

파이썬 프로그래밍

12. 파일 처리

❖ 수업 목표

- 파일을 열고 닫을 수 있다.
- 파일을 입출력 할 수 있다.
- 파일 내용을 추가 할 수 있다.
- 자동으로 파일을 닫을 수 있다.

❖ 세부 목표

- 12.1 파일 개요
- 12.2 파일 열기와 닫기
- 12.3 파일 쓰기와 읽기
- 12.4 파일 내용 추가
- 12.5 자동 파일 닫기

1. 파일 개요

❖ 파일 개요

- 파일(File): 컴퓨터에서 데이터를 저장하는 기본 단위
 - 관련 정보 자료들의 집합에 이름을 명명한 것
 - 텍스트, 이미지, 비디오 등의 정보를 포함 가능
 - 파일은 일반적으로 보조기억장치(HDD, SSD, USB 등)에 저장
 - 파일 이름과 확장자로 식별
- 파일 구성
 - 이름(Name): 파일 식별을 위한 문자열
 - 데이터(Data): 파일의 내용. 예) 텍스트, 이미지, 비디오 등 다양한 형태
 - 메타데이터(Metadata): 파일의 정보 예) 파일 크기, 생성날짜, 수정날짜, 접근 권한 등

1. 파일 개요

❖ 파일 분류

- 텍스트 파일 : 사람이 읽을 수 있는 문자 형태로 저장된 파일
 - .txt / .csv / .html 등
 - 텍스트 에디터를 사용하여 내용 확인 가능
- 바이너리 파일 : 기계가 이해할 수 있는 이진 형식으로 저장된 파일
 - .exe / .jpg / .mp3 등
 - 내용 확인을 위해서는 특화된 전용 프로그램 필요

❖ 파일 시스템(File System)

- 운영 체제에서 파일을 저장, 관리, 검색하는 시스템
- 파일에 접근하려면 관리 도구나 프로그래밍 언어를 사용
- 파일을 처리하기 위해서 파일 이름과 파일 경로가 필요

1. 파일 개요

❖ 디렉터리(폴더)

- 운영체제에서 다수의 파일을 관리하기 위한 목적으로 파일을 분류하는 공간
- 디렉터리에는 다수의 파일 또는 서브 디렉터리가 존재
- 윈도우즈에는 드라이브라는 폴더와 파일을 분류할 수 있는 구조가 존재

1. 파일 개요

❖ 파일 경로(File Path)

■ 절대 경로

- 최상위 폴더부터 파일이 있는 위치까지 모두 포함하여 표현
- 드라이브부터 표현
 - 예) C:/rokey/ch19/pandas/data.csv

■ 상대 경로

- 현재 작업 디렉터리를 기준으로 상대적인 위치를 표현
- 상대 경로의 표시를 위해서 [.] 또는 [..]을 사용
 - [.] 은 현재 경로를 의미
 - [...] 은 상위 경로를 의미
 - » 예) 전체 경로가 C:/rokey/ch19/pandas/data.csv 인 경우
 - » 현재 프로그램 실행 경로가 C:/rokey/ch19 라고 하면
 - » data.csv 파일의 위치를 상대경로로 표현하면 다음과 같음
 - » ./pandas/data.csv
 - » 예) 전체 경로가 C:/rokey/a.txt 인 경우
 - » 현재 프로그램 실행 경로가 C:/rokey/ch19 라고 하면
 - » a.txt 파일의 위치를 상대경로로 표현하면 다음과 같음
 - » ../a.txt

1. 파일 개요

❖ 파일 경로(File Path)

- 파일 코드를 사용하여 현재 작업 디렉터리를 확인하는 방법
 - import os
 - print(os.getcwd())
 - 실행 결과
 - D:\rokey

