

# 03

## 자료형과 변수

- + 학습 목표**
  - 자료형의 종류와 특성을 알고, 적합한 자료형을 선택하여 프로그램을 작성할 수 있다.
  - 변수의 개념을 알고 이를 활용한 프로그램을 작성할 수 있다.
- + 학습 요소**
  - 자료형, 수치형, 문자열, 불 자료형, 변수

### 생각 깨우기

다음 글과 그림을 보고 생각해 보자.


한국이는 친구에게 줄 비스킷과 초콜릿을 서로 다른 크기의 그릇에 담고 있다.



프로그래밍을 할 때는 어떤 데이터를 사용하며, 이러한 데이터는 어디에 보관할까?

# 1 | 자료형의 종류와 특성

우리는 1, 10, 100 등은 '수'라고 부르고, 'Hello', '정보'는 '문자'라고 부른다. 파이썬에도 다양한 형태의 자료가 있으며, 자료형에 따라 데이터를 처리하고 저장하는 방식이 달라지기도 한다. 파이썬은 자동으로 자료형을 판별하지만, 프로그램에서 데이터를 올바르게 처리하기 위해서는 사용하려는 값의 종류에 따라 알맞은 자료형을 선택해야 한다. 파이썬의 기본적인 자료형으로는 수치, 문자열, 불 등이 있으며, 파이썬 내장 함수 `type()` 함수를 사용하여 데이터의 자료형을 확인해 볼 수 있다.

 230~231쪽 (부록 2)  
파이썬 프로그래밍 준비하기를 참고하자.


수치	문자열	불
123, 3.14, -10, 0	'안녕', 'Python', '123'	True, False


▲ 파이썬의 자료형

## 01 수치 자료형

수치 자료형이란 숫자 형태로 이루어진 자료형이다. 우리가 일상생활에서 사용하는 키와 몸무게, 성적, 상품의 가격 등 수학 시간에 접하는 수들이 모두 수치형에 해당한다. 파이썬에서 수치 자료형은 소수점을 포함하는지의 여부에 따라 크게 정수형과 실수형으로 구분한다.

구분	설명	예시
정수형	일반적인 정수로 소수점이 없는 숫자	1, 0, -1, 123
실수형	소수점이 포함된 숫자	3.14, -1.23, 5.0

 컴퓨터 프로그래밍에서 실수형 자료형을 흔히 'float'라 부른다.

 `print()` 함수  
괄호 안에 작성된 내용을 모니터에 출력한다.

### 실습 1


123과 3.14의 자료형을 확인해 보는 프로그램을 작성하고 실행해 보자.

```

프로그램
1 print(type(123))
2 print(type(3.14))
    
```

```

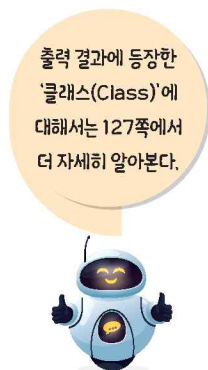
실행 결과
<class 'int'>
<class 'float'>
    
```

 **설명** 1행의 123은 소수점을 포함하지 않은 정수형이기 때문에 정수(integer)를 짧게 표현한 'int'가 출력되고, 2행의 3.14는 소수점을 포함한 실수형이기 때문에 실수형(floating-point)을 짧게 표현한 'float'가 출력된다.

```

프로그램
print('안녕하세요')

실행 결과
안녕하세요
    
```



## 02 문자열 자료형

문자열은 문자들의 집합을 의미하며, 파이썬에서 작은따옴표 또는 큰따옴표 1개 또는 3개를 사용하여 감싸서 표현한 데이터를 문자열이라고 한다. 일상생활에서 사용하는 이름, 과목명, 메시지의 내용 등이 문자열 자료형에 해당한다.

❗ 따옴표 3개를 사용하면 줄바꿈을 포함한 여러 줄에 걸친 문자열을 만들 수 있다.

### 문자열 예시

```
'Hello',    '''123''',
"홍길동",   """문자열 입니다."""
```

### 실습 2

'안녕하세요'의 자료형을 확인해 보는 프로그램을 작성하고 실행해 보자.

