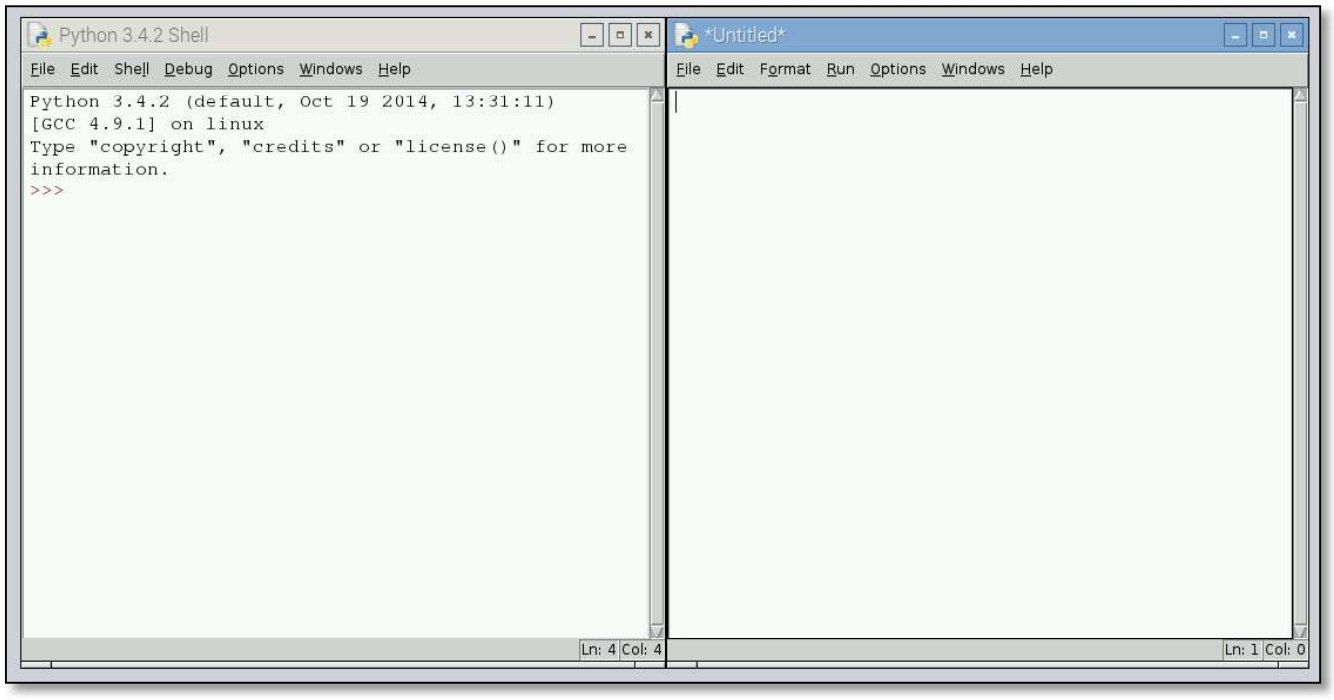
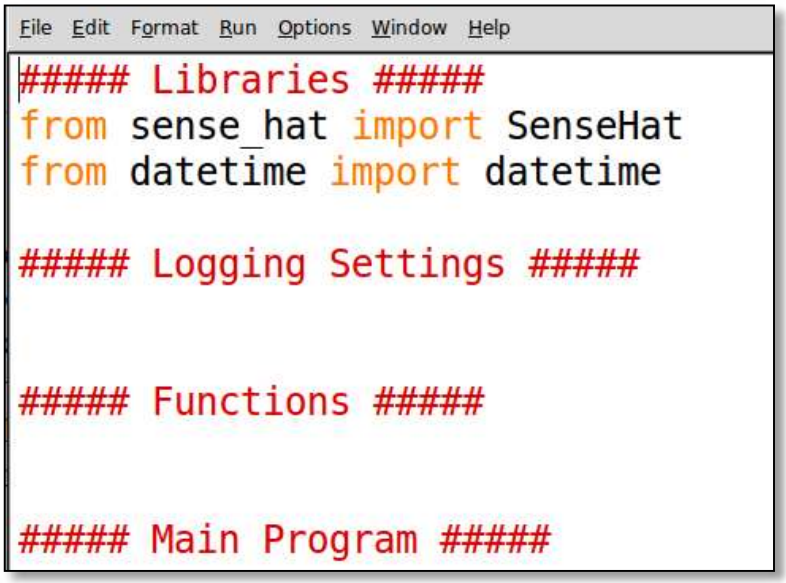
# Approach 2: Storing readings in Device itself as csv file

In this approach, python script is used which takes the readings from device and stores it in comma separated values file. To begin your script you need to boot your Raspberry Pi into desktop mode and run Idle for Python 3 from the programming section of the menu. Once Idle has loaded you will need to select **File** and then **New File** which will load a separate window in which you can write your code.




In your right hand window, add the following lines of python code. The lines starting with a # symbol are **comments** and are ignored by the computer. We have used comments here to break the code into four sections, which will make it easier to build your program as it gets more complex.




