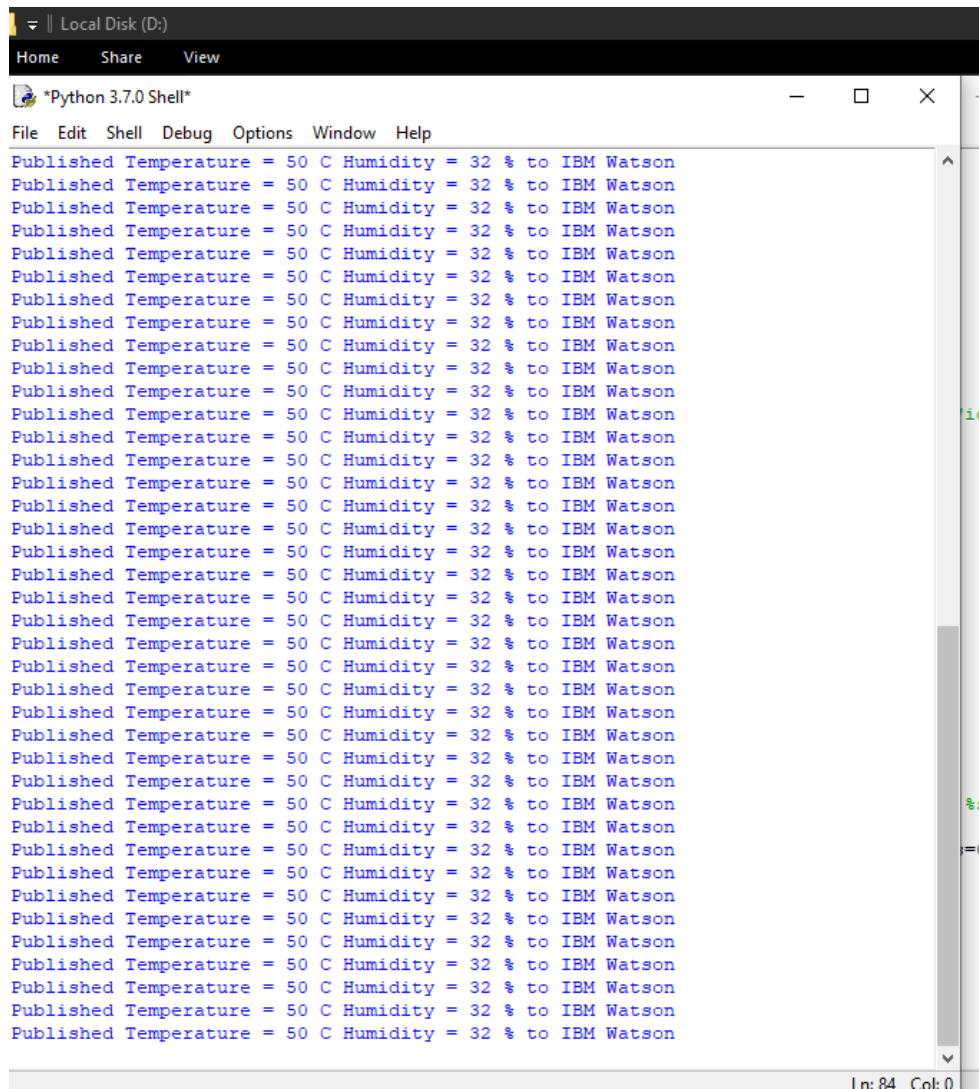


DEVELOP PYTHON SCRIPT

Date	19 September 2022
Team ID	PNT2022TMID45392
Project Name	SMART WASTE MANAGEMENT SYSTEM IN METROPOLITAN CITIES

DEVELOP A PYTHON SCRIPT



The screenshot shows a Windows File Explorer window titled "Local Disk (D:)" with tabs for Home, Share, and View. Below it is a Python 3.7.0 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The shell window displays 30 identical lines of code in blue text:

```
Published Temperature = 50 C Humidity = 32 % to IBM Watson
```

The code is repeated 30 times. The status bar at the bottom right of the shell window shows "Ln: 84 Col: 0".

File Edit Format Run Options Window Help

```

#Provide your IBM Watson Device Credentials
organization = "tubusr"
deviceType = "Evangel151"
deviceId = "trainingid"
authMethod = "token"
authToken = "vqHfrv0*Jf3RB5hoJ8"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='lighton':
        print("LIGHT ON")
    elif cmd.data['command'] == 'lightoff':
        print("LIGHT OFF")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:
    T=50;
    H=32;
    #Send Temperature & Humidity to IBM Watson
    data = { 'Temperature': T, 'Humidity': H }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "to IBM Watson")

    success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Ln:1 Col:0

PROGRAM:

```

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "i3869j"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

```

```

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud
as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(90,110)
    Humid=random.randint(60,100)

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s
%%" % Humid, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTf")
            time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```