#### 프로그램

```
1 print(type('안녕하세요!'))
```

#### 실행 결과

```
<class 'str'>
```

🔍 설명 type( ) 함수 안에 따옴표로 감싸진 문자열 '안녕하세요!'가 입력되었기 때문에 문자열 (string)을 짧게 표현한 'str'이 출력된다.



## 해 보기 1 수치 자료형 실습하기

🗨️ -10, 0.0, 0의 자료형을 예상해 보고, 직접 실행하여 결과를 확인해 보자.

데이터	자료형
-10	
0.0	
0	

#### 프로그램

```
1 print(type(-10))
2 print(type(0.0))
3 print(type(0))
```

#### 실행 결과

```
<class 'int'>
<class 'float'>
<class 'int'>
```

파이썬은 따옴표로 감싼 부분은 모두 문자열로 인식한다. 산술 연산을 수행하고 그 결과를 출력하려면 문자열이 아닌 형태로 작성해야 한다.

### 실습 3

데이터의 자료형과 출력 결과를 확인해 보는 프로그램을 작성하고 실행하여 결과를 비교해 보자.

<pre>1 print('2+3') 2 print(type('2+3'))</pre>	<pre>2+3 &lt;class 'str'&gt;</pre>
<pre>1 print(2+3) 2 print(type(2+3))</pre>	<pre>5 &lt;class 'int'&gt;</pre>

**설명** 따옴표로 감싼 부분은 문자열이기 때문에 문자열의 내용 그대로 '2+3'이 출력되며, 따옴표 없이 쓰인 2와 3은 수치 자료형이므로 정수 2와 3을 덧셈 연산한 결과인 5가 출력된다.

파이썬에서는 문자열 자료형의 일부 연산을 지원한다. 덧셈 기호(+)를 사용하면 문자열을 연결하는 연산을 수행할 수 있으며, 이때는 문자열끼리만 연산이 가능하다. 곱셈 기호(\*)를 사용하면 문자열을 여러 번 반복하는 연산을 수행할 수 있으며 '문자열 \* 정수' 또는 '정수 \* 문자열'의 형태로 사용한다.

### 실습 4

문자열 연산을 활용한 프로그램을 작성하고 실행해 보자.

<pre>1 print('안녕' + '하세요')      # 문자열 + 문자열 연산 2 print('파이썬' * 3)          # 문자열 * 정수 연산</pre>
<pre>안녕하세요 파이썬파이썬파이썬</pre>

**설명** 1행에서 문자열 '안녕'과 문자열 '하세요'가 덧셈 기호(+)를 통해 연결되어 '안녕하세요'가 되어 출력되고, 2행에서 '파이썬'이 곱셈 기호(\*)에 의해 3번 반복된 결과가 출력된다.

## 03 불(Boolean) 자료형

불 자료형이란 참과 거짓을 나타내는 자료형으로 True(참) 또는 False(거짓)의 값만 가질 수 있으며, 프로그램의 실행 흐름을 제어할 때 유용하게 사용한다. True와 False는 파이썬에 미리 만들어져 있는 예약어로, 첫 번째 글자는 반드시 대문자로 입력해야 한다.

### 파이썬 주석

주석은 프로그램 수행에 영향을 주지 않는 문장으로, 주로 프로그램에 대한 설명을 작성할 때 사용한다. 파이썬에서는 # 뒤에 작성된 문장을 주석으로 처리한다.

**예약어**는 이미 문법적인 용도로 사용되는 단어로, 변수나 함수의 이름으로 사용할 수 없다.



자료형	값	참/거짓
수치형	0이 아닌 숫자	참
	0	거짓
문자열	'string'	참
	" "	거짓

수치 자료형 중 0과 비어 있는 문자열 자료형 ' '는 False(거짓) 값을 가지며, 이를 제외한 나머지 숫자와 문자열은 True(참) 값을 가진다.

bool( ) 함수를 사용하면 특정 데이터의 True(참) 또는 False(거짓) 값을 확인할 수 있다.

알고 가기의 프로그램에서 사용된 '선택 구조'에 대해서는 111쪽에서 더 자세히 알아본다.



