UNISTUDY

파이썬

28강. 텍스트 파일

1. 텍스트 파일

- 텍스트 파일을 대상으로 텍스트 자료를 읽어오는 방법과 읽어온 자료를 처리하고, 처리 결과를 파일에 저장하는 방법에 대해서 알아본다.

2. 텍스트 파일 입출력

- 텍스트 파일을 읽고 쓰기 위해서는 io 모듈에서 제공하는 open()함수를 이용한다. open()함수의 형식은 다음과 같다. file, mode, encoding 등의 주요 파라미터를 사용하여 텍스트 파일을 읽어올 수 있고, 파일에 자료를 기록할 수 있다.

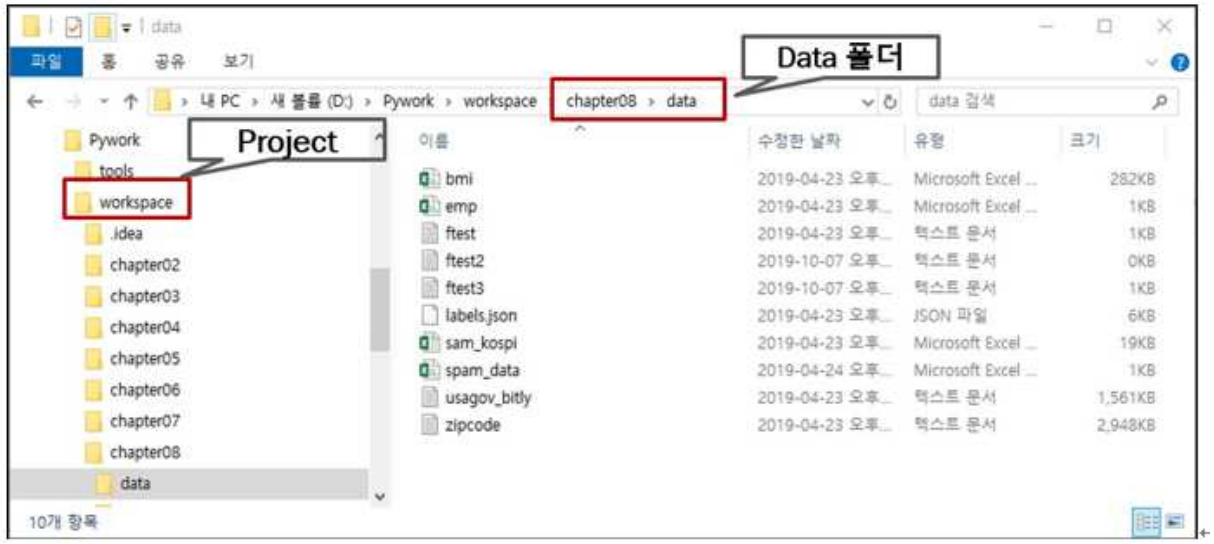
```
open(file, mode, encoding)
```

2. 텍스트 파일 입출력

파라미터	의미
file	파일의 경로와 파일명을 지정한다.
mode	읽기 모드, 쓰기 모드, 쓰기+추가 모드 등을 정해진 문자(character)로 지정한다. 문자의 의미는 다음과 같다. 'r' : 읽기용 파일 객체를 연다.(기본값) 'w' : 쓰기용 파일 객체를 연다. 'x' : 쓰기용 파일을 새로 만들어서 연다.(기존 파일이 있으면 error) 'a' : 기존 파일의 맨 마지막에 추가용 파일 객체를 연다. 'b' : 이진파일(binary file)형식으로 읽기/쓰기 파일 객체를 연다.
encoding	인코딩 또는 디코딩에 사용되는 인코딩의 이름을 지정하는 속성으로 텍스트 모드에서만 사용한다. 기본 인코딩은 플랫폼에 따라 다르지만 파이썬에서 지원하는 인코딩 목록은 codec 모듈을 참조한다.

2. 텍스트 파일 입출력

- 다음은 텍스트 파일을 읽고, 쓰기 위해서 open()함수를 이용하여 파일 객체를 생성하는 예문이다.
- 예문에서 사용되는 실습환경으로 파이참(PyCharm)의 프로젝트 폴더와 data 폴더의 경로는 다음과 같다.



2. 텍스트 파일 입출력

chapter08.lecture.step02_text_file.py

```
# (1) 현재 작업디렉터리
import os
print('\n현재 경로 :', os.getcwd()) # D:\Pywo
rk\workspace


# (2) 예외처리
try :
    # (3) 파일 읽기
    ftest1 = open('chapter08/data/ftest.tx
t', mode = 'r')
    print(ftest1.read()) # 파일 전체 읽기

    # (4) 파일 쓰기
    ftest2 = open('chapter08/data/ftest2.t
xt', mode = 'w')
    ftest2.write('my first text~~~') # 파일

쓰기

    # (5) 파일 쓰기 + 내용 추가
    ftest3 = open('chapter08/data/ftest2.t
xt', mode = 'a')
    ftest3.write('\nmy second text ~~~') #
파일 쓰기(추가)
```

