

## 2장. 변수 & 연산자



# 변수(variable)

## ■ 변수란?

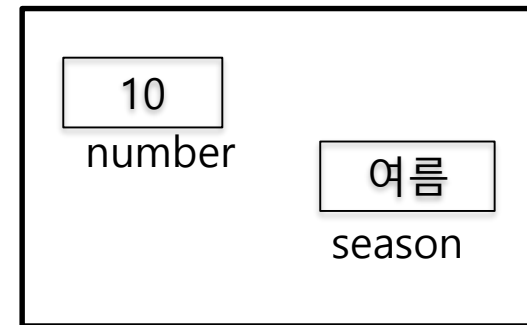
- 프로그램에서 사용되는 자료를 저장하기 위한 메모리 공간(영역)
- 프로그램 실행 중에 값 변경 가능, variable 이라 함

## ■ 변수의 선언 및 초기화

- 변수 선언은 어떤 타입의 데이터를 저장할 것인지 그리고 변수이름은 무엇인지를 결정한다. (자료형)은 생략함.

(자료형) 변수이름 = 초기값;

예) number = 10 , season = "여름"



# 변수(variable)

- 변수 사용 예제

```
# 정수 - 상수(변하지 않는 수)
"""
print(4 + 5)
print(4 - 5)
print(4 * 5)
print(4 / 5)
"""

# 변수 - 자료를 저장하기 위한 메모리 공간
n1 = 4 # n1이라는 변수에 4를 저장(기억)
n2 = 5

print("n1 + n2 =", n1 + n2)
print("n1 - n2 =", n1 - n2)

season = "여름" # season이라는 변수에 '여름'을 저장
print(season)
```

# 변수(variable)

- 변수명(이름) 작성시 주의

- 변수 이름은 영문, 한글, 특수문자, 숫자의 결합으로 만든다.
- 변수 이름은 숫자로 시작할 수 없고, 공백이 있으면 안됨.
- 변수 이름은 대, 소문자를 구분함
- 예약어는 사용할 수 없음.

```
# 변수 이름 만들때 오류의 예  
1n = 4 # 숫자로 시작 불가  
season 2 = "봄" # 공백 불가  
class = 100 # 예약어 불가
```

# 자료형(Type)

## ● 자료형이란?

- 사용할 데이터의 종류에 따라 메모리 공간을 적절하게 설정해 주는 것
- 파이썬에서는 표기하지 않음 (예, num=10, name="홍길동", pi=3.14)

분류	자료형	설명	예
정수	int	소수점이 없는 수	-2, -1, 0, 1, 2
실수	float	소수점(.)이 있는 수, 부동소수점수라고도 불린다.	-3.5, 0.0, 1.25
문자열	str	알파벳과 다른 문자로 이루어진 문장	"a", 'hello', "비"
논리형	bool	참과 거짓을 표현	True, False(2가지 값만 있음)

# type() 함수

- 자료형을 반환해주는 type() 함수

```
# 자료형 반환 - type() 함수
# 숫자형
print(type(n1))
print(type(3.14))

# 문자형
print(type(season))
print(type('K'))

# 논리형
print(4 > 5)
print(4 < 5)

print(type(4 > 5))
print(type(4 < 5))
```

```
n1 + n2 = 9
n1 - n2 = -1
여름
<class 'int'>
<class 'float'>
<class 'str'>
<class 'str'>
False
True
<class 'bool'>
<class 'bool'>
```

# 컴퓨터에서 데이터 표현하기

## ■ 비트(binary digit)

bit(비트) : 컴퓨터가 표현하는 데이터의 최소 단위로 2진수 하나의 값을 저장할 수 있는 메모리의 크기

컴퓨터는 0과 1로만 데이터를 저장함(0-> 신호꺼짐, 1-> 신호켜짐)

## ■ 비트로 표현할 수 있는 수의 범위

비트수	표현할 수 있는 범위(십진수)	
1bit	0, 1(0~1)	$2^1$
2bit	00, 01, 10, 11(0~3)	$2^2$
3bit	000, 001, 010, 011, 100, 101, 110, 111(0~7)	$2^3$

