8장 파일 입출력

대용량의 데이터를 한 번에 처리하기 위해서는 파일을 통한 입출력이 필수적이다. 이번 장에서는 파일 입출력의 기본에 대해여 학습한다. 파일 입출력은 지금까지 해온 콘솔(화면)에서 키보드를 통한 입출력과 개념이 매우 비슷하다. 키보드 대신 파일의 내용을 한 단위씩 읽어서 프로그램 내부의 변수나 리스트와 같은 컨테이너에 저장하여 필요한 작업을 하는 방식이다.

콘솔에서 입력할 때 _____ 함수를 사용했다면 파일 입력과 관련해서는 read(), readline(), readlines()와 같은 함수를 사용한다. 이 함수들은 파이썬에서 기본으로 제공하고 있다. 또한 콘솔에서 출력은 주로 _____ 함수를 사용하였는데, 파일 출력과 관련해서는 write(), writelines()와 같은 함수가 사용된다.

8.1 파일 입출력의 기본 과정

파일 입출력을 위해서는 다음의 3단계를 거쳐야 한다. 1단계는 _____이며, 2단계는 _____, 3단계는 _____이다. 말 그대로 파일을 열어서 읽거나 쓴 다음 파일을 닫는 과정을 진행하면 된다.

<1단계> 파일 열기

```
읽기용: 변수명 = open("파일명", "r")
쓰기용: 변수명 = open("파일명", "w")
```

읽기용은 open() 함수의 마지막 파라미터로 ____의 약자인 r을 사용하고, 쓰기용은 _____의 약자인 w를 사용하는 것을 제외하고는 동일하다. 이외에도 r+, a, t, b와 같은 매개 변수를 사용할 수 있는데, r+는 읽기/쓰기 겸용 모드를 말하고, a는 append의 약자로 기존 파일에 이어서 쓴다는 뜻이다. t는 텍스트모드를 말하고, b는 binary의 약자로 이진 파일을 처리한다. 예를 들어 일반 텍스트 파일의 읽기 모드는 r 또는 rt를 사용하고, 이진 파일을 출력할 때는 wb를 사용한다.

<2단계> 파일 처리
read() 나 write() 같은 함수로 읽기 및 쓰기를 진행한다. 자세한 설명은 추후 예

제를 통해 하겠다.

<3단계> 파일 닫기
파일을 닫은 때는 open() 함수로 파일을 열 때 이용한 변수명을 통하여 '변수명.close()'를 하면 된다.

8.2 텍스트 파일의 입력

텍스트 파일을 입력할 때에는 read(), readline(), readlines() 함수를 사용한다. 이 중 readline()은 텍스트 파일을 한 줄씩 읽어 들인다. 먼저 메모장에 아래의 내용을 입력한 후 저장하자. readline() 함수를 이용해서 아래 내용의 텍스트 파일을 읽어 들여 화면에 출력하는 프로그램을 작성할 것이다. 텍스트 파일은 c:\Temp에 text.txt라고 가정하자.

```
가을이 오면 눈부신 아침 햇살에 비친
그대의 미소가 아름다워요
눈을 감으면 싱그런 바람 가득한
그대의 맑은 숨결이 향기로와요
```

다음은 파일을 읽어서 그대로 화면에 출력하는 프로그램이다. 파일이 4행의 텍스트로 구성되어 있기 때문에 readline() 함수를 __ 번 호출하여 읽어 들인다. 텍스트 파일의 경로를 지정할 때 역슬래시(\)가 아닌 _____를 사용한다는 점에 주의하자.

```
inStr = ""    #읽어온 문자열을 저장할 변수
inFp = open("C:/Temp/text.txt", "r")

inStr = inFp.readline()
print(inStr, end="")
inStr = inFp.readline()
print(inStr, end="")
inStr = inFp.readline()
print(inStr, end="")
inStr = inFp.readline()
print(inStr, end="")
```

```
inFp.close()
```
```
<실행 결과>
가을이 오면 눈부신 아침 햇살에 비친
그대의 미소가 아름다워요
눈을 감으면 싱그런 바람 가득한
그대의 맑은 숨결이 향기로와요
```

