
라즈베리파이 파이썬

제작 임형택

목차

CONTENTS

01

파이썬의 특징

02

작성 규칙

03

기본 지식

04

기본 자료형

05

상황에 따른 처리

06

파일 저장

07

함수 만들기

08

모듈

01 파이썬의 특징

다른 C 언어와 다른 파이썬의 특징입니다.

- **가상 머신으로 실행**

C 언어로 작성한 프로그램은 컴퓨터가 실행할 수 있게 기계 언어로 컴파일한 후 실행해야 하지만 파이썬은 바로 실행할 수 있습니다. 그래서 특정 플랫폼에서 작성하여도 다른 플랫폼에서도 똑같이 동작하는 장점이 있습니다.

- **풍부한 라이브러리**

프로그래밍 언어 상당수는 자주 쓰는 처리를 프로그램으로 준비해 두고, 자유롭게 쓸 수 있게 합니다. 이런 프로그램을 라이브러리라고 합니다.

- **목적에 맞는 객체 지향 절차 전환**

객체 지향이란 프로그램과 데이터를 객체(object) 단위로 묶어 여러 객체로 전체 기능을 만들어 내는 것인데, 파이썬은 객체 지향 언어이지만 절차형으로 작성할 수 있다는 장점이 있습니다.

- **들여쓰기 블록 구조**

다른 프로그래밍 언어는 함수 정의나 if 문 같은 구문 범위를 중괄호({})를 주로 사용하지만 파이썬은 들여쓰기로 표현하는 특징이 있습니다.

02 작성 규칙

다른 프로그래밍 언어도 마찬가지지만, 파이썬은 프로그램을 작성할 때 사용하면 좋거나 필요한 규칙이나 있습니다.

2.1 | 파일명은 영문, 숫자 기호만 사용

프로그래밍 언어의 공통 규칙인데 프로그램에는 기본적으로 영문, 숫자, 기호만 사용합니다. 특수문자를 입력하면 오류가 발생하므로 주의해야 합니다.

2.2 | 공백

문장 시작 부분에 공백을 입력하면 안 됩니다(들여쓰기 필요한 부분 별개).
명령어나 변수 등의 요소 사이에 들어가는 공백은 몇 칸이 들어가는 상관은 없습니다.

2.3 | 프로그램의 빈 줄

프로그램 안에 빈 줄을 넣을 수 있습니다. 프로그램을 읽기 쉽게 작성하기 위해 적절한 곳에 빈 줄을 넣어 두면 좋습니다.

02 작성 규칙

다른 프로그래밍 언어도 마찬가지지만, 파이썬은 프로그램을 작성할 때 사용하면 좋거나 필요한 규칙이나 있습니다.

2.4 | 주석

프로그램을 작성하다 보면 숫자의 의미나 처리 내용 등을 설명하고 싶을 때가 있습니다. 파이썬은 #부터 다음 줄의 앞까지 주석으로 처리해서 프로그램으로 해석하지 않으므로 그 안에 자유롭게 주석을 작성할 수 있습니다.

예시)

```
1 from sklearn.model_selection import GridSearchCV
2
3 param_grid = [
4     # 12(=3×4)개의 하이퍼파라미터 조합을 시도합니다.
5     {'n_estimators': [3, 10, 30], 'max_features': [2, 3, 4]},
6     # bootstrap은 False로 하고 6(=2×3)개의 조합을 시도합니다.
7     {'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3, 4]},
8 ]
```

2.5 | 긴 줄 하나를 여러 줄로 작성

프로그램을 작성하다가 줄이 너무 길어지면 백슬래시(\)를 입력하고 줄 바꿈을 하면 됩니다. (참고로 백슬래시가 키보드에 안보일 수 있는데 (**W**) 이게 백슬래시 입니다. 깊은 사연이 있어요 ㅋㅋ~)

03 기본 지식

파이썬뿐만 아니라 어떤 프로그래밍 언어를 배우더라도 중요한 기본 지식을 배워보자.

3.1 | 컴퓨터에서 다루는 데이터 단위와 자료형

비트(bit): 컴퓨터에서 다루는 정보의 최소 단위 1비트 = '1 or 0'

바이트(byte): 8비트, 메모리 용량을 나타낼 때 사용하는 단위

자료형: 자료를 정수, 문자, 소수점 등 그 자료가 무엇을 표현하는지 분류한 단위

ex) int 형(integer, 정수)이라는 자료형은 정수를 나타내고, int형이 저장할 수 있는 크기는 32비트지만, 파이썬은 값이 커져 원래 크기를 초과하면 자동으로 비트 범위를 확장하므로 자료형의 크기를 신경 써야 할 일이 거의 없다.