**실습 5**

True, False 값의 자료형을 확인해 보는 프로그램을 작성하고 실행해 보자.

```

프로그램
1 print(type(True))
2 print(type(False))
    
```

```

실행 결과
<class 'bool'>
<class 'bool'>
    
```

**설명** type( ) 함수 안에 입력된 True와 False는 불 자료형이므로, 불 자료형(boolean)을 짧게 표현한 'bool'이 출력된다.

**알고 가기** ▶ 실행 흐름 제어



프로그램은 기본적으로 주어진 명령이 위에서 아래로 순차적으로 실행되지만, 문제 해결을 위해 같은 명령을 여러 번 반복하거나 조건에 따라 서로 다른 명령을 실행하기 위해 실행 흐름을 제어할 수 있다. 이때 불 자료형이 유용하게 사용되며 자료형의 참, 거짓 값에 따라 실행의 흐름이 달라진다.

다음 프로그램의 경우 '5<10'은 참이고, '5 >10'은 거짓이기 때문에 '5가 10보다 작다.'만 출력된다.

```

프로그램
1 if 5 < 10:
2     print('5가 10보다 작다.')    # 5<10이 참일 때만 출력
3 if 5 > 10:
4     print('5가 10보다 크다.')    # 5>10이 참일 때만 출력
    
```

```

실행 결과
5가 10보다 작다.
    
```

**해 보기 2** 불 자료형 실습하기

프로그램을 작성하여 결과를 확인해 보고 그렇게 실행되는 이유를 짚과 함께 이야기해 보자.

```

프로그램
1 print(bool('10/2'))
2 print(bool(3))
3 print(bool(0))
4 print(bool(''))
5 print(bool('Hello'))
    
```

```

실행 결과
    
```

## 2 | 변수와 연산자

컴퓨터를 이용하여 다양한 문제를 해결하기 위해서는 덧셈, 뺄셈, 곱셈, 나눗셈과 같은 사칙 연산뿐만 아니라 비교 연산과 논리 연산도 이해하고 활용할 수 있어야 한다. 또한 연산을 통해 얻어진 결과 값을 저장하고 활용하기 위해서는 변수의 개념을 이해하고 어떻게 사용되는지 알아야 한다.

### 01 변수

변수는 프로그램에서 값을 저장하기 위한 기억 장소로, 여기에 이름을 붙인 것을 변수명이라고 한다. 변수에 대입된 값은 하나로 고정되지 않고 필요에 따라 변할 수 있다. 파이썬에서는 변수에 저장되는 값에 따라 변수의 자료형이 자동으로 결정된다.

```
프로그램
1 var = 10 # 변수 var에 10 대입
2 print('var:', var) # var 출력
3 print(type(var)) # var의 자료형 출력

실행 결과
var: 10
<class 'int'>
```

```
프로그램
1 var = '정보' # 변수 var에 '정보' 대입
2 print('var:', var)
3 print(type(var))

실행 결과
var: 정보
<class 'str'>
```

**설명** 변수는 한 번에 하나의 값만 저장할 수 있기 때문에 변수 var에 10을 넣으면 var에는 10이 저장되고 자료형은 정수형이 된다. 그 후에 '정보'를 넣으면 var에는 '정보'가 저장되고 자료형이 문자열로 변경된다. 변수에 값을 할당할 때에는 대입 연산자(=)를 사용한다.

### 02 연산자

파이썬에서 사용하는 연산자는 기능에 따라 산술 연산자, 비교 연산자, 논리 연산자 등으로 분류된다.

#### 1 산술 연산자

파이썬에서는 다양한 산술 연산자를 사용하여 덧셈, 뺄셈, 곱셈, 나눗셈 등의 산술 연산을 수행할 수 있다.

산술 연산자의 종류							
산술 연산자	+	-	*	/	//	%	**
기능	덧셈	뺄셈	곱셈	나눗셈	정수 나눗셈(몫)	나머지	거듭제곱

#### 변수명 작성 방법

변수명은 저장된 값을 알 수 있는 의미 있는 이름을 사용하는 것이 좋으며, 다음과 같은 규칙을 지켜야 한다.

- 1 영문자와 한글, 숫자, '\_'를 사용할 수 있다.
- 2 영문자는 대문자와 소문자를 구분한다.
- 3 숫자부터 시작할 수 없다.
- 4 파이썬에서 미리 약속된 키워드(if, for, while, and, or 등)는 사용할 수 없다.
- 5 특수 문자(+, -, \* 등)와 공백은 사용할 수 없다.
- 6 여러 단어로 구성된 경우 '\_'를 사용하여 가독성을 높일 수 있다.

