Internet of Things (IoT) with the IoT Greenhouse

Overview:

In this introductory lesson you will experience the Internet of Things (IoT) using the IoT Greenhouse. This system is based on the Raspberry Pi computer and uses temperature sensors to monitor the greenhouse. When the temperature hits threshold values, system components such as louvers, fan, and heaters are activated. The system also posts greenhouse data to cloud services enabling other devices to consume the data.

Prerequisites:

This is the introductory lesson. There are no prerequisites other than basic computer and keyboard use.

Performance Outcomes:

1. Build a model greenhouse
2. Experience electronic sensors and actuators connected to a Raspberry Pi
3. Write and run code using Python
4. Post data to the Internet
5. View Greenhouse data in the IoT Greenhouse mobile app

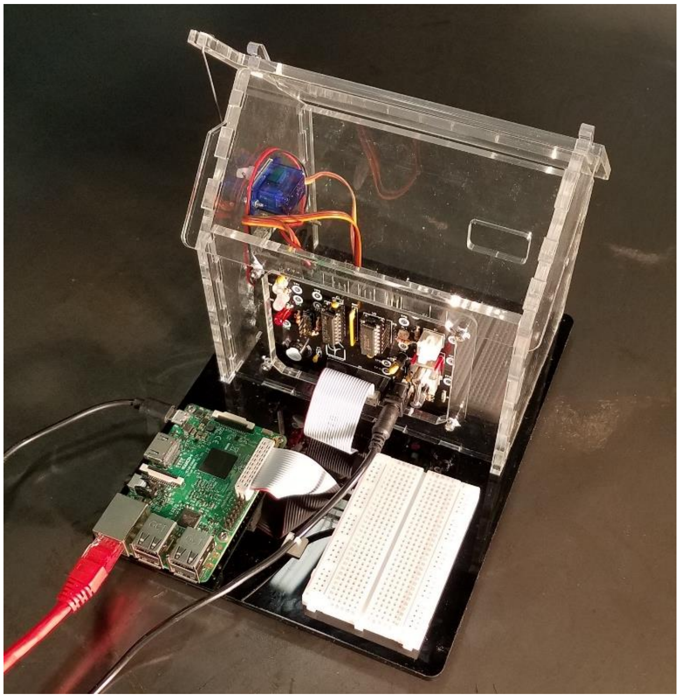
Resources:

1. IoT Greenhouse system including HDMI monitor, keyboard, and mouse
2. 2.4 GHz Wi-Fi

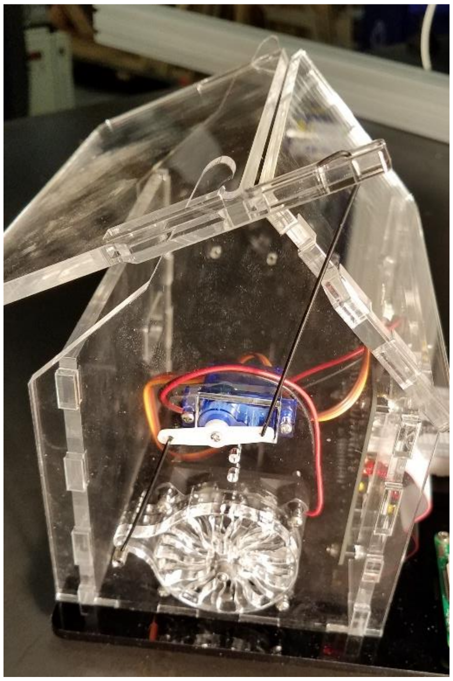
Activity:

Step 1: Build the IoT Greenhouse!

