

## 2장 변수와 자료형

프로그래밍을 할 때에는 수시로 필요한 변수들을 선언하고 변수에 값을 저장한 다음 저장된 값을 계산 또는 처리하여 프로그램의 목적을 달성시키는 것이 일반적이다. 프로그래밍을 하다 보면 필요한 값들을 메모리 공간에 저장해야 하는데, 변수(variable)는 값을 저장하는 \_\_\_\_\_이다. 변수는 값을 저장하는 공간이기 때문에 보통 그릇에 비유한다. 변수와 변수에 저장되는 값을 혼동하는 경우가 많은데, 이 둘의 관계를 정확히 이해할 필요가 있다.

### 2.1 변수의 선언

그릇의 종류도 어떠한 음식을 담을 것인지에 따라 여러 종류가 존재하듯이, 파이썬에서 사용하는 변수의 종류도 여러 가지가 있다. 가장 많이 사용하는 것은 불(boolean)형, 정수형, 실수형, 문자열형 네 가지이다.

파이썬에서 다음과 같이 코딩하면 변수(그릇) 4개가 메모리에 준비된다. 4개의 변수의 종류는 순서대로 불형, 정수형, 실수형, 문자열형으로 서로 다르기 때문에 메모리상에 존재하는 그릇의 형태는 서로 다르다. 변수(그릇)의 종류를 자료형(data type) 또는 데이터타입이라 한다.

변수(그릇) 하나에는 \_\_\_\_\_의 값만이 저장된다. 만약 동일한 변수에 다른 값을 대입한다면 기존의 값은 사라진다. 아래 코드에서는 변수를 선언함과 동시에 대입 연산자(=) 우측의 값이 저장된다.

아래의 코드에서 boolVar, intVar, floatVar, strVar는 변수의 \_\_\_\_\_이다. (변수 자체가 아님을 주목하자.) 파이썬에서는 변수를 선언할 때 변수의 이름 옆에 자료형을 명시하지 않는다. 변수에 대입하는 값의 종류에 따라서 데이터 타입이 결정되는 방식이다. 이를 \_\_\_\_\_이라 한다.

```
boolVar = True
intVar = 0
floatVar = 0.0
strVar = ""
```

변수명 규칙은 다음과 같다.

- 대소문자를 \_\_\_\_\_.

- 문자, 숫자, 언더바(\_)를 포함할 수 \_\_\_\_\_. 단, 숫자로 시작은 되지 않는다. (예, var1(O), \_var(O), new\_var(O), 3var(X))
- 예약어는 변수명이 될 수 없다. (True, False, if, else, elif, for, while 등)

## 2.2 변수의 사용

변수는 값을 대입하면 사용할 수 있다. 값을 대입하는 방법은 변수명 다음에 대입연산자(=)를 쓰고 값을 넣으면 된다. 그러면 기존의 값은 없어지고 새로운 값으로 변경된다. 수학에서의 '=' 연산자는 같다는 뜻인데 파이썬에서는 우측의 값을 왼쪽 변수에 \_\_\_\_\_는 뜻이다. (이 부분은 매우 중요하며 혼동하지 말자.)

변수에는 직접 값을 넣을 수도 있고, 다른 변수에 있는 값을 대입할 수도 있다.

```
var1 = 200
var2 = var1
```

위의 코드는 var1에 정수 200을 대입한다. 그리고 var2에는 우측의 var1 변수에 있는 값을 대입시키라는 명령이기 때문에 현재 var1에 들어있는 200을 대입한다. 그래서 실행 후 var1, var2 모두 정수 200을 저장하게 된다.

```
var1 = 100 + 100
var2 = var1 + 100
```

위의 첫 번째 줄은 100+100의 결과인 200을 var1에 \_\_\_\_시키라는 것이다. 대입 연산자(=)는 우측의 결과 값을 왼쪽에 대입시키라는 명령임을 상기하자. (단순히 왼쪽과 오른쪽이 같다는 뜻이 아니다.) 두 번째 줄은 변수 var1의 값과 정수 100을 더한 결과를 var2에 대입시키라는 뜻이다. var1의 값과 100을 더하는 것은 내부적으로 수행된다. 그래서 실행 후 var1은 200의 값을 가지고, var2는 300을 저장하게 된다.

```
var1 = var2 = 100
```

위의 코드는 대입연산자 여러 개가 함께 있어서 다소 생소해 보일 것이다. 대입 연산자는 우측의 값을 왼쪽 변수에 대입한다고 설명하였는데, 대입 순서는 오른쪽 끝부터 왼쪽으로 수행된다. 그래서 위의 코드는 `var2 = 100`이 먼저 실행되고, 그 다음 `var1 = var2`가 순차적으로 실행된다. 그래서 `var2`에 100이 먼저 대입되고, `var1`에 `var2`가 가지고 있는 100을 대입하게 되어 두 변수 모두 정수 100을 저장한다.

```
var1 = 100
var1 = var1 + 200
```

위의 첫 번째 줄은 `var1`에 100이 대입되는 것을 쉽게 알 수 있다. 두 번째 줄은 자신의 값을 연산에 이용한 후 자기 자신에게 대입하는 형태이다. 그러나 특별할 것 없이 앞에서 배운 규칙을 그대로 적용하면 결과를 알 수 있다. 먼저 `var1`에 대입하기 전에 우측의 연산을 실행해야 하는데, `var1`의 현재 값이 첫 번째 줄에서 100이기 때문에 `100+200`을 먼저 수행하고, 계산 결과인 300을 `var1`에 다시 대입하라는 명령이다. 그래서 최종적으로 `var1`에는 300이 대입된다.

위의 표현은 프로그래밍 중에 상당히 많이 볼 수 있는데, 주로 자신의 값을 누적할 경우에 사용한다. 다시 말해 위의 표현은 `var1`의 현재 값에 200을 누적하라는 표현이 된다.

## 2.3 기본 데이터형

\_\_\_\_\_은 소수점이 없는 수를 저장하는 자료형이다. `int`는 가장 기본적인 정수형 타입이며, 파이썬 3.x에서는 크기의 제한이 없어 아주 큰 수를 부담 없이 사용해도 된다. 다음은 정수형(`int`) 사용의 예이다. 변수 `a`에는 100이 대입되고, `b`에는  $10^{100}$ 에 해당하는 매우 큰 값이 저장된다.

```
a = 100
b = 10 ** 100
```

다음은 실수형 사용의 예인데 소수점이 포함된 값을 대입하면 자동으로 실수형으로 저장된다. 그리고 코드의 `1.414e3`은  $1.414 \times 10^3$ 을 의미한다.

```
a = 1.414
b = 1.141e3
```

\_\_\_\_\_ 변수는 참(True)이나 거짓(False)을 저장할 수 있다. 다음은 사용 예이다. 변수 a는 True 값이 대입되며, b는 연산  $10 > 100$ 의 결과인 False가 대입된다. 파이썬에서 논리 연산(>, <, >=, <= 등)의 결과는 True 또는 False이다. 불형은 주로 if문 while문 등의 조건 부분에 주로 사용된다.

```
a = True
b = (10 > 100)
```

\_\_\_\_\_ (string)은 문자집합을 의미한다. 문자열은 양쪽을 큰 따옴표(“”)나 작은 따옴표(‘’)로 감싸야 한다. 문자열은 String의 약자인 str로 표현된다. 아래는 사용 예를 보여주고 있다. 변수 b는 정수 100(백)이 아닌 문자열 100(일영영)임을 주목할 필요가 있다.

```
a = “파이썬 프로그래밍”
b = ‘100’
```