# 파이썬 기초 문법 마스터하기

2023.01.05 목요일 10:05

파이썬 기초 문법을 실습해보자. 파이썬 학습과 관련한 자료와 영상들이 온라인에 많이 있다. 본인의 경험 상 파이썬을 학습하는데 가장 효율적인 방법은 유튜브 영상과 온라인 자료를 따라하면서 기초 문법을 빠르게 익힌 후 관심 있는 프로젝트를 실제로 하는 것이다. 어느정도 숙련도가 높아진 이후에는 본인과 관련한 분야에 데이터 과학을 적용하는 학술 교재 혹은 학술 논문을 읽으며 응용하다 보면 상당한 성취를 이룰 수 있다.
본인의 경우 유튜브 영상들과 '점프투파이썬' 온라인 교재(https://wikidocs.net/book/1) 로 기본적인 사용법을 익힌 이후, 부동산 가격 지수/부동산 자동감정평가/부동산 가격 예측을 정형 데이터와 파이썬 라이브러리인 사이킷런이 제공하는 머신러닝 알고리즘을 통해 실습하였다. 이후 '퀀트 투자를 위한 머신러닝, 딥러닝 알고리즘 트레이딩'을 포함한 여러 권의 퀀트 투자 관련 책들을 학습하였다. 이러한 과정을 통해 파이썬에 대한 숙련도가 상당히 높아졌다.
이번 글은 파이썬의 핵심 문법을 다룬다. 필수 문법 대부분을 다루고 있어 해당 내용만 숙지해도 파이썬을 통한 데이터 활용이 가능하다. 처음에 낯설고 이해가 안되더라도 반복하다 보면 친숙해질 수 있다. 학습을 돕기 위해 유투브 영상으로 코딩 과정을 녹화하였다. 또한, 코딩의 결과물을 다음 링크에서 다운 받을 수 있다.
파이썬 코딩 파일 다운로드
https://www.notion.so/13b6cfa756de436c8fe282d01b6169bd
https://github.com/charmingcity/realestate_fintech/blob/main/B1_Python.ipynb

## 파이썬 핵심 문법

글에서는 개념 설명에 초점을 두었다. 핵심 문법의 종류와 개념적 이해를 돕기 위함이다. 실제 코딩을 살펴보고 따라하기 위해서는 앞선 글에서 설명한 "주피터 노트북" (혹은 다른 파이썬 개발환경)에서 실행해야 한다. 위의 링크에서 파이썬 파일을 다운로드 받아서 따라하면 도움이 된다.
기본 자료형으로 숫자형, 문자형이 있으며 구조와 관련된 자료형으로 **[배열형(리스트, 튜플), 집합, 사전, 부울]**이 있다. 자료형을 제어하는 제어문으로 **[조건문(if), 반복문(for/ while)]**이 있다. 프로그램의 반복적인 입력과 출력을 실행하는 **[함수]**가 있다. 마지막으로 기존의 개발해 놓은 코딩 파일을 가져다 사용할 수 있는 도구들로 **[클래스, 모듈, 패키지]**가 있다.

### 1. 자료형

어떤 프로그래밍 언어든 해당 언어의 '자료형'을 이해하면 절반 이상을 이해한 것과 같다는 말이 있다. 자료형이란 프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료 형태로 사용되는 모든 것을 의미한다. 프로그래밍의 기본이자 핵심 단위이다. 계산 프로그램을 만들기 위해서는 어떤 것을 계산할지 알아야 하며, 데이터베이스 프로그램을 만들려면 어떤 자료를 저장할지 아는 것부터 알아야 하듯 자료형은 프로그래밍의 시작이다.

| 자료형 객체 | 의미 | 사용 |
|---|---|---|
| 숫자형 int | 자연수 객체 | 자연수 |
| 숫자형 float | 실수 객체 | 실수 |
| 문자 자료형 str | 문자열 객체 | 글자 |
| 리스트형 list | 가변 컨테이너 | 변화하는 레코드 집함 |
| 튜플형 tuple | 불변 컨테이너 | 고정된 객체 집합, 레코드 |
| 집합형 set | 가변 컨테이너 | 유일한 객체 집합 |
| 사전형 dict | 가변 컨테이너 | 키와 값의 저장소 |
| 부울형 bool | 비교, 논리연산 | 참 또는 거짓 |

**1.1 기본형: 숫자 자료형**

숫자형이란 말 그대로 숫자 형태로 이루어진 자료형이다.

