

# GCSE Programming

## Remembering Python





# Starter



3 mins

Quickly research online how we can store data from our python programs even when they are not running.

# R:\Computing\ 10\Writing to files

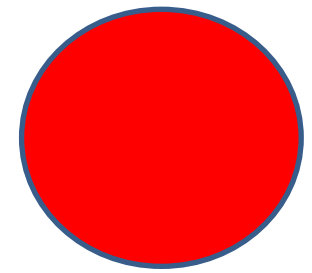
- Copy the folder “Writing to files” to Z:/My Documents/Computing
- Open the Word File
- Watch the tutorials and complete the challenges

# R:\Computing\ 10\Writing to files

## Tutorials

- Part 1: <https://youtu.be/m0o0CkYsDzl>
- Part 2: <https://youtu.be/Olq86NN6-Ko>

# Do Now



- Find the errors:

```
1 name=input("What is your name?")
2 age=input("What is your age?")
3 newfile=open(name+".txt", "w")
4 newFile.write(name, ",", age)
5 newfile.close
6
7 newfile=open(name+".txt", "r")
8 print("Now outputting the contents of the "\
9       + name + ".txt file")
10 print(newfile.read())
```

- Explain what the program does



# Activity 2



30 mins

Complete the tasks set out in the 'reading and writing to text files' activity sheet.

Task 4 correction: Write an extra piece of code on the end of your careers finder program which includes the user's name and their suggested careers into a text file.

# Reading and Writing to Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Text Files and Python

In python, like all other languages, it is possible to store in **text files** data held **within** the program (such as data in variables and lists).

This is great as it means that when our program closes, we can still keep our data.

# Writing to text files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Creating the file and opening it in write mode

Before we can write data into a text file we first need to create the text file.

We need to use a new function, called '**open()**'. This creates the file and allows us to write text to it.

Firstly, we create a variable that python will use to identify the text file, then we use the open function to assign the external file to it.

```
* newfile = open("The New File.txt", "w")
```

We type in the name of the new file, and what type it is. For this guide we will be using note-pad files. Notepad files are designated as .txt.

We also need to tell the computer that we will be writing to the new file. This is why we have the , "w" after the file name and file type. The 'w' means 'write'.



# Writing to text files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Writing to the File

To start writing to the file we have created we need to add `.write` to the end of our new variable followed by brackets.

In this brackets we add our text.

```
• newfile.write("Hello World")
```

If we are going to write more than one line to the file, we need to tell the computer when to start a new line. To do this, we use `\n` at the end of each line.

We don't need to put this in for the last line because there is no other line coming after it.

```
• newfile.write("Hello World\n")  
• newfile.write("Hello World Again\n")  
• newfile.write("No More Hello World")
```

# Writing to text files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Closing the File

We need to close the file after we have written to it otherwise we will not be able to see it once we are finished.

All we have to do to close the file is add '.close' to the end of our variable.

```
newfile.close()
```

*If all we are doing is writing to the file, we do not need an input() at the end of the programme, because we are not displaying anything or inputting any values.*

*The program must be saved and then run externally (Alt+F9) for the file to be written to.*

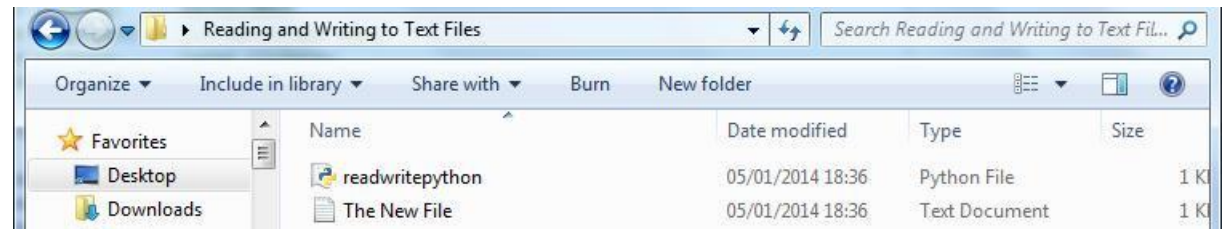
# Writing to text files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## The Result

