

## Develop A Python Script

Team ID	PNT2022TMID16088
Project Name	Smart Waste Management System

**Step 1:** Open python idle

**Step 2:** 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

### Python Script :

```
import requests
```

```
import json
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import time
```

```
import random
```

```
import sys
```

```
# watson device details
```

```
organization = "RMK Engineering College"
```

```
devicType = "NodeMCU"
```

```
deviceId = "4076"
```

```
authMethod= "token"
```

```
authToken= "zs4P1axSjkUg+0QG-("
```

```
#generate random values for random 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,"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()

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 %":
```

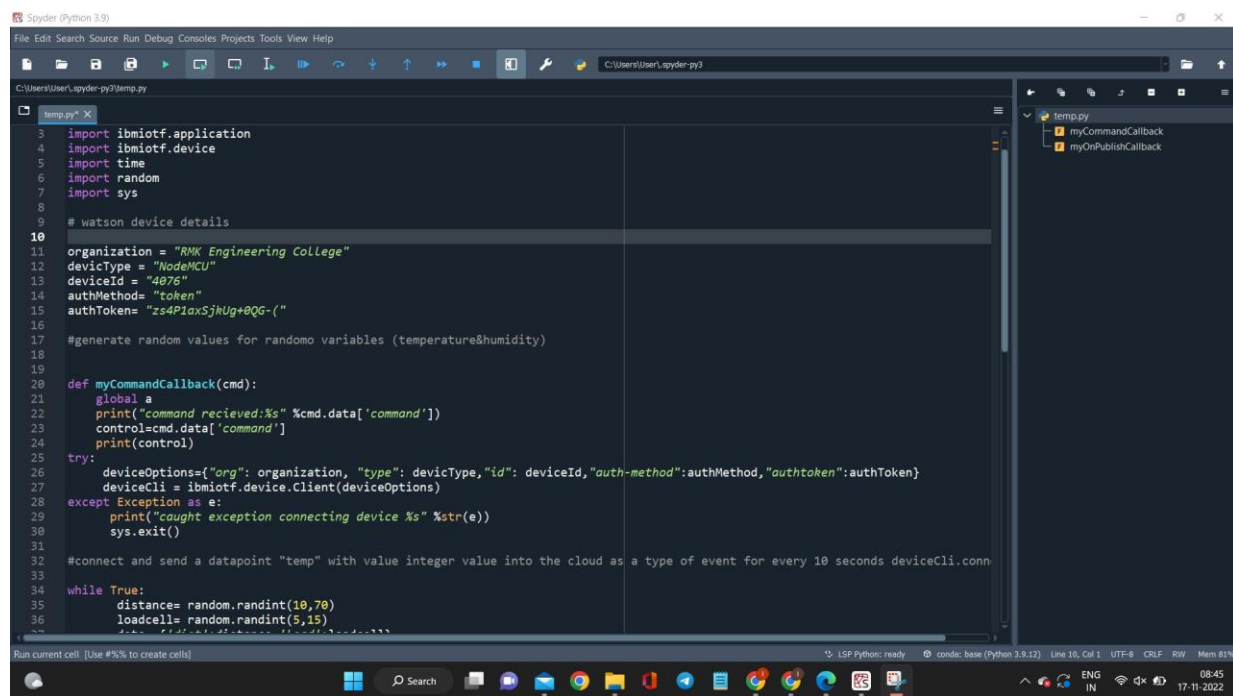
```

        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

```



The screenshot shows the Spyder Python IDE interface. The main editor displays a Python script named 'temp.py' with the following code:

```

3  import ibmiotf.application
4  import ibmiotf.device
5  import time
6  import random
7  import sys
8
9  # watson device details
10
11 organization = "RMK Engineering College"
12 deviceType = "NodeMCU"
13 deviceId = "4876"
14 authMethod= "token"
15 authToken= "zs4P1axSjhUg+0QG-"
16
17 #generate random values for random variables (temperature&humidity)
18
19
20 def myCommandCallback(cmd):
21     global a
22     print("command recieved:%s" %cmd.data['command'])
23     control=cmd.data['command']
24     print(control)
25 try:
26     deviceOptions={"org": organization, "type": deviceType,"id": deviceId,"auth-method":authMethod,"authtoken":authToken}
27     deviceCli = ibmiotf.device.Client(deviceOptions)
28 except Exception as e:
29     print("caught exception connecting device %s" %str(e))
30     sys.exit()
31
32 #connect and send a datapoint "temp" with value integer value into the cloud as a type of event for every 10 seconds deviceCli.conn
33
34 while True:
35     distance= random.randint(10,70)
36     loadcell= random.randint(5,15)

```

The right sidebar shows the file explorer with the following structure:

- temp.py
  - myCommandCallback
  - myOnPublishCallback

The status bar at the bottom indicates the current cell is ready, the conda base (Python 3.9.12) environment is active, and the file is at Line 10, Col 1. The system clock shows 08:45 on 17-11-2022.

```

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 %'

    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)

```

```

        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

```