**1.1.1 정수형(int)**

1, 5, 10 와 같이 소수점이 없는 정수형 자료이다.

```
a = 10
type(a)
```

```
int
```

**1.1.2 실수형(float)**

실수 2.5와 같이 정수형 값에 소수점 밑에 수가 있는 실수형 자료이다.

```
a = 10     #정수형(int)
b = 2.5    #실수형(float)

print(type(a), type(b))
```

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

**1.1.3 문자 자료형 (str)**

말 그대로 문자를 의미, 문자의 양 옆에 큰 따옴표(") 혹은 작은 따옴표(')로 감싸서 정의한다. 숫자 양 옆을 따옴표를 감싸면 문자로 취급할 수 있다.

```
x = "python"
y = '서울역'
z = '2002'    #숫자형
print(x, y, z)
```

```
python 서울역 2002
```

문자 자료형에서 메소드를 활용하여 다양한 기능을 수행할 수 있다.

| 문자열 메소드 | 인수 | 반환 값 |
|---|---|---|

| 문자열 메소드 | 인수 | 반환 값 |
|---|---|---|
| count | (sub[, start[, end]]) | 문자열이 나타난 횟수 |
| find | (sub[, start[, end]]) | 특정 문자열이 처음으로 나타난 위치 |
| join | (seq) | 시퀀스 seq 문자열을 이어 붙인 문자열 |
| replace | (old, new[, count]) | 문자열 old를 처음부터 count 수만큼 문자열 new로 변환 |
| split | ([sep[, maxsplit]]) | 문자열을 sep 구분자로 구분한 리스트 |

```
#문자열에서 일부를 바꿀 때 replace 메소드 사용

s = "인공지능을 부동산 금융에 접목합니다."
print(s.replace("인공지능", "블록체인"))
```

```
블록체인을 부동산 금융에 접목합니다.
```

```
# split 메소드 : 문자열에 포함된 부분 문자열을 기

s = "A₩tB₩tC"
print(s.split("₩t"))
```

```
['A', 'B', 'C']
```

```
# format 메소드 : 문자열의 빈칸을 채우는 데 사용

A = 10
B = 20
C = 30
print("A= {}, B = {}, C = {}".format(A, B, C))
```

```
A= 10, B = 20, C = 30
```

### 1.3 자료의 구조: 배열형

#### 1.3.1 리스트(list)

리스트형은 가변형의 객체 모음으로 다양한 메소드를 제공한다. 자료의 변경이 가능하며 다양한 자료형을 포함한다. 자료의 순서대로 배열된다.

```
a = [1,2,3,4,5,6]
print(a)
print(type(a)) # <class 'list'>

b = ['a','hello',1,3.5,a,3.5,1]
print(b)
```

```
[1, 2, 3, 4, 5, 6]
<class 'list'>
['a', 'hello', 1, 3.5, [1, 2, 3, 4, 5, 6], 3.5, 1]
```

### 1.3.2 자료의 구조: 튜플(tuple)

튜플형은 불변형의 객체 모음으로 사용할 수 있는 메소드가 상대적으로 제한적이다. 튜플과 리스트는 거의 비슷하지만 값을 변화시킬 수 있느냐 없느냐 여부에 차이가 있다. 프로그램이 실행되는 동안 값이 변하지 않기를 바란다면 튜플을 사용해야 하며 반대로 값을 수시로 변환시켜야 하는 경우라면 리스트를 사용해야 한다.

```
a = (1,2,3,4,5)
print(a,type(a)) # <class 'tuple'>
```

```
(1, 2, 3, 4, 5) <class 'tuple'>
```

```
# indexing
print(a[0],a[1],a[2],a[-1],a[-2],a[True],a[False])

# a[0] = 10  # TypeError  , 요소 변경 불가
```

```
1 2 3 5 4 2 1
```

```
b = list(a)
b[0] = 10
b = tuple(b)
print(b)
```

```
(10, 2, 3, 4, 5)
```

