# Weather Display With Sense Hat

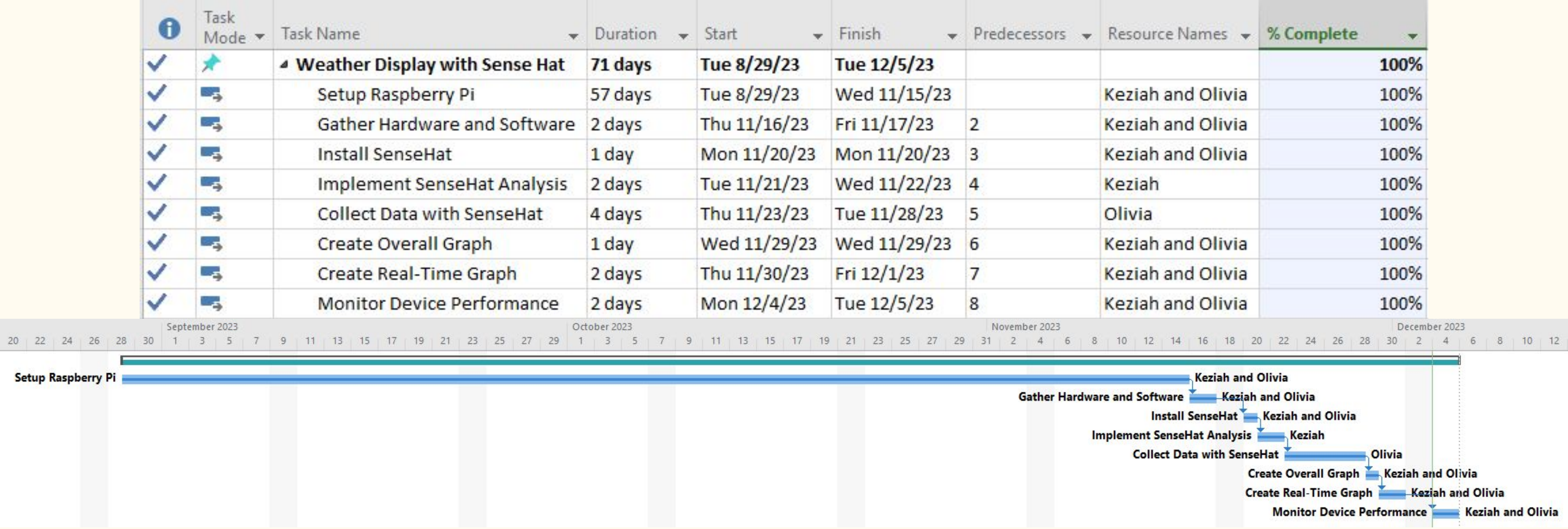
By Olivia French and Keziah John

Group E

# Description of project

The Weather Display with Sense HAT on a Raspberry Pi 4 is a compact device equipped with sensors to measure environmental parameters. It includes a gyroscope for orientation and movement, an accelerometer for detecting speed changes, a magnetometer for Earth's magnetic field orientation, a barometric pressure sensor for altitude estimation and short-term weather prediction, and a humidity sensor for measuring air moisture content. Overall, this device can track and analyze temperature, humidity, and pressure, allowing for the visualization of these parameters over time through graphical representation.

# Project Activities



| ℹ | Task Mode | Task Name | Duration | Start | Finish | Predecessors | Resource Names | % Complete |
|---|---|---|---|---|---|---|---|---|
| ✓ | | **Weather Display with Sense Hat** | **71 days** | **Tue 8/29/23** | **Tue 12/5/23** | | | **100%** |
| ✓ | | Setup Raspberry Pi | 57 days | Tue 8/29/23 | Wed 11/15/23 | | Keziah and Olivia | 100% |
| ✓ | | Gather Hardware and Software | 2 days | Thu 11/16/23 | Fri 11/17/23 | 2 | Keziah and Olivia | 100% |
| ✓ | | Install SenseHat | 1 day | Mon 11/20/23 | Mon 11/20/23 | 3 | Keziah and Olivia | 100% |
| ✓ | | Implement SenseHat Analysis | 2 days | Tue 11/21/23 | Wed 11/22/23 | 4 | Keziah | 100% |
| ✓ | | Collect Data with SenseHat | 4 days | Thu 11/23/23 | Tue 11/28/23 | 5 | Olivia | 100% |
| ✓ | | Create Overall Graph | 1 day | Wed 11/29/23 | Wed 11/29/23 | 6 | Keziah and Olivia | 100% |
| ✓ | | Create Real-Time Graph | 2 days | Thu 11/30/23 | Fri 12/1/23 | 7 | Keziah and Olivia | 100% |
| ✓ | | Monitor Device Performance | 2 days | Mon 12/4/23 | Tue 12/5/23 | 8 | Keziah and Olivia | 100% |

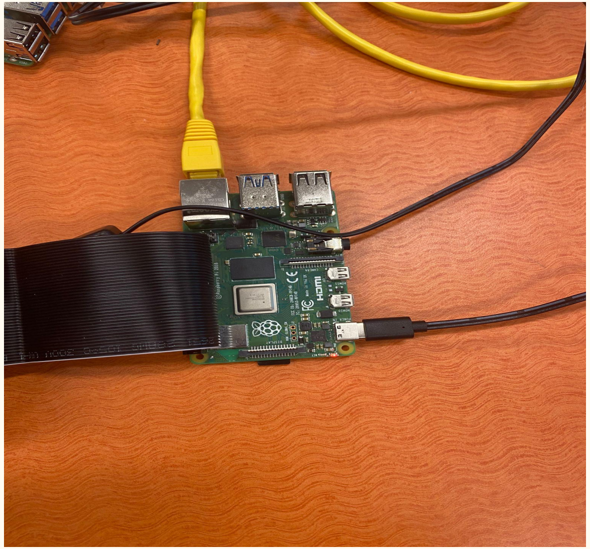
# Hardware Setup

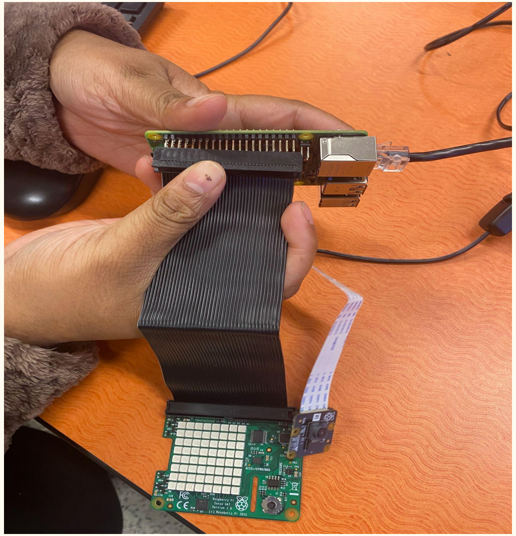
Needed Hardware: Raspberry Pi 4, Sense Hat, Raspberry pi 4 power supply, and RJ45 ethernet cable.