예) student\_name

#### print() 함수

- 괄호 안에 변수명을 작성하면 변수에 저장된 값이 출력된다.
- 문자열과 변수를 함께 출력하기 위해서는 쉼표(,)로 구분하여 작성한다.

### 여러 개의 변수 생성하기

변수1, 변수2 = 값1, 값2

위와 같이 작성하면 변수1에 값1이 대입되고, 변수2에 값2가 대입된다.

예) name, grade = '한국이', 1

### ⚠️ 잠깐

다음 프로그램을 작성하여 결과를 확인해 보고, 그 이유를 짝과 함께 이야기해 보자.

#### 프로그램

```
print(type(10/2))
print(type(10/2.0))
```

## 실습 6

산술 연산자를 사용하여 변수 num1과 num2의 산술 연산 결과를 출력하는 프로그램을 작성해 보자.

#### 프로그램

```
1 num1, num2 = 10, 3          # 변수 num1과 num2에 각각 10, 3 대입
2 print(num1 + num2)
3 print(num1 - num2)
4 print(num1 * num2)
5 print(num1 / num2)
6 print(num1 // num2)
7 print(num1 % num2)
8 print(num1 ** num2)
```

#### 실행 결과

```
13
7
30
3.3333333333333335
3
1
1000
```

## 해 보기 3 자료형과 산술 연산자

반지름의 길이가 3인 원의 넓이를 구하는 프로그램을 작성하려고 한다. 변수의 역할에 따라 적절한 자료형을 선택하여 프로그램을 완성해 보자. **※원의 넓이=반지름 길이<sup>2</sup>×원주율**

각 변수에 어떤 데이터가 들어갈지 고민하여 변수의 자료형을 선택해 보고, 변수 area에 원의 넓이가 대입되도록 변수와 산술 연산자를 사용하여 프로그램을 작성해 보자.

변수명	변수의 역할	자료형
r	반지름의 길이	
pi	원주율	
area	원의 넓이	

#### 실행 결과

원의 넓이: 28.26

#### 프로그램

```
pi = 3.14
r =
area =
print('원의 넓이:', area)
```

## 2 비교 연산자

비교 연산자를 사용하여 피연산자의 값 두 개를 비교할 수 있으며, 연산의 결과는 불 자료형으로 표현된다.

비교 연산자의 종류						
비교 연산자	>	>=	<	<=	==	!=

### 실습 7

비교 연산자를 사용하여 변수 num1과 num2의 비교 연산 결과를 출력하는 프로그램을 작성해 보자.

프로그램	실행 결과
<pre> 1 num1, num2 = 5, 10 # 변수 num1, num2에 각각 5, 10 대입 2 print(num1 &gt; num2) # 5가 10보다 크다 : 거짓 3 print(num1 &lt;= num2) # 5가 10보다 작거나 같다 : 참 4 print(num1 == num2) # 5와 10이 같다 : 거짓 5 print(num1 != num2) # 5와 10이 같지 않다 : 참 </pre>	<pre> False True False True </pre>

### 비교 연산자의 의미

표현	의미
a==b	a와 b가 같다
a!=b	a와 b가 다르다
a<b	a가 b보다 작다
a>b	a가 b보다 크다
a<=b	a가 b보다 작거나 같다
a>=b	a가 b보다 크거나 같다

## 3 논리 연산자

논리 연산자는 불 자료형의 값인 True(참) 또는 False(거짓)에 대한 연산에 사용되며, 연산의 결과도 불 자료형으로 표현된다.

논리 연산자의 종류			
논리 연산자	and	or	not
기능	모두 True일 때 True	둘 중 하나만 True여도 True	True면 False, False면 True

### 논리 연산의 진리표

A	not A
False	True
True	False

### 실습 8

논리 연산자를 사용하여 변수 a와 b의 논리 연산 결과를 출력하는 프로그램을 작성해 보자.

