

# Ask Django

파일에 저장하고 읽어오기

# 데이터를 영속성있게 저장하려면???

- 메모리 내 데이터는 프로세스가 종료되면, bye bye ~
- 영속성있게 저장하려면 파일에 저장해야합니다.
  - 파이썬에서는 open함수를 통해 파일 읽기/쓰기 지원

# 파일 모드 (**Read, Write, Append**)

- [모드 R] 기존 파일 읽기
- [모드 W 혹은 A] 새 파일 생성해서 쓰기
- [모드 W] 기존 파일 내용 제거하고, 처음부터 쓰기
- [모드 A] 기존 파일에 추가하기

# 파일의 종류

- TEXT : 문자열 데이터
  - 자동 인코딩/디코딩이 있으면, 더 편리하겠죠?
- BINARY : 바이너리 데이터
  - 자동 인코딩/디코딩을 구지 수행하지 않습니다.
  - 대개 문자열이 아닌, 이미지/PDF/XLS 포맷 등
  - TEXT 데이터여도 BINARY로 열수도 있겠죠.

# open

# **open (파일 쓰기/읽기 함수) #doc**

```
file_obj = open(파일경로, mode='r', encoding=None, 그외옵션생략)
readed_data = file_obj.read() # 파일 내용 처음부터 끝까지 모두 읽기
file_obj.close()
```

- file object 주요 멤버함수
  - .write 함수 : 파일에 쓰기
  - .read 함수 : 파일 읽기
  - .close 함수 : 파일 닫기

# encoding 옵션

- 자동 인코딩/디코딩 인코딩 옵션
- text mode 시에만 지정 가능. binary 모드에서는 지정불가
- 미지정시에 OS 설정에 따라, 다른 인코딩이 지정
  - `locale.getpreferredencoding(False)`
  - 한글 윈도우 : cp949
  - 맥/리눅스 : 대개 utf-8

가급적이면 모든 파일은 *utf-8* 인코딩으로 생성합니다.

-- AskDjango

- 요즘 소스코드 편집기의 디폴트 인코딩은 UTF-8
- 다양한 인코딩이 혼재하면, 정신건강에 좋지 않아요.
- 특별히 cp949인코딩이 필요한 경우가 아니라면, utf-8 로 대동단결 !!!
  - ex) 한글 윈도우 엑셀에서 쓸 csv 파일 만들기
    - (utf8인코딩으로도 할 수는 있어요... - BOM사용)



# 파일을 열 때, 5가지 모드

- r (read), w (write), a (append)
- 인코딩 모드
  - t (text) : 자동 인코딩/디코딩 모드
    - read() 반환타입은 str
    - write() 인자로 str 타입 필요
  - b (binary) : 바이너리 모드
    - read() 반환타입은 bytes
    - write() 인자로 bytes 타입 필요
- 지정 예
  - rt (read + text), rb, wt, wb, at, ab

## r (read)

```
filecontent_unicode = open('filepath.txt', 'rt', encoding='utf8').read()
```

- 지정 경로에 파일이 없을 경우 **IOError** 예외 발생
- 지정 경로의 파일에 대해 읽기권한이 없을 경우 **PermissionError** 예외 발생

## w (write)

```
open('filepath.txt', 'wt', encoding='utf8').write('가나다') # 유니코드 문자열 (str)
```

- 지정 경로 파일이 없을 경우, 해당 내용으로 새 파일 생성
- 지정 경로 파일이 존재할 경우, 해당 내용을 무시하고, 새로이 파일 생성
- 지정 경로의 파일이 존재하지만, 쓰기권한이 없을 경우  
**PermissionError** 예외 발생
- 지정 경로 내에 없는 디렉토리가 지정된 경우, **FileNotFoundError**  
예외 발생

