

데이터 과학을 위한

파이썬 프로그래밍



02. 변수와 자료형

목차

1. 변수의 이해
2. 자료형과 기본 연산
3. 자료형 변환

01

변수의 이해

01. 변수의 이해

- 다음 코드는 Chapter 01에서 작성한 코드이다.

```
>>> professor = "Sungchul Choi"
>>> print(professor)
Sungchul Choi
>>> a = 7
>>> b = 5
>>> print(a + b)
12
>>> a = 7
>>> b = 5
>>> print("a + b")
a + b
```

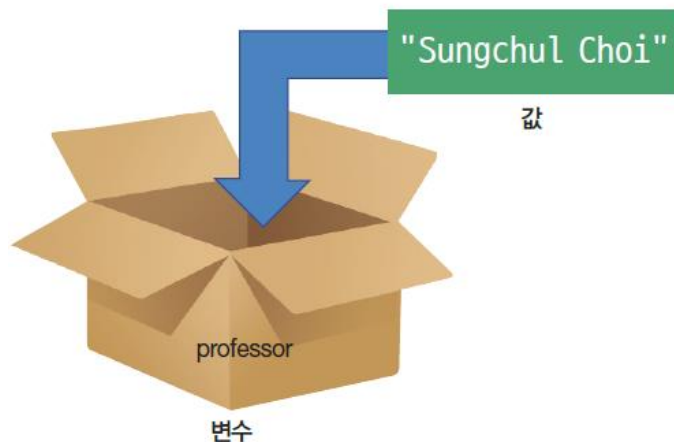
01. 변수의 이해

■ 변수와 값

- 앞의 코드를 하나씩 살펴보자. 먼저 1~3줄 코드는 다음과 같다.

```
>>> professor = "Sungchul Choi"  
>>> print(professor)  
Sungchul Choi
```

- 첫 번째 줄의 professor = "Sungchul Choi"라는 코드는 "professor라는 변수에 Sungchul Choi라는 값을 넣으라"는 뜻이다.



[변수와 값]

01. 변수의 이해

■ 변수와 값

- 다음 코드를 살펴보자. `print(a + b)`와 `print("a + b")`의 차이는 따옴표의 사용 여부에 따른 의미 차이에 있다.

```
>>> a = 7
>>> b = 5
>>> print(a + b)
12
>>> a = 7
>>> b = 5
>>> print("a + b")
a + b
```

코드	의미
<code>print(a + b)</code>	a 변수에 있는 값과 b 변수에 있는 값을 더해 화면에 출력하라.
<code>print("a + b")</code>	"a + b"의 값을 그대로 화면에 출력하라.

[따옴표의 사용 여부에 따른 의미 차이]

01. 변수의 이해

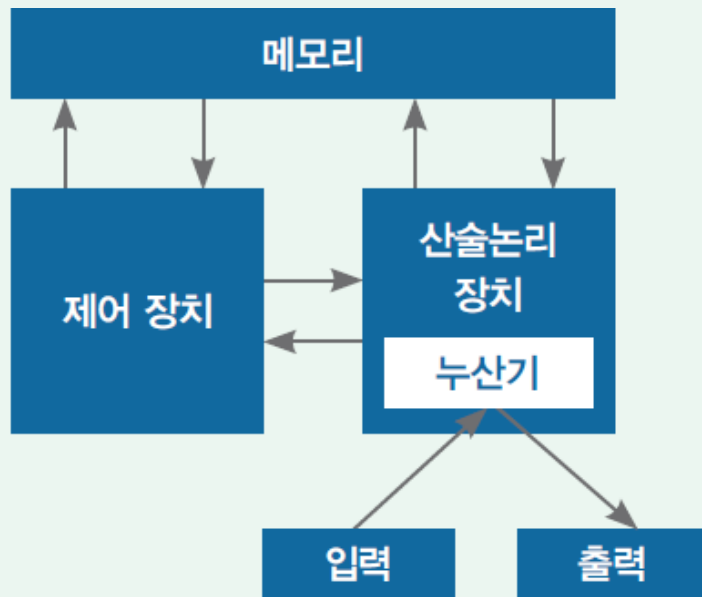
■ 변수와 메모리

- 프로그래밍에서 변수는 어떠한 값을 저장하는 장소라는 뜻으로 사용된다.
- 변수에 값이 저장되는 공간을 메모리라고 한다.
- 변수에 값을 넣으라고 선언하는 순간, 물리적으로 메모리 어딘가에 물리적인 공간을 확보할 수 있게 운영체제와 파이썬 인터프리터가 협력하여 메모리 저장 위치를 할당한다. 이 위치를 메모리 주소라고 한다.

