

Develop a python script

Team ID	PNT2022TMID02004
Project Name	Smart waste management system for metropolitan cities

Step 1: Open python idle Step2: Type the program Step 3: Then click on file and save the document Step 4: Then click on Run then Run Module Step 5: output will be appeared in the idle window

Python script

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

# watson device details
organization = "4yi0vc"
devicType    = "BIN1"
deviceId     = "BIN1ID"
authMethod= "token"
authToken="123456789"

#generate random values for random variables (temperature&humidity)

def
myCommandCallback(cmd):
    global a
    print("command recieved:%s"
%cmd.data['command'])
    control=cmd.data['command']
    print(control)
try:
```

```

        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"auth-
method":authMethod,"authtoken":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 "temp" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

```

```

    while

```

```

True:

```

```

        distance= random.randint(10,70)
loadcell= random.randint(5,15)      data=
{'dist':distance,'load':loadcell}
        if loadcell < 13 and loadcell
> 15:

```

```

            load = "90 %"
elif loadcell < 8 and loadcell > 12:
            load = "60 %"
elif loadcell < 4 and loadcell > 7:
            load = "40 %"

```

```

else:

```

```

            load = "0 %"
if distance < 15:
            dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'
                elif

```

```

distance < 40 and distance >16:
            dist = 'Risk warning:' 'dumpster is above 60%'
                elif distance < 60

```

```

and distance > 41:                dist =
'Risk warning:' '40 %'          else:
            dist = 'Risk warning:' '17 %'

```

```

                                if
load == "90 %" or distance == "90 %":
            warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

```

```

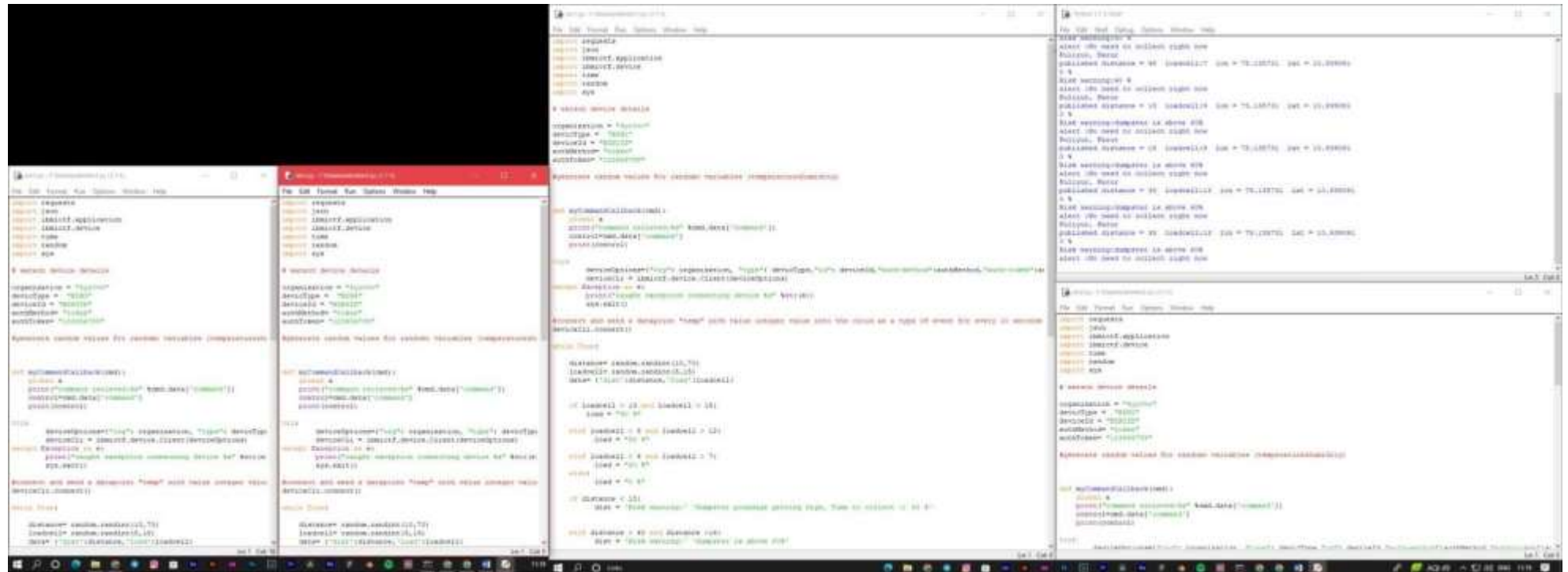
        elif load == "60 %" or
distance == "60 %":
            warn = 'alert :'
'dumpster is above 60%'    else :
            warn = 'alert :' 'No need to collect right now '
def myOnPublishCallback(lat=10.678991,long=78.177731):
    print("Gandigramam, Karur")    print("published distance = %s " %distance,"loadcell:%s "
%loadcell,"lon = %s " %long,"lat = %s" %lat)    print(load)    print(dist)    print(warn)

    time.sleep(10)
        success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)    success=deviceCli.publishEvent
("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)
        if not success:
print("not connected to ibmiot")
time.sleep(30)

deviceCli.commandCallback=myCommandCallback
#disconnect the device deviceCli.disconnect

```

Screenshots Python script:



The image displays three screenshots of a Python script, likely a web scraper or data collector, written in a code editor. The script is organized into several sections, each with a comment header in Russian.

Section 1: Initialization and Configuration

```
# Инициализация переменных
import requests
import time
import random
import sys
import os
import logging

# Настройка логирования
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S',
    filename='log.txt',
    filemode='a'
)

# URL-адрес для скачивания
url = 'http://example.com'

# Заголовки для имитации браузера
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149 Safari/537.36'
}
```

Section 2: Main Loop and Data Collection

```
# Основной цикл
def main():
    # Скачивание файла
    response = requests.get(url, headers=headers)

    # Проверка статуса ответа
    if response.status_code == 200:
        # Сохранение файла
        with open('data.txt', 'wb') as f:
            f.write(response.content)

        # Вывод информации
        logging.info(f'Файл успешно скачан. Размер: {len(response.content)} байт.')
    else:
        logging.error(f'Ошибка при скачивании файла. Статус: {response.status_code}.')

    # Задержка перед следующим запросом
    time.sleep(random.randint(1, 10))

# Запуск основного цикла
if __name__ == '__main__':
    main()
```

Section 3: Error Handling and Cleanup

```
# Обработка ошибок
def handle_error(error):
    logging.error(f'Ошибка: {error}')
    sys.exit(1)

# Проверка наличия файла
def check_file_exists(filename):
    if not os.path.exists(filename):
        logging.error(f'Файл {filename} не найден.')
        return False
    return True

# Проверка наличия папки
def check_directory_exists(directory):
    if not os.path.exists(directory):
        logging.error(f'Папка {directory} не найдена.')
        return False
    return True
```