3.2 | 논리 연산

논리 연산이란 컴퓨터에서 자주 사용하는 연산으로 '**참**'과 '**거짓**' 조건을 논리곱(AND), 논리합(OR), 부정(NOT)으로 연산하는 것이다.(&&, ||, ~는 파이썬에서 안 쓰입니다.)

AND
OR
진리표

AND	A	B	Z
1)	T	T	T
2)	T	F	F
3)	F	T	F
4)	F	F	F

OR	A	B	Z
1)	T	T	T
2)	T	F	T
3)	F	T	T
4)	F	F	F

03 기본 지식

파이썬뿐만 아니라 어떤 프로그래밍 언어를 배우더라도 중요한 기본 지식을 배워보자.

3.3 | 문자 정보 처리

복잡할 수도 있지만 간단하게 말하자면 문자 코드는 미국에서 정한 'ASCII'로 사용되는데, 한국에서도 한글을 다루기 위해 'EUC-KR', 'CP949' 등을 사용하였다.

하지만 호환성 문제 때문에 제대로 표시되지 않기도 하여서 유니코드 중 'UTF-8'이라는 방식을 사용하게 되었다(책에서도 UTF-8을 사용한다고 전제함).

3.4 | 함수, 메소드, 모듈 사용 방법

· 함수

함수는 어떤 단위로 묶어서 나눌 수 있는 처리를 하나의 블록으로 만든 것입니다. 함수에는 '인수'와 '반환 값'이 있는데 이는 수학의 함수 구조와 비슷합니다.

내장함수: 라이브러리 등을 불러오지 않고 바로 함수 형태로 이용할 수 있는 함수

수학의 함수

$$y = f(x)$$

함수 f 는 입력 값 x 에 따라 출력 값 y 를 결정함

프로그래밍의 함수

$$y = \text{int}('5')$$

함수 `int`는 인수인 문자열 '5'를 정수로 변환한 후 값을 반환 값으로 y 에 할당함.

03 기본 지식

파이썬뿐만 아니라 어떤 프로그래밍 언어를 배우더라도 중요한 기본 지식을 배워보자.

· 메서드

메서드란 객체 지향 프로그래밍에서 객체에 실행하는 어떤 '조작'입니다. 객체는 객체에 관련된 정보(데이터 또는 프로퍼티라고도 함)와 객체를 조작하는 메서드로 구성됩니다.

ex) 데이터 - 문자열, 메서드 - index(), int(), len() 등 → 데이터 + 메서드 = 객체

프로그램에서 내장 메서드를 사용하려면 메서드를 사용할 객체와 메서드명 사이에 마침표(.)를 쓰면 됩니다.

ex) 객체.메서드()

```
>>> letters = 'Python'
# index()는 지정한 문자가 문자열의 몇 번째인지를 조사함.
>>> letters.index('y')
# P는 0번째 자리, 보통 C언어에서는 0번째부터 자리 숫자를 셈.
1
```


03 기본 지식

파이썬뿐만 아니라 어떤 프로그래밍 언어를 배우더라도 중요한 기본 지식을 배워보자.

· 모듈

모듈은 시스템을 구성하는 독립성 높은 요소 1개를 가리킵니다.

파이썬 표준 라이브러리에는 복잡한 처리나 전문적인 내용에 특화된 다양한 객체, 함수, 상수를 정의해 둔 파일이 있는데, 이런 파일을 '모듈'이라고 합니다.

파이썬에서 모듈 기능을 사용하려면 프로그램 시작 부분에서 사용하려는 모듈을 **임포트(import)**해야 합니다.

ex) **import 모듈명**

임포트한 모듈에 포함된 함수를 사용하려면 모듈명을 지정해야 합니다. 메서드를 사용하는 것과 마찬가지로 모듈명과 함수명 사이에 마침표(.)를 써서 함수를 불러옵니다.

ex) **모듈.함수()**

```
>>> import math
# math 모듈의 sqrt(a)는 a의 제곱근을 돌려줌.
>>> math.sqrt(4)
# 4의 제곱근인 2.0이 반환됨.
2.0
```

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

4.1 | 파이썬의 내장 자료형

파이썬에 미리 준비된 자료형을 **내장 자료형**이라 합니다.

