

# DEVELOP A PYTHON SCRIPT

Date	17 <sup>th</sup> November 2022
Team ID	PNT2022TMID47535
Project Name	Personal assistance for Seniors who are self-reliant

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

# Provide your IBM Watson Device Credentials
organization = "ca0tfo"
deviceType = "node"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# 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 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
    dolo = random.randint(0, 5)
    cetrecin = random.randint(0, 5)

    data = {'dolo - 26/11/22': dolo, 'cetrecin - 26/11/22': cetrecin}

    # print data
    def myOnPublishCallback():
        print("Published temperature = %s C" % dolo, "Humidity = %s %%" % cetrecin, "to IBM
Watson")

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

    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```