# Lesson 13 – Activity Sheet

# Getting Started

The micro:bit contains approximately 30k of **storage** which can be accessed and used as a file system: A **file system** is defined as ‘organising data in a persistent manner - any data stored in a file system should survive restarts of the device.’ You can use the micro:bit to store and organise files.

Research some of these file extensions and complete the table below

|  |  |
| --- | --- |
| .ppt |  |
| .mov |  |
| .mp3 |  |
| .vlc |  |
| .txt |  |

The micro:bit uses a **flat-file** system which means that all of the files are stored in on one place. Imagine if you kept all your clothes in one drawer, this is a flat-file system. Most computers and devices use a **directory** structure, one drawer for each type of clothing, jumpers, socks, t-shirts.

## **Creating the file**

To develop a file, you need to create a file, name it and then set it to **write** mode, which means you can write text to it. This uses the code: with open('hello.txt', 'w') as newFile:

Then we can write to the file using the command newFile.write("Hello my name is")

Where you replace the ("Hello my name is") with your own text.

Enter the first part of the program code below, (the parts shown in bold are the names that you can customise and use your own.)

# Add your Python code here. E.g.

from microbit import \*

import os

with open(**'hello.txt'**, 'w') as newFile:

newFile.write("**Hello my name is**")

If you download and run the program it will create a file named hello and then add the text ‘Hello my name is’ to the file. This file is now stored on the micro:bit.

## **Reading the file**

At this stage you cannot use a computer to view the files on your micro:bit as it requires additional software called REPL. This requires a downloaded version of the Mu editor. However, we can write a program to **read** the contents of the file and display it on the micro:bit LEDs. This uses code with open() to open the file and then we read the line of text and save it to a variable.

Then use the code display.scroll(output) to scroll the contents of the file. Add the lines of code below to your program.

with open('hello.txt') as file:

output=file.readline()

display.scroll(output)

Save and download your program to your micro:bit. Swap with another student and plug it in to retrieve the contents of the text file.

Flashing your micro:bit will DESTROY ALL YOUR DATA since it re-writes all the flash memory used by the device and the file system is stored in the flash memory. However, if you switch off your device the data will remain intact until you either delete it or re-flash the device.

## **The Complete Program Code**

# Add your Python code here. E.g.

from microbit import \*

import os

with open('hello.txt', 'w') as newFile:

newFile.write("Hello my name is")

with open('hello.txt') as file:

output=file.readline()

