

python #4

▼ 문제

1단계: .env 작성

- env 파일을 만들고 다음 키-값을 저장
 - LOG_LEVEL=???
 - APP_NAME=???
- LOG_LEVEL에는 DEBUG 또는 INFO,
- APP_NAME에는 자유로운 앱 이름 입력

2단계: 환경 변수 로드

- python-dotenv 모듈 설치
 - pip install python-dotenv
- .env 파일을 읽어오는 코드 작성 (힌트: load_dotenv() 사용)
- os.getenv("LOG_LEVEL") 로 변수 읽기

3단계: logging 설정

- 로그 파일명: app.log
- 로그 포맷: 시간 [레벨] 메시지
- 로그 레벨: .env에서 가져온 값 사용 (getattr(logging, log_level))
- 콘솔 + 파일 동시에 출력되도록 핸들러 구성

4단계: 로그 출력 테스트

- INFO 레벨 메시지: "앱 실행 시작"
- DEBUG 레벨 메시지: "환경 변수 로딩 완료"
- ERROR 레벨 메시지: ZeroDivisionError 예외 발생 시 출력

5단계: 실행 후 확인

- 콘솔 화면 출력 확인
- app.log 파일 내용 확인

콘솔 화면의 결과와 env 파일 내용, app.log에 기록된 내용, 그리고 개별로 작업하셨던 폴더의 구조에 대해 화면 Capture 하셔서 댓글로 올려주세요.

▼ 코드

1. .env 작성

```
LOG_LEVEL=DEBUG
APP_NAME=MyCoolApp
```

2. main.py 작성

▼ 전체 코드

```
import os
import logging
from logging import StreamHandler, FileHandler
from dotenv import load_dotenv

def configure_logging(level, filename="app.log"):
    """
    콘솔과 파일 모두에 로그를 출력하도록 설정하는 함수
    """
```

```

# 로그 메시지 포맷 지정: 시간, [로그레벨], 메시지
fmt = "%(asctime)s [%(levelname)s] %(message)s"
formatter = logging.Formatter(fmt, "%Y-%m-%d %H:%M:%S")

# 콘솔 출력 핸들러 생성
sh = StreamHandler()
sh.setFormatter(formatter)

# 파일 출력 핸들러 생성
fh = FileHandler(filename, encoding="utf-8")
fh.setFormatter(formatter)

# 루트 로거에 핸들러 적용 (기본 설정)
logging.basicConfig(level=level, handlers=[sh, fh])

def main():
    # 1) .env 파일 로드
    load_dotenv()

    # 2) 환경 변수에서 로그 레벨과 앱 이름 읽기
    # .env에서 LOG_LEVEL을 읽어와 logging 모듈 상수로 변환
    log_level = getattr(logging, os.getenv("LOG_LEVEL"), logging.INFO)
    app_name = os.getenv("APP_NAME", "MyApp")

    # 3) 로깅 설정 적용
    configure_logging(log_level)

    # 4) 로그 메시지 출력
    logging.info("앱 실행 시작") # INFO 레벨
    logging.debug(f"환경 변수 로딩 완료 (APP_NAME={app_name})") # DEBUG 레벨

    # 5) 예외 발생 예시
    try:
        1 / 0 # ZeroDivisionError
    except ZeroDivisionError:
        # ERROR 레벨로 예외 메시지와 스택 트레이스 출력
        logging.error("예외 발생 예시", exc_info=True)

if __name__ == "__main__":
    # main() 함수 실행
    main()

```

▼ `configure_logging` 함수

```

def configure_logging(level, filename="app.log"):
    """
    콘솔과 파일 모두에 로그를 출력하도록 설정하는 함수
    """
    # 로그 메시지 포맷 지정: 시간, [로그레벨], 메시지
    fmt = "%(asctime)s [%(levelname)s] %(message)s"
    formatter = logging.Formatter(fmt, "%Y-%m-%d %H:%M:%S")

    # 콘솔 출력 핸들러 생성
    sh = StreamHandler()
    sh.setFormatter(formatter)

```

```
# 파일 출력 핸들러 생성
fh = FileHandler(filename, encoding="utf-8")
fh.setFormatter(formatter)

# 루트 로거에 핸들러 적용 (기본 설정)
logging.basicConfig(level=level, handlers=[sh, fh])
```

