파이썬 스터디

# 스터디 일정

* 일정 : 2023년 5월 ~ 모든 과정이 끝날 때 까지
* 횟수 : 주 1회 2시간
* 장소 : 비투엔 회의실

# 스터디 목표

* 파이썬으로 자유롭게 전처리하고, 데이터를 다루는 디테일 능력 향상
* 컴퓨터 사이언스(Computer Science) 지식 및 프로그램 아키텍처의 기본 이해

# 기본 준비물

* Laptop (OS : Window)
* IDE : Visual Studio Code

# 1. Python Project Env

* Workspace Environmet Management : venv, virtualenv, conda
* Package Management Tool : pip, conda
* Package Dependency Management: requirements.txt
* Useful VSCode Extension : Python Docs, python snippets, Python Type Hint
* Useful Jupyter Magic Keyword (%autoreload, %memit)

# 2. Memory Management

* Memory Layout : Code, Data, Heap, Stack
* Everything is an object : Object, First-Class Function, Class & Instance
* Python Implementation : CPython
  + Global Interpreter Lock
  + Garbage Collector
* Mutable/Immutable (Call-By-Value & Call-By-Reference)

# 3. Python Basic Grammar

* Python Built-in Functions
* Comment (Line & Paragraph)
* Don’t memorize function parameters
* If Grammar (If ~ else) & (Ternary Operator)
* Loop Grammar (for ~ else) & (while) & (list/dict Comprehension)
* Garbage Code Example

# 4. Primitive Data Type

* Primitive Data Type(Literal)
  + int / str/ float / bool
  + nan
* None
* Type Inference & Cast Operation (type casting)
* Detail Float Operation : Decimal

# 5. Container Data Type

* Python Builtin Container Type
  + List (Linked List)
  + tuple
  + dict (Hash Table & Hash Collision Resolution)
  + set
* Collections (Sepecial Container Data Type)
  + Counter, deque, namedtuple, OrderedDict, defaltDict
* 3rd Party Library Data Type
  + numpy.ndarray
  + pandas.Series
  + pandas.DataFrame
* Container Type Function : lambda, map, apply, applymap

# 6. 파이썬 변수/함수/클래스/모듈/패키지 관리

* Naming (Pascal Case, Camel Case, Snake case, Hungarian Notation)
* variable, function (+ magic method), class, module, package
* OOP – Encapsulation, Inheritance, Hiding, Polymorphism
* 3rd party document reading / Source code peeking

# 7. Useful Built-in Modules

* os (absolute path, relative path)
* sys (import paths management) & Env Files: Export PYTHONPATH
* subprocess

# 8. Deep Python

* Object-Oriented Programming
* iterator / iterable / generator / decorator
* logging
* Parallel Processing 1: Multi-Threading, Multi-Processing
* Parallel Processing 2: Inter-Process-Communication Programming
* Exception Handling

# 특별부록 1 : 분석가 전처리 Detail Up

* numpy
  + array, matrix, n-dimension
* pandas
  + Series, DataFrame
    - column dtype(object, np.int, np.Int, np.float ..)
    - str preprocessing (str.function, regex)
  + pandas preprocessing functions

# 특별부록 2 : Computer Science

* CPU, Memory, Disk
* Operating System & Application Architecture
* Process & Thread ( + namespace, docker)
* Network : IP, Port, session, socket, driver, etc..
* Git