

Python Code

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

#provide your IBM Watson Device Credentials
organization = "pol6f4"
deviceType = "ESP-32IoT"
deviceId = "100100C40A24"
authMethod = "EnuF+Tgx40@Y!"
authToken = ""

#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 data point "hello" with value "world" into the cloud
deviceCli.connect()

while True:
    #Get sensor data from DBT11
    temp=random.randint(90,110)
    Hunid=random.randint(60,100)

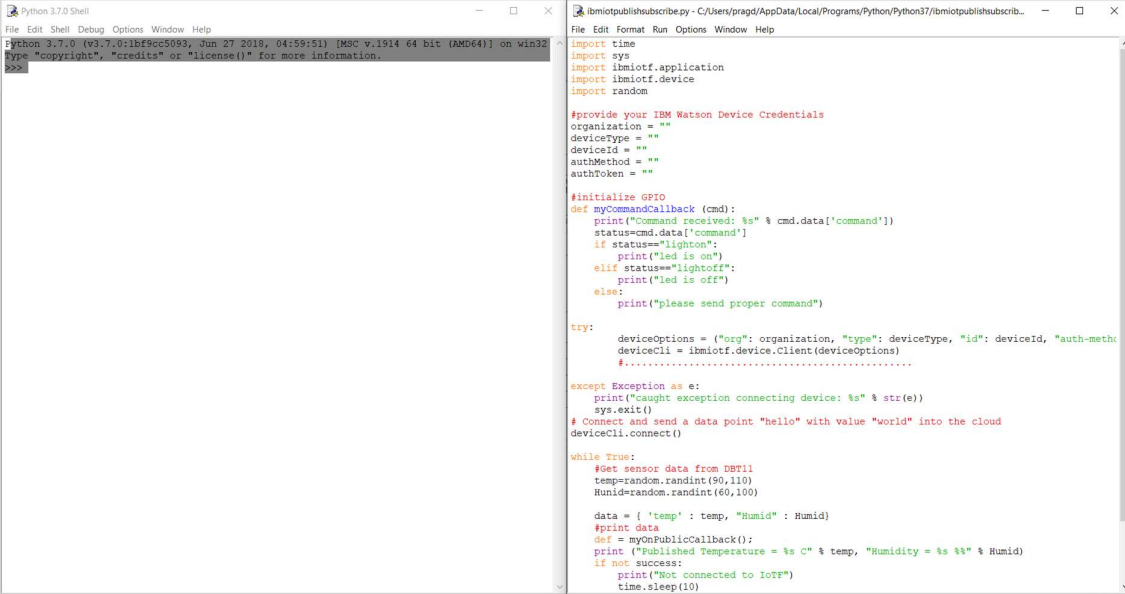
    data = { 'temp' : temp, "Humid" : Humid}
    #print data
    def = myOnPublicCallback();
    print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid)
```

```
if not success:
    print("Not connected to IoTf")
    time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device
devCli.disconnect()
```

Code using Python IDLE



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:bf5f50083, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

ibmiotpublishsubscribe.py - C:/Users/pragd/AppData/Local/Programs/Python/Python37/ibmiotpublishscrib...
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 = ""
deviceType = ""
deviceId = ""
authMethod = ""
authToken = ""

#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-meth": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    # Connect and send a data point "hello" with value "world" into the cloud
    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 myOnPublicCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %% " % Humid)
    if not success:
        print("Not connected to IoTf")
        time.sleep(10)
```