# 10진수를 2진수로 바꾸기

## ■ 진수 표현

10진수	2진수	16진수	10진수	2진수	16진수
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F
8	1000	8	16	10000	10

자리 올림 발생



# 아스키 코드와 유니코드

- 숫자, 문자 표현 – 컴퓨터 내부에서는 숫자뿐만 아니라 문자도 2진수로 표현

- 아스키 코드(ASCII)**

숫자, 영문알파벳 등 128개의 문자를  
표기하도록 정한 코드값

- 유니코드(Unicode)**

한글, 중국어 등 아스키 코드로 표현할 수  
없는 문자 표기 가능

예) 한글은 약 11,000자

- 총 65536개의 문자 표현 가능

문자	코드값	문자	코드값
	32	A	65
!	33	B	66
"	34	C	67
...	...	...	...
0	48	a	97
1	49	b	98
2	50	c	99

# 진수 표현

## ▪ 10진수, 2진수, 16진수

※ 10진수를 2진수로 바꾸기

10 = 1010(2)

8 4 2 1

1 0 1 0 ( $1 \times 2^3 + 1 \times 2^1 \rightarrow 8 + 2$ )

# 진수 표현하기

```
num = 10  
b_num = 0b1010  
h_num = 0xA
```

```
print(num)  
print(b_num)  
print(h_num)
```

#진수 표현 함수

```
print(bin(10))  
print(bin(65))  
print(hex(10))  
print(hex(65))
```

```
10 10 10  
0b1010  
0b1000001  
0xa  
0x41
```

# 아스키 코드값과 문자

```
print(ord('0'))  
print(ord('A'))  
print(chr(48))  
print(chr(65))
```

```
48  
65  
0  
A
```

## 문자열 다루기

- 문자열 안에 따옴표 사용하기 - 서로 중복이 되지 않게 함
- 여러 줄로 출력하기 - 쌍따옴표 또는 홀따옴표 3개 사용(""" ~ ~ """)
- 이스케이프 문자 사용하기

코드	내 용
\n	줄바꿈
\t	탭 - 문자열 간격
\'	어포스트로피(')

# 문자열 다루기

## ■ 문자열 예제

```
say1 = "'힘내세요!' \n라고 말했습니다."  
print(say1)  
  
say2 = "No, it doesn't"  
print(say2)  
  
say3 = "문자 \"12\"를 정수 12로 변환하세요"  
print(say3)  
  
table = ""  
상품명\t가격\t수량  
키보드\t20000\t100  
마우스\t25000\t100  
모니터\t80000\t50  
""
```

```
'힘내세요!'  
라고 말했습니다.  
No, it doesn't  
문자 "12"를 정수 12로 변환하세요  
  
상품명    가격    수량  
키보드     20000   100  
마우스     25000   100  
모니터     80000   50
```

# 항과 연산자

## ■ 항(operand)

- 연산에 사용되는 값

## ■ 연산자(operator)

- 연산에 사용되는 기호  
예)  $3 + 7$  (3과 7은 항, '+'는 연산자)



## ■ 연산자의 종류

구분	연산자	연산 예
대입연산자	'='	num = 10
산술연산자	+, -, *, /, %	3 + 7
비교연산자	>, >=, <, <=, ==, !=	7 == 3
논리연산자	and, or, not	7 >= 3 and 3 != 3
복합대입연산자	+=, -=, *=, /=	n += 1

# 대입 연산자

## ■ 대입 연산자

- 오른쪽의 값을 왼쪽의 변수에 대입
- '=' 연산자를 사용.

age = 20

user\_id = "abc123"

- 연산의 순서

오른쪽 > 왼쪽

산술 연산 > 대입 연산

total = num1 + num2



# 대입 연산자 연습문제

## ■ 변수 값 교환하기

```
# 변수값 교환
x = 1
y = 2

print("==== 교환전 =====")
print("x =", x, ", y =", y)

# 교환
"""
temp = x
x = y
y = temp
"""

# 직접 교환
x, y = y, x

print("==== 교환후 =====")
print("x =", x, ", y =", y)
```

```
==== 교환전 =====
x = 1 , y = 2
==== 교환후 =====
x = 2 , y = 1
```