2. 파일 열기와 닫기

❖ 파일 열기(생성)하기

- **open()** : 내장클래스

- 파일 객체 생성. 이후, 파일 객체의 멤버 함수를 사용하여 입출력 작업 수행

- 파일객체명 = **open(경로/파일이름, 파일열기모드)**

- `f = open("F:/rokey/ch12/file1.txt", 'w')`

❖ 파일 열기 모드

- **r** : 읽기모드 – 파일을 읽을 때 사용

- **w** : 쓰기모드 – 파일에 내용을 쓸때 사용

- 파일이 이미 존재하는 경우, 원래 내용이 삭제
 - 파일이 없는 경우, 새로운 파일 생성

- **a** : 추가모드 – 파일의 마지막에 새로운 내용을 추가할 때 사용

2. 파일 열기와 닫기

❖ 인코딩(Encoding)

- 데이터를 컴퓨터가 이해할 수 있는 바이너리 형식으로 변환하는 것
 - 예) 대문자 A는 65를 이진수로 표현하고 소문자 a는 97을 이진수로 표현해야 기계가 이해 가능
- 인코딩 규칙을 표로 정리
 - ASCII (American Standard Code for Information Interchange, 미국 정보 교환 표준 부호)
 - 미국에서 표준화한 정보 교환용 7비트 부호 체계
 - 유니코드 (Unicode)
 - 16비트 32비트 문자 체계, 모든 문자를 다루도록 설계된
 - 국제 표준 문자 전산 처리방식

❖ 디코딩(Decoding)

- 바이너리 형식에서 사람이 이해할 수 있는 문자로 변환하는 것

2. 파일 열기와 닫기

❖ 인코딩(Encoding)

■ 한글 인코딩

- CP949(확장 완성형)
 - CP949는 이전에 유닉스 계열의 EUC-KR에 생긴 문제를 해결하기 위해 제작
 - ‘MS949, WINDOWS949’ 라고도 불림
- UTF-8(유니코드)
 - 유니코드를 인코딩하는 방식 중 하나

2. 파일 열기와 닫기

❖ 파일 입출력과 문자 인코딩

- 파일을 저장 할 때에는 내용을 특정 방식을 통해 인코딩 함
- 파이썬은 운영체제에서 사용하는 방법으로 기본 인코딩 수행
 - 윈도우즈 인코딩 방식: CP949
 - 기본 인코딩 방식이 아닌 다른 형태로 파일을 읽고 쓰려면 인코딩 방식을 지정해야 함
 - 비주얼 스튜디오 코드는 기본 인코딩은 UTF-8
 - 만약, VSCode에서 CP949로 인코딩된 텍스트 파일을 열면 한글이 깨져서 보임

2. 파일 열기와 닫기

❖ 파일 닫기

■ **close()** 메서드

- 열려 있는 파일 객체를 닫고 사용한 자원을 반환

■ 파일객체명.**close()**

- **f = open("F:/rokey/ch12/file1.txt", 'w')**
- **f.close()**

3. 파일 쓰기와 읽기

❖ 파일에 데이터 쓰기

■ **write()** 메서드

- 프로그램의 출력값을 파일에 쓰기
- 리턴 값은 저장된 문자의 바이트 수

■ 파일객체명.write(data)

- `f = open("F:/rokey/ch12/새파일.txt", 'w')`
- `for i in range(1, 11):`
- `data = "%d번째 줄입니다.\n" % i`
- `f.write(data)`
- `f.close()`

3. 파일 쓰기와 읽기

❖ 파일 읽기

■ **f.readline()**

- 파일을 line을 읽어서 결과값을 반환

■ 파일객체명.readline()

- **path = "./ch12/file1.txt"**
- **f = open(path, 'r')**
- **line = f.readline()**
- **print(line)**
- **f.close()**

3. 파일 쓰기와 읽기

❖ 파일 읽기

■ **f.readlines()**

- 파일의 모든 줄을 읽어서 각각의 줄을 요소로 갖는 리스트로 반환

■ 파일 객체명.readlines()

- path = "F:/rokey/ch12/file1.txt"
- f = open(path, 'r')
- lines = f.readlines()
- for line in lines:
- print(line, end='')
- f.close()

3. 파일 쓰기와 읽기

❖ 파일 읽기

■ **f.read()**

- 파일의 내용 전체를 문자열로 반환

■ 파일 객체명.read()

