을 더해주어야 한다.

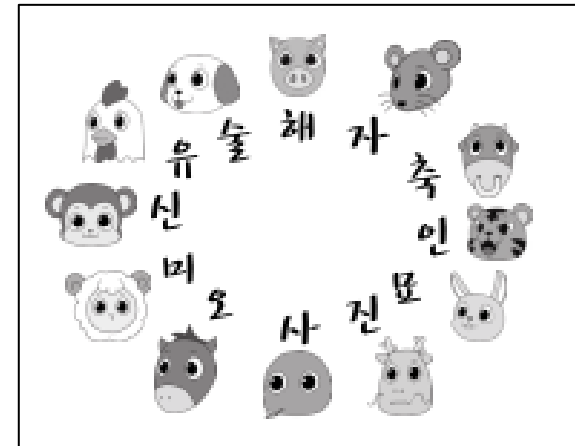
[소스코드] 4-8.py

```
data = ['쥐', '소', '범', '토끼', '용', '뱀', '말', '양', '원숭이', '닭',  
'개', '돼지']
```

```
birthYear = int(input("태어난 년도 입력>> "))  
year = (birthYear + 8) % 12  
print("당신은 " + data[year] + "띠입니다.")
```

[실행결과]

```
태어난 년도 입력>> 2000  
당신은 용띠입니다.
```



따라 해보기 - 나이 구하기

• 태어난 년도로부터의 나이 구하기

- time() 모듈을 작업 환경으로 import 해야 함
- time() 함수
 - 1970년 1월 1일 0시 0분 0초를 시작으로 현재 시각까지를 초 단위로 구해주는 함수
 - 적용 공식

$$\text{현재년도} = 1970 + \frac{\text{time()}}{1\text{년에 대한 초의 값}(365 \times 24 \times 1\text{시간초값 } 3600)}$$

[소스코드] 4-9.py

```
import time

now = time.time()
thisYear = int(1970 + now // (365*24*3600))
birthYear = int(input("태어난 년도 입력>> "))
age = thisYear - birthYear + 1

print("올해: ", thisYear)
print("당신의 나이: ", age)
```

[실행결과]

```
태어난 년도 입력>> 2000
올해: 2018
당신의 나이: 19
```



Python

실습_코딩하기2

따라 해보기 - 온도 변환 윈도우 프로그램

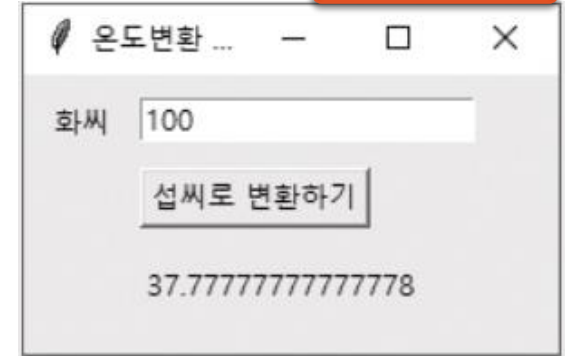
화씨온도를 입력 받아 섭씨온도로 바꾸는 윈도우 프로그램

변환 공식

$$C = \frac{5}{9}(F - 32)$$

C: 섭씨온도, F: 화씨온도

[실행화면]



시작

from tkinter import *

온도를 변환하기 위해 함수 정의

```
def convert() :
    f = float(ent.get())
    temp = float( 5*(f-32)/9 )
    lbl2.configure(text=temp)
```

```
root = Tk()
root.title("온도변환")
root.geometry("230x120")
```

이어서

[소스코드] 4-10.py

```
lbl1 = Label(root, text="화씨")
ent = Entry(root)
lbl2 = Label(root, text="섭씨")
btn = Button(root, text="섭씨로 변환하기", command = convert )
```

이벤트 처리위해 정의한 함수 등록

```
lbl1.place(x=10, y=10)
ent.place(x=50, y=10)
btn.place(x=50, y=40)
lbl2.place(x=50, y=80)
```

```
root.mainloop()
```

지정한 위치 좌표에 위젯을
배치하는 place() 배치관리자