

## Developing A Python Script

|               |  |
|---------------|--|
| Date          | 16 November 2022                                       |
| Team ID       | PNT2022TMID33206                                       |
| Project Name  | IoT Based Smart Crop Protection System for Agriculture |
| Maximum Marks | 4 Marks  |

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

#Provide your IBM Watson Device Credentials
organization = "us27lh"
deviceType = "CROP"
deviceId = "KEERTHIKA123"
authMethod = "token"
authToken = "keekee123"

# 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 IoTTF")
```

```
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

The screenshot displays the IBM Watson IoT Platform dashboard on the left and a Python 3.7.0 Shell terminal window on the right.

**IBM Watson IoT Platform Dashboard:**

- Header:** IBM Watson IoT Platform, ID: us27lh
- Navigation:** Browse, Action, Device Types, Interfaces, Add Device (+)
- Search:** Search by Device ID, Device Simulator (toggle), Filter icon
- Table:**

| Device ID    | Status    | Device Type | Class ID | Date Added      |
|--------------|-----------|-------------|----------|-----------------|
| KEERTHIKA123 | Connected | CROP        | Device   | Oct 31, 2021 PM |

- Identity:** Device Information, Recent Events, State, Location
- Message:** The recent events listed show the live stream of data that is coming and going from the device.
- Table:**

| Event     | Value                   | Format |
|-----------|-------------------------|--------|
| IoTSensor | {"temp":106,"Humid":75} | json   |
| IoTSensor | {"temp":100,"Humid":79} | json   |
| IoTSensor | {"temp":92,"Humid":90}  | json   |

**Python 3.7.0 Shell:**

```
File Edit Shell Debug Options Window Help
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: D:/ibm python/python script.py =====
2022-11-16 19:52:33,189 ibmiotf.device.Client INFO Connected successfully: d:us27lh:
h:CROP:KEERTHIKA123
Published Temperature = 93 C Humidity = 76 % to IBM Watson
Published Temperature = 95 C Humidity = 99 % to IBM Watson
Published Temperature = 92 C Humidity = 90 % to IBM Watson
Published Temperature = 100 C Humidity = 79 % to IBM Watson
Published Temperature = 106 C Humidity = 75 % to IBM Watson
```