- **path = "F:/rokey/ch12/file1.txt"**
- **f = open(path, 'r')**
- **data = f.read()**
- **print(data)**
- **f.close()**

4. 파일 내용 추가

❖ 파일 내용 추가

■ 기존 데이터를 유지하면서 새로운 값을 추가

- `path = "F:/rokey/ch12/file1.txt"`
- `f = open(path,'a')`
- `for i in range(11, 20):`
- `data = "%d번째 줄입니다.\n" % i`
- `f.write(data)`
- `f.close()`

5. 자동 파일 닫기

❖ **with .. as 문**

- 파일을 열고 닫는 것을 자동으로 처리
- **with open(파일경로, 모드) as 파일객체:**
- 처리코드블록
 - **path = "F:/rokey/ch12/file2.txt"**
 - **mode = "w"**
 - **with open(path, mode) as f:**
 - **f.write("No pain, no gain.")**

5. 자동 파일 닫기

❖ with 문

syntax : with open()

파일 쓰기

〈open() 함수〉

```
1. vfileobj = open("vfilesaved.txt", "w")
2. vfileobj.write(vwordwrite)
3. vfileobj.close()
```

〈with open() 함수〉

```
with open("vfilesaved.txt", "w") as vfileobj: -> ①
    vfileobj.write(vwordwrite)
```

파일 읽기

〈open() 함수〉

```
1. vfileobj = open("vfilesaved.txt", "r")
2. vb=vfileobj.read()
3. vfileobj.close()
print(vb)
```

〈with open() 함수〉

```
with open("vfilesaved.txt", "r") as vfileobj: -> ②
    vb = vfileobj.read()
print(vb)
```

6. 연습 문제

- ❖ rokey 폴더 내 file 폴더를 생성하고
 - 해당 폴더에 '계좌1.txt' 파일을 생성한 후,
 - 예금주와 계좌번호를 파일에 작성하세요.
 - 예)
 - 김삿갓 597-89-000089
 - 이수근 343-64-000064
 - 박혁거세 136-97-000097

6. 연습 문제

- ❖ 앞서 생성한 file 폴더의 '계좌1.txt' 읽어서 계좌번호를 리스트로 반환 및 저장하세요.

6. 연습 문제

- ❖ 앞서 생성한 file 폴더의 '계좌1.txt' 읽어서 예금주와 계좌번호를 딕셔너리로 저장해보세요.
 - 예금주를 key로, 계좌번호를 value로 저장하시오.

6. 연습 문제

- ❖ 앞서 생성한 file 폴더의 '계좌1.txt' 파일에 다음 예금주와 계좌번호를 추가하세요.
 - 강호동 147-12-002093
 - 유재석 146-22-102093

6. 연습 문제

❖ 피자 주문 프로그램을 주문 내역 파일로 저장하기

- 터미널을 통해 주문 입력
- 주문 내역은 order.txt 파일로 저장
- 다음과 같이 주문 내역 쓰기
 - 주문 내역:
 - 치즈피자 (3200원) x 3 9600원
- 주문 내역 읽어 화면에 출력하기

6. 연습 문제

❖ 피자 주문 프로그램을 주문 내역 파일로 저장하기

- 터미널을 통해 주문 입력
- 주문 내역은 order.txt 파일로 저장
- 다음과 같이 주문 추가하기
 - 주문 내역:
 - 치즈피자 (3200원) x 3 9600원
 - 사이다 (1500원) x 2 3000원
 - 주문 내역 읽어 화면에 출력하기

❖ 과제

- 1. 파일을 열고 닫는 코드 작성하기
- 2. 파일을 쓰고 읽는 코드 작성하기
- 3. 파일 내용을 추가하는 코드 작성하기
- 4. 파일을 자동으로 닫는 코드 작성하기

❖ 다음 수업 내용

- 예외 처리, 문자열, 람다 함수, map 함수
 - 인덱싱, 슬라이싱, 문자열 함수