내장 자료형	설명	작성법	불변/가변
정수	소수가 없는 숫자	<code>a = 10, a = -10</code>	불변
부동소수	소수가 있는 숫자	<code>a = 10.5</code>	불변
불	True 또는 False 값을 가진 자료형	<code>a = True</code>	불변
문자열	문자 데이터를 순서대로 나열한 데이터. 문자가 1개뿐이어도 문자열이라고 부름	<code>a = 'Python'</code> 문자열의 0번째(첫 글자) 문자는 <code>a[0]</code> 으로 지정	불변
리스트	숫자나 문자열 등의 요소를 순서대로 나열한 구조체	<code>a = ['p', 'y', 't', 'h', 'o', 'n']</code> 리스트 <code>a</code> 의 0번째(첫 글자) 문자는 <code>a[0]</code> 으로 지정	가변
튜플	리스트와 비슷하지만 불변성 때문에 바뀌 쓸 수 없음.	<code>a = ('p', 'y', 't', 'h', 'o', 'n')</code> 리스트 <code>a</code> 의 0번째(첫 글자) 문자는 <code>a[0]</code> 으로 지정	불변
딕셔너리	각 데이터에 이름(키)이 있는 데이터 구조	<code>a = {'happy': '^', 'sad': '(ToT)'}</code> 딕셔너리 <code>a</code> 의 키 <code>happy</code> 에 대응하는 값은 <code>a['happy']</code> 로 지정	가변

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

4.2 | 변수

일반적으로 **변수**는 데이터를 저장하는 영역 또는 데이터 그 자체를 가리킵니다.
데이터를 변수에 할당하면 변수 명이 데이터 고유의 이름이 됩니다.
파이썬에서는 모든 객체를 변수에 할당할 수 있습니다.

· 변수에 할당하기

C나 자바는 변수를 사용하려면 변수명과 자료형을 미리 선언해야 하지만, 파이썬은 굳이 할당할 필요는 없다(알아서 정해지기 때문). 하지만 변수에 데이터를 할당해야 하는 상황이 언제나 있다. 변수에 데이터를 할당하려면 밑과 같은 방법으로 사용하면 된다.

변수 = 데이터(숫자나 문자 같은 다양한 객체)

공백은 있어도 되고 없어도 됨.

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 '숫자' 혹은 '문자'로 다루기 위해 정한 데이터의 종류를 말한다.

변수에 데이터를 할당하면 그 변수는 데이터 자체를 다루게 된다.

밑의 대화형 세션으로 변수 할당과 계산 예제를 참고해보자.

```
>>> a = 2
>>> b = 3
>>> a + b
5
```

파이썬에서 변수의 자료형은 할당할 때 정해집니다. 그런데 이미 데이터를 할당해서 사용한 변수에 다른 자료형 데이터를 할당하면 변수의 자료형이 나중에 할당한 데이터의 자료형으로 변합니다. 이처럼 할당한 데이터에 맞게 자료형이 유연하게 변하는 변수를 **동적 자료형 변수**라고 합니다.

```
>>> a = 5 # 숫자 할당
>>> a
5
```



```
>>> a = 'python' # 문자열 할당
>>> a
'python'
```

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

· 변수 이름 규칙

파이썬의 변수명은 다음과 같은 규칙을 따라야 된다.

1. 변수명에는 영문, 숫자, 밑줄(_)만 사용할 수 있음.
2. 대문자와 소문자를 구별함
3. 변수명은 숫자로 시작할 수 없음
4. 파이썬 예약어는 사용할 수 없음
5. 파이썬 내장 메서드나 내장 함수와 같은 이름은 피하는 것을 추천

이러한 규칙만 지키면 변수명을 자유롭게 사용할 수 있다.

하지만 어느 정도 규모가 있는 프로그램을 작성할 때는 나중에 알아보기 쉽게 어떤 데이터가 들어 있는지 추측할 수 있는 변수명을 붙이는 것이 좋다.

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

4.3 | 정수형, 부동소수형, 불형

· 정수형

정수란 소수점 이하를 포함하지 않는 숫자입니다.

- 기호를 붙이면 음수가 되고 아무것도 붙이지 않으면 양수가 됩니다.

파이썬의 정수형은 숫자의 크기 제한이 없습니다.

할당된 메모리가 허용하는 한 자릿수가 큰 숫자도 얼마든지 다룰 수 있습니다.

정수형 데이터는 10진수 외에도 2진수, 8진수, 16진수로 표기하기도 합니다.

표기법	접두어	사용할 수 있는 문자
10진수(Decimal)	0 외의 숫자	0~9
2진수(Binary)	0b 또는 0B	0, 1
8진수(Octal)	0o 또는 0O 0(숫자)와 알파벳 소문자 o 또는 대문자 O	0~7
16진수(Hexadecimal)	0x 또는 0X	0~9, a~f(또는A~F)