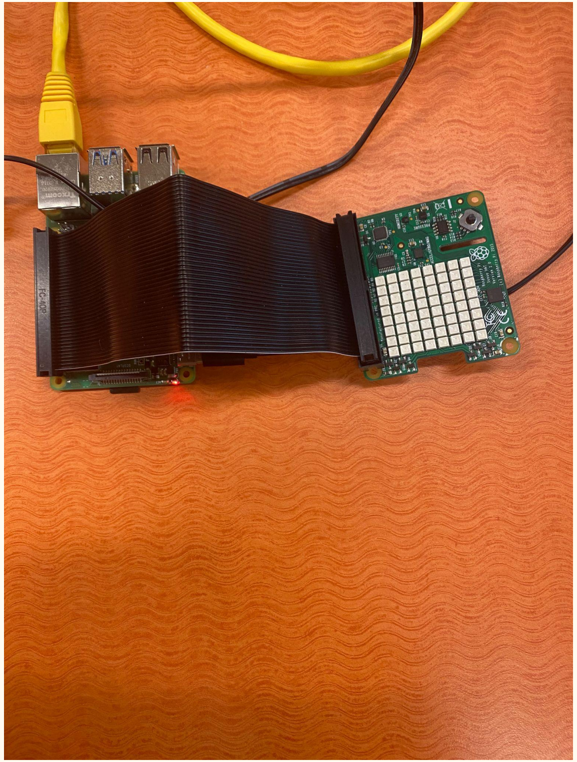
Step 1: Connect RJ45 ethernet cable from ethernet port to raspberry pi 4

Step 2: Connect sense hat (sensor)

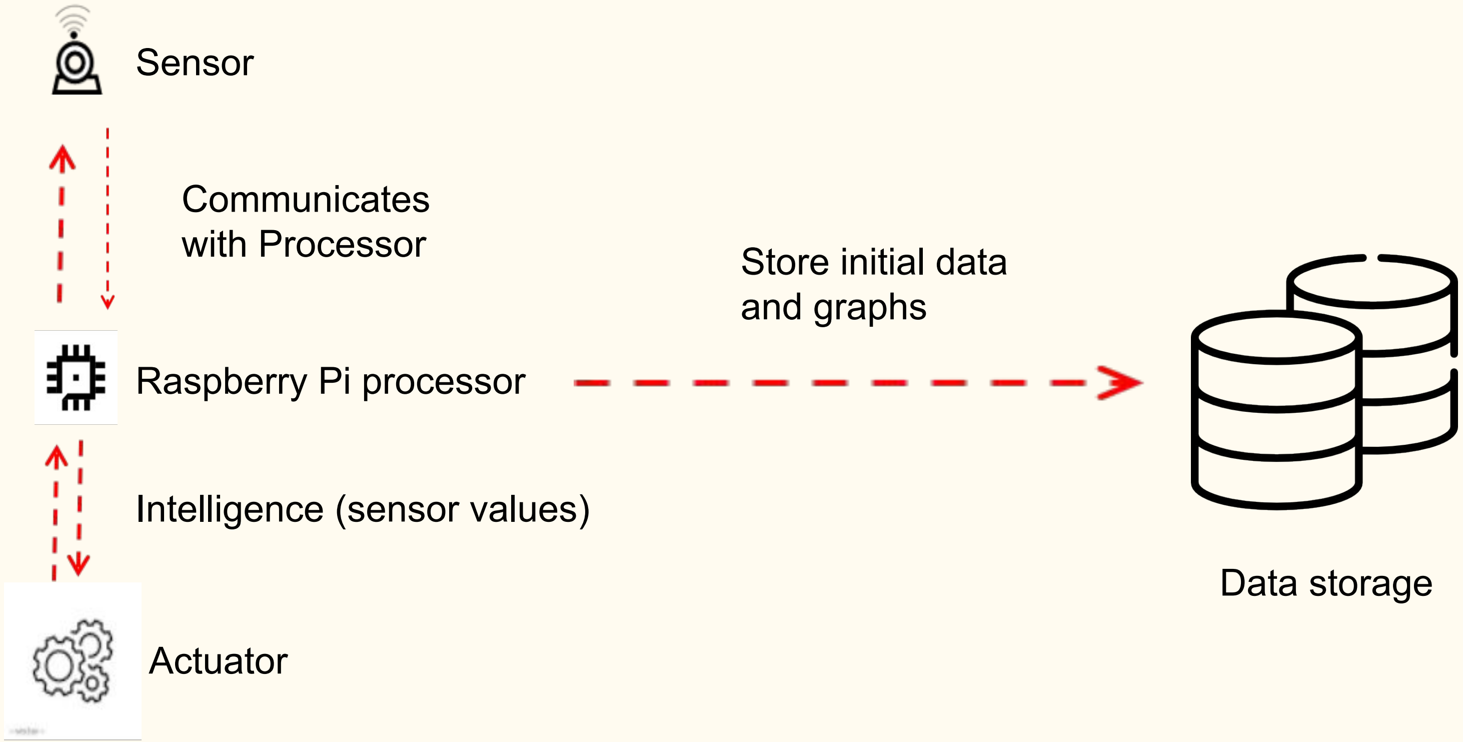
Step 3: Connect the power supply.







# Avatar



Sensor

Communicates
with Processor

Store initial data
and graphs

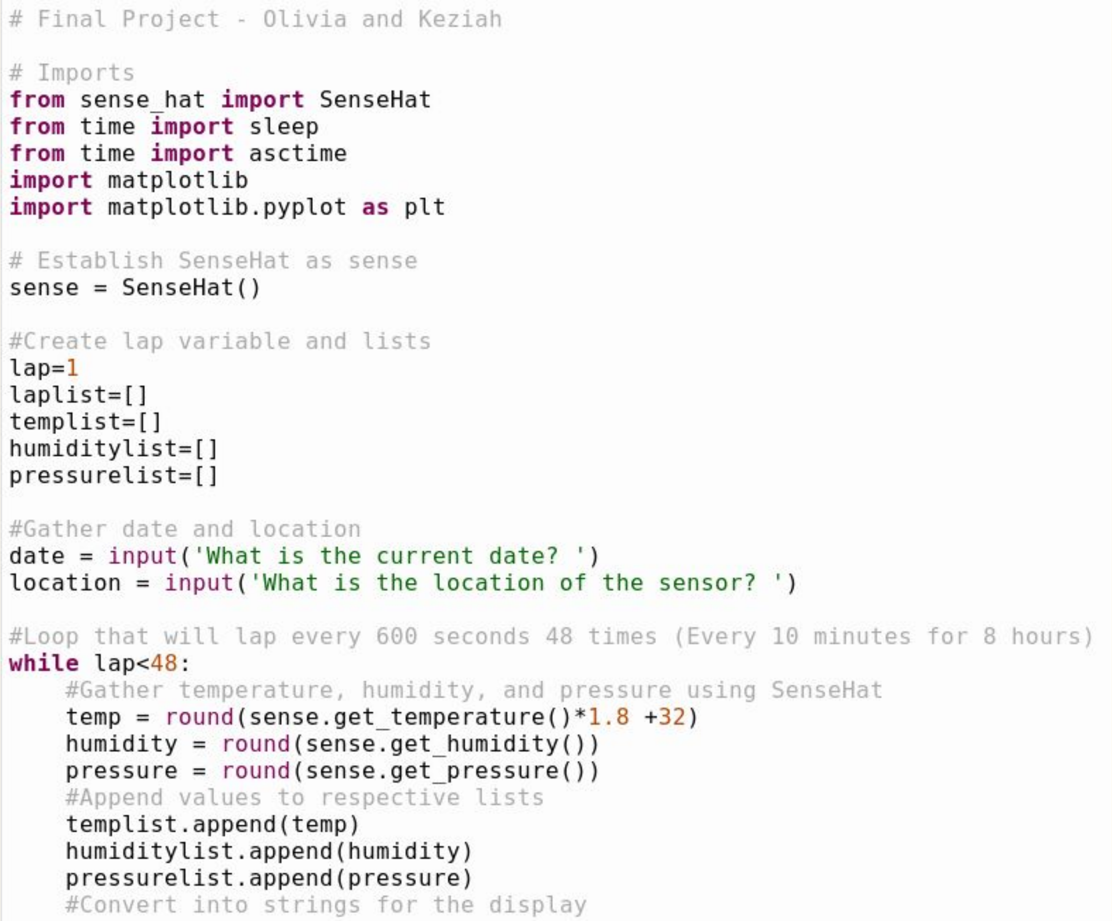
Raspberry Pi processor

Intelligence (sensor values)