# 산술 연산자

## ■ 산술 연산자

연산자	연산 작업	설명
+	$n1 + n2$	더하기
-	$n1 - n2$	빼기
*	$n1 * n2$	곱하기
/	$n1 / n2$	나누기
//	$n1 // n2$	몫
%	$n1 \% n2$	나머지
**	$n1 ** n2$	거듭제곱



# 산술 연산자

## ■ 산술 연산자

```
n1 = 10  
n2 = 4
```

```
print("n1 + n2 =", n1 + n2)  
print("n1 - n2 =", n1 - n2)  
print("n1 * n2 =", n1 * n2)  
print("n1 / n2 =", n1 / n2) # 나누기  
print("n1 // n2 =", n1 // n2) # 몫  
print("n1 % n2 =", n1 % n2) # 나머지  
print("n1 ** n2 =", n1 ** n2) # 거듭제곱
```

```
n1 + n2 = 14  
n1 - n2 = 6  
n1 * n2 = 40  
n1 / n2 = 2.5  
n1 // n2 = 2  
n1 % n2 = 2  
n1 ** n2 = 10000
```

# 복합 대입 연산자

## ■ 복합 대입 연산자

연산자	연산 작업
+=	val += 10
-=	val -= 10
*=	val *= 10
/=	val /= 10
%=	val %= 10

```
val = 20

val += 10  # val = val + 10
print(val)

val -= 10  # val = val - 10
print(val)

val *= 10  # val = val * 10
print(val)

val /= 10  # val = val / 10
print(val)

val %= 10  # val = val % 10
print(val)
```

# 비교 연산

## ■ 비교 연산자

연산자	의미	예	결과
<	보다 작다	9 < 10	True
>	보다 크다	9 > 10	False
<=	작거나 같다	9<=10	True
>=	크거나 같다	9>=10	False
==	같다	9==10	False
!=	같지 않다	9!=10	True
is	같다(객체)	a is b	False
is not	같지 않다(객체)	a is not b	True

# 논리 연산

## ■ 논리 연산자

연산자	규칙
x and y	x, y 가 모두 참이면 참, 나머지는 거짓
x or y	x, y 중 둘 중 하나가 참이면 참
not x	x가 참이면 거짓, 거짓이면 참

## Python Docs > Library Reference > Built-in Types

Operation	Meaning
<	strictly less than
<=	less than or equal
>	strictly greater than
>=	greater than or equal
==	equal
!=	not equal
is	object identity
is not	negated object identity

### Boolean Operations — and, or, not ¶

These are the Boolean operations, ordered by ascending priority:

Operation	Result	Notes
x or y	if x is false, then y, else x	(1)
x and y	if x is false, then x, else y	(2)
not x	if x is false, then True, else False	(3)

# 논리 연산

## ■ 비교 및 논리 연산자

```
x = 10  
y = -10
```

```
print(x > 0)  
print(y > 0)  
print()
```

```
True  
False
```

```
print(x > y)  
print(x < y)  
print()
```

```
True  
False
```

```
print(x == 10)  
print(x == y)  
print(x != y)  
print(x is y)  
print(x is not y)  
print()
```

```
True  
False  
True  
False  
True
```

```
print(x > 0 and y > 0)  
print(x > 0 or y > 0)  
print(not x > 0)
```

```
False  
True  
False
```

# 비트 연산

## ■ 비트 이동 연산자

연산자	규칙
$a \ll 2$	a를 왼쪽으로 2비트 이동
$b \gg 3$	b를 오른쪽으로 3비트 이동

원본 데이터      옮길 비트의 수

```
num = 5 #00000101
val_1 = (num) << (2) #00010100
print(val_1)

val_2 = (num >> 1) #00000010
print(val_2)
```

