Chapter 7

How to work with file I/O

Objectives

Applied

1. Use text, CSV, or binary files to save and retrieve the data that's used by your programs.

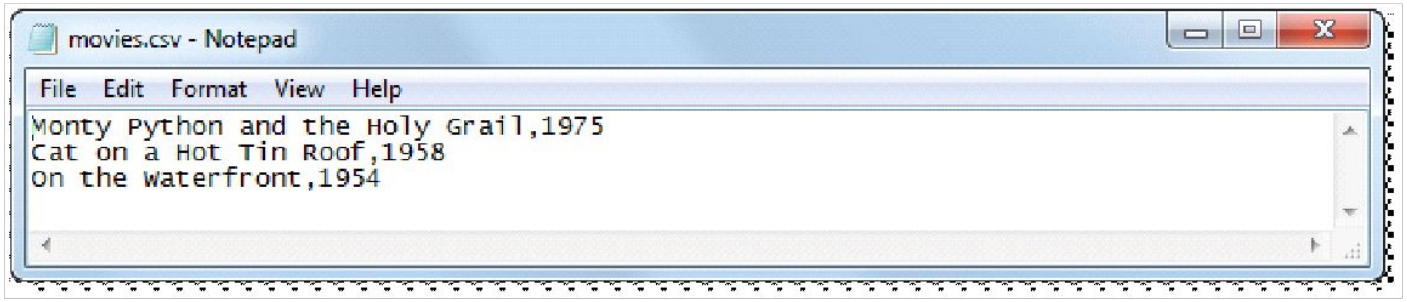
Knowledge

1. Differentiate between text and binary files.
2. Describe the benefit of using a with statement for opening and closing a file.
3. Describe the use of the csv module, writer objects, and reader objects for writing a list of lists to a CSV file and reading a list of lists from a CSV file.
4. Describe the use of the pickle module and the load() and dump() methods for saving a list of lists to a binary file and reading a list of lists from a binary file.

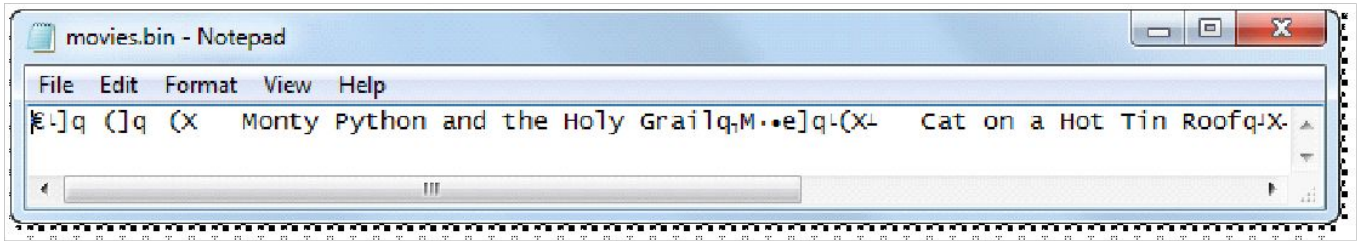
Two types of files

- Text
- Binary

A text file that's opened by a text editor



A binary file with the same data



The sequence of file operations

1. *Open* the file.
2. *Write* data to the file or *read* data from the file.
3. *Close* the file.

The built-in open() function

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

A few of its modes

Character	Mode
`r`	Read
`w`	Write
`a`	Append
`b`	Binary

The close() method of a file object

```
close()
```

How to open a file in write mode and close the file manually

```
outfile = open("test.txt", "w")
outfile.write("Test")
outfile.close()
```

How to use with statements to open and close files

The syntax of the with statement for file I/O

```
with open(file, mode) as file_object:
    statements...
```

Code that opens a text file in write mode and automatically closes it

```
with open("test.txt", "w") as outfile:
    outfile.write("Test")
```

Code that opens a text file in read mode and automatically closes it

```
with open("test.txt", "r") as infile:
    print(infile.readline())
```

The write() method of a file object

```
write(str)
```

How to write one line to a text file

```
with open("members.txt", "w") as file:
    file.write("John Cleese\n")
```

How to append one line to a text file

```
with open("members.txt", "a") as file:
    file.write("Eric Idle\n")
```

The contents of the text file after the two lines have been written

```
John Cleese\nEric Idle\n
```

The contents of the text file when viewed in a text editor

```
John Cleese
Eric Idle

```

Three read methods of a file object

```
read()
readlines()
readline()
```

How to use a loop to read each line of the file

```
with open("members.txt") as file:
    for line in file:
        print(line, end="")
    print()
```