Actuator

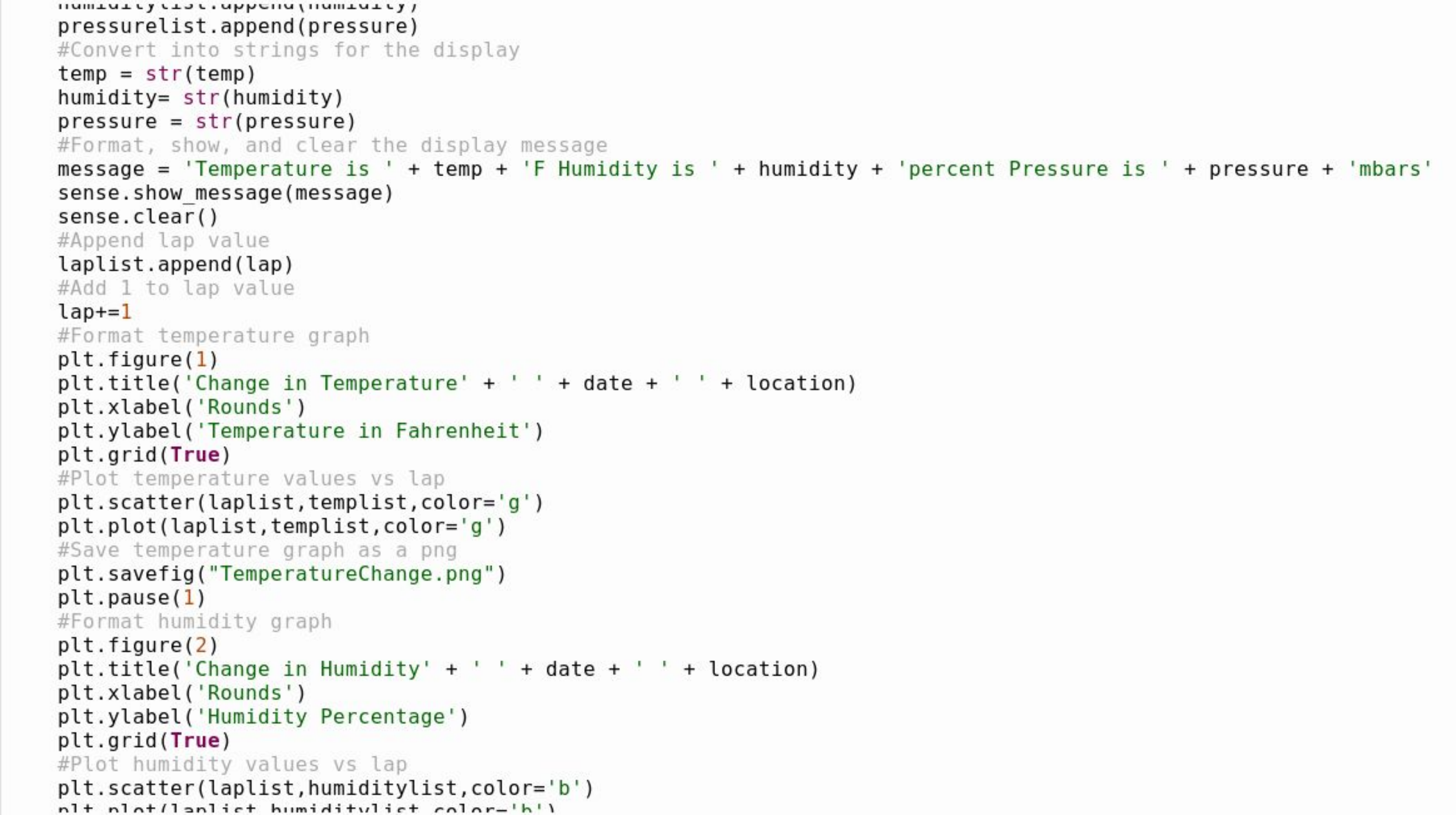
Data storage

# Python Code - WeatherGraph.py



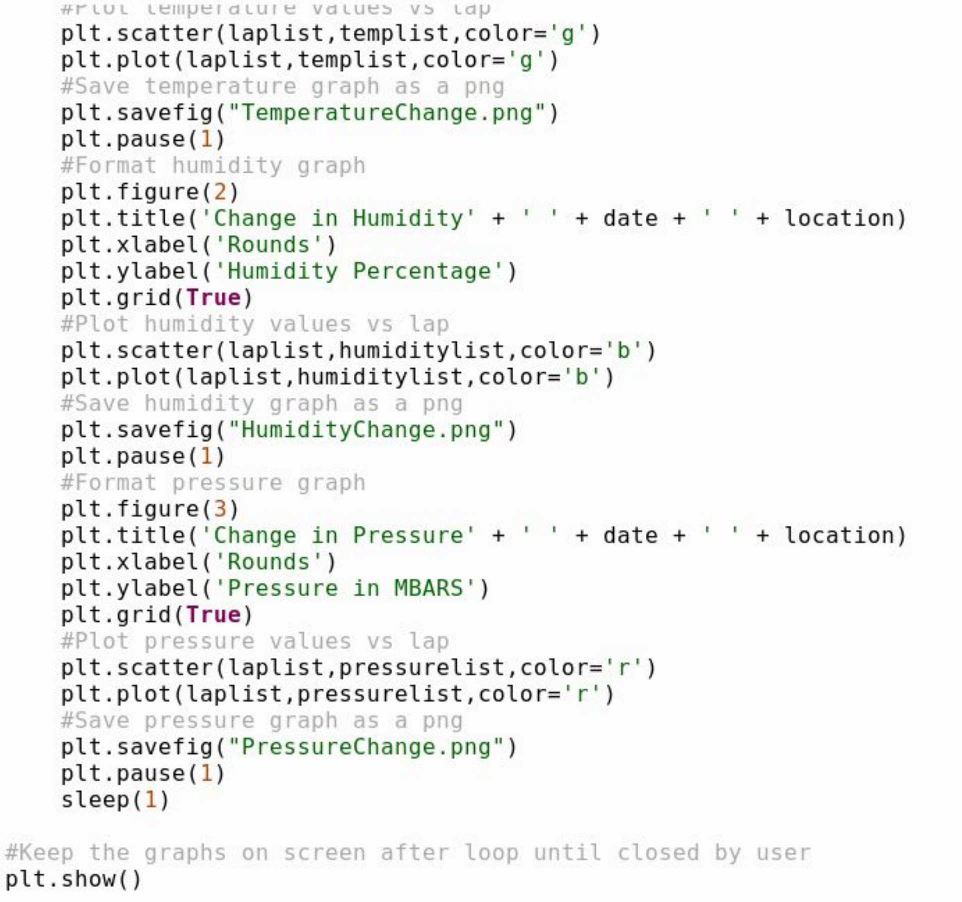
```python
# Final Project - Olivia and Keziah

# Imports
from sense_hat import SenseHat
from time import sleep
from time import asctime
import matplotlib
import matplotlib.pyplot as plt

# Establish SenseHat as sense
sense = SenseHat()

#Create lap variable and lists
lap=1
laplist=[]
templist=[]
humiditylist=[]
pressurelist=[]

#Gather date and location
date = input('What is the current date? ')
location = input('What is the location of the sensor? ')

#Loop that will lap every 600 seconds 48 times (Every 10 minutes for 8 hours)
while lap<48:
    #Gather temperature, humidity, and pressure using SenseHat
    temp = round(sense.get_temperature()*1.8 +32)
    humidity = round(sense.get_humidity())
    pressure = round(sense.get_pressure())
    #Append values to respective lists
    templist.append(temp)
    humiditylist.append(humidity)
    pressurelist.append(pressure)
    #Convert into strings for the display
```