04 기본 자료형

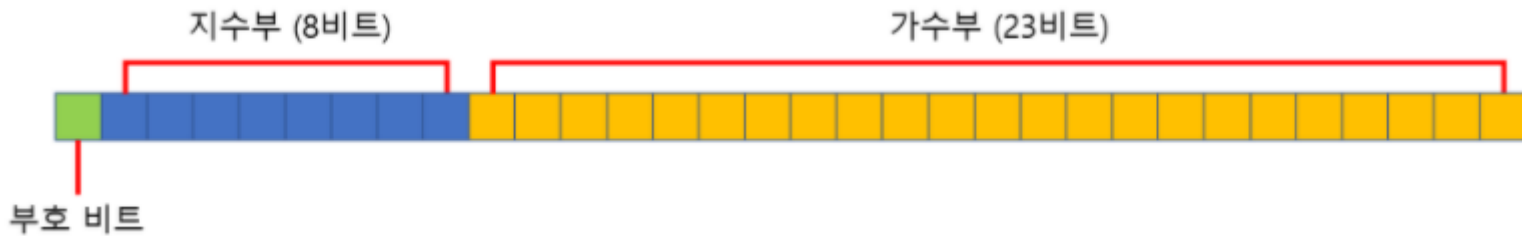
자료형이란 0과 1의 비트 나열을 프로그래밍에서 '숫자' 혹은 '문자'로 다루기 위해 정한 데이터의 종류를 말한다.

· 부동소수형

부동소수형은 소수점 이하를 포함하는 숫자로 가수부와 지수부를 조합하여 표시합니다.

예를 들어 123.456을 부동소수형으로 표시하면 가수부가 1.23456이고 지수부가 10^2 입니다.

같은 메모리 크기라도 소수점의 자릿수를 그대로 메모리에 저장하는 고정소수점보다 더 큰 자릿수를 다룰 수 있으므로 컴퓨터에서는 자주 사용하는 방식입니다.



내장 함수 `round()`를 사용하면 부동소수의 소수점 이하를 반올림할 수 있습니다.

```
round(반올림할 숫자[, 자릿수])
```

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 '숫자' 혹은 '문자'로 다루기 위해 정한 데이터의 종류를 말한다.

· 불형

불형은 불 대수로 다루는 True(참)와 False(거짓) 상태를 가진 자료형입니다. 연산의 처리 결과를 나타내는 자료형으로 자주 사용됩니다.

파이썬에서 True는 1, False는 0과 같습니다.

그래서 True 대신 1을 사용하거나 False 대신 0을 사용할 수 있지만, True/False 중 하나의 값만 가진 불형을 사용하면 프로그램을 쉽게 해석할 수 있습니다.

파이썬의 각 객체는 불형으로 다룰 때 True/False가 미리 정해져 있는데 다음 객체는 False가 됩니다.

- 숫자 0
- 빈 문자열, 리스트, 튜플, 딕셔너리

그 외의 객체는 모두 True가 됩니다.

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

4.4 | 숫자 값 계산

· 산술 연산자

일반 숫자 계산과 마찬가지로 파이썬에서도 +, -를 사용합니다.

연산자	의미 (우선순위가 낮은 것부터 작성)
$x + y$	x와 y를 더함(덧셈)
$x - y$	x에서 y를 뺌(뺄셈)
$x * y$	x와 y를 곱함(곱셈)
x / y	x를 y로 나눔(나눗셈. 정수끼리 계산하면 소수점 이하는 버림 소수 결과를 얻으려면 한쪽은 소수로 작성해서 계산해야함)
$x // y$	나눗셈의 몫을 구함
$x \% y$	나눗셈의 나머지를 구함
$x ** y$	제곱을 구함. X^Y (x의 y승)을 의미

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

· 복합 할당 연산자

복합 할당 연산자란 연산과 할당을 하나로 합친 것입니다.

연산식	의미
$x += y$	$x + y$ 의 결과를 x 에 할당
$x -= y$	$x - y$ 의 결과를 x 에 할당
$x *= y$	$x * y$ 의 결과를 x 에 할당
$x /= y$	$x \div y$ 의 결과를 x 에 할당
$x \% = y$	$x \div y$ 의 나머지를 x 에 할당
$x >> = y$	x 를 y 비트만큼 오른쪽으로 시프트해서 x 에 할당
$x << = y$	x 를 y 비트만큼 왼쪽으로 시프트해서 x 에 할당

변수에 복합 할당 연산자를 사용할 때에는 그 변수에 값이 미리 할당되어 있어야 오류가 발생하지 않습니다.

04 기본 자료형

자료형이란 0과 1의 비트 나열을 프로그래밍에서 ‘숫자’ 혹은 ‘문자’로 다루기 위해 정한 데이터의 종류를 말한다.

· 비트 연산자

비트 연산자는 정수형 값을 일반적인 숫자가 아닌 2진수 숫자로 보고 연산을 합니다.

비트 연산자에는 논리합, 논리곱, 부정, 배타적 논리합, 반전, 모든 비트 위치를 하나씩 옆으로 옮기는 **시프트 연산자**가 있습니다.