K-DT BigData#1 (Python)

[K-DT 강사 아카데미] Big Data Track #1. Python Programming

[K-DT#01_230612] Python Programming

(1) Python 기초

- 개요: Interpreter 언어. Guido van Rossum이 개발(1990)
  ※파르노스 산의 동굴에 살던 큰 뱀(고대 신화)
- 편집 및 실행 환경
  - Python IDLE(*.py): 파이썬 설치 시 기본적으로 설치
  - IDE 계열(*.py): Visual Studio Code (Python Extension 설치 필요), PyCharm (Python 전용 Editor)
  - Notebook 계열(*.ipynb): Google Colab (Colaboratory), Jupyter Notebook
  - 가상 환경(Virtual Environment) 생성: python -m venv 이름
- 자료형
  - 숫자형(Number): 정수형(integer), 실수형(floating-point), 복소수(complex)
  - 문자열(String): len(a) 길이, 인덱싱(각 문자별), 슬라이싱
  - 리스트: [1, 2, 3], 인덱싱, 슬라이싱, 변경(append, insert, sort, pop, del)
  - 튜플: (1, 2, 3) ※리스트와는 달리 요솟값을 지우거나 변경할 수 없음
  - 딕셔너리: Key와 Value를 한 쌍으로 갖는 자료형 → {Key1:Value1, Key2:Value2, ...}
  - 집합: s1 = set([1,3,2]) → 중복 미허용, 순서 없음(Unordered) ※인덱싱이 무의미
  - 불(bool):True/False의 두 가지 값만 가짐
- 제어문
  - 조건문(if): ① if 조건문: ₩ else: ② if 조건1: ₩ elif 조건2: ₩ elif 조건3: ₩ ...₩ else:
  - 반복문(while, for)
- 함수
  - def 함수명(매개변수): return 결과값 → 매개변수(parameter)는 여러 개 가능하나, 결과값은 하나 (혹은 없음)
    ※ 함수 호출 시 전달하는 입력값은 인수(arguments)로 구분함
- 파일 읽고 쓰기
  - f = open("새파일.txt", 'w') ₩ f.close()

    - 모드: r(읽기), w(쓰기), a(추가), b(binary file) → 예: rb, rt(textdml 't'는 생략 가능)
- 패키지(packages)
    - 도트(.)를 사용하여 Python 모듈을 계층적으로 관리할 수 있게 해 줌
    - import: import pandas as pd ₩ from myPack import calculator
    - 해당 패키지가 없으면 설치: pip install 패키지
- 예외 처리 - try: ₩ except: ₩ else: ₩ finally:
- 데이터 구조: Scalar(0차원), Vector(1차원), Matrix(2차원), Tensor(n차원)
    - Pandas: DataFrame과 Series라는 자료형과 데이터 분석을 위한 다양한 기능을 제공하는 Python 라이브러리 → Table, 머신러닝, 숫자/문자 혼합, 열 머리글

      import pandas as pd
      df = pd.read_csv('./data/아파트실거래가.csv')
    - NumPy: 배열, 딥러닝, 숫자, 행렬 연산
    - BeautifulSoup: 데이터 구조화

(2) Algorithm

- 정렬
    - 선택 정렬: for i in range(len(lst)): → i번째 이후 숫자 중 최소의 숫자를 제일 앞(i번째)으로 이동 저장
    - 삽입 정렬: for i in range(1, len(lst)): → i로 선택된 부분에 해당하는 전체 비교 후, 큰 수를 뒤로 밀어냄
    - 퀵정렬: 둘로 나눈 후 각각 다시 재귀함수로 계속해서 각각 순서 정렬
    - Pandas의 Series 정렬을 이용하면 바로 정렬됨
- 탐색
    - 1차원 탐색, 2차원 탐색
    - Pandas 탐색(2차원), Pandas의 DataFrame 정렬(2차원)

