Objective 10:
Understand serial files

In this objective you learn how to read and write data to files so it can be saved and loaded again later.

Tasks

1. Try entering the following program and see what happens:

```
#Open a text file for writing
text_file = open("data.txt", "w")
text_file.write("This is a simple way to save data\n")
text_file.write("one line at a time")
text_file.close()
print("File created.")
```

You should find a new file called data.txt in the same folder that you saved the program in. Open it and examine the contents.

Note the "w" in line 2. This means write data to the file.
Note the \n in line 3. This is known as an escape sequence and adds a carriage return to start a new line.

2. Try entering the following program and see what happens:

```
#Open a text file for reading
text_file = open("data.txt", "r")
print(text_file.read())
text_file.close
```

3. Change the program so that 3 lines of text are written to a file and read back in again.

It is often useful to read in all the data from a file line by line until you reach the end of the file. To do this we would want to combine an iteration with a file read.

1. Try entering the following program and see what happens:

   ```
   #Open a text file for reading
   text_file = open("data.txt", "r")

   #Read each line into a variable and output
   while True:
       a = text_file.readline()
       if not a: break
       print(a)
   ```

Note how we can use an infinite loop and a break command to read all the lines in the file until no more lines are found.

Each line in the text file has a hidden \n new line escape sequence to indicate the next item of data is on a new line. This has to be removed before selection statements can be used because C is not the same as C\n.

The method `.strip` will remove the escape sequence from the string. E.g.

```
x = text_file.readline()
x = answer.strip()
```

Objective 10: Key learning points

Understand serial files

- Serial files store data with no order to the data maintained.
- To search data from a serial file you begin at the start of the file and read all the data until the item is found.
- Data cannot be deleted from a serial file without creating a new file and copying all the data except the item you want to delete.
- Data cannot be changed in a serial file without creating a new file, copying all the data across to a new file, inserting the change at the appropriate point.
- Data can be appended to a serial file.
- A file can be open for reading or writing, but not reading and writing at the same time.
- Serial files are quite limiting, but are useful for simple data sets and configuration files. Other types of files include: sequential files where order of the data is maintained, index sequential files for large data sets and random files which allow you to access any item without searching through the file from the start.

Objective 10: Key words

open()

Example code: `text_file = open("filename", "w")`

Opens a file specified by the path and filename.

`"r"`	- for reading data
`"w"`	- for writing data, over-writing a file that may already exist
`"a"`	- for adding data to the end of the file without overwriting existing contents

textfile.read()

Example code: `x = textfile.read()`

the contents of the file is copied to x.

textfile.readline()

Example code: `variable = LineInput(1)`

Reads a single line of data from an open file into x.

textfile.write()

Example code: `textfile.write("data to write")`

Writes a single item of data to an open file.

textfile.close

Example code: `textfile.close`

Commits any data in the file buffer to storage and closes the file. If a file is not closed, data will not be written.