01. 변수의 이해

여기서 잠깐! 컴퓨터의 구조: 폰 노이만 아키텍처

- 모든 컴퓨터에서는 값이 CPU로 가기 전에 반드시 메모리 공간에 저장되는데, 이 값을 CPU가 하나하나 돌아가면서 처리하는 구조가 오늘날 컴퓨터의 기본 구조인 폰 노이만 아키텍처이다

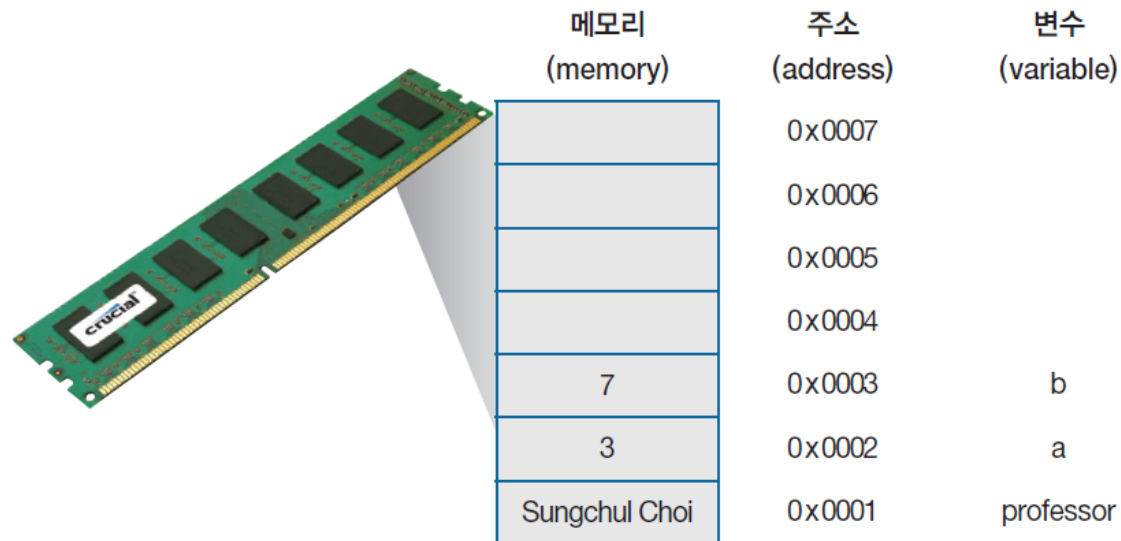


[폰 노이만 아키텍처]

01. 변수의 이해

■ 변수와 메모리

- professor = "Sungchul Choi", a = 3, b = 7과 같은 변수를 선언하면, 아래 그림과 같이 메모리 어딘가에 주소값을 할당받아 저장한다.



[메모리의 주소 할당]

01. 변수의 이해

■ 변수명 선언

- 알파벳, 숫자, 밑줄(_)로 선언할 수 있다.
- 변수명은 의미 있는 단어로 표기하는 것이 좋다.
- 변수명은 대소문자가 구분된다.
- 특별한 의미가 있는 예약어는 사용할 수 없다.

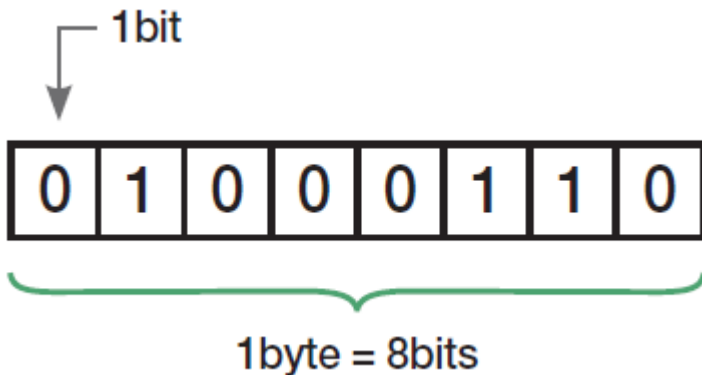
02

자료형과 기본 연산

02. 자료형과 기본 연산

■ 메모리 공간

- 하나의 변수를 메모리에 저장할 때, 그 변수의 크기만큼 공간(일정한 용량)을 할당받는다.
- 이진수 한 자리를 비트(bit)라고 하며, 8개의 비트는 1바이트(byte), 1,024바이트는 1킬로바이트(kilobyte, KB), 1,024킬로바이트는 1메가바이트(megabyte, MB)이다. 이러한 개념을 메모리 공간이라고 한다.