### 1.4 자료의 구조: 집합 자료형(set)

집합형은 집합과 관련된 연산을 편리하게 한다. 집합 자료형은 집합의 원소가 되는 객체가 중복되지 않으며 순서가 없다.

```
a = [1,2,3,4,5,3,4,2,1,2,4,2,3,1,4,1,5,1]

a_set = set(a)

print(a_set)
```

```
{1, 2, 3, 4, 5}
```

```
# 집합 자료형에서는 교집합, 합집합, 차집합

# 교집합 연산: &, intersection 메서드
s1 = {1, 2, 3, 4}
s2 = {3, 4, 5, 6}
print(s1 & s2)
print(s1.intersection(s2))

# 합집합 연산: |, union 메서드
print(s1|s2)
print(s1.union(s2))

#차집합 연산: -, difference 메서드
print (s1 - s2)
print(s1.difference(s2))
```

```
{3, 4}
{3, 4}
{1, 2, 3, 4, 5, 6}
{1, 2, 3, 4, 5, 6}
{1, 2}
{1, 2}
```

### 1.5 자료의 구조: 사전 자료형(dictionary)

사전형은 [키(key)-값(value)] 두 개가 쌍으로 이루어진 저장소 객체이다. 사전형은 리스트와 튜플처럼 자료의 순서대로 '순차적'으로 출력하지 않고 키와 값을 통해 출력한다.

```
X = {"A":1, "B":2, "C":3}
print(X["A"])


X["D"] = 4
X["B"] = 20
print(X)

#del 함수

del(X["A"])
print(X)


# 딕셔너리 구성 요소를 확인하는 데 사용하는
# key()는 딕셔너리의 모든 키를, values()는

X = {1:"A", 2:"B", 3:"C"}
print(X.keys())
print(X.values())
print(X.items())
```

```
1
{'A': 1, 'B': 20, 'C': 3, 'D': 4}
{'B': 20, 'C': 3, 'D': 4}
dict_keys([1, 2, 3])
dict_values(['A', 'B', 'C'])
dict_items([(1, 'A'), (2, 'B'), (3, 'C')])
```

**1.6 자료의 구조: 부울 자료형 (bool)**

비교나 논리연산을 통해 [참과 거짓] 값을 출력한다.

```
a = True
b = False
print(a, b)
print(type(a), type(b))
```

```
True False
<class 'bool'> <class 'bool'>
```

```
# 부울 연산자

print(True and True)
print(True and False)
print(False & False)
print(False & True)
print(True or False)
print(True | False)
print(False | False | False)
print(True + True)    # 사칙연산을 할 때 True = 1, False = 0
print(True*False)
print(False - True)
print(True/True)
```

```
True
False
False
False
True
True
False
2
0
-1
1.0
```

**2. 제어 구조: 조건문 (if문)**

조건문에는 if라는 키워드를 사용한다. If 다음에 '조건'이 존재하는데 조건이 참(True)이면 들여쓰기한 문장이 실행된다. If 문의 끝에는 콜론(:)을 입력한다. If 문의 조건이 참(True)일 때 실행되는 문장은 들여쓰기해야 한다.

```
cond = True
if cond:
    print("참")
else:
    print("거짓")
```

```
참
```

**3. 제어 구조: 반복문 (for문, while 문)**

**3.1 for문**

반복적인 작업을 수행할 때 for문을 사용한다.

```
T = (1, 2, 3, 4, 5)
for a in T:
    x = a ** 2
    y = a ** 3
    print(a, x, y)
```

```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
```

```
X = [[1, "A"], [2, "B"], [3, "C"]]
for a, b in X:
    print(a, b)
```

```
1 A
2 B
3 C
```

**3.2 while 문**

일반적으로 for문은 반복 횟수가 미리 정해져 있거나 리스트, 튜플, 사전과 같은 파이썬 자료구조와 함께 사용된다. 반면 while 문은 반복해야 할 횟수가 특별히 정해지지 않고 특정한 조건을 충족하는 동안만 실행할 때 사용한다.

```
x = 1
while x <=5 :
    print(x)
    x +=1
```

```
1
2
3
4
5
```

