

DEVELOP THE PYTHON SCRIPT

Team ID	PNT2022TMID12734
Project Name	Real-Time River Water Quality Monitoring And Control System

File Edit Format Run Options Window Help

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

#ibm watson device credentials

organization="rj0qwb"
deviceType="RivWatQuality"
deviceid="RivWatQuality"
authMethod="token"
authToken="UFT_EB+dHA3k)0_pA7"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "MotorON":
        print("motor in on")
    else :
        print ("motor is off")

#generate random values for pH and turbity

def myCommandCallback(cmd):
    print ("command received: %s" %cmd.data['command'])
    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 sending data of pH Values and Turbidity
```

```
deviceCli.connect()
```

```
while True:
```

```
    time.sleep(2)
```

```
    Ph=random.randint(0,14)
```

```
    Turb=random.randint(0,10)
```

```
    data={'Ph':Ph,'Turb':Turb}
```

```
    print(data)
```

```
def myOnPublishCallBack():
```

```
    print("pH Value of Water %s " %Ph)
```

```
    print("Turb Value of Water %s " %Turb)
```

```
success=deviceCli.publishEvent("IoTSensor","json",data,qos=0,on_publish=myOnPublishCallBack)
```

```
if not success:
```

```
    print ("Not connected to IoT")
```

```
time.sleep(1)
```

```
deviceCli.commandCallback=myCommandCallback
```

```
#disconnect the device from the cloud
```

```
deviceCli.connect()
```