



F7.1 Reading and writing files

Files and databases · GCSE · OCR J277 2.2.3, Edexcel 1CP2 6.4.2 · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Open, read, write, append and close: a run log in a CSV file.

- The files on this page
- Reading a file
- Line by line
- Writing a file
- The whole file as a list of records
- When files go wrong

Questions

6 marks in all

1. Put the steps for reading a file in order. [1 mark]

Number the lines 1 to 3 to put them in the right order.

___ Open the file

___ Read from the file

___ Close the file

2. What does opening a file with mode "w" do to a file that already exists? [1 mark]

- ☐ A Empties it before writing
- ☐ B Adds to the end of it
- ☐ C Reads it
- ☐ D Nothing until close is called

3. Which mode adds new lines to the end of a file and keeps what was there? [1 mark]

- ☐ A "a"
- ☐ B "w"
- ☐ C "r"
- ☐ D "x"

4. What does this program print? [1 mark]

```
line = "3,square,80,9.4\n"
fields = line.strip().split(",")
print(fields[1], float(fields[2]) + 1)
```

5. Why use `with open(...)` as `f`?

[1 mark]

- ☐ A The file is closed automatically at the end of the block
- ☐ B It reads the file faster
- ☐ C It makes the file bigger
- ☐ D It is the only way to write

6. Why store data in a file rather than only in variables?

[1 mark]

- ☐ A Variables are lost when the program stops; a file keeps the data
- ☐ B Files are faster than variables
- ☐ C Variables cannot hold numbers
- ☐ D Files cannot be changed

The task: log the run

Drive forward 25 cm, timing it with `clock()`. Then **append** a line to `runs.csv` for this run: run number 4, task wall, 25 cm, and the seconds rounded to one decimal place, in the same format as the other lines. Finally read the file and print 4 runs logged, counting the lines after the header rather than typing 4. The task starts from the original four-line `runs.csv` every time it runs, whatever earlier runs added to the copy above.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

start = clock()
forward(50, distance=25)
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f7-1-reading-and-writing-files/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Write a program that prints the total distance of all the runs in the file.
2. Log a whole session: drive three different distances, appending a line for each.
3. What happens to `runs.csv` if you use mode `"w"` instead of `"a"`? Try it, then reset the file.