# Python Code - WeatherGraph.py



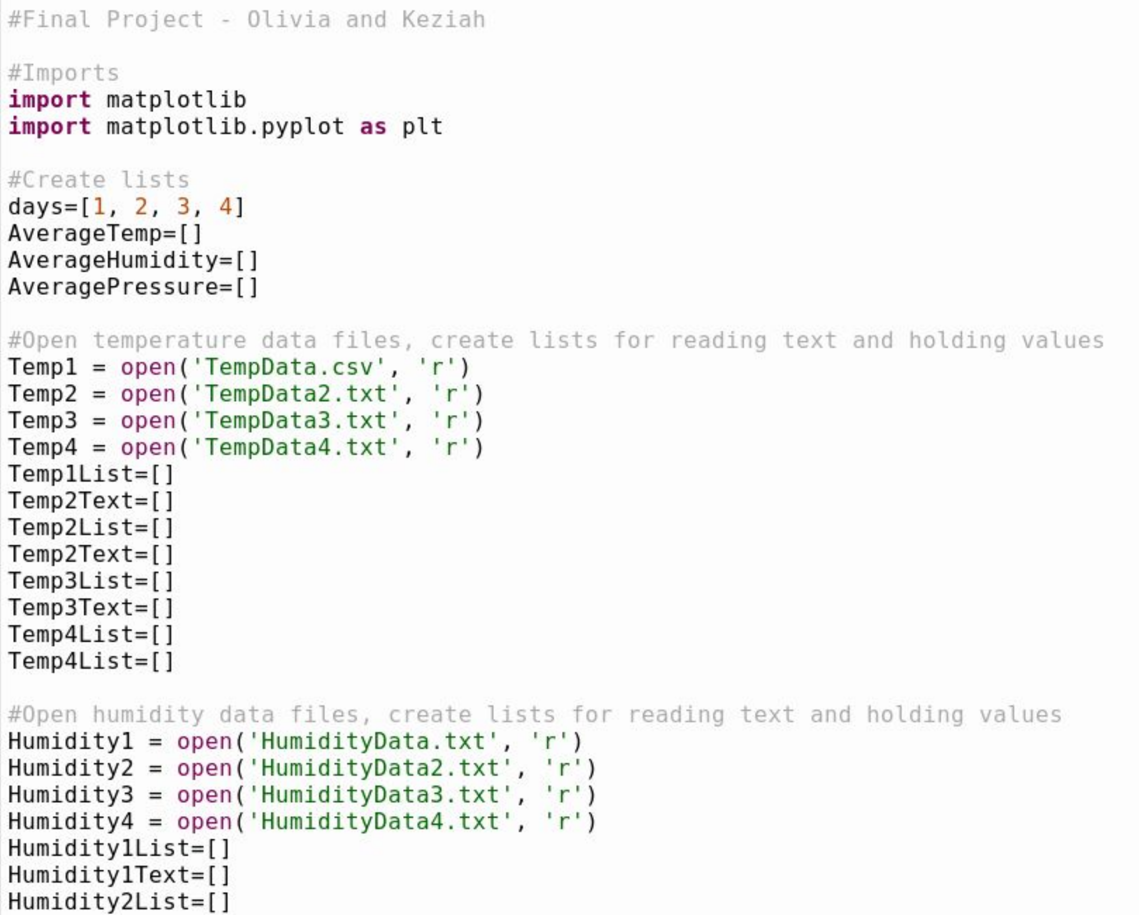
```
humiditylist.append(humidity)
pressurelist.append(pressure)
#Convert into strings for the display
temp = str(temp)
humidity= str(humidity)
pressure = str(pressure)
#Format, show, and clear the display message
message = 'Temperature is ' + temp + 'F Humidity is ' + humidity + 'percent Pressure is ' + pressure + 'mbars'
sense.show_message(message)
sense.clear()
#Append lap value
laplist.append(lap)
#Add 1 to lap value
lap+=1
#Format temperature graph
plt.figure(1)
plt.title('Change in Temperature' + ' ' + date + ' ' + location)
plt.xlabel('Rounds')
plt.ylabel('Temperature in Fahrenheit')
plt.grid(True)
#Plot temperature values vs lap
plt.scatter(laplist,templist,color='g')
plt.plot(laplist,templist,color='g')
#Save temperature graph as a png
plt.savefig("TemperatureChange.png")
plt.pause(1)
#Format humidity graph
plt.figure(2)
plt.title('Change in Humidity' + ' ' + date + ' ' + location)
plt.xlabel('Rounds')
plt.ylabel('Humidity Percentage')
plt.grid(True)
#Plot humidity values vs lap
plt.scatter(laplist,humiditylist,color='b')
plt.plot(laplist,humiditylist,color='b')
```