## a (append)

```
open('filepath.txt', 'at', encoding='utf8').write('가나다') # 유니코드 문자열 (str)
```

- w (write) 와 유사
- 지정 경로 파일이 존재할 경우, 해당 내용에 이어서, 내용 추가

## t (text)

```
with open('filepath.txt', 'wt', encoding='utf8') as f:  
    f.write('가나다')
```

- 지정 **encoding**으로, 자동 인코딩/디코딩과 함께 파일 쓰기/읽기

## b (binary)

```
with open('filepath.txt', 'wb') as f:  
    f.write('가나다'.encode('utf8'))
```

- 자동 인코딩/디코딩없이 파일 쓰기/읽기
- encoding 옵션 지정 불가
- 문자열이 아닌 파일을 읽어들이 때에는 인코딩/디코딩을 수행하면 안 되므로, 필히 binary모드를 지정

```
with open('myphoto.jpg', 'rb') as f:  
    photo_data = f.read() # bytes 타입
```

# 파일에 접근하는 다양한 방법

```
f = open('sample.txt', 'rt', encoding='utf8')
print(f.read())
f.close()
```

```
f = open('sample.txt', 'wt', encoding='utf8')
f.write('hello ')
print('world', file=f) # file에 넘겨진 인자는 .write 멤버함수만 지원하면 OK (오리 타이핑)
f.close()
```



**열려진 파일은 항상 닫아줘야합니다.**



- 닫고 싶어도 닫기 전에 예외가 발생하면 닫을 수가 없 ...

```
f = open('sample.txt', 'wt', encoding='utf8')
f.write('hello ')
1/0 # ZeroDivisionError 예외가 발생
f.close()
```

- 다음과 같이 예외처리를 통해, 꼭 닫아줘야합니다.

```
f = open('sample.txt', 'wt', encoding='utf8')
try:
    f.write('hello ')
    1/0 # ZeroDivisionError 예외가 발생
finally: # 예외 발생여부에 상관없이 무조건 실행
    f.close()
    print('file closed.')
```

# with 절

- 특정 block을 **with절**을 통해, 해당 block의 실행전/실행후/예외발생시의 처리를 with절을 통해 처리가능
- open함수에서 with절을 지원하지만, with절을 한 번 만들어보자.
- class를 통한 with절 지원도 가능 (클래스 EP에서 다뤄집니다.)

```
from contextlib import contextmanager

@contextmanager
def myopen(filepath, mode, encoding):
    f = open(filepath, mode, encoding)
    try:
        yield f      # with절의 as에 넘겨줍니다.
    finally:
        f.close()

with myopen('helloworld.txt', 'wt', 'utf8') as f:
    f.write('hello ')
    f.write('world ')
```

## **open** 함수에서의 **with**절 지원 #ref

```
with open('sample.txt', 'rt') as f:  
    file_content = f.read()  
  
print(file_content)
```

## **file object** 는 순회 가능(**Iterable**)한 객체 #ref

줄 (line) 단위로 순회

```
with open('sample.txt', 'rt', encoding='utf8') as f:
    for line in f:
        print(line)
```

## 소스파일 저장은 필히 **UTF-8**로 해주세요.

- 요즘 소스코드 에디터의 디폴트 인코딩은 UTF-8
- 파이썬에서는 각 소스파일 최상단에 **소스코드 자체의 인코딩 지정 가능**
  - Python 3 : 디폴트 utf-8
  - Python 2 : 디폴트 **ascii** (이 디폴트 인코딩이 원흉 !!!)

```
# -*- coding: utf-8 -*-
```

- 파이썬은 소스코드를 실행하기 전에, 소스파일의 내용을 먼저 **디코딩**
- 소스코드 인코딩을 파이썬에게 잘못 알려주면, `SyntaxError` 예외 발생
- 이는 주석이라고 해서 예외가 없습니다. 파일 내 모든 문자열이 해당
- 참고 : UTF8, 유니코드에 대한 오해

*Life is short,  
use Python3/Django.*