프로그램	실행 결과
<pre> 1 a, b = True, False # 변수 a, b에 각각 True, False 대입 2 print(a and a) 3 print(a and b) 4 print(b and b) 5 print(a or a) 6 print(a or b) 7 print(b or b) 8 print(not b) </pre>	<pre> True False False True True False True True </pre>

A	B	A and B	A or B
False	False	False	False
False	True	False	True
True	False	False	True
True	True	True	True



## 해 보기 4 공배수 판별하기

아래 프로그램을 보고 각 질문에 답해 보자.

```

프로그램
1 print( ㉠ )
2 print( ㉡ )
3 print( ㉢ )

```

1 28이 4의 배수이면 True, 그렇지 않으면 False를 출력하도록 프로그램의 빈칸 ㉠을 작성해 보자.

4의 배수는 4로 나누었을 때의 나머지가 0이다.

2 28이 7의 배수이면 True, 그렇지 않으면 False를 출력하도록 프로그램의 빈칸 ㉡을 작성해 보자.

7의 배수는 7로 나누었을 때의 나머지가 0이다.

3 28이 4와 7의 공배수이면 True, 그렇지 않으면 False를 출력하도록 프로그램의 빈칸 ㉢을 작성해 보자.

4와 7의 공배수는 4의 배수이면서 7의 배수인 수이다.

### 소단원 1 요약

- 1 파이썬에는 다양한 형태의 자료가 있으며, 프로그램에서 데이터를 올바르게 처리하기 위해서는 사용하려는 값의 종류에 따라 적절한 자료형을 선택해야 한다. 파이썬의 기본 자료형에는 수치형, 문자열, 불 등이 있다.
- 2 변수는 프로그램에서 값을 저장하기 위한 기억 공간을 의미하며, 값이 고정되어 있지 않고 변할 수 있다.
- 3 다양한 문제를 해결하기 위해 연산자를 사용하며, 파이썬에는 산술 연산자, 비교 연산자, 논리 연산자 등을 제공한다.

### 소단원 자기 평가

평가 항목	평가 기준		
	잘함	보통	노력
1. [지식이해] 변수의 개념을 알고 설명할 수 있다.			
2. [지식이해] 자료형의 종류와 특성을 설명할 수 있다.			
3. [과정기능] 문제 해결에 적합한 자료형을 선택하여 프로그램을 작성할 수 있다.			



## 기초 대사량 계산 프로그램 작성하기

기초 대사량이란 생명을 유지하는 데 필요한 최소한의 에너지량을 의미한다. 기초 대사량 계산식을 참고하여 프로그램을 작성해 보자.

성별	계산식
남성	$66.47 + (13.75 \times \text{체중}) + (5 \times \text{키}) - (6.76 \times \text{나이})$
여성	$665.1 + (9.56 \times \text{체중}) + (1.85 \times \text{키}) - (4.68 \times \text{나이})$

### 1 알고리즘 설계하기

- ① 체중, 키, 나이를 변수에 대입한다. ② 기초 대사량을 계산한다. ③ 기초 대사량을 출력한 후 프로그램을 종료한다.

### 2 프로그램에서 사용되는 변수 설계하기

프로그램에 필요한 변수를 생각해 보고, 변수의 이름과 역할, 적합한 자료형을 선택하여 표를 완성해 보자.

변수명	변수의 역할	자료형
예) BMR	기초 대사량	실수형

### 3 프로그램 작성하기

2번에 작성한 변수와 산술 연산자를 활용하여 기초 대사량 계산식에 따라 계산하여 출력하는 프로그램을 작성해 보자.

```

프로그램

print('<기초 대사량 계산 프로그램>')
# 체중, 키, 나이를 변수에 대입하기

# 기초 대사량 계산하기
BMR =

# 출력하기
    
```

```

출력 예시

<기초 대사량 계산 프로그램>
체중 52.5 키 163 나이 18인 여성의 기초 대사량: 1384.31
    
```

탐구 활동 자기 평가	평가 항목	평가 기준		
		잘함	보통	노력
1. 문제를 해결하기 위해 적합한 자료형을 선택할 수 있는가?				
2. 변수와 자료형을 활용하여 문제 해결을 위한 프로그램을 작성할 수 있는가?				

• 잘함 (내용을 이해하고 설명함) • 보통 (내용을 이해함) • 노력 (내용을 부분적으로 이해함)