ibm iot.py - C:\Users\subis\AppData\Local\Programs\Python\Python37\ibm iot.py (3.7.0)

File Edit Format Run Options Window Help

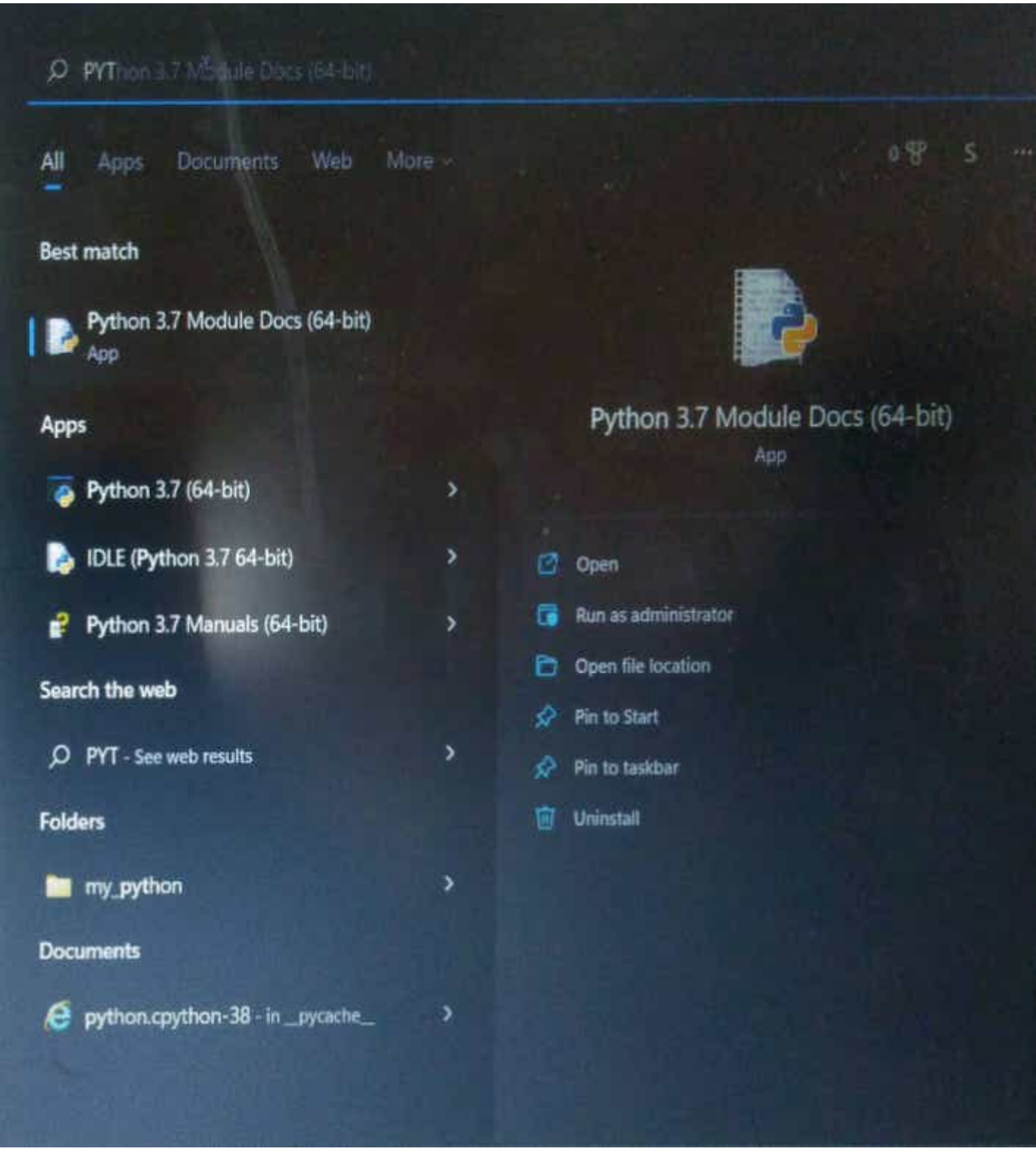
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "nym2po"
deviceType = "weather_device"
deviceId = "weather_today"
authMethod = "token"
authToken = "4c9LEolzH!97!juTxI"
#Initialize GPIO
temp=random.randint(0,100)
pulse=random.randint(0,100)
oxygen=random.randint(0,100)
lat= 17
lon=18


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data[ 'command'])
    print(cmd)



try:
      deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"ayth-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
       data = {"d":{ 'temp' : temp, 'pulse' : pulse , 'oxygen' : oxygen,"lat":lat,"lon":lon}}
       #print data
       def myonPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")
       success = devicecli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
       if not success:
          print("Not connected to IOTF")
```

ibm iot.py - C:\Users\subis\AppData\Local\Programs\Python\Python37\ibm iot.py (3.7.0)

File Edit Format Run Options Window Help

```
organization = "nym2po"
deviceType = "weather_device"
deviceId = "weather_today"
authMethod = "token"
authToken = "4c9LEolzH!97!juTxI"
#Initialize GPIO
temp=random.randint(0,100)
pulse=random.randint(0,100)
oxygen=random.randint(0,100)
lat= 17
lon=18


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data[ 'command'])
    print(cmd)



try:
       deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"ayth-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
       data = {"d":{ 'temp' : temp, 'pulse' : pulse , 'oxygen' : oxygen,"lat":lat,"lon":lon}}
       #print data
       def myonPublishCallback():
           print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")
       success = devicecli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
       if not success:
           print("Not connected to IOTF")
       time.sleep(1)

       devicecli.commandcallback = myCommandCallback

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

```
e  Edit  Format  Run  Options  Window  Help
ganization = "nym2po"
viceType = "weather_device"
viceId = "weather_today"
thMethod = "token"
thToken = "4c9LEolzH197!juTxI"
nitialize GPIO
mp=random.randint(0,100)
lse=random.randint(0,100)
ygen=random.randint(0,100)
t= 17
n=18


f myCommandCallback(cmd):
   print("Command received: %s" % cmd.data[ 'command'])
   print(cmd)



y:
      deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"ayth-method": authMethod,"auth-token":authtoken}
      deviceCli  = ibmiotf.device.Client(deviceOptions)
      #.....................................................

cept 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
eviceCli.connect()

hile True:
      #Get Sensor Data from DHT11
      data = {"d":{ 'temp' : temp, 'pulse' : pulse , 'oxygen' : oxygen,"lat":lat,"lon":lon}}
      #print data
      def myOnPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")
      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
      if not success:
          print("Not connected to IOTF")
      time.sleep(1)

      deviceCli.commandCallback = myCommandCallback

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


PYThon 3.7 Module Docs (64-bit)
All
Apps
Documents
Web
More
Best match
Python 3.7 Module Docs (64-bit)
App
Apps
Python 3.7 (64-bit)
IDLE (Python 3.7 64-bit)
Python 3.7 Manuals (64-bit)
Search the web
PYT - See web results
Folders
my_python
Documents
python.cpython-38 - in __pycache__
Python 3.7 Module Docs (64-bit)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall