Develop a python script

Team ID	PNT2022TMID48693
Project Name	Smart waste management system for metropolitan cities

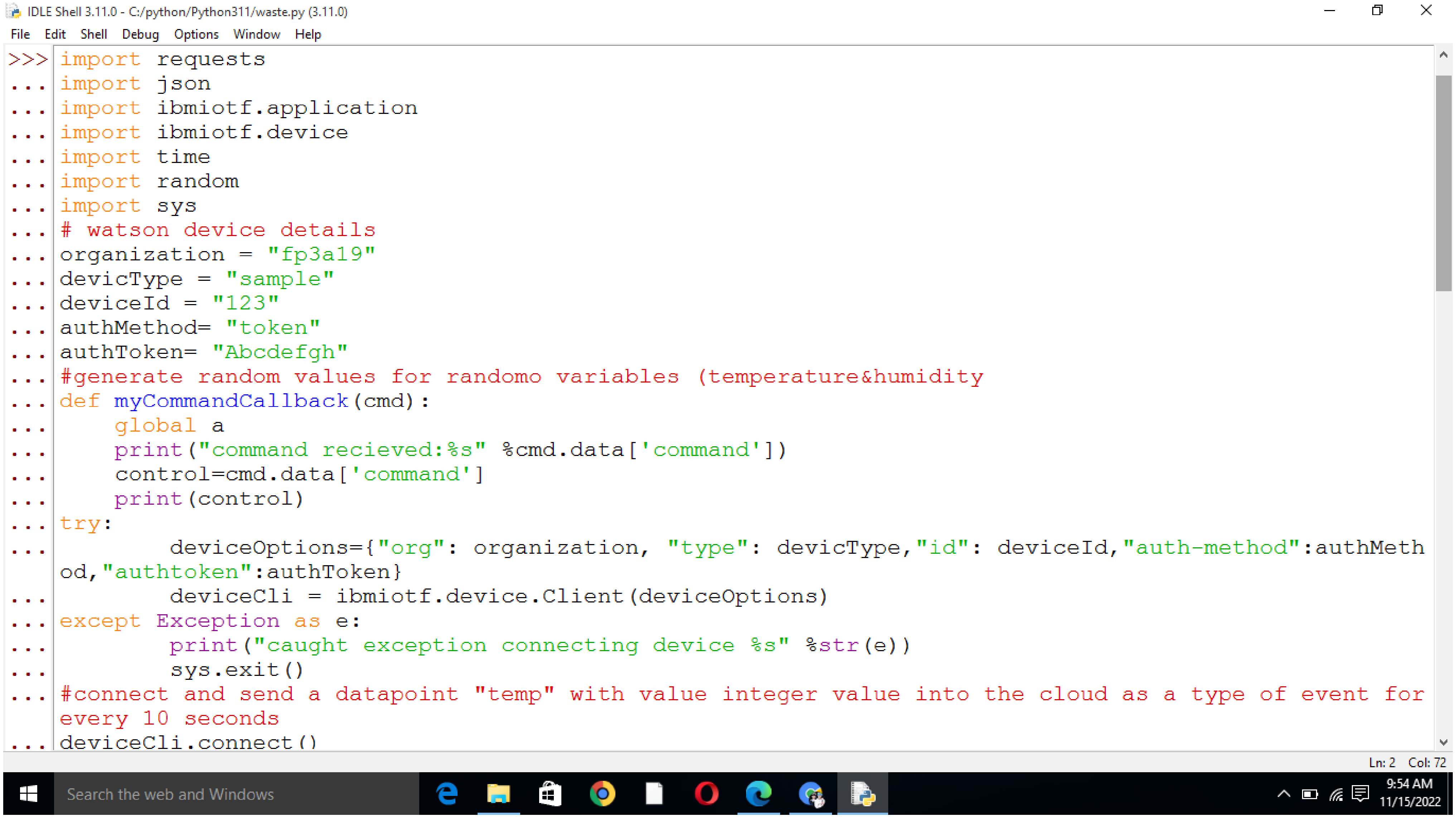
Python script

```
import requests
import json
import
ibmiotf.application
import ibmiotf.device
import time
import random
import sys

 # watson device details

organization =
"4yi0vc" devicType =
            "BIN1"
deviceId = "BIN1ID"
authMethod= "token"
authToken= "123456789"

 #generate random values for randomo variables (temperature&humidity)
```

```
def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try
:       deviceOptions={"org": organization, "type": devicType,"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 "temp" with value integer value into the cloud as a type of event for every
10 seconds deviceCli.connect()

while True:

    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
        load = "90 %"

    elif loadcell < 8 and loadcell >
          12: load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if distance < 15:
          dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'


    elif distance < 40 and distance >16:
          dist = 'Risk warning:' 'dumpster is above 60%'

    elif distance < 60 and distance > 41:
          dist = 'Risk warning:' '40 %'
    else
    :     dist = 'Risk warning:' '17 %'




    if load == "90 %" or distance == "90 %":
          warn  = 'alert :' ' Dumpster poundage getting high, Time to

    collect :)' elif load == "60 %" or distance == "60 %":

          warn = 'alert :' 'dumpster is above
    60%' else :
          warn = 'alert :' 'No need to collect right now
    ' def
    myOnPublishCallback(lat=10.678991,long=78.177731):
        print("Gandigramam, Karur")
        print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s " %long,"lat =
        %s" %lat) print(load)
        print(dist)
        print(warn)

    time.sleep(10)
```