Python Console

```

현재 경로:
 D:\Pywork\workspace


programming is fun
very fun!
have a good time
mouse is input device
keyboard is input device
computer is input output sys
tem
```

2. 텍스트 파일 입출력

```
except Exception as e:
    print('Error 발생 : ', e)

finally:
    ftest1.close() # 파일 객체 닫기
    ftest2.close()
    ftest3.close()
```

2. 텍스트 파일 입출력

- 텍스트 파일 입출력 예
- # (1) 현재 작업디렉터리
- os 모듈에서 제공하는 getcwd()함수를 이용하여 현재 기본 작업 디렉터리를 확인할 수 있다. 파이참 은 기본적으로 프로젝트로 지정된 폴더를 기본 작업 디렉터리로 인식한다. 예문에서 사용된 파이참의 프로젝트 폴더의 경로는 확인할 수 있다.
- # (2) 예외처리
- 파일 입출력을 위해서 try ~ except ~ finally 블록을 이용한다. try 블록에서는 파일 입출력에 관련된 코드를 작성하고 except 블록에서는 파일 입출력 과정에서 발생되는 예외를 처리하는 코드를 작성한다. 그리고 finally 블록은 파일 입출력을 위해서 생성된 객체를 닫는 코드를 작성한다.

2. 텍스트 파일 입출력

- # (3) 파일 읽기
- open()함수의 첫 번째 파라미터는 data 폴더의 ftest.txt 파일에 대한 경로와 파일명을 지정한다. 이때 기본 작업 디렉터리(D:₩Pywork₩workspace) 이후부터 폴더명과 파일명을 순서대로 지정한다. 폴더와 폴더, 폴더와 파일의 구분자는 '/' 또는 '₩₩' 기호를 이용한다. 예문에서 사용된 data 폴더의 경로는 확인할 수 있다. 두 번째 파라미터는 파일 읽기 모드인 'r'을 지정한다. mode='r'은 기본값이기 때문에 생략이 가능하다.
- # (4) 파일 쓰기
- open()함수의 첫 번째 파라미터는 텍스트 파일이 생성될 위치와 파일명을 지정한다. 예문에서는 data 폴더에 ftest2.txt 파일 이름으로 생성된다. 두 번째 파라미터는 파일 쓰기 모드인 'w'를 지정한다. mode='w'는 가장 최근의 값으로 덮어쓴다.

2. 텍스트 파일 입출력

- # (5) 파일 쓰기 + 내용 추가
- open()함수의 첫 번째 파라미터는 기존 텍스트 파일의 위치와 파일명을 지정한다. 두 번째 파라미터는 파일 쓰기와 내용을 추가할 수 있는 모드인 'a'를 지정한다. mode='a'는 기존 파일의 내용 마지막 부분에 새로운 내용을 추가한다.

UNISTUDY

3. 텍스트 자료 읽기

- 텍스트 파일의 자료를 읽어오는 방법에 대해서 알아본다. 읽기 모드(mode='r')로 객체를 생성할 경 우에는 다음 <표> 8-2와 같은 텍스트 자료 읽기 관련 함수를 제공한다.

파라미터	의미
read()	전체 텍스트 자료를 한 번에 읽어온다. 읽어온 자료는 문자열(str) 자료형으로 반환된다.
readlines()	전체 텍스트 자료를 줄 단위로 읽어온다. 읽어온 자료는 리스트(list) 자료형으로 반환된다.
readline()	한 줄 단위로 읽어온다. 읽어온 자료는 문자열(str) 자료형으로 반환된다.

UNISTUDY

3. 텍스트 자료 읽기

- 텍스트 파일의 자료를 읽어오는 방법에 대해서 알아본다. 읽기 모드(mode='r')로 객체를 생성할 경 우에는 다음 <표> 8-2와 같은 텍스트 자료 읽기 관련 함수를 제공한다.

파라미터	의미
read()	전체 텍스트 자료를 한 번에 읽어온다. 읽어온 자료는 문자열(str) 자료형으로 반환된다.
readlines()	전체 텍스트 자료를 줄 단위로 읽어온다. 읽어온 자료는 리스트(list) 자료형으로 반환된다.
readline()	한 줄 단위로 읽어온다. 읽어온 자료는 문자열(str) 자료형으로 반환된다.

3. 텍스트 자료 읽기

- 텍스트 파일을 하나의 문단으로 해석하고, 텍스트 파일에서 줄 단위로 읽어온 자료는 문장 으로 해석하여 자연어를 처리하는 방식에 적용한 예문이다.

3. 텍스트 자료 읽기

chapter08.lecture.step02_text_file.py

```
# 파일 읽기 관련 함수
try :

    # (1) read() : 전체 텍스트 자료 읽기
    ftest = open('chapter03/data/ftest.txt
', mode='r')
    full_text = ftest.read()
    print(full_text)
    print(type(full_text))


    # (2) readlines() : 전체 텍스트 줄 단위 읽기
    ftest = open('chapter03/data/ftest.txt
', mode='r')
    lines = ftest.readlines() #  list 반환
    print(lines)
    print(type(lines)) # <class 'list'>
    print('문단 수 :', len(lines)) # 문단 수 :
    6


    # (3) list -> 문장 추출
```

Python Console

```
programming is fun
very fun!
have a good time
mouse is input device
keyboard is input device
computer is input output sys
tem
<class 'str'>

['programming is fun\n', 've
ry fun!\n', 'have a good tim
e\n', 'mouse is input device
\n', 'keyboard is input devi
```

3. 텍스트 자료 읽기

```
        print(line.strip())  # text만 출력
        docs.append(line.strip())

    print(docs)  # 문장 출력

    # (4) readline() : 한 줄 읽기
    ftest = open('chapter08/data/ftest.txt
', mode='r')
    line = ftest.readline() #  한 줄 읽기
    print(line)
    print(type(line))  # <class 'list'>

except Exception as e:
    print('Error 발생 : ', e)

finally:
    # 파일 객체 닫기
    ftest.close()
```

```
tput system']
<class 'list'>
문단 수 : 6

programming is fun
very fun!
have a good time
mouse is input device
keyboard is input device
computer is input output sys
tem
programming is fun
<class 'str'>
```

3. 텍스트 자료 읽기

- 텍스트 자료 처리 예
- # (1) read() : 전체 텍스트 자료 읽기
- read()함수는 ftest.txt 파일의 전체 내용을 문자열로 읽어온다.
- # (2) readlines() : 전체 텍스트 줄 단위 읽기
- readlines()함수는 전체 텍스트를 줄 단위로 읽어서 리스트 자료형으로 반환한다. 따라서 리스트 원소 한 개가 텍스트 파일의 한 줄이다.
- # (3) list -> 문장 추출
- 텍스트 파일에서 한 줄의 끝 부분에는 줄바꿈('₩n')의 이스케이프 문자가 포함된다. 이러한 이스케이 프 문자를 불용어로 보고, 이를 제거하기 위해서는 문자열 객체에서 지원하는 strip()함수를 이용한 다. 다음은 strip()함수를 이용하여 x변수의 문자열 끝 부분에 오는 '₩n', '공백', '₩t', '₩r' 등의 이스케이프 문자를 제거하는 예문이다.
- x = "sgat3sgaga₩n ₩t₩r" print(x.strip()) # sgat3sgaga

3. 텍스트 자료 읽기

- # (4) readline() : 한 줄 읽기
- readline()함수는 텍스트 파일에서 순서대로 한 줄 단위로 읽어서 문자열(str)을 반환한다. 만약 6줄 전체를 읽기 위해서는 다음과 같은 for문을 이용하여 6번 반복하여 readline()함수를 호출하면 된다.
- for i in range(6) :
- line = ftest.readline() print(line.strip())

- ※ 텍스트 파일을 3가지 방법을 읽어올 때 마다 open()함수를 이용하여 매번 같은 파일을 읽어오는 이유는 한 번 파일을 읽을 때 마다 커서위치가 변경되기 때문이다. 따라서 맨 첫 줄부터 다시 읽기 위해서는 텍스트 파일을 새롭게 읽어야 한다.

4. with 블록과 인코딩방식

- with 명령어를 이용하여 블록 형식으로 파일 객체를 생성하는 방법과 한글과 같은 유니코드를 사용할 경우 인코딩 방식을 지정하여 텍스트 파일을 작성하는 방법을 예문으로 알아본다. 다음은 with 블록 을 이용하여 파일 객체를 생성하는 형식이다.

```
with open(file, mode, encoding) as 참조변수 :
    pass
```

4. with 블록과 인코딩방식

- with 블록을 이용하면 블록 내에서 참조변수를 이용하여 파일 객체를 사용할 수 있고, 블록을 벗어나 면 자동으로 객체가 close된다. 아래 예문에서 첫 번째 with 블록은 data 폴더에 ftest3.txt 파일을 쓰기 모드로 파일 객체를 생성하고 참조변수인 ftest에 주소를 할당한다. 이때 파일에 한글을 쓰기 위해서 인코딩 방식(utf-8)을 파라미터로 지정하였다.
  두 번째 with 블록은 동일한 파일을 대상으로 읽기 모드와 인코딩 파라미터를 이용하여 파일 객체를 생성하였다. read()함수를 호출하면 파일에 저장된 두 줄의 한글 텍스트가 출력된다. with 블록을 벗어나면 자동으로 객체가 소멸되기 때문에 finally 블록의 내용을 생략한다

4. with 블록과 인코딩방식

```
try :
  with open('chapter08/data/ftest3.txt', mode='w', encoding='utf-8') as  ft
est :
        ftest.write('파이썬 파일 작성 연습')
        ftest.write('\n파이썬 파일 작성
        연습2')# with 블록 벗어나면 자동 close
    with open('chapter08/data/ftest3.txt', mode='r', encoding='utf-8') as
ftest :
        print(ftest.read())
except Exception as e:
    print('Error 발생 : ', e)
finally:
    pass
```

UNISTUDY

THANK YOU