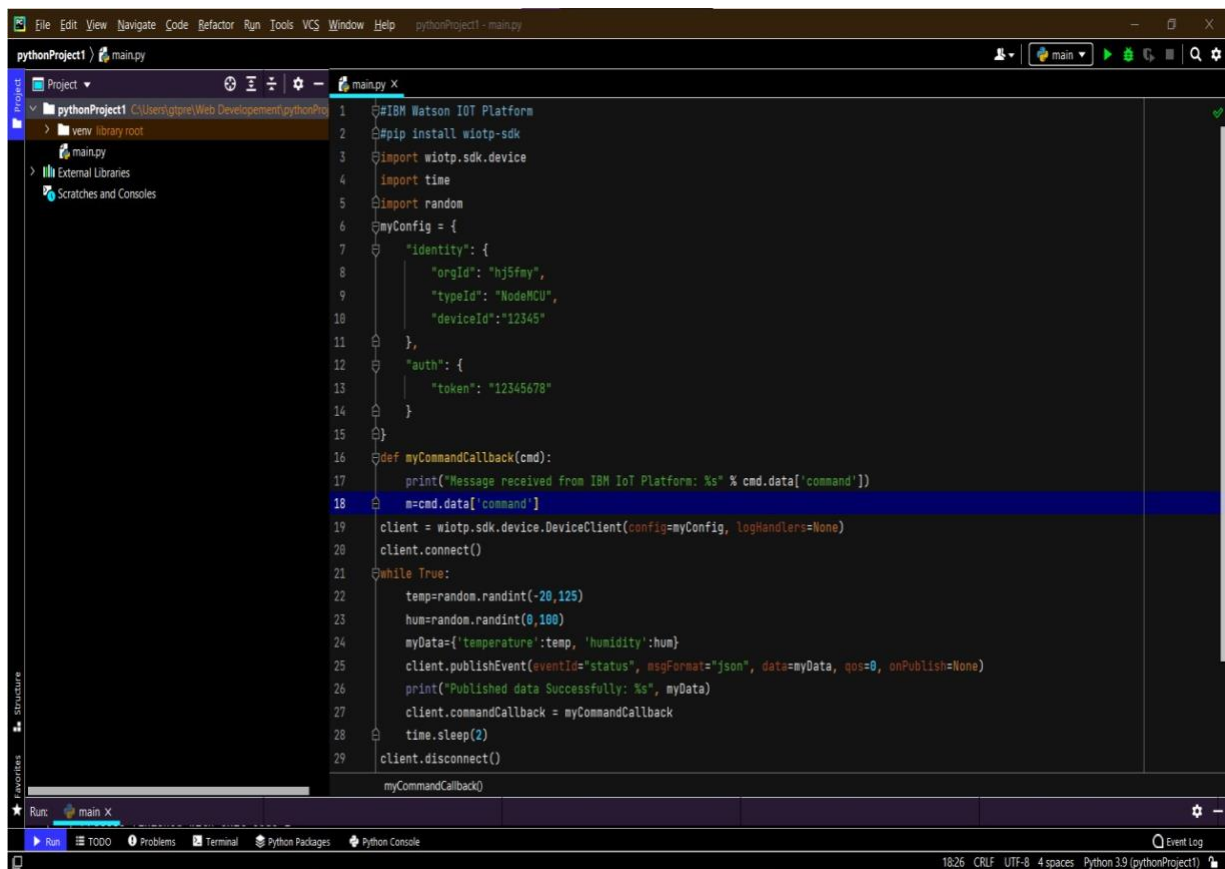


Develop a Python Script



The screenshot shows a code editor with a Python script for connecting to the IBM Watson IoT Platform. The script is named `main.py` and is located in a project named `pythonProject1`. The script includes comments for installing the `wiotp-sdk` and importing the necessary modules. It defines a configuration dictionary `myConfig` with identity and authentication details. A callback function `myCommandCallback` is defined to handle incoming commands. The main logic consists of creating a `DeviceClient`, connecting to the platform, and entering a loop where it publishes random temperature and humidity data every 2 seconds.

```
1 #IBM Watson IoT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
7     "identity": {
8         "orgId": "hj5fmy",
9         "typeId": "NodeMCU",
10        "deviceId": "12345"
11    },
12    "auth": {
13        "token": "12345678"
14    }
15 }
16 def myCommandCallback(cmd):
17     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
18     m=cmd.data['command']
19 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
20 client.connect()
21 while True:
22     temp=random.randint(-20,125)
23     hum=random.randint(0,100)
24     myData={'temperature':temp, 'humidity':hum}
25     client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
26     print("Published data Successfully: %s", myData)
27     client.commandCallback = myCommandCallback
28     time.sleep(2)
29 client.disconnect()
30 myCommandCallback)
```