# Python Code - WeatherGraph.py



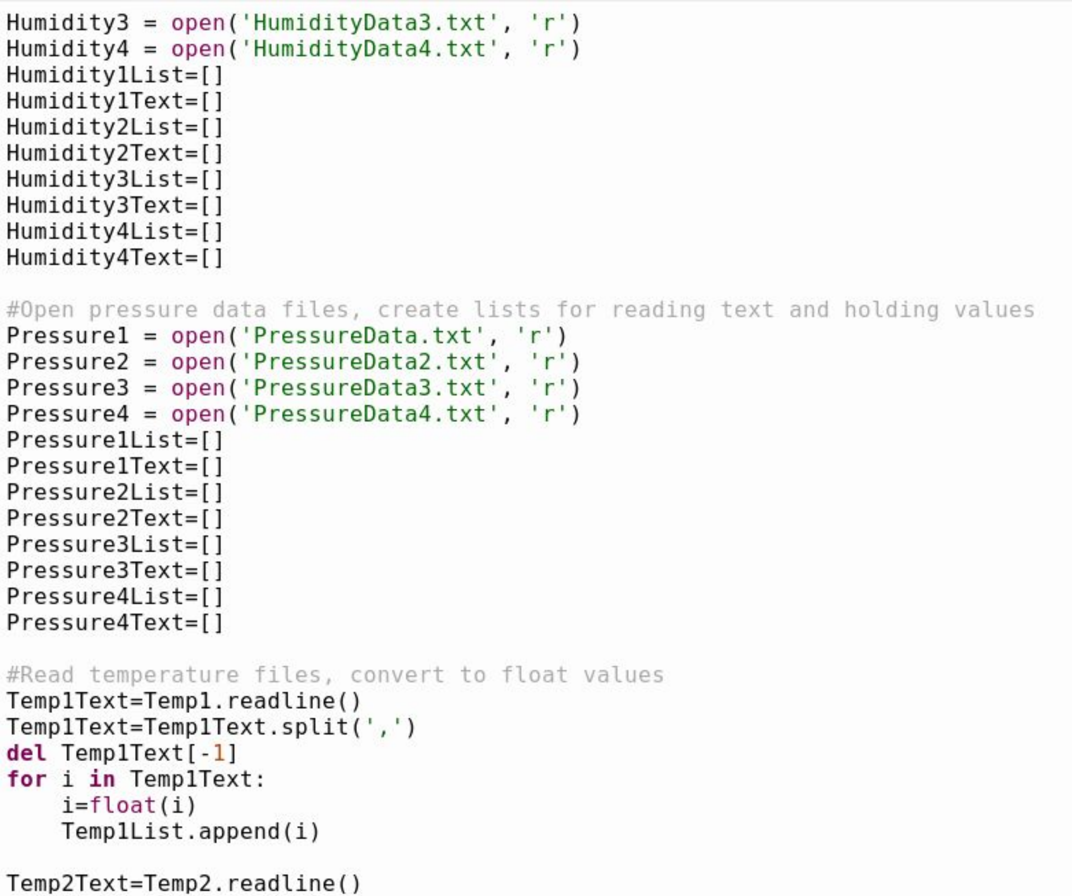
```
    #Plot temperature values vs lap
    plt.scatter(laplist,templist,color='g')
    plt.plot(laplist,templist,color='g')
    #Save temperature graph as a png
    plt.savefig("TemperatureChange.png")
    plt.pause(1)
    #Format humidity graph
    plt.figure(2)
    plt.title('Change in Humidity' + ' ' + date + ' ' + location)
    plt.xlabel('Rounds')
    plt.ylabel('Humidity Percentage')
    plt.grid(True)
    #Plot humidity values vs lap
    plt.scatter(laplist,humiditylist,color='b')
    plt.plot(laplist,humiditylist,color='b')
    #Save humidity graph as a png
    plt.savefig("HumidityChange.png")
    plt.pause(1)
    #Format pressure graph
    plt.figure(3)
    plt.title('Change in Pressure' + ' ' + date + ' ' + location)
    plt.xlabel('Rounds')
    plt.ylabel('Pressure in MBARS')
    plt.grid(True)
    #Plot pressure values vs lap
    plt.scatter(laplist,pressurelist,color='r')
    plt.plot(laplist,pressurelist,color='r')
    #Save pressure graph as a png
    plt.savefig("PressureChange.png")
    plt.pause(1)
    sleep(1)

#Keep the graphs on screen after loop until closed by user
plt.show()
```

# Python Code - AverageWeather.py



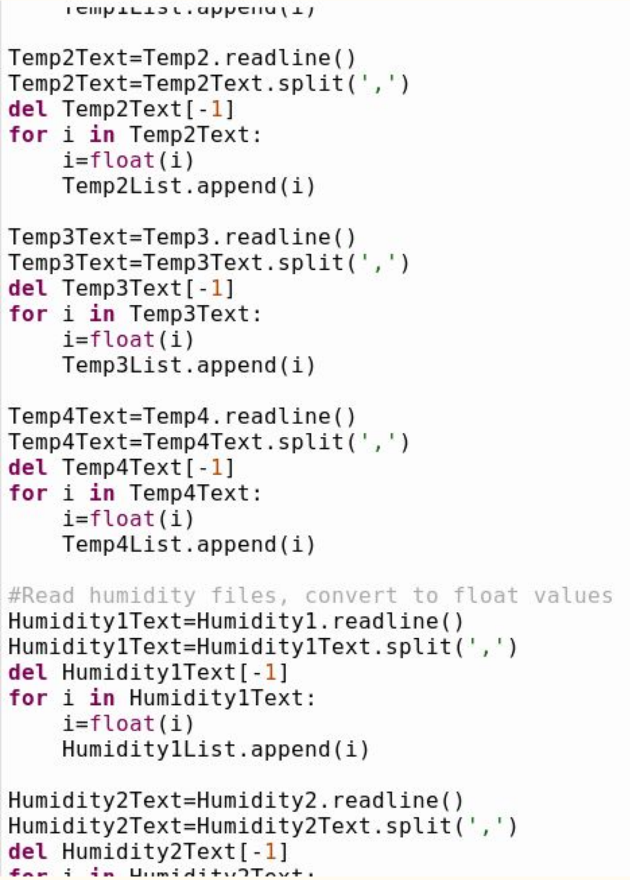
```
#Final Project - Olivia and Keziah

#Imports
import matplotlib
import matplotlib.pyplot as plt

#Create lists
days=[1, 2, 3, 4]
AverageTemp=[]
AverageHumidity=[]
AveragePressure=[]

#Open temperature data files, create lists for reading text and holding values
Temp1 = open('TempData.csv', 'r')
Temp2 = open('TempData2.txt', 'r')
Temp3 = open('TempData3.txt', 'r')
Temp4 = open('TempData4.txt', 'r')
Temp1List=[]
Temp2Text=[]
Temp2List=[]
Temp2Text=[]
Temp3List=[]
Temp3Text=[]
Temp4List=[]
Temp4List=[]

#Open humidity data files, create lists for reading text and holding values
Humidity1 = open('HumidityData.txt', 'r')
Humidity2 = open('HumidityData2.txt', 'r')
Humidity3 = open('HumidityData3.txt', 'r')
Humidity4 = open('HumidityData4.txt', 'r')
Humidity1List=[]
Humidity1Text=[]
Humidity2List=[]
```