위의 예제에서 보듯이 readline() 함수를 실행하면 한 줄 전체가 inStr 변수에 문자열로 대입된다. 그래서 inStr에 저장된 문자열을 그대로 print() 함수를 이용하여 화면에 출력하면 텍스트 파일의 한 줄이 출력된다. 전체가 네 줄이기 때문에 이를 네 번 실행하면 전체를 출력할 수 있다. 그런데 readline() 함수를 실행하면 텍스트 파일의 마지막 줄바꿈 문자까지 변수 inStr에 포함이 된다. 그리고 print() 함수는 매개변수로 전달한 문자열을 모두 출력한 다음 자동으로 줄바꿈을 하기 때문에 'print(inStr)'을 실행하면 행과 행 사이에 빈 줄이 출력되게 된다. 이를 방지하기 위하여 'print(inStr, end="")'과 같이 해준다.

그런데 위의 코드는 사용자가 읽어 들이는 텍스트 파일이 몇 줄로 구성되어 있는지를 알고 있을 때 사용할 수 있는 방법이다. 만약 텍스트 파일의 줄 수를 알 수 없는 상황이 일반적인데, 이를 위해서 다음과 같은 방법을 이용해야 한다.

```
inStr = ""    #읽어온 문자열을 저장할 변수
inFp = open("C:/Temp/text.txt", "r")

while True:
    inStr = inFp.readline()
    if inStr == "":
        break
    print(inStr, end="")

inFp.close()
```

다음은 readlines() 함수를 이용해 모두 한 번에 읽어 들이는 방법이다.

```
inList = ""    #읽어온 문자열을 저장할 변수
inFp = open("C:/Temp/text.txt", "r")
```

```
inList = inFp.readlines()
print(inList)

inFp.close()
```

```
<실행 결과>
['가을이 오면 눈부신 아침 햇살에 비친\n', '그대의 미소가 아름다워요\n', '눈을 감으면 싱그
런 바람 가득한\n', '그대의 맑은 숨결이 향기로와요\n']
```

위의 결과를 보면 텍스트 파일 전체가 _____ 형태로 저장이 되었고 각 항목은 문자열 형태임을 확인할 수 있다. 즉 readlines()의 반환 자료형은 문자열의 리스트이다. 그래서 이를 원래 파일 그대로 화면에 출력하려면 리스트의 각 항목을 출력하는 방식으로 하면 된다.

```
inList = ""    #읽어온 문자열을 저장할 변수
inFp = open("C:/Temp/text.txt", "r")

inList = inFp.readlines()
for inStr in inList:
    print(inStr, end="")

inFp.close()
```

```
<실행 결과>
가을이 오면 눈부신 아침 햇살에 비친
그대의 미소가 아름다워요
눈을 감으면 싱그런 바람 가득한
그대의 맑은 숨결이 향기로와요
```

8.3 텍스트 파일의 출력

텍스트 파일을 출력하는 방법은 콘솔에서 _____ 함수를 사용하는 것과 매우 유사하다. 다음은 콘솔에 한 줄씩 입력한 내용을 텍스트 파일에 그대로 쓰는 예제이다.

```
outList = ""    #키보드로 입력한 문자열을 저장할 변수
```

```
outFp = open("C:/Temp/text2.txt", "w")

while True:
    outStr = input("입력할 내용:")
    if outStr != "":
        outFp.writelines(outStr+"\n")
    else:
        break

outFp.close()
print("파일 쓰기 완료")
```

```
<실행 결과>
입력할 내용:안녕하세요.                    ← 사용자입력
입력할 내용:파이썬을                        ← 사용자입력
입력할 내용:열심히 공부하고 있습니다.       ← 사용자입력
입력할 내용:                                ← 사용자입력
파일 쓰기 완료
```

다음은 텍스트 파일에 쓰여진 결과이다.