[비트(bit)와 바이트(byte)]

02. 자료형과 기본 연산

여기서 잠깐! 컴퓨터가 이진수를 사용하는 이유

- 컴퓨터는 왜 이진수를 사용할까? 컴퓨터의 메모리는 실리콘으로 만든 반도체이다. 반도체의 가장 큰 특징은 특정 자극을 주었을 때 전기가 통할 수 있어 전류의 흐름을 제어할 수 있다는 것이다. 이러한 성질을 이용해 반도체에 전류가 흐를 때 1, 흐르지 않을 때 0이라는 숫자로 표현할 수 있다. 따라서 메모리는 전류의 흐름을 이진수로 표현할 수 있다.

02. 자료형과 기본 연산

■ 기본 자료형

- 정수형(integer type) : 자연수를 포함해 값의 영역이 정수로 한정된 값.
- 실수형(floating-point type) : 소수점이 포함된 값.
- 문자형(string type) : 값이 문자로 출력되는 자료형.
- 불린형(boolean type) : 논리형으로, 참(True) 또는 거짓(False)을 표현할 때 사용.

유형	자료형	설명	예	선언 형태
수치형	정수형	양수와 정수	1, 2, 3, 100, -9	data = 1
	실수형	소수점이 포함된 실수	10.2, -9.3, 9.0	data = 9.0
문자형	문자형	따옴표에 들어가 있는 문자형	abc, a20abc	data = 'abc'
논리형	불린형	참 또는 거짓	True, False	data = True

02. 자료형과 기본 연산

여기서 잠깐! 동적 타이핑

- 동적 타이핑(dynamic typing)은 변수의 메모리 공간을 확보하는 행위가 실행 시점에서 발생하는 것을 뜻한다. 조금 어렵게 느낄 수도 있지만, 일반적으로 C나 자바는 `int data = 8` 과 같이 `data`라는 변수가 정수형이라고 사전에 선언한다. 그에 비해 파이썬은 `data = 8` 형태로 선언한다. 즉, `data`라는 변수의 자료형이 정수(integer)인지 실수(float)인지를 프로그래머가 아닌 인터프리터가 스스로 판단하는 것이다. 그리고 그것을 실행 시점에 동적으로 판단하므로 파이썬 언어가 동적으로 자료형의 결정을 지원한다.
- 다른 언어들과 달리 파이썬은 매우 유연한 언어로, 할당받는 메모리 공간도 저장되는 값의 크기에 따라 동적으로 다르게 할당받을 수 있다.

02. 자료형과 기본 연산

■ 기본 자료형

- 다음 코드를 파이썬 셸에 입력하여 실제 값이 화면에 출력되는지 확인한다.

```
>>> a = 1                # 정수형
>>> b = 1                # 정수형
>>> print(a, b)
1 1
>>> a = 1.5              # 실수형
>>> b = 3.5              # 실수형
>>> print(a, b)
1.5 3.5
>>> a = "ABC"            # 문자형
>>> b = "101010"         # 문자형
>>> print(a, b)
ABC 101010
>>> a = True             # 불린형
>>> b = False            # 불린형
>>> print(a, b)
True False
```


02. 자료형과 기본 연산

■ 간단한 연산 : 사칙연산

- 덧셈 기호(+), 뺄셈 기호(-), 곱셈 기호(*), 나눗셈 기호(/)

```
>>> 25 + 30
```

```
55
```

```
>>> 30 - 12
```

```
18
```

```
>>> 50 * 3
```

```
150
```

```
>>> 30 / 5
```

```
6.0
```

02. 자료형과 기본 연산

■ 간단한 연산 : 제곱승

- 2개의 별표 기호(**)

```
>>> print(3 * 3 * 3 * 3 * 3)    # 3을 다섯 번 곱함
243
>>> print(3 ** 5)              # 3의 5승
243
```

02. 자료형과 기본 연산

■ 간단한 연산 : 나눗셈의 몫과 나머지 산출 연산

- 몫을 반환하는 연산자는 2개의 빗금 기호(/), 나머지 연산자는 백분율 기호(%)

```
>>> print(7 // 2)           # 7 나누기 2의 몫
3
>>> print(7 % 2)           # 7 나누기 2의 나머지
1
```