# Python Code - AverageWeather.py

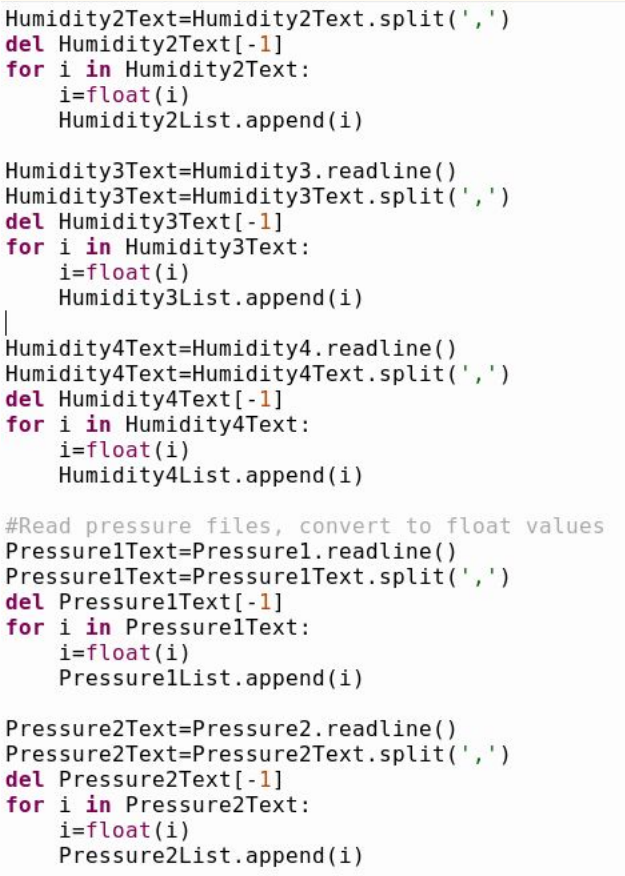


```
Humidity3 = open('HumidityData3.txt', 'r')
Humidity4 = open('HumidityData4.txt', 'r')
Humidity1List=[]
Humidity1Text=[]
Humidity2List=[]
Humidity2Text=[]
Humidity3List=[]
Humidity3Text=[]
Humidity4List=[]
Humidity4Text=[]

#Open pressure data files, create lists for reading text and holding values
Pressure1 = open('PressureData.txt', 'r')
Pressure2 = open('PressureData2.txt', 'r')
Pressure3 = open('PressureData3.txt', 'r')
Pressure4 = open('PressureData4.txt', 'r')
Pressure1List=[]
Pressure1Text=[]
Pressure2List=[]
Pressure2Text=[]
Pressure3List=[]
Pressure3Text=[]
Pressure4List=[]
Pressure4Text=[]

#Read temperature files, convert to float values
Temp1Text=Temp1.readline()
Temp1Text=Temp1Text.split(',')
del Temp1Text[-1]
for i in Temp1Text:
    i=float(i)
    Temp1List.append(i)

Temp2Text=Temp2.readline()
```

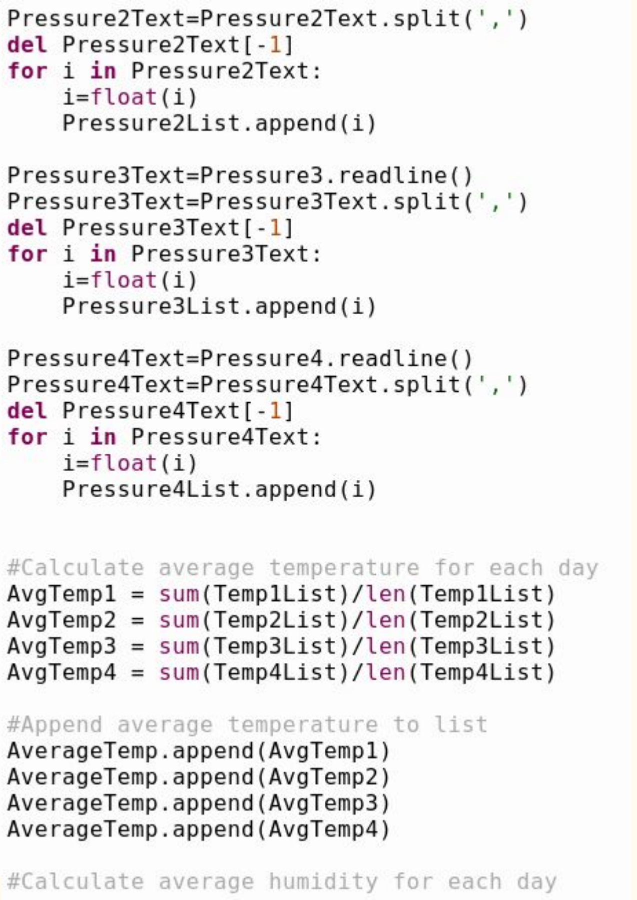
# Python Code - AverageWeather.py



```
    Temp1List.append(i)

Temp2Text=Temp2.readline()
Temp2Text=Temp2Text.split(',')
del Temp2Text[-1]
for i in Temp2Text:
    i=float(i)
    Temp2List.append(i)

Temp3Text=Temp3.readline()
Temp3Text=Temp3Text.split(',')
del Temp3Text[-1]
for i in Temp3Text:
    i=float(i)
    Temp3List.append(i)

Temp4Text=Temp4.readline()
Temp4Text=Temp4Text.split(',')
del Temp4Text[-1]
for i in Temp4Text:
    i=float(i)
    Temp4List.append(i)

#Read humidity files, convert to float values
Humidity1Text=Humidity1.readline()
Humidity1Text=Humidity1Text.split(',')
del Humidity1Text[-1]
for i in Humidity1Text:
    i=float(i)
    Humidity1List.append(i)

Humidity2Text=Humidity2.readline()
Humidity2Text=Humidity2Text.split(',')
del Humidity2Text[-1]
for i in Humidity2Text:
```

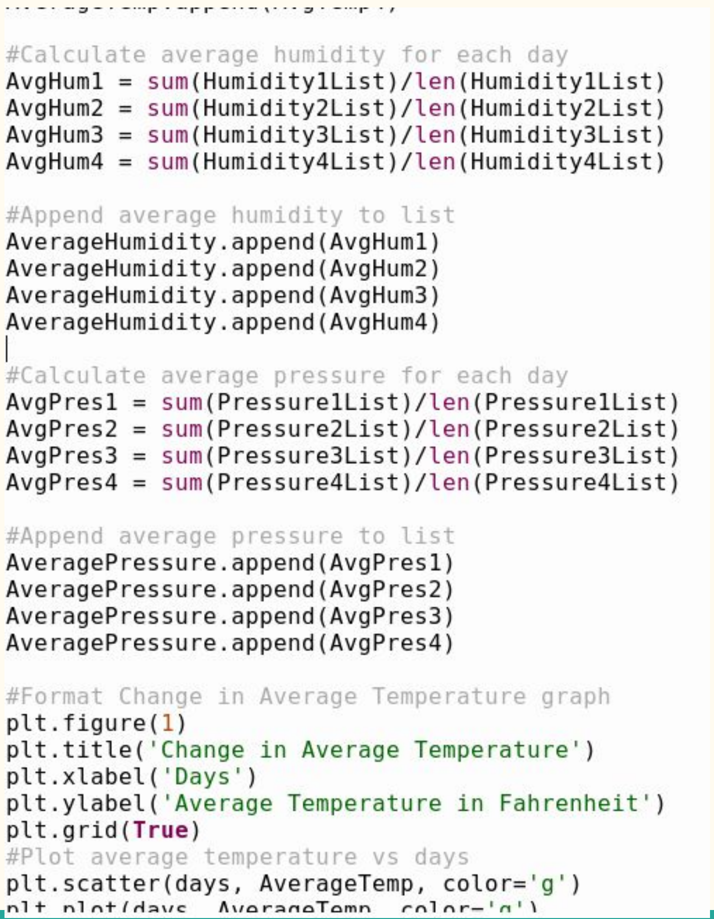


```
Humidity2Text=Humidity2Text.split(',')
del Humidity2Text[-1]
for i in Humidity2Text:
    i=float(i)
    Humidity2List.append(i)

Humidity3Text=Humidity3.readline()
Humidity3Text=Humidity3Text.split(',')
del Humidity3Text[-1]
for i in Humidity3Text:
    i=float(i)
    Humidity3List.append(i)

Humidity4Text=Humidity4.readline()
Humidity4Text=Humidity4Text.split(',')
del Humidity4Text[-1]
for i in Humidity4Text:
    i=float(i)
    Humidity4List.append(i)

#Read pressure files, convert to float values
Pressure1Text=Pressure1.readline()
Pressure1Text=Pressure1Text.split(',')
del Pressure1Text[-1]
for i in Pressure1Text:
    i=float(i)
    Pressure1List.append(i)

Pressure2Text=Pressure2.readline()
Pressure2Text=Pressure2Text.split(',')
del Pressure2Text[-1]
for i in Pressure2Text:
    i=float(i)
    Pressure2List.append(i)
```

