

Text Files: Saving a Playlist and Reading It Back

Why?

Every variable in a program disappears when the program ends, so any data worth keeping has to be saved somewhere. In this activity, you will save a playlist to a text file and read the playlist back, which is how many real programs store and load their data.

Prerequisites

- Write for loops over lists, including lists of tuples.
- Convert strings to numbers with `int()` or `float()`.
- Format output with f-strings and escape sequences.

Learning Objectives

Content:

- Predict the exact contents of a text file produced by a sequence of `write()` calls, including where `\n` and `\t` place line and field breaks.
- Read a text file line by line inside a `with` statement, and explain why each line read from the file ends in a newline character.
- Use `split()` to break a line into fields, and convert fields to numbers to compute a result from the file.

Process:

- Compare a program's code with the file the program produced to infer what each part of the code contributes (Information Processing).
- Predict a result, check the prediction by running code, and revise the team's explanation when the prediction was wrong (Critical Thinking).

Model 1: Saving a Playlist

Each program below creates a new file named `playlist.txt` and writes text into the file. The programs are separate: each one runs by itself and creates `playlist.txt` from scratch.

Program A

```
file = open("playlist.txt", "w")
file.write("Hey Jude")
file.write("Respect")
file.write("Dreams")
file.close()
```

Contents of `playlist.txt`:

```
Hey JudeRespectDreams
```

Program B

```
file = open("playlist.txt", "w")
file.write("Hey Jude\n")
file.write("Respect\n")
file.write("Dreams\n")
file.close()
```

Contents of `playlist.txt`:

```
Hey Jude
Respect
Dreams
```

Program C

```
file = open("playlist.txt", "w")
file.write("Hey Jude\tThe Beatles\t7\n")
file.write("Respect\tAretha Franklin\t2\n")
file.write("Dreams\tFleetwood Mac\t4\n")
file.close()
```

Contents of `playlist.txt`:

```
Hey Jude      The Beatles      7
Respect Aretha Franklin 2
Dreams Fleetwood Mac   4
```

Program D

```
with open("playlist.txt", "w") as file:
    file.write("Hey Jude\tThe Beatles\t7\n")
    file.write("Respect\tAretha Franklin\t2\n")
    file.write("Dreams\tFleetwood Mac\t4\n")
print("Done")
```

Contents of `playlist.txt`: exactly the same as Program C.

Note: until a file is closed, some of the text written to the file may still be waiting in memory rather than saved in the file.

1. The `print()` function always moves to a new line after printing its output. Compare the files from Programs A and B. Does `write()` behave the same way? Explain using evidence from the two files.
2. In the Thonny Shell, predict the value of `len("Hey\tJude\n")`, then type the expression to check. Record both your prediction and the actual result.
3. How many characters does each escape sequence, such as `\t` or `\n`, add to a string? Explain how your result from question 2 shows this.
4. Program D never calls `close()`, yet the file from Program D is the same as the file from Program C. At what point while Program D runs is the file closed, and how can you tell from the code?
5. Why might a programmer prefer to write Program D rather than Program C? Use the note at the end of Model 1 in your answer.
6. The list below holds a longer playlist. Write a program that uses a `with` statement, a loop, and an f-string to save songs to `playlist.txt` in the same format as Program C. Run the program, then open `playlist.txt` in Thonny to check the file contents. Keep this file; you will read it in Model 2.

```
songs = [  
    ("Hey Jude", "The Beatles", 7),  
    ("Respect", "Aretha Franklin", 2),  
    ("Dreams", "Fleetwood Mac", 4),  
    ("Bohemian Rhapsody", "Queen", 6),  
]
```

Model 2: Loading the Playlist

Each program below opens the `playlist.txt` file from question 6 and reads the file one line at a time. The programs are separate: each one runs by itself, and none of them changes the file.

Program E

```
with open("playlist.txt") as file:
    for line in file:
        print(line)
```

Output:

```
Hey Jude      The Beatles      7
Respect Aretha Franklin 2
Dreams Fleetwood Mac 4
Bohemian Rhapsody Queen 6
```

Program F is the same as Program E, except the last line is `print(repr(line))`. The `repr()` function shows a string the way the string would be typed in Python code, so escape sequences are visible.

Output:

```
'Hey Jude\\tThe Beatles\\t7\\n'
'Respect\\tAretha Franklin\\t2\\n'
'Dreams\\tFleetwood Mac\\t4\\n'
'Bohemian Rhapsody\\tQueen\\t6\\n'
```

Program G

```
with open("playlist.txt") as file:
    for line in file:
        fields = line.split("\\t")
        print(fields)
```

Output:

```
['Hey Jude', 'The Beatles', '7\\n']
['Respect', 'Aretha Franklin', '2\\n']
['Dreams', 'Fleetwood Mac', '4\\n']
['Bohemian Rhapsody', 'Queen', '6\\n']
```

Program H

```
total = 0
with open("playlist.txt") as file:
    for line in file:
        fields = line.split("\\t")
        total += int(fields[2])
print(total)
```

Output:

```
19
```

7. The programs in Model 2 call `open()` with only a file name, not `"w"`. Based on Model 1, what would happen to the playlist if Program E used `open("playlist.txt", "w")` instead?
8. Compare the output of Program E with the contents of `playlist.txt`. What is different about how the songs appear?
9. Use the output of Program F to explain where the blank lines in Program E's output come from.
10. In Program G, describe the value that `line.split("\t")` produces for the first line of the file.
11. In Program G, the last string in each list ends with `\n`, but the other strings do not. Using your answer to question 9, explain why.
12. Write a program that reads `playlist.txt` and prints the title of each song that is longer than 5 minutes.

13. *Optional, if your team has time.* Predict what Program H would do if line 5 were `total += fields[2]`, without `int()`. Then make the change and run Program H to check.
14. *Optional, if your team has time.* Change Program E so that the songs print without blank lines between them.