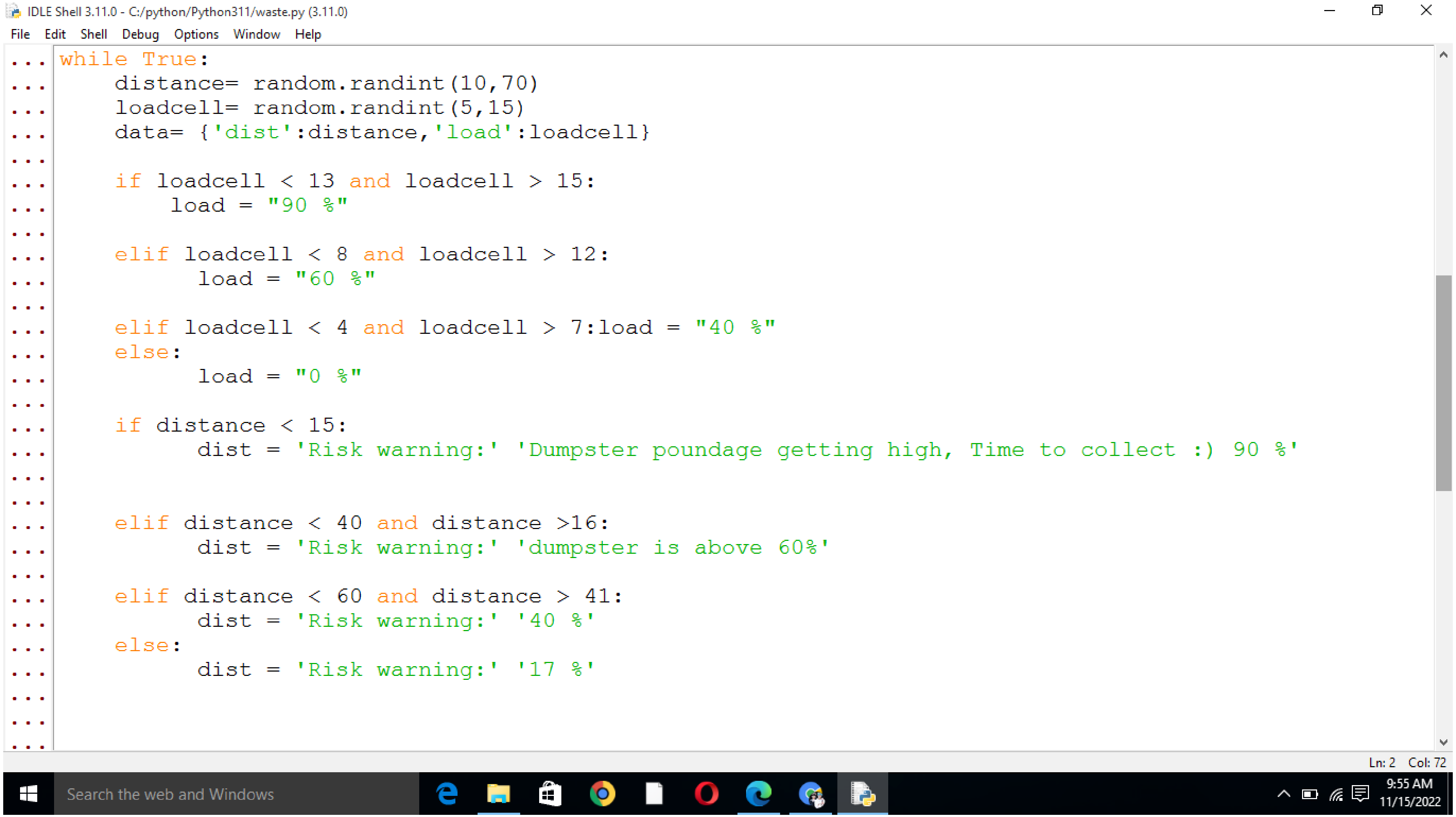
```
 success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback

 success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(30)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect
```