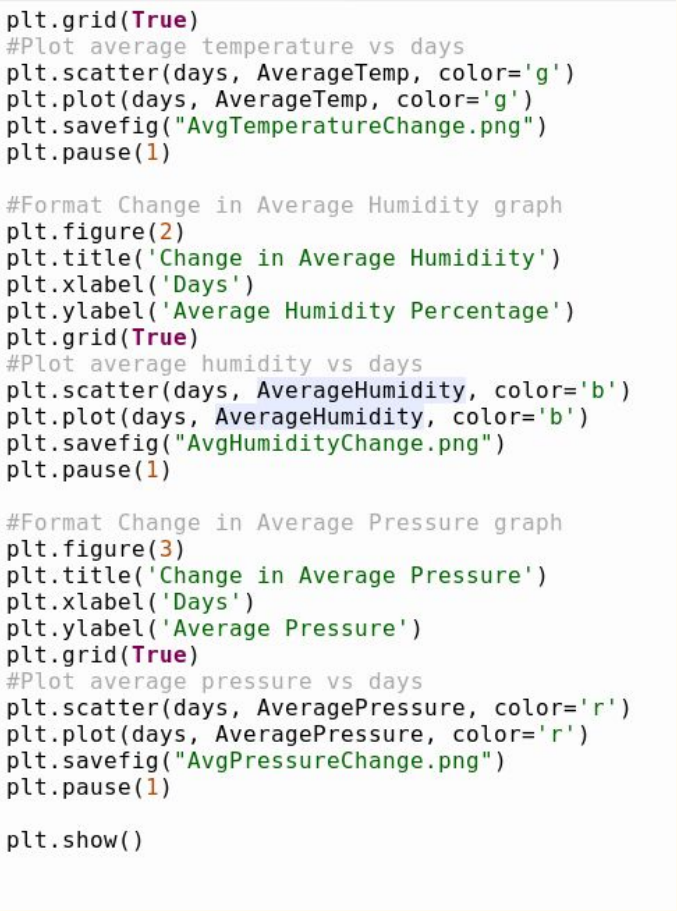


```
Pressure2Text=Pressure2Text.split(',')
del Pressure2Text[-1]
for i in Pressure2Text:
    i=float(i)
    Pressure2List.append(i)

Pressure3Text=Pressure3.readline()
Pressure3Text=Pressure3Text.split(',')
del Pressure3Text[-1]
for i in Pressure3Text:
    i=float(i)
    Pressure3List.append(i)

Pressure4Text=Pressure4.readline()
Pressure4Text=Pressure4Text.split(',')
del Pressure4Text[-1]
for i in Pressure4Text:
    i=float(i)
    Pressure4List.append(i)


#Calculate average temperature for each day
AvgTemp1 = sum(Temp1List)/len(Temp1List)
AvgTemp2 = sum(Temp2List)/len(Temp2List)
AvgTemp3 = sum(Temp3List)/len(Temp3List)
AvgTemp4 = sum(Temp4List)/len(Temp4List)

#Append average temperature to list
AverageTemp.append(AvgTemp1)
AverageTemp.append(AvgTemp2)
AverageTemp.append(AvgTemp3)
AverageTemp.append(AvgTemp4)

#Calculate average humidity for each day
```

# Python Code - AverageWeather.py



```python
AverageTemp.append(AvgTemp)

#Calculate average humidity for each day
AvgHum1 = sum(Humidity1List)/len(Humidity1List)
AvgHum2 = sum(Humidity2List)/len(Humidity2List)
AvgHum3 = sum(Humidity3List)/len(Humidity3List)
AvgHum4 = sum(Humidity4List)/len(Humidity4List)

#Append average humidity to list
AverageHumidity.append(AvgHum1)
AverageHumidity.append(AvgHum2)
AverageHumidity.append(AvgHum3)
AverageHumidity.append(AvgHum4)

#Calculate average pressure for each day
AvgPres1 = sum(Pressure1List)/len(Pressure1List)
AvgPres2 = sum(Pressure2List)/len(Pressure2List)
AvgPres3 = sum(Pressure3List)/len(Pressure3List)
AvgPres4 = sum(Pressure4List)/len(Pressure4List)

#Append average pressure to list
AveragePressure.append(AvgPres1)
AveragePressure.append(AvgPres2)
AveragePressure.append(AvgPres3)
AveragePressure.append(AvgPres4)

#Format Change in Average Temperature graph
plt.figure(1)
plt.title('Change in Average Temperature')
plt.xlabel('Days')
plt.ylabel('Average Temperature in Fahrenheit')
plt.grid(True)
#Plot average temperature vs days
plt.scatter(days, AverageTemp, color='g')
plt.plot(days, AverageTemp, color='g')
```



```python
plt.grid(True)
#Plot average temperature vs days
plt.scatter(days, AverageTemp, color='g')
plt.plot(days, AverageTemp, color='g')
plt.savefig("AvgTemperatureChange.png")
plt.pause(1)

#Format Change in Average Humidity graph
plt.figure(2)
plt.title('Change in Average Humidiity')
plt.xlabel('Days')
plt.ylabel('Average Humidity Percentage')
plt.grid(True)
#Plot average humidity vs days
plt.scatter(days, AverageHumidity, color='b')
plt.plot(days, AverageHumidity, color='b')
plt.savefig("AvgHumidityChange.png")
plt.pause(1)

#Format Change in Average Pressure graph
plt.figure(3)
plt.title('Change in Average Pressure')
plt.xlabel('Days')
plt.ylabel('Average Pressure')
plt.grid(True)
#Plot average pressure vs days
plt.scatter(days, AveragePressure, color='r')
plt.plot(days, AveragePressure, color='r')
plt.savefig("AvgPressureChange.png")
plt.pause(1)

plt.show()
```

# Instructions

Step 1: Get required items: Hardware (Slide 4). Software: Raspberry pi os, python 3, SenseHat, and matplotlib.

Step 2: Setup hardware (slide 4)

Step 3: Setup Software: Open terminal and install software with this command:

```
sudo apt install python3-matplotlib
```

Step 4: Place SenseHat where you would like to collect data.

Step 5: Save the code into a python file named "WeatherGraph.py" in your chosen directory.

Step 6: Navigate to chosen directory in terminal and run the command:

```
sudo python WeatherGraph.py
```

Step 7: Once the loop is finished, closing the figures will save them in to the same directory as the python file WeatherGraph.py.

# Instructions

Step 8: Navigate to directory where collected data is stored

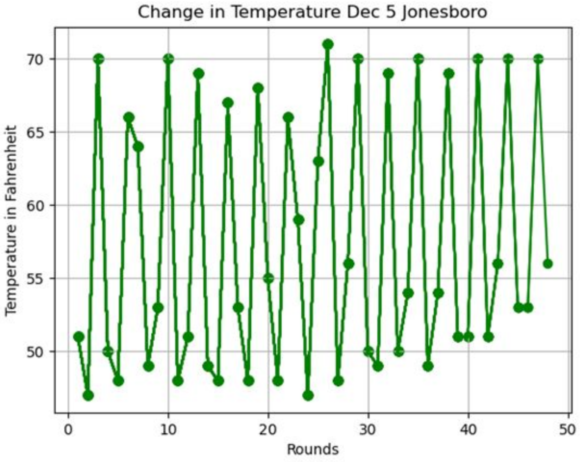
Step 9: Save code into a python file named “AverageWeather.py” in the same directory.

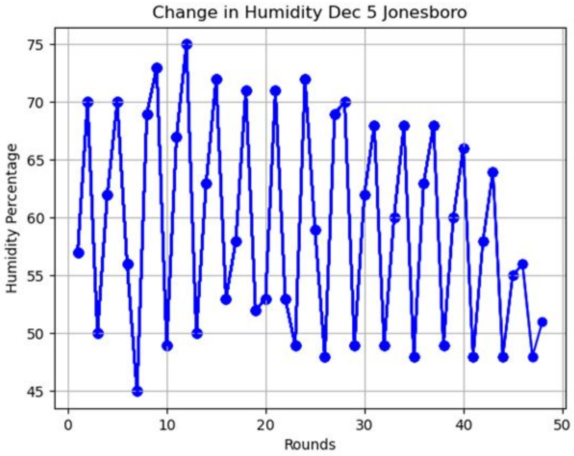
Step 10: Run the command in terminal:

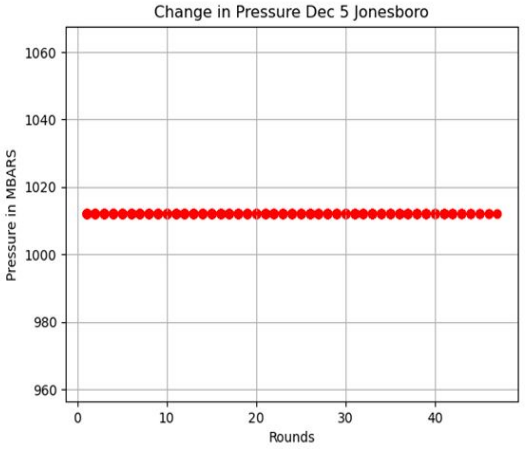
```
sudo python AverageWeather.py
```

# Output Analysis

## WeatherGraph.py outputs ( Real-time Output )

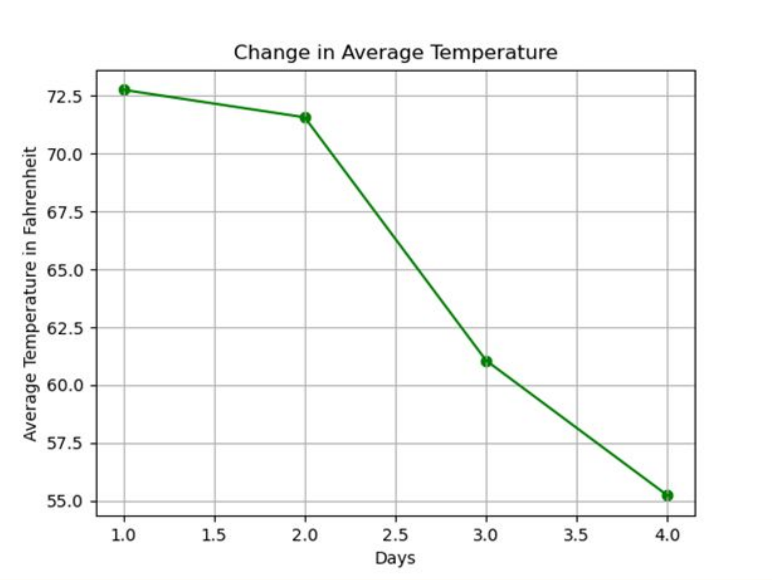


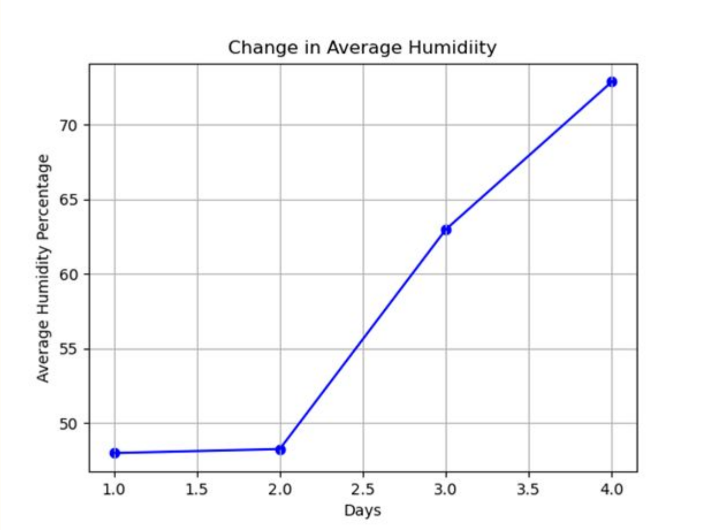


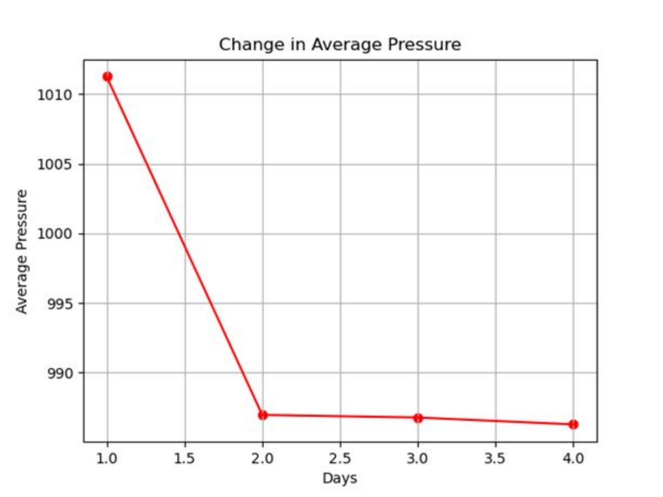


# Output Analysis

## Collected output data (4 days)







## Business Use of the output

**The output which this device created can be used for many business field such as;**

**Agriculture:**

Monitor and analyze environmental conditions in agricultural settings.

Optimize irrigation schedules based on humidity and temperature data.

**Integrate environmental monitoring into smart building systems.**

Adjust heating, ventilation, and air conditioning (HVAC) systems based on real-time weather and humidity data.

Enhance energy efficiency by optimizing systems according to environmental conditions.

**Monitor and control environmental conditions in industrial settings.**

Implement predictive maintenance strategies based on sensor data.

Ensure optimal working conditions for equipment and personnel.

# Any Questions?