- The first section, **Libraries**, is where you will import code that will give your program extra abilities. The line from sense_hat import SenseHat allows your program to use the Sense-HAT hardware. The line from datetime import datetime allows your program to use the time module.
- The section headed **Logging Settings** is where you will be able to control different features of your logger program.
- The third section, **Functions**, will contain short "chunks" of reusable code which do a specific job, such as writing the current data to a file.
- The final section, **Main Program**, is the part of your code which uses each of the functions in the right sequence to run the whole program.

In order to get data from the Sense HAT you will need to write a function called **get_sense_data** which will check each sensor in turn and store the sensor data in a list. The function should be added to the **Functions** section.

```
def get_sense_data():
  sense_data=[]


  sense_data.append(sense.get_temperature_from_humidity())
  sense_data.append(sense.get_temperature_from_pressure())
  sense_data.append(sense.get_humidity())
  sense_data.append(sense.get_pressure())
```

The first line defines your function name, and the second sets up an empty **list** structure into which you will add your collected data.

The next four lines get data from some of the sensors and adds (or appends) them to the `sense_data` list.

Next we'll need to add some lines to your **Main Program** Section, this will need to do two things:

- create a sense object, which represents the Sense HAT
- repeatedly **get_sense_data** from the sensors and display it

Add the following code to the **Main Program** section:

```
sense = SenseHat()
while True:
  sense_data = get_sense_data()
  print(sense_data)
```

3. Another function you will need is the file_setup function which will create a list of headings that will be written to the file before any data. The function is shown below and needs to be added to your Functions section.

```
def file_setup(filename):
  header =["temp_h","temp_p","humidity","pressure",
  "pitch","roll","yaw",
  "mag_x","mag_y","mag_z",
  "accel_x","accel_y","accel_z",
  "gyro_x","gyro_y","gyro_z",
  "timestamp"]
  with open(filename,"w") as f:
   f.write(",".join(str(value) for value in header)+ "\n")
```

This function is slightly different to the previous as it needs an input (or parameter) in order to work; in this case the input has been called filename. When the main program calls this function it must also give the function the name of the file to write to. If it were called like this file_setup("output.csv") the function would create output.csv

The function itself creates a list of header names called header. It then opens a file in write mode (which will overwrites any previous data) and refers to that file as f. whilst the file is open it joins all the list headings together using commas and writes that line to the file.

4. The two functions and the settings you added now need to be used in the main program.

Straight after the lines that read:

```
##### Main Program #####

sense = SenseHat()
```

add the following:

```
batch_data= []

if FILENAME == "":

  filename = "SenseLog-"+str(datetime.now())+".csv"

else:

  filename = FILENAME+"-"+str(datetime.now())+".csv"

file_setup(filename)
```

The first line here creates an empty list that the program will keep adding sense_data lines to until it reaches 50 (or whatever value is set by WRITE_FREQUENCY). The if/else block checks whether a FILENAME has been set, if it hasn't then the default of "SenseLog" is used. The current date and time is also added to the filename.Finally the file_setup functions is called and given the filename that was decided upon in the previous if / else block.

5. The last step is to change some of the logic inside the while True: loop.

You need to make it collect sense_data

Then use the log_data function to convert the sense_data into csv form and add the the current batch_data. Once the data is logged, the program checks whether the size of batch_data exceeds the WRITE_FREQUENCY setting, if so the data is written to a file and batch_data is reset.

Your while True: loop should be updated to look like this:

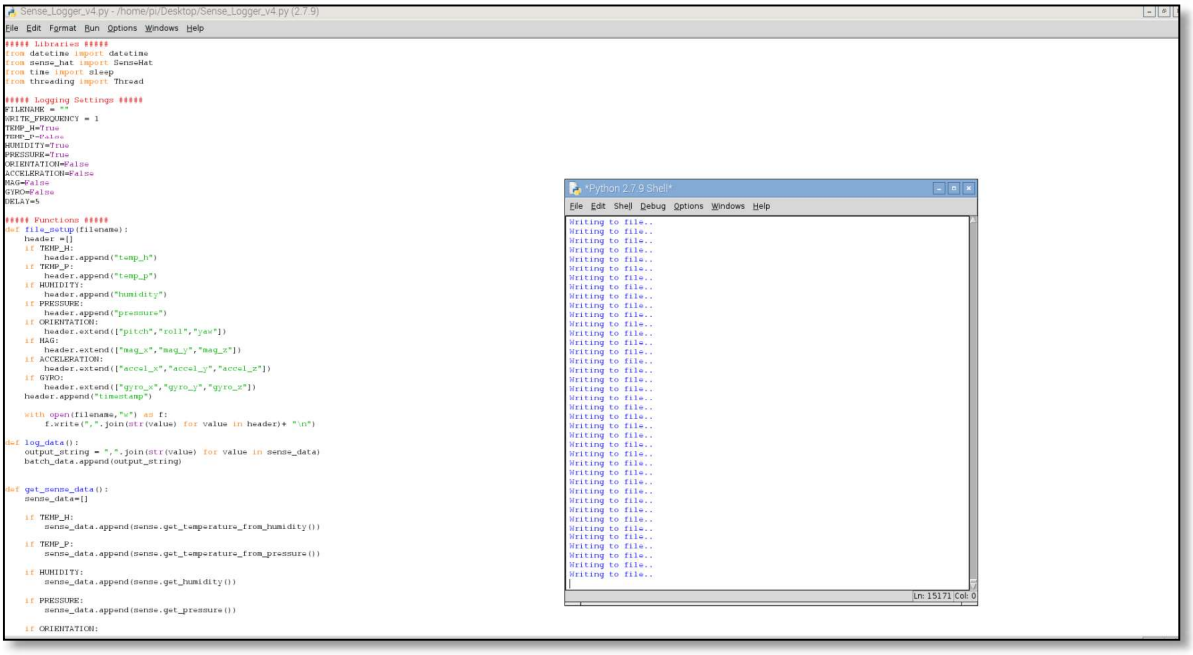
```
while True:
  sense_data = get_sense_data()
  log_data()


  if len(batch_data) >= WRITE_FREQUENCY:
   print("Writing to file..")
    with open(filename,"a") as f:
    for line in batch_data:
            f.write(line + "\n")
        batch_data = []
```