Screenshots Python script:

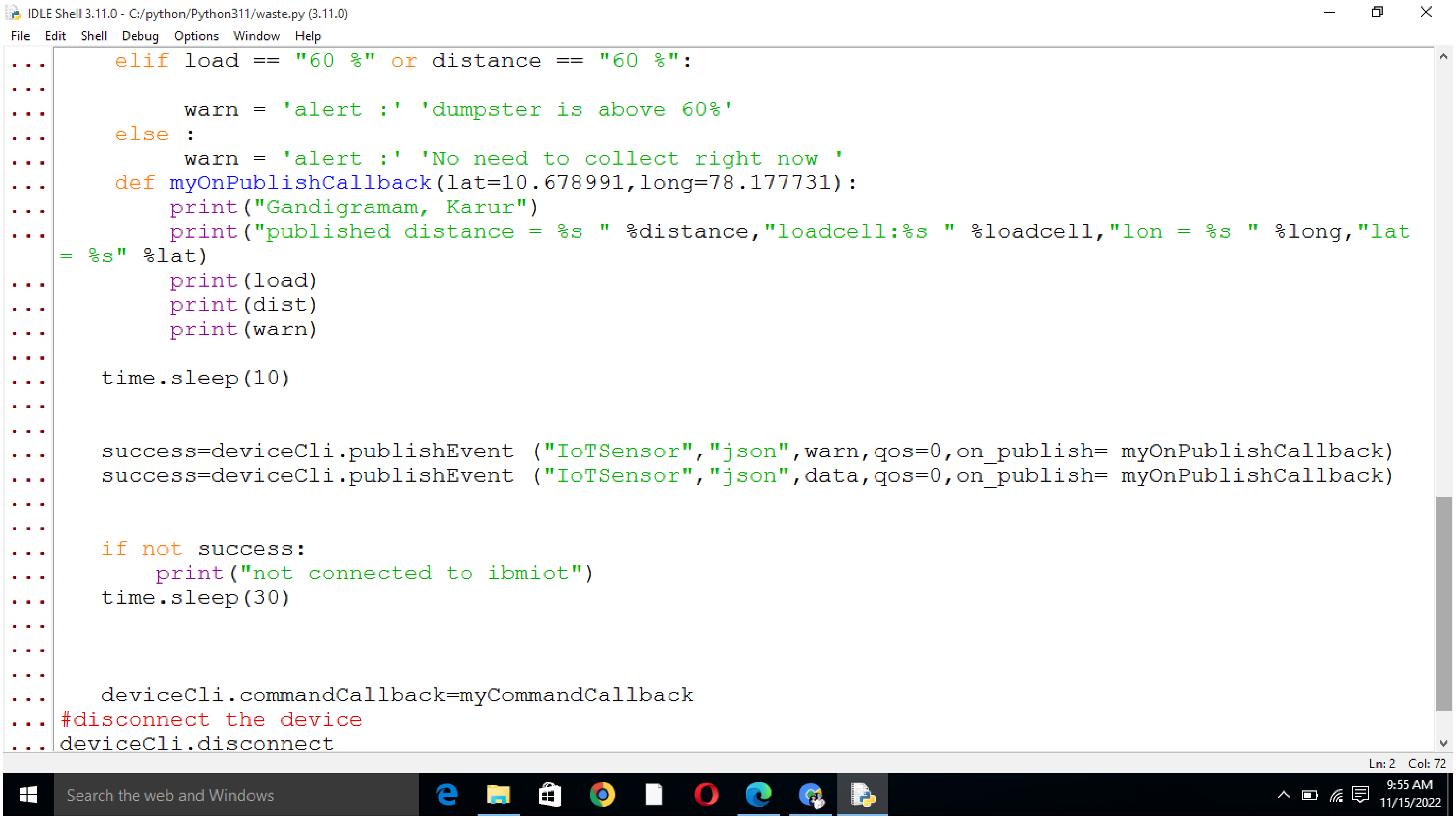


IDLE Shell 3.11.0 - C:/python/Python311/waste.py (3.11.0)
File Edit Shell Debug Options Window Help
Ln: 2 Col: 72
Search the web and Windows
9:54 AM
11/15/2022

