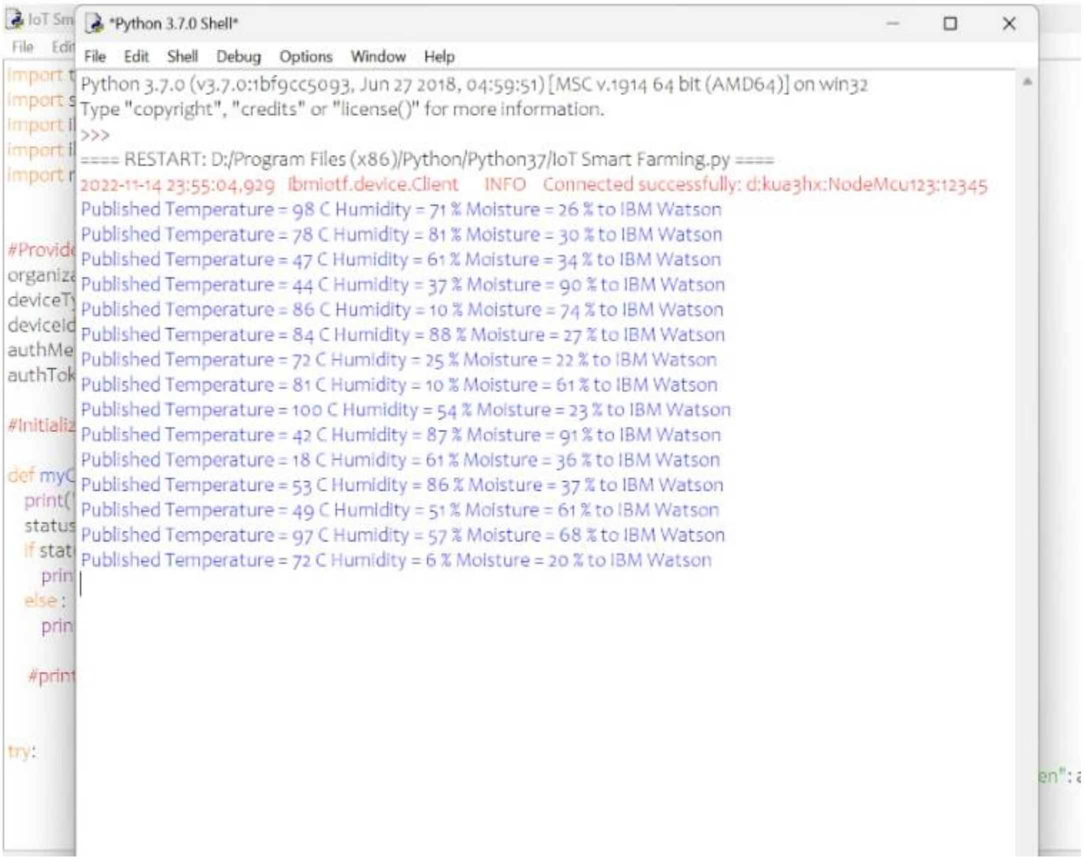
PYTHON SCRIPT:



```
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:/Program Files (x86)/Python/Python37/IoT Smart Farming.py ====
2022-11-14 23:55:04,929  ibmiotf.device.Client     INFO    Connected successfully: d:kua3hx:NodeMcu123:12345
Published Temperature = 98 C Humidity = 71 % Moisture = 26 % to IBM Watson
Published Temperature = 78 C Humidity = 81 % Moisture = 30 % to IBM Watson
Published Temperature = 47 C Humidity = 61 % Moisture = 34 % to IBM Watson
Published Temperature = 44 C Humidity = 37 % Moisture = 90 % to IBM Watson
Published Temperature = 86 C Humidity = 10 % Moisture = 74 % to IBM Watson
Published Temperature = 84 C Humidity = 88 % Moisture = 27 % to IBM Watson
Published Temperature = 72 C Humidity = 25 % Moisture = 22 % to IBM Watson
Published Temperature = 81 C Humidity = 10 % Moisture = 61 % to IBM Watson
Published Temperature = 100 C Humidity = 54 % Moisture = 23 % to IBM Watson
Published Temperature = 42 C Humidity = 87 % Moisture = 91 % to IBM Watson
Published Temperature = 18 C Humidity = 61 % Moisture = 36 % to IBM Watson
Published Temperature = 53 C Humidity = 86 % Moisture = 37 % to IBM Watson
Published Temperature = 49 C Humidity = 51 % Moisture = 61 % to IBM Watson
Published Temperature = 97 C Humidity = 57 % Moisture = 68 % to IBM Watson
Published Temperature = 72 C Humidity = 6 % Moisture = 20 % to IBM Watson
```

CODE:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "kua3hx"

deviceType = "NodeMcu123"

deviceId = "12345"

authMethod = "token"

authToken = "1234567890"
```

```
#Initialize GPIO
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="Motor ON":
 print ("Motor is ON")
 else :
 print ("Motor is OFF")

 #print(cmd)

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(0,100)
 Humid=random.randint(0,100)
 Moist=random.randint(0,100)


 data = { 'temperature' : temp, 'humidity': Humid , 'moisture': Moist}
 #print data
 def myOnPublishCallback():
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,
"Moisture = %s %%" % Moist, "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 the application from the cloud
deviceCli.disconnect()
```