How to read the entire file as a string

```
with open("members.txt") as file:
    contents = file.read()
    print(contents)
```

The result that's printed to the console

```
John Cleese
Eric Idle

```

How to read the entire file as a list

```
with open("members.txt") as file:
    members = file.readlines();
    print(members[0], end="")
    print(members[1])
```

How to read each line of the file

```
with open("members.txt") as file:
    member1 = file.readline();
    print(member1, end="")
    member2 = file.readline();
    print(member2)
```

The result that's printed to the console

```
John Cleese
Eric Idle

```

How to write and read a list of strings

How to write the items in a list to a file

```
members = ["John Cleese", "Eric Idle"]
with open("members.txt", "w") as file:
    for m in members:
        file.write(m + "\n")            # adds new line
```

How to read the lines in a file into a list

```
members = []
with open("members.txt") as file:
    for line in file:
        line = line.replace("\n", "") # removes new line
        members.append(line)
print(members)
```

The result that's printed to the console

```
['John Cleese', 'Eric Idle']
```

How to write and read a list of numbers

How to write the items in a list to a file

```
years = [1975, 1979, 1983]
with open("years.txt", "w") as years_file:
    for year in years:
        years_file.write(str(year) + "\n")  # converts
int to str
```

How to read the items in a list from a file

```
years = []
with open("years.txt") as file:
    for line in file:
        line = line.replace("\n", "")
        years.append(int(line))             # converts
str to int
print(years)
```

The result that's printed to the console

```
[1975, 1979, 1983]
```

The user interface for the Movie List 1.0 program

```
The Movie List program

COMMAND MENU
list - List all movies
add -  Add a movie
del -  Delete a movie
exit - Exit program

Command: list
1. Monty Python and the Holy Grail
2. Cat on a Hot Tin Roof
3. On the Waterfront

Command: add
Movie: Casablanca
Casablanca was added.

Command: list
1. Monty Python and the Holy Grail
2. Cat on a Hot Tin Roof
3. On the Waterfront
4. Casablanca

Command: del
Number: 4
Casablanca was deleted.
```

The code

```
FILENAME = "movies.txt"

def write_movies(movies):
    with open(FILENAME, "w") as file:
        for movie in movies:
            file.write(movie + "\n")

def read_movies():
    movies = []
    with open(FILENAME) as file:
        for line in file:
            line = line.replace("\n", "")
            movies.append(line)
    return movies

def list_movies(movies):
    for i in range(len(movies)):
        movie = movies[i]
        print(str(i+1) + ". " + movie)
    print()
```

The code (cont.)

```
def add_movie(movies):
    movie = input("Movie: ")
    movies.append(movie)
    write_movies(movies)
    print(movie + " was added.\n")

def delete_movie(movies):
    index = int(input("Number: "))
    movie = movies.pop(index - 1)
    write_movies(movies)
    print(movie + " was deleted.\n")

def display_menu():
    print("The Movie List program")
    print()
    print("COMMAND MENU")
    print("list - List all movies")
    print("add -  Add a movie")
    print("del -  Delete a movie")
    print("exit - Exit program")
    print()
```

The code (cont.)

```
def main():
    display_menu()
    movies = read_movies()
    while True:
        command = input("Command: ")
        if command == "list":
            list_movies(movies)
        elif command == "add":
            add_movie(movies)
        elif command == "del":
            delete_movie(movies)
        elif command == "exit":
            print("Bye!")
            break
        else:
            print("Not a valid command. Please try again.")

if __name__ == "__main__":
    main()
```

CSV files

The writer() function of the CSV module

`writer(file)` Returns a CSV writer object for the file. This writer object converts the data into comma separated values.

The writerows() method of the CSV writer object

`writerows(rows)`

Writes all specified rows to the file specified by the writer object using the CSV format specified by the writer object.

A 2-dimensional list with 3 rows and 2 columns

```
movies = [["Monty Python and the Holy Grail", 1975],
          ["Cat on a Hot Tin Roof", 1958],
          ["On the Waterfront", 1954]]
```

How to import the csv module

```
import csv
```

How to write the list to a CSV file

```
with open("movies.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerows(movies)
```

The contents of the CSV file

```
Monty Python and the Holy Grail,1975
Cat on a Hot Tin Roof,1958
On the Waterfront,1954
```

The reader() function of the csv module

reader(*file*)

```
Returns a CSV reader object for the file.
This reader object gets the data from the
CSV file.
```

The reader() function of the csv module

```
reader(file)
```

How to read data from a CSV file

```
with open("movies.csv", newline="") as file:
reader = csv.reader(file)
for row in reader:
print(row[0] + " (" + str(row[1]) + ")")
```