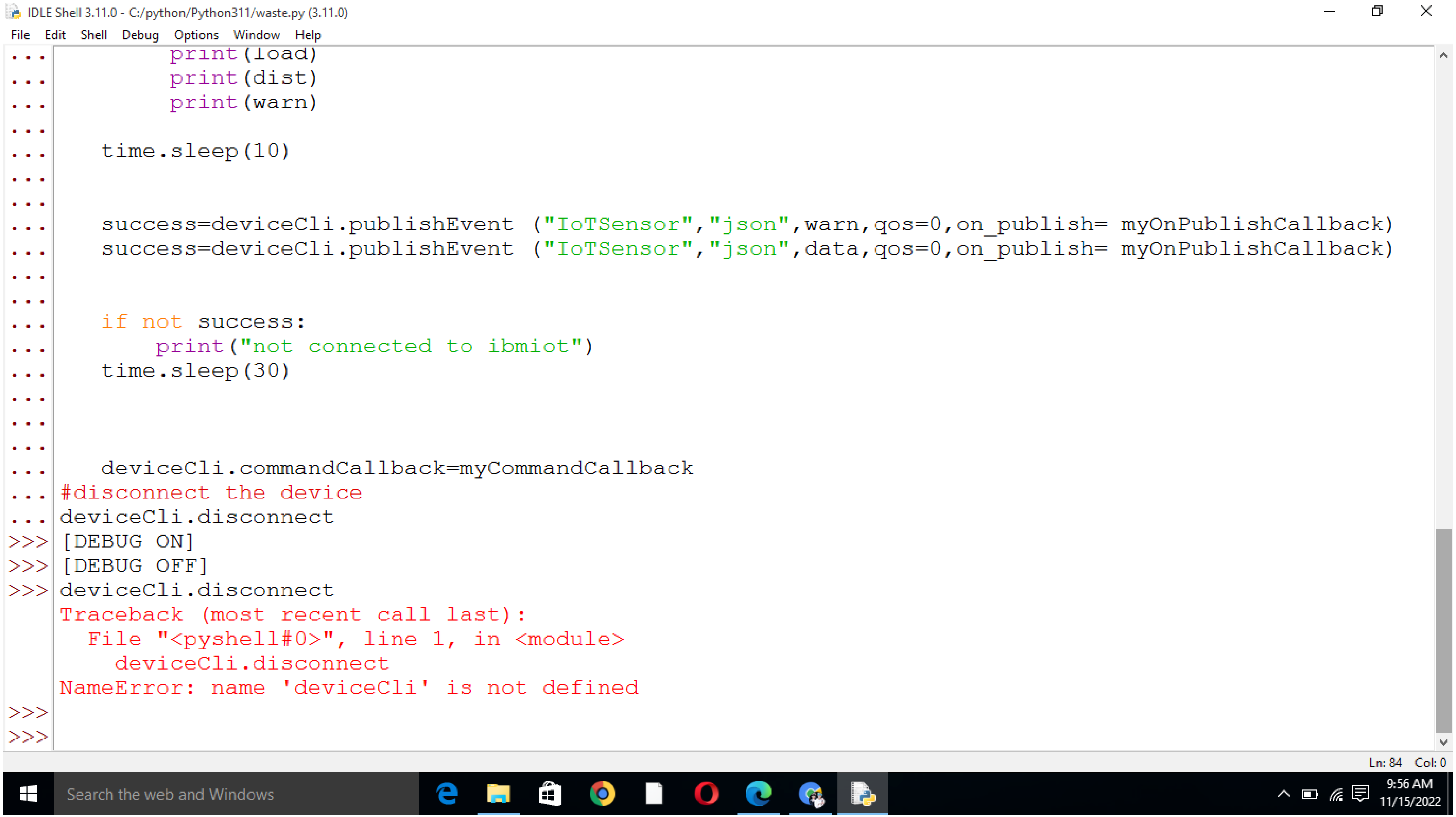
```
>>> import requests
... import json
... import ibmiotf.application
... import ibmiotf.device
... import time
... import random
... import sys
... # watson device details
... organization = "fp3a19"
... devicType = "sample"
... deviceId = "123"
... authMethod= "token"
... authToken= "Abcdefgh"
... #generate random values for randomo variables (temperature&humidity
... def myCommandCallback(cmd):
...     global a
...     print("command recieved:%s" %cmd.data['command'])
...     control=cmd.data['command']
...     print(control)
... try:
...         deviceOptions={"org": organization, "type": devicType,"id": deviceId,"auth-method":authMeth
od,"authtoken":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 "temp" with value integer value into the cloud as a type of event for
every 10 seconds
... deviceCli.connect()
```


IDLE Shell 3.11.0 - C:/python/Python311/waste.py (3.11.0)
File
Edit
Shell
Debug
Options
Window
Help
Ln: 2
Col: 72
Search the web and Windows
9:55 AM
11/15/2022

```
... while True:
...     distance= random.randint(10,70)
...     loadcell= random.randint(5,15)
...     data= {'dist':distance,'load':loadcell}
...
...     if loadcell < 13 and loadcell > 15:
...         load = "90 %"
...
...     elif loadcell < 8 and loadcell > 12:
...           load = "60 %"
...
...     elif loadcell < 4 and loadcell > 7:load = "40 %"
...     else:
...           load = "0 %"
...
...     if distance < 15:
...           dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'
...
...
...     elif distance < 40 and distance >16:
...           dist = 'Risk warning:' 'dumpster is above 60%'
...
...     elif distance < 60 and distance > 41:
...           dist = 'Risk warning:' '40 %'
...     else:
...           dist = 'Risk warning:' '17 %'
...
...
...
```

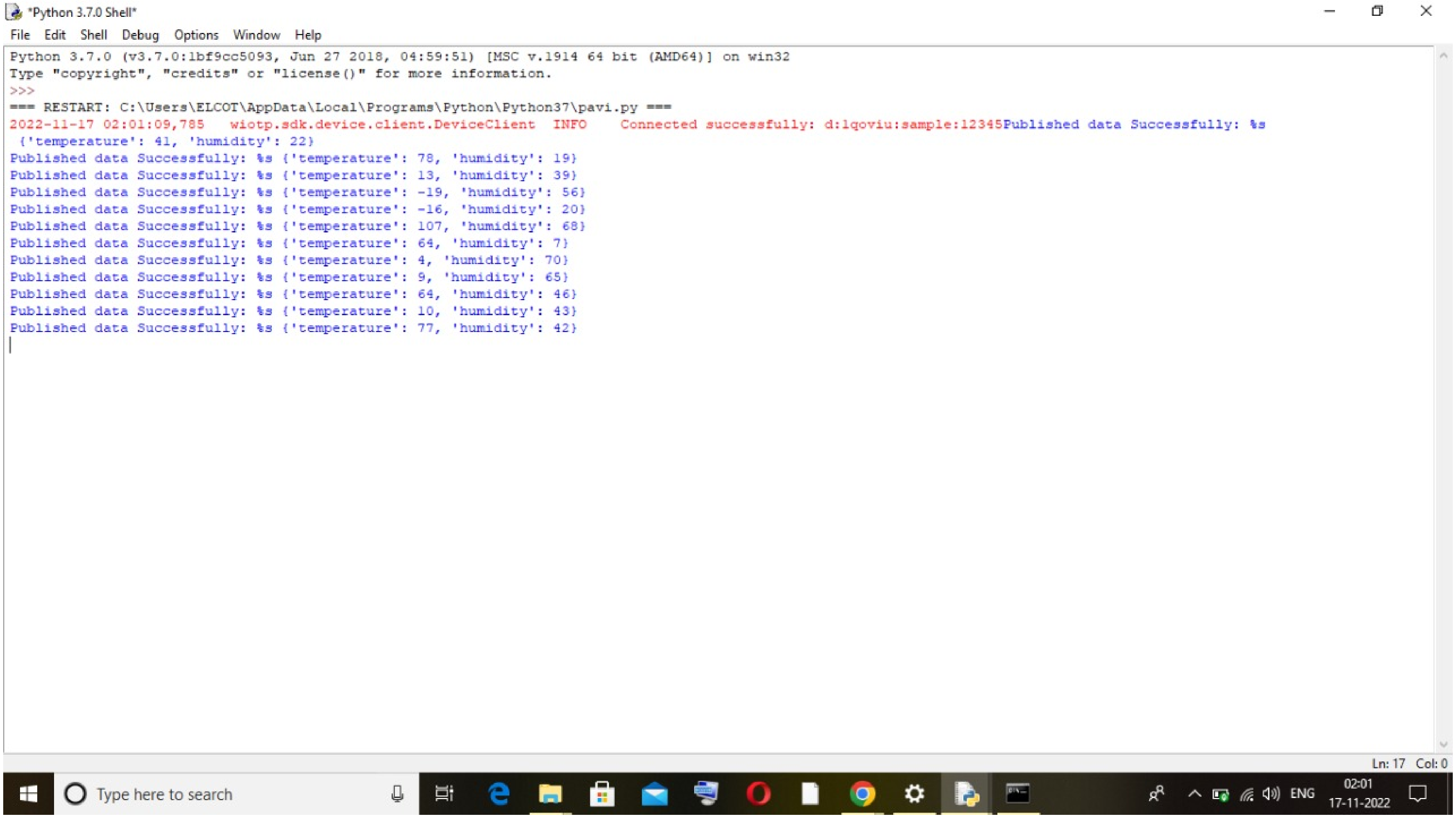

IDLE Shell 3.11.0 - C:/python/Python311/waste.py (3.11.0)
File
Edit
Shell
Debug
Options
Window
Help
Ln: 2
Col: 72
Search the web and Windows
9:55 AM
11/15/2022

```
...     elif load == "60 %" or distance == "60 %":
... 
...          warn = 'alert :' 'dumpster is above 60%'
...     else :
...          warn = 'alert :' 'No need to collect right now '
...     def myOnPublishCallback(lat=10.678991,long=78.177731):
...         print("Gandigramam, Karur")
...         print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s " %long,"lat
= %s" %lat)
...         print(load)
...         print(dist)
...         print(warn)
... 
...    time.sleep(10)
... 
... 
...    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)
...    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)
... 
... 
...    if not success:
...        print("not connected to ibmiot")
...    time.sleep(30)
... 
... 
... 
...    deviceCli.commandCallback=myCommandCallback
... #disconnect the device
... deviceCli.disconnect
```



IDLE Shell 3.11.0 - C:/python/Python311/waste.py (3.11.0)
File Edit Shell Debug Options Window Help
Ln: 84 Col: 0
Search the web and Windows
9:56 AM
11/15/2022

```
...         print(load)
...         print(dist)
...         print(warn)
...
...    time.sleep(10)
...
...
...    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)
...    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)
...
...
...    if not success:
...        print("not connected to ibmiot")
...    time.sleep(30)
...
...
...
...    deviceCli.commandCallback=myCommandCallback
... #disconnect the device
... deviceCli.disconnect
>>> [DEBUG ON]
>>> [DEBUG OFF]
>>> deviceCli.disconnect
    Traceback (most recent call last):
      File "<pyshell#0>", line 1, in <module>
        deviceCli.disconnect
    NameError: name 'deviceCli' is not defined
>>>
>>>
```



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=== RESTART: C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\pavi.py ===
2022-11-17 02:01:09,785   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:lqoviu:sample:12345Published data Successfully: %s
 {'temperature': 41, 'humidity': 22}
Published data Successfully: %s {'temperature': 78, 'humidity': 19}
Published data Successfully: %s {'temperature': 13, 'humidity': 39}
Published data Successfully: %s {'temperature': -19, 'humidity': 56}
Published data Successfully: %s {'temperature': -16, 'humidity': 20}
Published data Successfully: %s {'temperature': 107, 'humidity': 68}
Published data Successfully: %s {'temperature': 64, 'humidity': 7}
Published data Successfully: %s {'temperature': 4, 'humidity': 70}
Published data Successfully: %s {'temperature': 9, 'humidity': 65}
Published data Successfully: %s {'temperature': 64, 'humidity': 46}
Published data Successfully: %s {'temperature': 10, 'humidity': 43}
Published data Successfully: %s {'temperature': 77, 'humidity': 42}
```

CONCLUSION:

Hence we created the python script.