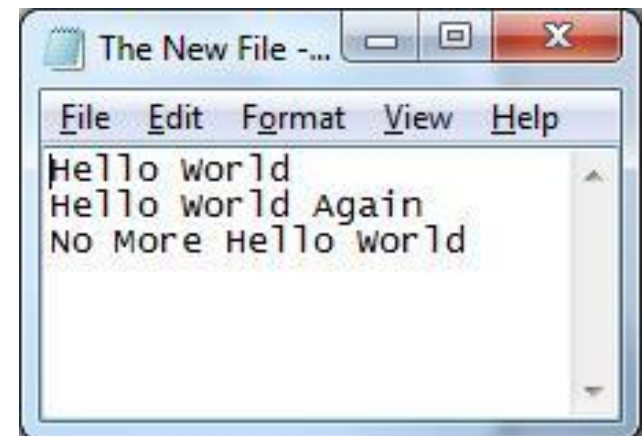
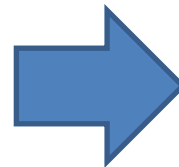
The file that is created and written to will be in the same folder as the program that you made to create the text file.



```

• newfile = open("The New File.txt", "w")
• newfile.write("Hello World\n")
• newfile.write("Hello World Again\n")
• newfile.write("No More Hello World")
• newfile.close()

```



# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Reading a File Opening a file in read mode

Reading a file in python is even easier than writing to a file.

We first need to open the file we are using in read mode. *To be able to read a file you need to have created one beforehand.*

```
• newfile = open("The New File.txt", "r")
```

We do this by typing in the file name and file type, followed by a comma and "r". This tells the computer that we will be opening the file in order to read its contents.

*In this example I am using the same file that I wrote to earlier on.*

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Reading the File

All we have to do to actually see the text within the file is print the text file's variable with `.read()` attached to the end of the variable name.

```
* print(newfile.read())
```

In contrast to when we wrote to a file, we need an `input()` statement at the end of our code because we want to see the file's contents before it closes.

# Reading from Text Files in Python

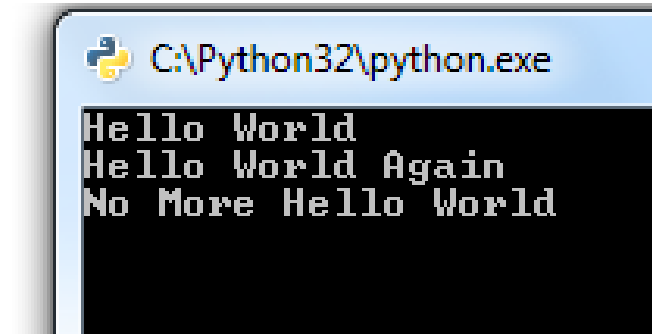
## Learning Objectives:

Reading and Writing data into Text Files

## The Result

Here is the code and the resulting program.

```
* newfile = open("The New File.txt", "r")  
* print(newfile.read())  
* input()
```



A screenshot of a Python command prompt window. The title bar shows the Python logo and the path 'C:\Python32\python.exe'. The window contains the following text:

```
Hello World  
Hello World Again  
No More Hello World
```

# Reading from Text Files in Python

## Learning Objectives:

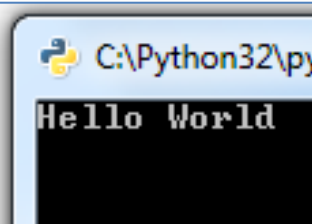
Reading and Writing data into Text Files

## Reading Only One Line at a Time

To read only one line at a time we use `‘.readline()’` instead of `.read()`

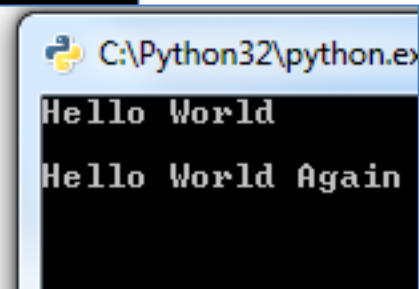
```
• print(newfile.readline())
```

```
• newfile = open("The New File.txt", "r")  
• print(newfile.readline())  
• input()
```



```
C:\Python32\py  
Hello World
```

```
• newfile = open("The New File.txt", "r")  
• print(newfile.readline())  
• print(newfile.readline())  
• input()
```



```
C:\Python32\python.ex  
Hello World  
Hello World Again
```

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Reading Only Specific Lines

To read specific lines we use `.readlines()` followed by the line number in square brackets.

```
* print(newfile.readlines()[2])
```

```
* newfile = open("The New File.txt", "r")  
* print(newfile.readlines()[2])  
* input()
```

C:\Python32\python.exe

No More Hello World





# Activity 2



30 mins

Complete the tasks set out in the 'reading and writing to text files' activity sheet.

Task 4 correction: Write an extra piece of code on the end of your careers finder program which includes the user's name and their suggested careers into a text file.

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Problems from opening a file in Python

Sometimes, if we don't close a file that we have opened in our program, errors can occur.

So, there is another method of opening a file which could be considered superior because with this method we don't have to remember to close the file when we are done with it.

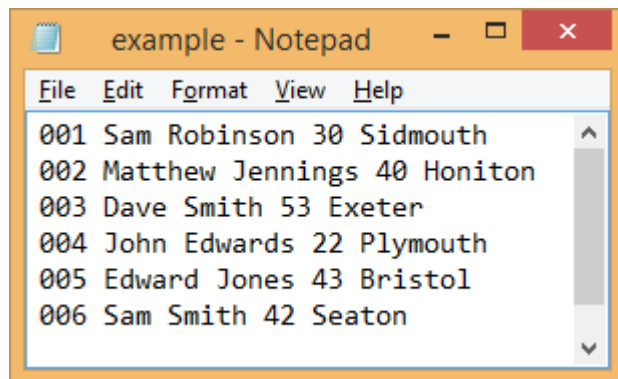
This 'better' method is done using the '**with**' statement.

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

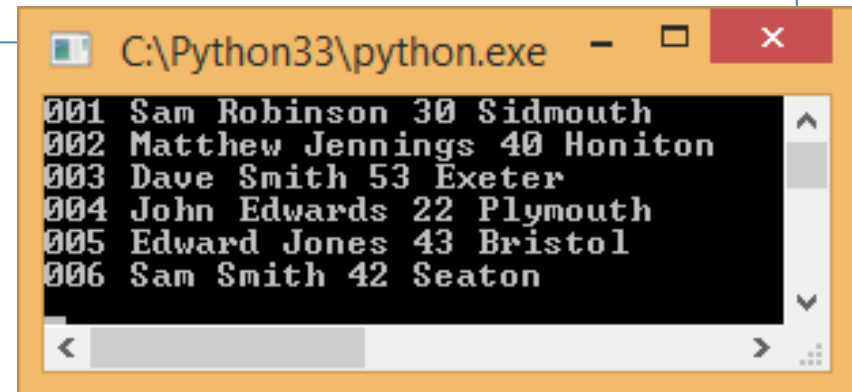
## The WITH statement



```
example - Notepad
File Edit Format View Help
001 Sam Robinson 30 Sidmouth
002 Matthew Jennings 40 Honiton
003 Dave Smith 53 Exeter
004 John Edwards 22 Plymouth
005 Edward Jones 43 Bristol
006 Sam Smith 42 Seaton
```

```
with open('example.txt', "r") as a_file:
    print(a_file.read())

input()
```



```
C:\Python33\python.exe
001 Sam Robinson 30 Sidmouth
002 Matthew Jennings 40 Honiton
003 Dave Smith 53 Exeter
004 John Edwards 22 Plymouth
005 Edward Jones 43 Bristol
006 Sam Smith 42 Seaton
```