The console

```
Monty Python and the Holy Grail (1975)
Cat on a Hot Tin Roof (1958)
On the Waterfront (1954)
```

Some optional arguments that can be used to change the CSV format

```
quoting=csv.QUOTE_MINIMAL
quotechar='"'
delimiter=","
```

Code that changes the delimiter for the writer object

```
writer = csv.writer(file, delimiter="\t")
```

Code that changes the delimiter for the reader object

```
reader = csv.reader(file, delimiter="\t")
```

The user interface for the Movie List 2.0 program

```
The Movie List program

COMMAND MENU
list - List all movies
add -  Add a movie
del -  Delete a movie
exit - Exit program

Command: list
1. Monty Python and the Holy Grail (1975)
2. Cat on a Hot Tin Roof (1958)
3. On the Waterfront (1954)

Command: add
Name: Gone with the Wind
Year: 1939
Gone with the Wind was added.

Command: list
1. Monty Python and the Holy Grail (1975)
2. Cat on a Hot Tin Roof (1958)
3. On the Waterfront (1954)
4. Gone with the Wind (1939)

Command: del
Number: 4
Gone with the Wind was deleted.
```

The code

```
import csv

# a file in the current directory
FILENAME = "movies.csv"

def write_movies(movies):
    with open(FILENAME, "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerows(movies)

def read_movies():
    movies = []
    with open(FILENAME, newline="") as file:
        reader = csv.reader(file)
        for row in reader:
            movies.append(row)
    return movies

def list_movies(movies):
    for i in range(len(movies)):
        movie = movies[i]
        print(str(i+1) + ". " + movie[0] + " (" + movie[1] + ")")
    print()
```

The code (cont.)

```
def add_movie(movies):
    name = input("Name: ")
    year = input("Year: ")
    movie = []
    movie.append(name)
    movie.append(year)
    movies.append(movie)
    write_movies(movies)
    print(name + " was added.\n")

def delete_movie(movies):
    index = int(input("Number: "))
    movie = movies.pop(index - 1)
    write_movies(movies)
    print(movie[0] + " was deleted.\n")

def display_menu():
    print("The Movie List program")
    print()
    print("COMMAND MENU")
    print("list - List all movies")
    print("add -  Add a movie")
    print("del -  Delete a movie")
    print("exit - Exit program\n")
```

The code (cont.)

```
def main():
    display_menu()
    movies = read_movies()
    while True:
        command = input("Command: ")
        if command.lower() == "list":
            list_movies(movies)
        elif command.lower() == "add":
            add_movie(movies)
        elif command.lower() == "del":
            delete_movie(movies)
        elif command.lower() == "exit":
            break
        else:
            print("Not a valid command. Please try again.\n")
    print("Bye!")

if __name__ == "__main__":
    main()
```

Two methods of the pickle module

```
dump(object, bfile)
load(bfile)
```

A 2-dimensional list with 3 rows and 2 columns

```
movies = [["Monty Python and the Holy Grail", 1975],
          ["Cat on a Hot Tin Roof", 1958],
          ["On the Waterfront", 1954]]
```

How to import the pickle module

```
import pickle
```

How to write an object to a binary file

```
with open("movies.bin", "wb") as file:  # write binary
    pickle.dump(movies, file)
```

How to read an object from a binary file

```
with open("movies.bin", "rb") as file: # read binary
    movie_list = pickle.load(file)
    print(movie_list)
```

The console

```
[['Monty Python and the Holy Grail', 1975], ['Cat
on a Hot Tin Roof', 1958], ['On the Waterfront',
1954]]
```

The user interface for the Movie List 3.0 program

```
The Movie List program

COMMAND MENU
list - List all movies
add -  Add a movie
del -  Delete a movie
exit - Exit program

Command: list
1. Monty Python and the Holy Grail (1975)
2. Cat on a Hot Tin Roof (1958)
3. On the Waterfront (1954)

Command: add
Name: Gone with the Wind
Year: 1939
Gone with the Wind was added.
```

The code for the two file I/O functions

```
import pickle

FILENAME = "movies.bin"

def write_movies(movies):
    with open(FILENAME, "wb") as file:
        pickle.dump(movies, file)

def read_movies():
    movies = []
    with open(FILENAME, "rb") as file:
        movies = pickle.load(file)
    return movies
```

The code for the list_movies() function

```
def list_movies(movies):
    for i in range(len(movies)):
        movie = movies[i]
        print(str(i+1) + ". " + movie[0] + " ("
            + str(movie[1]) + ")")
    print()
```