

## DEVELOP THE PYTHON SCRIPT

TEAM ID	PNT2022TMID02419
PROJECT TITLE	Real-time river water quality monitoring and control system

### PROGRAM:

```
File Edit Format Run Options Window Help
import random
import time
import sys
import ibmiotf.application
import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "xfptfb" # replace it with organization ID
deviceType = "NodeMCU" # replace it with device type
deviceId = "19141" # replace with device id
authMethod = "use-token-auth"
authToken = "1914137383010209" # replace with token

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == 'lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT 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()

deviceCli.connect()

while True:
    pH = random.randint(0,100)
```

Ln: 1 Col: 0

File Edit Format Run Options Window Help

```
print("LIGHT 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()

deviceCli.connect()

while True:
    pH = random.randint(0,100)
    conductivity = random.randint(0,100)
    T = random.randint(0,100)
    oxygen = random.randint(0,100)
    turbidity = random.randint(0,100)
    # Send Temperature & Humidity to IBM Watson
    data = {'temperature': T, 'ph': pH, 'conductivity': conductivity, 'oxygen': oxygen, 'turbidity': turbidity}

    # print data
    def myOnPublishCallback():
        print("Published data", data, "to IBM Watson")

    success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
```

Ln: 1 Col: 0

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

```
#Provide your IBM Watson Device Credentials
```

```
organization = "xfptfb"
deviceType = "NodeMCU"
deviceId = "19141"
authMethod = "use-token-auth"
authToken = "1914137383010209"
```

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("Light on")
    else :
        print ("Light 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
    pH=random.randint(0,100)
    conductivity=random.randint(0,100)
    T=random.randint(0,100)
    oxygen=random.randint(0,100)
    turbidity=random.randint(0,100)

    data = { 'temperature' : T, 'pH': pH, 'conductivity':
conductivity, 'oxygen':oxygen,'turbidity' :turbidity }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "ph = %s %" % ph,"turbidity = %s NTU " % turb ,"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()

```