

[Data트랙] 월말평가1 – Python



| Background

- ✓ 자료구조에 대한 이해와 사용
- ✓ 알고리즘 문제 해결 능력 향상
- ✓ 파이썬 기초 문법에 대한 이해

| Goal

- ✓ 파이썬을 사용하여 알고리즘 문제를 해결 할 수 있다.

| 환경 설정

1) 파일 이름 및 제출 방법

- 1, 2번 문제에 대한 소스 파일은 python문제번호_지역_반_이름.py로 만든다.
- 3번은 텍스트 파일로 작성한다.

python1_대전_4반_김싸피.py

python2_대전_4반_김싸피.py

python3_대전_4반_김싸피.md

python4_대전_4반_김싸피.md

- 위 3개의 파일만 지역_반_이름.zip으로 압축하여 제출한다.

대전_4반_김싸피.zip

(탐색기에서 파일 선택 후 오른쪽 클릭 – 보내기 – 압축(zip)폴더 선택)

3) 채점

- 주석이 없는 경우, 주석이 코드 내용과 맞지 않는 경우, 지정된 출력 형식을 만족하지 않는 경우 해당 문제는 0점 처리될 수 있다.
- import를 사용한 경우 해당 문제는 0점 처리될 수 있다. (import sys도 예외 없음)

4) 테스트케이스는 부분적으로 제공되며, 전체가 공개되지는 않는다.

5) 각 문제의 배점이 다르므로 표기된 배점을 반드시 확인한다.

- 1번 35점, 2번 35점, 3번 15점, 4번 15점

성실과 신뢰로 테스트에 임할 것 (부정 행위시 강력 조치 및 근거가 남음)

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시,

0점 처리 및 학사 기준에 의거 조치 실시 예정



[Data트랙] 월말평가1 – Python

| 문제 1 : 텍스트 전처리기 – 불량 데이터를 정제하라 (배점 35점)

아래는 로그 문자열 데이터를 정제하기 위한 함수들이다.

각 함수는 설명된 기능을 정확히 수행해야 하며, 최종적으로 process_logs() 함수를 실행하면 결과가 리스트로 반환되어야 한다. 주어진 스켈레톤을 반드시 유지하며 함수 내부만 구현하시오.

스켈레톤 코드 및 요구 조건

```
def strip_and_lowercase(text):
    """
    문자열의 앞뒤 공백을 제거하고, 소문자로 변환한 문자열을 반환한다.
    """
    pass

def is_valid_log(text):
    """
    로그 항목이 유효한지 검사한다.
    유효하지 않은 경우는 다음 조건 중 하나라도 해당되는 경우이다.
    - 공백 문자열
    - "error" 또는 "none" 이 포함된 문자열
    - 숫자로만 구성된 문자열
    """
    pass

def clean_log(text) :
    """
    '_'가 포함된 경우 이를 ' '로 바꾸고,
    각 단어의 첫 글자를 대문자로 바꾼다. (Title Case)
    """
    pass

def process_logs(log_list) :
    """
    전체 로그 리스트를 받아 유효한 항목만 정제하여 리스트로 반환한다.
    위 함수들을 적절히 활용해야 한다.
    """
    pass

# 아래 코드는 수정 할 수 없음
result = process_logs(raw_logs)
print(result)
```



[Data트랙] 월말평가1 – Python

| 문제 1 : 텍스트 전처리기 – 불량 데이터를 정제하라 (배점 35점)

입력 예시

```
raw_logs = [  
    " user_login ",  
    "ERROR_404",  
    " page_viewed",  
    "None",  
    "signup_success ",  
    " 1234 ",  
    " ",  
]
```

출력 예시

```
['User login', 'Page viewed', 'Signup success']
```

[Data트랙] 월말평가1 – Python



| 문제 2 : 클래스를 직접 설계하시오 (배점 35점)

도서 정보를 객체로 관리하는 Book 클래스를 작성하시오.

클래스 내부의 설계, 메서드 구분, 이름공간 활용 등을 평가하며, 다음 조건을 반드시 포함해야 한다.

생성자를 사용하여 책의 제목, 저자, 출판연도, 가격을 저장할 수 있어야 한다.

1. `__init__` 생성자: title, author, year, price를 인자로 받아 인스턴스 변수로 저장한다.
 - 가격은 0 이상만 허용하며, 0 미만일 경우 `ValueError`를 발생시킨다.
2. 클래스 변수 `book_count`:
 - 생성된 도서 인스턴스 수를 추적하며, 생성자 호출 시 1씩 증가한다.
3. 인스턴스 메서드 `get_info()`:
 - [제목 - 저자, 연도년, 가격원] 형식으로 문자열 반환
4. 정적 메서드 `is_expensive(price)`:
 - 가격이 30,000원 이상이면 `True`, 아니면 `False` 반환
5. 클래스 메서드 `get_book_count()`:
 - 현재까지 생성된 도서 객체 수를 반환
6. 클래스 메서드 `from_string(cls, info_str)`:
 - '제목/저자/연도/가격' 형태의 문자열을 받아 Book 인스턴스를 반환

입력 및 출력 예시

```
book1 = Book("파이썬 완벽 가이드", "홍길동", 2023, 35000)
book2 = Book.from_string("자료구조의 이해/이순신/2021/28000")

print(book1.get_info())
# [파이썬 완벽 가이드 - 홍길동, 2023년, 35000원]

print(Book.is_expensive(35000)) # True
print(Book.get_book_count())    # 2
```



[Data트랙] 월말평가1 – Python

| 문제 3 : 가변성과 복사 (배점 15점)

다음 코드의 출력 결과를 예상하고, 그 이유를 '가변성과 복사'의 개념을 중심으로 설명하시오.

```
a = [1, 2, 3]
b = a
b[0] = 100
print(a)
```

| 문제 4 : 이름공간 (배점 15점)

아래 코드의 출력 결과를 예상하고, 인스턴스 변수와 클래스 변수의 탐색 순서와 관련하여 그 이유를 설명하시오.

```
class MyClass:
    value = 10

obj = MyClass()
obj.value = 20

print(MyClass.value)
print(obj.value)
```