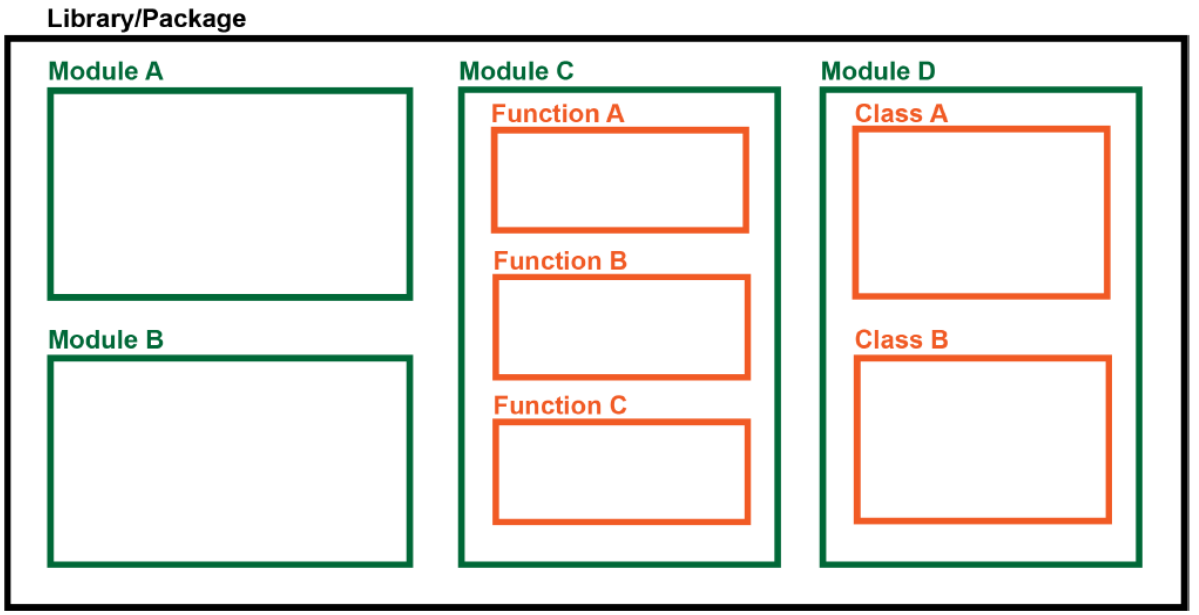
(3) Python 고급 기능

- 패키지, 모듈, 함수, 클래스
    - 패키지(package)
        - 라이브러리라고도 부름
        - 특정 기능과 관련된 여러 모듈을 한 그룹으로 묶은 것
        - 패키지 안에 패키지가 있을 수도 있다
        - import 패키지 | import 모듈 from 패키지
    - 모듈(module)
        - 여러 기능들이 뭉쳐진 하나의 .py 파일
        - 함수, 클래스, 변수 등 포함

        - import 모듈
    - 함수(function)
        - 하나의 기능을 가진 코드의 집합
        - (함수를 사용한다) = (함수를 호출한다)
        - 함수 사용 형태: 함수( ) | 함수(매개변수 또는 옵션)
    - 클래스(class)
        - Python은 객체 지향 언어로, 클래스를 기반으로 객체를 만들어 사용한다
        - 한 Python 파일(.py) 내에서 "class class명( ):"을 통해 클래스 선언
- Methods-Functions 비교

Methods	Functions
Objects와 연관됨	어떠한 Objects와도 연관되지 않음
이름으로 호출(invoke)할 수 없음	이름으로 호출(invoke) 가능
클래스(class)에 종속됨	클래스와 독립적
'self'를 필요로 함	'self'를 필요로 하지 않음

- 클래스 만들기
    - setdata방식 → def setdata(self, num1, num2):
    - init 방식 → def __init__(self, num1, num2):
    - method 방식(가장 많이 사용) → def __init__(self):
- 모듈 실습



    - 클래스 상속
    - 클래스 재정의
- 예제 코드: 지도학습(Supervised learning) 중 회귀(Regression)
    - 패키지 import : import statsmodels.api as sm | import pandas as pd | import joblib
    - DataFrame 정의 : df = pd.read_csv('trade_apt_api.csv')
    - 독립/종속변수 설정: y = df['Price'] ₩ X = df[[종속변수들]] ₩ Bias(절편) 추가, X = sm.add_constant(X)
    - Regression 수행: model = sm.OLS(y, X).fit( ) ※OLS = Ordinary Least Squares
    - 임의의 데이터에 대한 결과 예측: model.predict([임의의 종속변수값])
    - 모델 저장: joblib.dump(model, 'apt_price.pkl')

(4) 프로그램 실행

- [Pyinstaller] Windows 실행 파일(*.exe) 생성 : pyinstaller --onefile filr_name.py
- Argument와 Batch 처리: 매번 실행하는 작업을 단축해 주는 batch 파일
- [tkinter] GUI 구성: Python 기본 내장 ※Pyqt는 다양한 디자인 제공하지만, 별도 설치가 필요
- [xlwings] Excel 연동: VB 연동 및 Excel↔Python 양방향 제어 가능하지만 별도 설치가 필요. (openpyxl은 Python 기본 내장이지만, 기능이 제한적)
- Scheduler 연동: 반복적인 작업을 일정에 맞추어 실행하는 '작업 스케줄러' 사용하여 수집 작업 등을 규칙적으로 실행

(5) 프로그램 운영

- 웹서버 세팅: goormIDE 이용(https://ide.goorm.io/) → 컨테이너 생성 (이후 설정 및 실행)
- 웹서버 연동: Flask 기본 코드 실행

(End of Document)