The line print("Writing to file..") is optional, but it will show whenever data is being written.The line with open(filename,"a") as f: opens the file in append mode which adds the data at the end point of the file rather than overwriting. You can check your code against a full code listing here. When you running the program you should simply see the messages saying Writing to file.. every so often.

## Writing to a file

You can stop logging by pressing Ctrl+C




## Code:

##### Libraries #####

from datetime import datetime

from sense_hat import SenseHat

from time import sleep

from threading import Thread

##### Logging Settings #####

FILENAME = ""

WRITE_FREQUENCY = 1

TEMP_H=True

TEMP_P=False

HUMIDITY=True

PRESSURE=True

ORIENTATION=False

ACCELERATION=False

```
MAG=False

GYRO=False

DELAY=5


##### Functions #####

def file_setup(filename):

    header =[]

    if TEMP_H:

        header.append("temp_h")

    if TEMP_P:

        header.append("temp_p")

    if HUMIDITY:

        header.append("humidity")

    if PRESSURE:

        header.append("pressure")

    if ORIENTATION:

        header.extend(["pitch","roll","yaw"])

    if MAG:


        header.extend(["mag_x","mag_y","mag_z"])

    if ACCELERATION:

        header.extend(["accel_x","accel_y","accel_z"])

    if GYRO:

        header.extend(["gyro_x","gyro_y","gyro_z"])

    header.append("timestamp")
```

```
    with open(filename,"w") as f:
        f.write(",".join(str(value) for value in header)+ "\n")
def log_data():
    output_string = ",".join(str(value) for value in sense_data)
    batch_data.append(output_string)
def get_sense_data():
    sense_data=[]

    if TEMP_H:
        sense_data.append(sense.get_temperature_from_humidity())

    if TEMP_P:
        sense_data.append(sense.get_temperature_from_pressure())

    if HUMIDITY:
        sense_data.append(sense.get_humidity())

    if PRESSURE:
        sense_data.append(sense.get_pressure())

    if ORIENTATION:
        o = sense.get_orientation()
        yaw = o["yaw"]
        pitch = o["pitch"]
        roll = o["roll"]
        sense_data.extend([pitch,roll,yaw])
```

```
    if MAG:
        mag = sense.get_compass_raw()
        mag_x = mag["x"]
        mag_y = mag["y"]
        mag_z = mag["z"]
        sense_data.extend([mag_x,mag_y,mag_z])

    if ACCELERATION:
        acc = sense.get_accelerometer_raw()
        x = acc["x"]
        y = acc["y"]
        z = acc["z"]
        sense_data.extend([x,y,z])

    if GYRO:
        gyro = sense.get_gyroscope_raw()
        gyro_x = ["x"]
        gyro_y = ["y"]
        gyro_z = ["z"]
        sense_data.extend([gyro_x,gyro_y,gyro_z])
    sense_data.append(datetime.now())
    return sense_data
def timed_log():
    while True:
        log_data()
```

```
        sleep(DELAY)


##### Main Program #####
sense = SenseHat()
batch_data= []
if FILENAME  == "":
    filename = "SenseLog-"+str(datetime.now())+".csv"
else:
    filename = FILENAME+"-"+str(datetime.now())+".csv"
file_setup(filename)
if DELAY  > 0:
    sense_data = get_sense_data()
    Thread(target=  timed_log).start()
while True:
    sense_data = get_sense_data()
    if DELAY  == 0:
        log_data()
    if len(batch_data)  >= WRITE_FREQUENCY:
        print("Writing  to file..")
        with open(filename,"a")  as f:
            for line in batch_data:
                f.write(line + "\n")
            batch_data = []
```

# OUTPUT CSV FORMAT: