

Develop a python script

Team ID	PNT2022TMID14976
Project Name	Project :Smart Waste Management System for Metropolitan Cities

Step 1: Open python idle Step2: Type the program Step 3: Then click on file and save the document Step 4: Then click on Run then Run Module Step 5: output will be appeared in the idle window

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

```
#Provide your IBM Watson Device Credentials
organization = "up54lj"
deviceType = "Microcontroller"
deviceId = "Device07"
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")
    else :
        print ("led is off")
```

```
#print(cmd)
```

```

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

    level=random.randint(0,100)
    load=random.randint(5,100)

    data = { 'level' : level, 'load': load }
    #print data
    def myOnPublishCallback():
        print ("Published GARBAGE LEVEL = %s C" % level, "LOAD = %s %%" %
load)

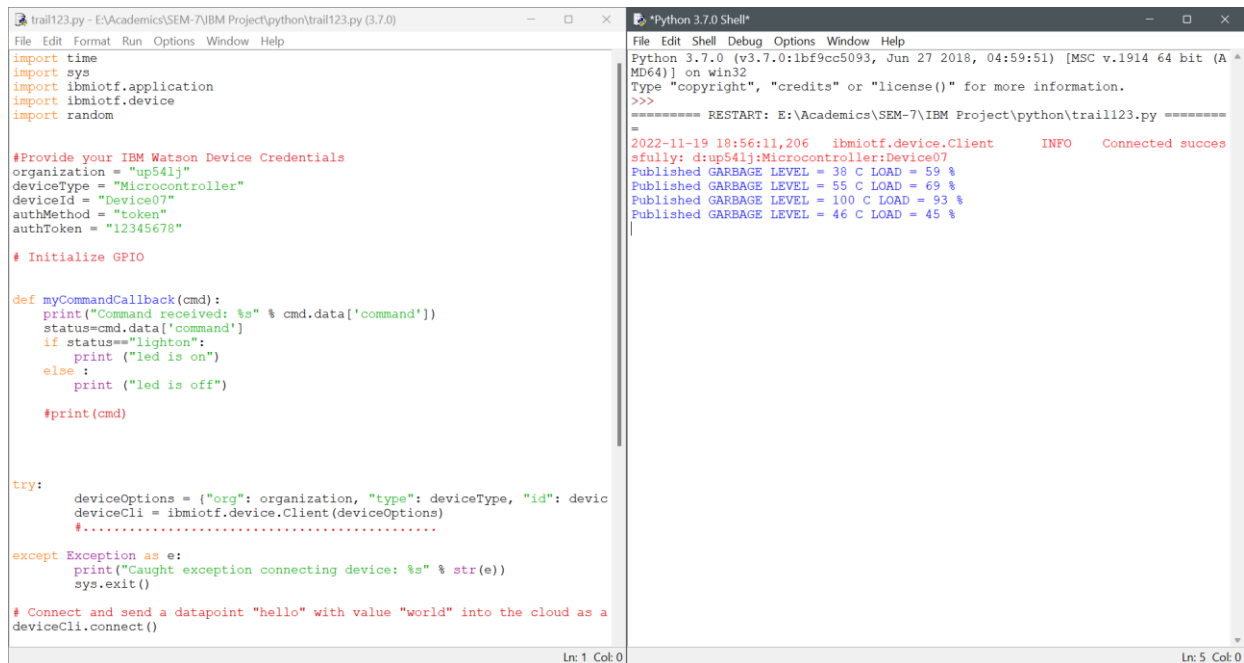
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(2)

    deviceCli.commandCallback = myCommandCallback

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

```

Screenshots Python script:



The image shows two side-by-side windows from a Windows operating system. The left window is a text editor titled 'trail123.py - E:\Academics\SEM-7\IBM Project\python\trail123.py (3.7.0)'. It contains a Python script that imports modules, sets up IBM Watson IoT credentials, initializes GPIO, and defines a command callback function. The script attempts to connect to a device and send a datapoint. The right window is a 'Python 3.7.0 Shell' titled 'Python 3.7.0 Shell'. It shows the execution of the script, including the restart command, the connection status 'Connected successfully', and the resulting datapoint values for temperature, load, and garbage levels.

```
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "up54lj"
deviceType = "Microcontroller"
deviceId = "Device07"
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")
    else :
        print ("led is off")

    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId}
    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 a
deviceCli.connect()
```

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: E:\Academics\SEM-7\IBM Project\python\trail123.py =====
2022-11-19 18:56:11,206 ibmiotf.device.Client INFO Connected successfully: d:up54lj:Microcontroller:Device07
Published GARBAGE LEVEL = 38 C LOAD = 59 %
Published GARBAGE LEVEL = 55 C LOAD = 69 %
Published GARBAGE LEVEL = 100 C LOAD = 93 %
Published GARBAGE LEVEL = 46 C LOAD = 45 %
|

Ln: 1 Col: 0

Ln: 5 Col: 0