- filename="app.log"
 - 로그 파일명: app.log
- fmt = "%(asctime)s [%(levelname)s] %(message)s"
 - 로그 포맷: 시간 | 로그레벨 | 메시지
- sh = StreamHandler(), fh = FileHandler(filename, encoding="utf-8")
 - 로그 출력: 콘솔 + 파일 모두
- logging.basicConfig(level=level, handlers=[sh, fh])
 - 콘솔 + 파일 동시에 출력되도록 핸들러 구성

▼ main 함수

```
def main():
    # 1) .env 파일 로드
    load_dotenv()

    # 2) 환경 변수에서 로그 레벨과 앱 이름 읽기
    # .env에서 LOG_LEVEL을 읽어와 logging 모듈 상수로 변환
    log_level = getattr(logging, os.getenv("LOG_LEVEL"), logging.INFO)
    app_name = os.getenv("APP_NAME", "MyApp")

    # 3) 로깅 설정 적용
    configure_logging(log_level)

    # 4) 로그 메시지 출력
    logging.info("앱 실행 시작") # INFO 레벨
    logging.debug(f"환경 변수 로딩 완료 (APP_NAME={app_name})") # DEBUG 레벨

    # 5) 예외 발생 예시
    try:
        1 / 0 # ZeroDivisionError
    except ZeroDivisionError:
        # ERROR 레벨로 예외 메시지와 스택 트레이스 출력
        logging.error("예외 발생 예시", exc_info=True)

if __name__ == "__main__":
    # main() 함수 실행
    main()
```

- load_dotenv()
 - .env 파일을 읽어오기
- os.getenv("LOG_LEVEL")
 - .env에서 LOG_LEVEL을 읽어오기
- getattr(logging, <string>, logging.INFO)

- <string>에 해당하는 속성을 logging 모듈에서 찾아 return.
ex. "DEBUG" → logging.DEBUG, "INFO" → logging.INFO
- os.getenv("APP_NAME", "MyApp")
 - .env에서 APP_NAME을 읽어오기
- configure_logging(log_level)
 - configure_logging 함수 실행
 - 콘솔 + 파일(app.log) 두 군데에 동시에 로그가 찍히도록 핸들러를 구성.
- logging.info("앱 실행 시작")
logging.debug(f"환경 변수 로딩 완료 (APP_NAME={app_name})")
 - 로그 메시지 출력
- logging.error("예외 발생 예시", exc_info=True)
 - 예외 발생 시 에러 로깅
- 파일 실행 시
 - main() 호출, 함수 수행.

▼ 결과물 제출

1. 콘솔 화면의 결과

```
(skala) yshmbid:env_logging_example yshmbid$ python main.py
2025-08-12 17:13:40 [INFO] 앱 실행 시작
2025-08-12 17:13:40 [DEBUG] 환경 변수 로딩 완료 (APP_NAME=MyCoolApp)
2025-08-12 17:13:40 [ERROR] 예외 발생 예시
Traceback (most recent call last):
  File "/Users/yshmbid/Documents/home/github/Data-ML0ps/env_logging_example/main.py", line 43, in main
    1 / 0 # ZeroDivisionError
    ~~~~
ZeroDivisionError: division by zero
```

2. env 파일 내용

```
.env
1 LOG_LEVEL=DEBUG
2 APP_NAME=MyCoolApp
```

3. app.log에 기록된 내용

```
.env main.py 1, M app.log M
1 2025-08-12 17:13:40 [INFO] 앱 실행 시작
2 2025-08-12 17:13:40 [DEBUG] 환경 변수 로딩 완료 (APP_NAME=MyCoolApp)
3 2025-08-12 17:13:40 [ERROR] 예외 발생 예시
4 Traceback (most recent call last):
5   File "/Users/yshmbid/Documents/home/github/Data-ML0ps/env_logging_example/main.py", line 43, in main
6     1 / 0 # ZeroDivisionError
7     ~~~~
8 ZeroDivisionError: division by zero
9
```

4. 작업 폴더 구조

```
env_logging_example
├── .env
├── app.log
└── main.py
```