# 비트 연산

## ■ 비트 논리 연산자

연산자	규칙
10 & 13	9와 8의 비트 논리곱을 수행 (1001 & 1000)
10   13	9와 8의 비트 논리합을 수행 (1001   1000)
~10	1은 0으로 0은 1로 반전

```
num1 = 10      # 00001010
num2 = 13      # 00001101

val_3 = num1 & num2 # 00001000
print(val_3)

val_4 = num1 | num2 # 00001111
print(val_4)
```



```
10 : 0 0 0 0 1 0 1 0
& 13 : 0 0 0 0 1 1 0 1
-----
      0 0 0 0 1 0 0 0
```



```
10 : 0 0 0 0 1 0 1 0
| 13 : 0 0 0 0 1 1 0 1
-----
      0 0 0 0 1 1 1 1
```

# 문자열 연산

- 문자열 더해서 연결하기(Concatenation)

+ : 연결 연산자

- 문자열 곱하기

\* : 곱하기 연산자

```
head = "Good"
tail = " Job!"
print(head + tail)
print(head * 3)

print('=' * 10)
print(head + tail)
print('=' * 10)
```

```
Good Job!
GoodGoodGood
=====
Good Job!
=====
```



# 형 변환(Type Conversion)

## ■ 형 변환

- 자료형은 각각 사용하는 메모리 크기와 방식이 다름
- **int**(문자) : 숫자로 변환함, **str**(숫자) : 문자로 변환함

```
# 문자를 숫자로 변환
val_1 = "123"
val_1 = int(val_1)
print(val_1)
print(val_1 + 10)

# 실수를 정수로 변환
val_2 = 3.1
print(int(val_2))

# 정수를 문자로 변환
val_3 = 123
val_3 = str(val_3)
print(val_3)
# print(val_3 + 10) 문자는 더할 수 없음
```

```
123
133
3
123
```

# 파이썬의 입력 처리

## ➤ input() – 입력 함수

```
"""  
print("문자 입력: ")  
ch = input()  
print(ch)  
"""
```

```
"""  
  
ch = input("문자 입력: ")  
print(ch)
```

문자 입력:  
happy  
happy

문자 입력: apple  
apple

```
# 숫자 입력  
num = input("숫자 입력: ")  
print(type(num))  
  
num = int(num) # 문자를 숫자로 변환  
print(type(num))  
print(num + 1)
```

숫자 입력: 8  
<class 'str'>  
<class 'int'>  
9

# 파이썬의 입력 처리

## ➤ input() – 입력 함수

```
print("이름을 입력해 주세요:")
name = input()
print(name + '님 반갑습니다.')

age = input('나이를 입력해 주세요: ')
age = int(age)
print("당신의 나이는 " + str(age) + "세 이군요!")
```

x, y는 str형이므로  
int형으로 형변환

```
print("첫째 수 입력: ")
x = int(input())
print("둘째 수 입력: ")
y = int(input())

sum = x + y
print("두 수의 합은", sum)
```

## 실습 문제

---

### 나이 계산 프로그램

나이를 입력 받아 아래의 결과처럼 계산하는 프로그램을 작성하세요.

---

☞ 실행 결과

```
태어난 연도를 입력하세요: 2000
2000에 태어난 사람의 나이는 25세 입니다.
```

## 실습 문제

### 사각형을 계산하는 프로그램

가로와 세로의 길이를 입력 받아 넓이를 계산하는 프로그램을 작성하세요.

☞ 실행 결과

가로의 길이 : 4  
세로의 길이 : 5  
가로 길이 : 4cm  
세로 길이 : 5cm  
면적 : 20cm

## 실습 문제 1 – 변수 사용

---

### 도형의 넓이 계산하기

1. 한 변의 길이가 5cm인 정사각형의 넓이
  2. 한 변의 길이가 5cm이고, 높이 7cm인 삼각형의 넓이
- 

👉 실행 결과

정사각형의 넓이: 25 cm  
삼각형의 넓이: 17.5 cm

## 실습 문제 2 – 산술 연산

---

### 몫과 나머지 계산하기

30개의 빵을 4명이 나눠 가질때 몫과 나머지를 구하세요.

---

☞ 실행 결과

빵의 개수	:	7
남은 빵의 개수	:	2