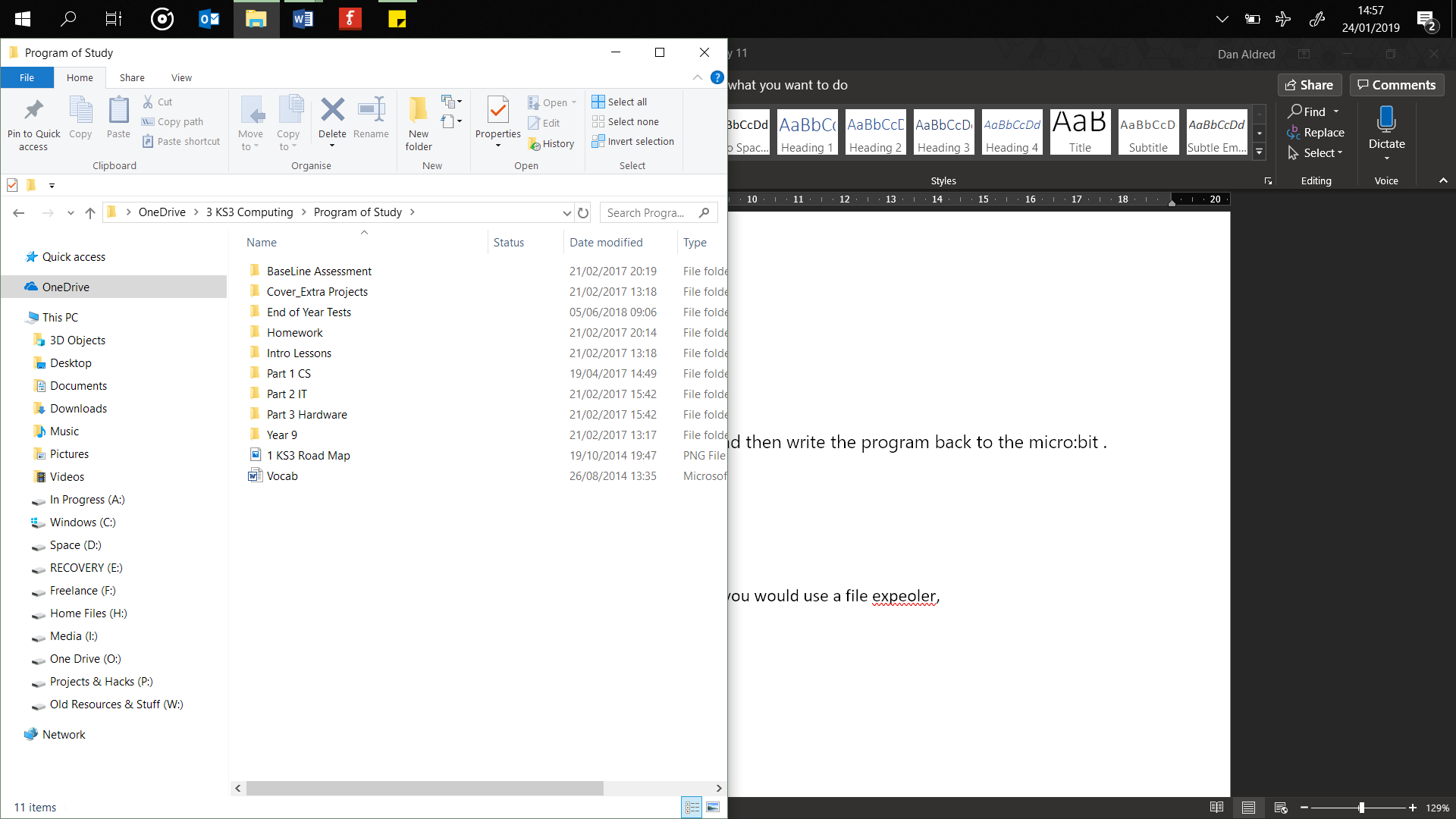
display.scroll(output)

## **What next?**

Change the contents of the file, add two lines of text and then write the program back to the micro:bit .

## **Listing the Files On the micro:bit**

It is possible to list the files on the micro:bit, the same way you would use a file explorer, except that the micro:bit has no graphic display, so the file names are scrolled across the LEDs.

To list the filenames use the operating system, **OS**, command listdir() and store the value, the file names in a varaiable. Then use the display.scoll() to scroll the names.

The files’ names are first converted into a string using the code str() This is because the filename is stored in a different format which cannot be scrolled.

Save and download your program to your micro:bit. Swap with another student and plug it in to retrieve the contents of the text file.

from microbit import \*

import os

with open('hello.txt', 'w') as newFile:

newFile.write("Hello my name is")

with open('hello.txt') as file:

output=file.readline()

display.scroll(output)

# a listing of the file directory

files = os.listdir()

display.scroll(str(files))

## Success Criteria

* Create, open and write text to a file on the micro:bit
* Read the contents of the file and display them.
* Write code to list the name of the files on the micro:bit

## Pro-tip

Always check for the correct indentation and use of brackets, missing one out or using too many will result in the program not running correctly.

You can create a Python file by adding the file name to the open()line of code, for example, with open('hello.py', 'w') as newFile: creates a Python program file named hello.

## Test Time

The micro:bit user guides give the following warning about flashing the micro:bit and working with files:

Flashing your micro:bit will DESTROY ALL YOUR DATA since it re-writes all the flash memory used by the device and the file system is stored in the flash memory. However, if you switch off your device the data will remain intact until you either delete it or re-flash the device.

## Stretch Tasks

Write a program that uses the two buttons. If Button A is pressed the first phrase it written to the file, if Button B is pressed then a different phrase is written to the file.

# Add your Python code here. E.g.

from microbit import \*

import os

while True:

with open('hello.txt', 'w') as newFile:

if button\_a.is\_pressed():

newFile.write("Arch")

elif button\_b.is\_pressed():

newFile.write("Linux")

else:

pass

with open('hello.txt') as file:

output=file.readline()

display.scroll(output)

## Final Thoughts

In this lesson you have learnt how to write to files that are saved onto your micro:bit. Creating a small storage device that can be used to write and read files from. This is similar to how you might use a USB memory stick.