DEVELOP THE PYTHON CODE

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

#Provide your IBM Watson Device Credentials

organization = "vbzdj5"
deviceType = "raspberrypi"
deviceId = "12345"
authMethod = "use-token-auth"
authToken= "12345678"

#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": organisation, "type": devicetype, "id": deviceid, "auth-method":
authMethod, "auth-token": authToken}
     deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
   print("Caught exceptions 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 DHTll

       temp=random.randint(0,100)
       pulse=random.randint(0,100)
       oxygen=random.randint(0,100)
       lat=17
       lon=18


       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()
```