**4. 입력과 출력: 함수**

수학 시간에 배운 함수(function)과 동일하게 input 값을 넣으면 output 값을 출력한다.

```
def f(x): #인자가 x인 함수 f를 정의
    y = 2*x - 1
    return y

print(f(1))
```

```
1
```

**5. 객체지향 프로그래밍: 클래스**

객체지향 프로그래밍이란 소프트웨어 개발에 필요한 모든 요소를 객체화하여 프로그래밍하는 기법을 말한다. 프로그램의 설계도에 해당하는 클래스를 작성한 뒤 클래스로부터 객체를 생성하여 원하는 동작을 수행한다.

**5.1 클래스**

클래스란 객체를 생성하는 틀로, 데이터 멤버(data member)라고 불리는 속성(attribute)과 동작을 수행하는 메소드(method)로 구성된다. 클래스로부터 객체를 생성하는 것을 인스턴스화(instantiation)이라고 한다. 생성된 인스턴스가 가지고 있는 속성과 메소드는 닷(.)을 사용해서 접근할 수 있다.

```
class Ourclass:
    pass

a = Ourclass()
b = Ourclass()

print(type(a))
print(type(b))
```

```
<class '__main__.Ourclass'>
<class '__main__.Ourclass'>
```

```
import pandas as pd
df = pd.DataFrame()
x = 1

print(type(df))
print(type(x))
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'int'>
```

**5.2 클래스 상속**

상속은 클래스가 가지고 있는 모든 속성과 메소드를 다른 클래스에 물려주는 방식이다. 상속을 해주는 클래스를 부모 클래스라고 하며 상속받는 클래스를 자식 클래스라고 일컫기도 한다.

```
class Stock:     # stock이라는 이름의 클래스 선언
    def __init__(self, name, market = "코스피"):
        self.name = name      # 속성 정의 1
        self.market = market    # 속성 정의 2

samsung = Stock("삼성전자", "코스피") #클래스 호출
print(samsung.name, samsung.market)
```

```
삼성전자 코스피
```

**6. 패키지와 모듈**

파이썬의 최고의 장점은 강력한 라이브러리를 사용할 수 있다는 것이다. 프로그램을 개발할 때 모든 기능을 처음부터 끝까지 직접 개발하는 경우는 드물다. 다른 컴퓨터 과학자가 모듈(module) 단위로 코드를 작성해 두어 가져다 쓸 수 있다. 모듈은 미리 만들어 놓은 파이썬(.py) 파일이다.
패지지(package)는 여러 모듈을 특정 디렉터리에 모아 둔다. 닷(.)을 사용하여 파이썬 모듈을 계층적으로 관리할 수 있게 해준다. 예를 들어, scipy.spatial가 있으면 scipy는 패키지 이름이고 spatial은 패키지의 모듈이다. 모듈이란 함수나 변수 또는 클래스를 모아 놓은 파일이다. 다른 파이썬 프로그램에서 불러와 사용할 수 있게 만들어졌다. 본인이 만들어 놓거나 다른 사람이 만들어 놓은 모듈을 가져와 사용할 수 있다.

**참고 자료**

코딩 1시간만에 배우기 – 파이썬, Teccboi Wonie / https://www.youtube.com/watch?v=M6kQTplqpLs&t=3882s
파이썬 코딩 무료 강의 (기본편) - 6시간 뒤면 여러분도 개발자가 될 수 있어요, 나도코딩 https://www.youtube.com/watch?v=kWiCuklohdY&t=142s

최신 파이썬 코딩 무료 강의 - 5시간만 투자하면 개발자가 됩니다 / 조코딩https://www.youtube.com/watch?v=KL1MIuBfWe0&t=3589s
점프 투 파이썬 / 박은용 지음 / 이지퍼블리싱
퀀트투자 with 파이썬 / Girl`s LAB 지음 / 위키북스
파이썬 머신러닝 판다스 데이터 분석 / 오승환 지음 / 정보문화사

---

**조철민**

SPI APAC 대표 부동산 금융 산업에 IT 기술을 도입하여 산업의 진보를 이끕니다. 거시 경제, 리츠, 퀀트, 공간 웹에 대한 연구를 기록합니다.

---