

Python Script

```
File Edit Format Run Options Window Help
#Importing Random function to generate the value and required IoT and System Libraries
import random as rand
import time
import ibmiotf.application
import ibmiotf.device
import sys

#defining credentials of device
organization = "0sqd61"
deviceType = "Surya"
deviceId = "surya7"
authMethod = "token"
authToken = "TqQ7dQl6)1I9CXx6!"

# Initialize GPIO
# code to activate the motor comes here in Sprint 4
def myCommandCallback(cmd):
    # Command Call back
    print("Command received: %s" % cmd.data['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()

deviceCli.connect()

while True:

    latArr = [11.55688,11.55635,11.55635,11.55186,11.55186]
    lonArr = [78.00328,78.00549,78.00549,77.99586,77.99586]

    latitude= rand.choice(latArr)
    longitude= rand.choice(lonArr)

    data = {"Latitude": latitude,
            "Longitude": longitude
            }
    #out area location

Ln: 27 Col: 0
```

```
File Edit Format Run Options Window Help
# Initialize GPIO
# code to activate the motor comes here in Sprint 4
def myCommandCallback(cmd):
    # Command Call back
    print("Command received: %s" % cmd.data['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()

deviceCli.connect()

while True:

    latArr = [11.55688,11.55635,11.55635,11.55186,11.55186]
    lonArr = [78.00328,78.00549,78.00549,77.99586,77.99586]

    latitude= rand.choice(latArr)
    longitude= rand.choice(lonArr)

    data = {"Latitude": latitude,
            "Longitude": longitude
            }

    #printing the values
    def myOnPublishCallback():
        print("Published all data to IBM Watson")

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

    deviceCli.commandCallback = myCommandCallback

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

Ln: 45 Col: 0
```