02. 자료형과 기본 연산

■ 간단한 연산 : 증가 연산과 감소 연산

- 증가 연산자는 +=이고, 감소 연산자는 -=이다.

```
>>> a = 1                # 변수 a에 1을 할당
>>> a = a + 1            # a에 1를 더한 후 그 값을 a에 할당
>>> print(a)             # a 출력
2
>>> a += 1               # a 증가 연산
>>> print(a)             # a 출력
3
>>> a = a - 1            # a에서 1을 뺀 후 그 값을 a에 할당
>>> a -= 1               # a 감소 연산
>>> print(a)             # a 출력
1
```

03

자료형 변환

03. 자료형 변환

■ 정수형과 실수형 간 변환

- `float()` 함수 : 정수를 실수형으로 변환해 주는 함수.

```
>>> a = 10                # a 변수에 정수 데이터 10을 할당
>>> print(a)              # a가 정수형으로 출력
10
>>> a = float(10)         # a를 실수형으로 변환 / 정수형인 경우 int()
>>> print(a)              # a를 출력
10.0                      # a가 실수형으로 출력됨
```

```
>>> a = 10                # a에 정수 데이터 10 할당
>>> b = 3                 # b에 정수 데이터 3 할당
>>> print(a / b)          # 실수형으로 a 나누기 b를 출력
3.3333333333333335        # 실수형 결과값 출력
```

03. 자료형 변환

■ 정수형과 실수형 간 변환

- `int()` 함수 : 실수형을 정수형으로 변환해 주는 함수.

```
>>> a = int(10.7)
>>> b = int(10.3)
```

```
>>> print(a + b)           # 정수형 a와 b의 합을 출력
20
>>> print(a)               # 정수형 a값 출력
10
>>> print(b)               # 정수형 b값 출력
10
```

03. 자료형 변환

여기서 잠깐! 형 변환을 하지 않아도 형 변환이 일어나는 경우

- '10 / 3'처럼 별도의 형 변환을 하지 않아도 자연스럽게 자료형이 변환되는 경우가 있다. 이것도 역시 파이썬의 대표적인 특징인 동적 타이핑 때문에 나타나는 현상 중 하나이다. 이러한 현상은 값의 크기를 비교할 때도 나타난다. 대표적인 예로 1은 정수형이고 True는 불린형인데, 이것을 '1 == True'라고 입력하면 결과는 True로 출력된다. 또한, 아무것도 넣지 않은 " " 같은 문자열을 불린형과 비교하면 False로 인식된다. 모두 파이썬의 특징에 의해 나타나는 현상이므로 기억해야 한다.

03. 자료형 변환

■ 숫자형과 문자형 간 변환

- 실수형 값을 문자형으로 선언하기 위해서는 반드시 따옴표를 붙여 선언해야 한다.

```
>>> a = '76.3'           # a에 문자열 76.3을 할당, 문자열을 의미
>>> b = float(a)         # a를 실수형으로 변환 후 b에 할당
>>> print(a)             # a값 출력
76.3
>>> print(b)             # b값 출력
76.3
>>> print(a + b)         # a와 b를 더함 → 문자열과 숫자열의 덧셈이 불가능하여 에러 발생
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (nor "float") to str
```

03. 자료형 변환

■ 숫자형과 문자형 간 변환

- 두 변수를 더하기 위해서는 다음과 같이 두 변수의 자료형을 통일해야 한다.

```
>>> a = float(a)           # a를 실수형으로 변환 후 a에 할당
>>> b = a                  # 실수형 a값을 b에 할당
>>> print(a + b)           # 두 실수형을 더한 후 출력
152.6
```

03. 자료형 변환

■ 숫자형과 문자형 간 변환

- **str() 함수** : 기존의 정수형이나 실수형을 문자열로 바꿔 준다. 문자형 간의 덧셈은 숫자 연산이 아닌 단순 붙이기(concatenate)가 일어난다.

```
>>> a = str(a)           # 실수형 a값을 문자열로 변환 후 a 할당
>>> b = str(b)           # 실수형 b값을 문자열로 변환 후 b 할당
>>> print(a + b)         # 두 값을 더한 후 출력
76.376.3                # 문자열 간 덧셈은 문자열 간 단순 연결
```

03. 자료형 변환

■ 자료형 확인하기

- `type()` 함수 : 자료형을 확인할 수 있는 함수.

```
>>> a = int(10.3)           # a는 정수형으로 10.3을 할당
>>> b = float(10.3)         # b는 실수형으로 10.3을 할당
>>> c = str(10.3)           # c는 문자형으로 10.3을 할당
>>>
>>> type(a)                 # a의 타입을 출력
<class 'int'>
>>> type(b)                 # b의 타입을 출력
<class 'float'>
>>> type(c)                 # c의 타입을 출력
<class 'str'>
```

Thank You !