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Printing a Particular Line

```
with open('example.txt', "r") as a_file:  
    print(a_file.readlines()[3])
```

```
input()
```



```
004 John Edwards 22 Plymouth
```



# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Printing each Line in Turn

```
with open('example.txt', "r") as a_file:  
    for a_line in a_file:  
        print(a_line)
```

```
input()
```



```
001 Sam Robinson 30 Sidmouth  
002 Matthew Jennings 40 Honiton  
003 Dave Smith 53 Exeter  
004 John Edwards 22 Plymouth  
005 Edward Jones 43 Bristol  
006 Sam Smith 42 Seaton
```



# Reading from Text Files in Python

## Learning Objectives:

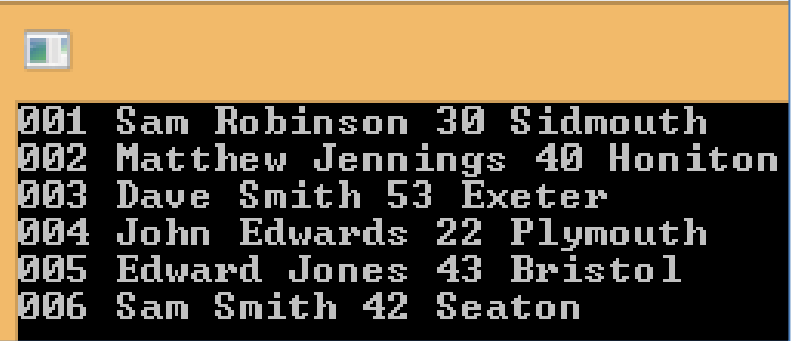
Reading and Writing data into Text Files

## Printing each Line in Turn

*(without the blank line between each item)*

```
with open('example.txt', "r") as a_file:  
    for a_line in a_file:  
        print(a_line, end="")
```

```
input()
```



```
001 Sam Robinson 30 Sidmouth  
002 Matthew Jennings 40 Honiton  
003 Dave Smith 53 Exeter  
004 John Edwards 22 Plymouth  
005 Edward Jones 43 Bristol  
006 Sam Smith 42 Seaton
```



# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Finding and Printing Lines based on 'Set Criteria'

```
line_number = 0

with open('example.txt', "r") as a_file:
    for a_line in a_file:
        line_number += 1
        if "Sam" in a_line:
            print(line_number, a_line, end="")
```

```
input()
```



```
C:\P
1 001 Sam Robinson 30 Sidmouth
6 006 Sam Smith 42 Seaton
```

# Reading from Text Files in Python

## Learning Objectives:

Reading and Writing data into Text Files

## Finding and Printing Lines based on 'User Defined Criteria'

```
line_number = 0
criteria = input("What would you like to search for? ")
with open('example.txt', "r") as a_file:
    for a_line in a_file:
        line_number += 1
        if criteria in a_line:
            print(line_number, a_line, end="")
```

input()



```
C:\Python33\p
What would you like to search for? Exeter
3 003 Dave Smith 53 Exeter
```



# Holiday Homework

- Gfscomputing.net
- Watch the tutorial 12
- Complete Activity 3

If you cannot install Python, use [skulpt.org](http://skulpt.org)

A\*\* extension: Watch tutorials 13-18 and also complete the programs shown in the videos



# Activity 3



30 mins

Create a text file with some lines of data (maybe your favourite computer games, their price and year of sale)

- 1) Create a program which uses the WITH statement, reads the text file and prints each line, one after the other, without any blank lines between, to the screen.
- 2) Create a program which asks the user to enter the name of one of the games in the file. It will then search the text file and print out the line of the text file which matches, to the screen.