1. Use the parts at your station to make your greenhouse model.

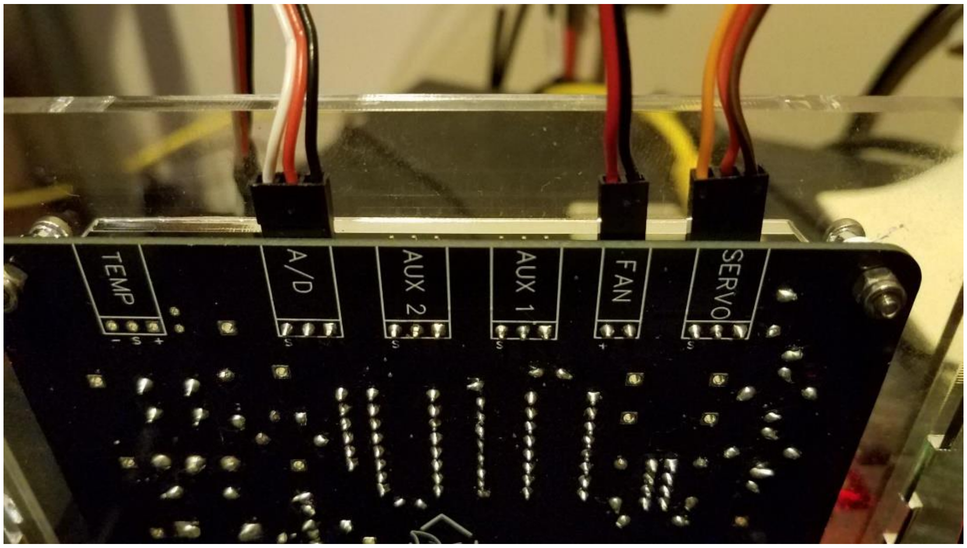


2. Attach the linkage to enable the servo motor to control the louver and the roof.

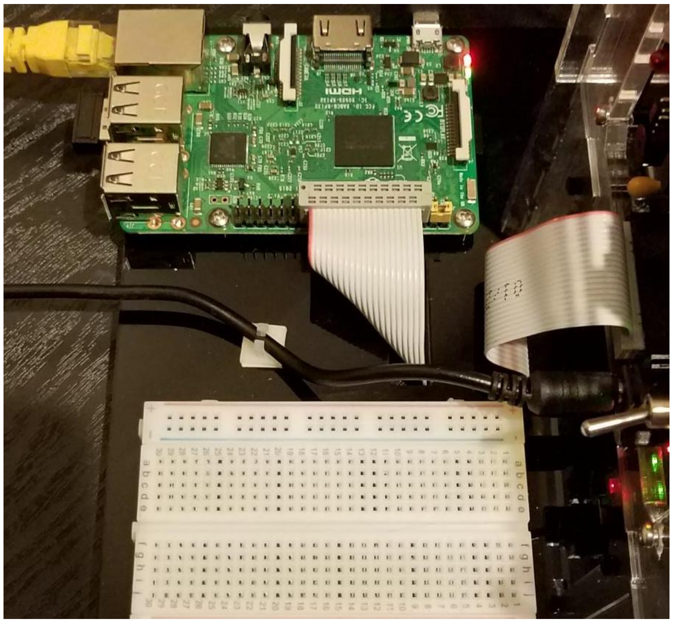


Step 2: Wire the IoT Greenhouse

3. Connect the servo, fan, and temperature sensor leads to the control board. The orientation of the connectors is critical. Reference the image below when making your connections.



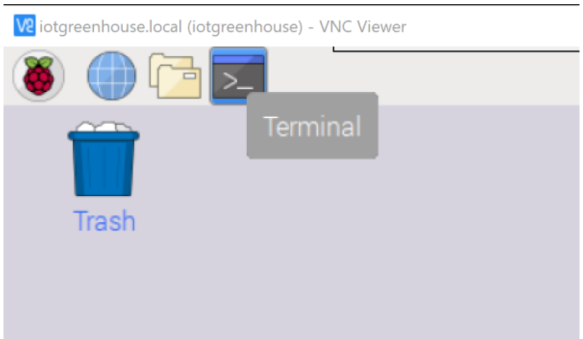
4. The IoT Greenhouse uses 20 of the 40 GPIO header on the Raspberry Pi. The ribbon cable must be connected as shown on the following page. The small two pin jumper covers the 5V connection points. These connections should not be used. The jumper also assists in indexing the correct position of the ribbon cable connector.



5. A breadboard area is included for future lessons. Components and the breadboard are not required when completing introductory activities.

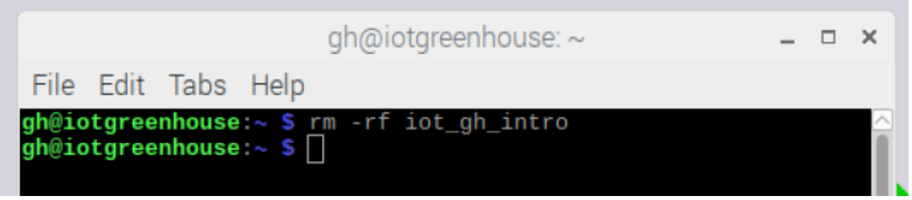
Step 3: Download Code to the Raspberry Pi

6. Boot the Raspberry Pi by applying power. Log in using the following credentials.
   a. User: **gh**
   b. Password:
7. Select the icon on the taskbar to open a Terminal window.



8. Enter the following command at the terminal prompt to remove any prior work associated with this introductory activity.

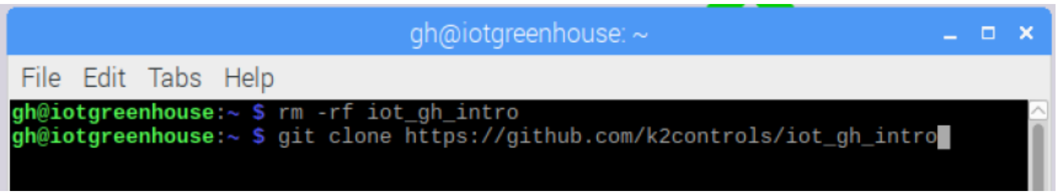
```
rm -rf iot_gh_intro
```



```
gh@iotgreenhouse:~ $ rm -rf iot_gh_intro
gh@iotgreenhouse:~ $ 
```

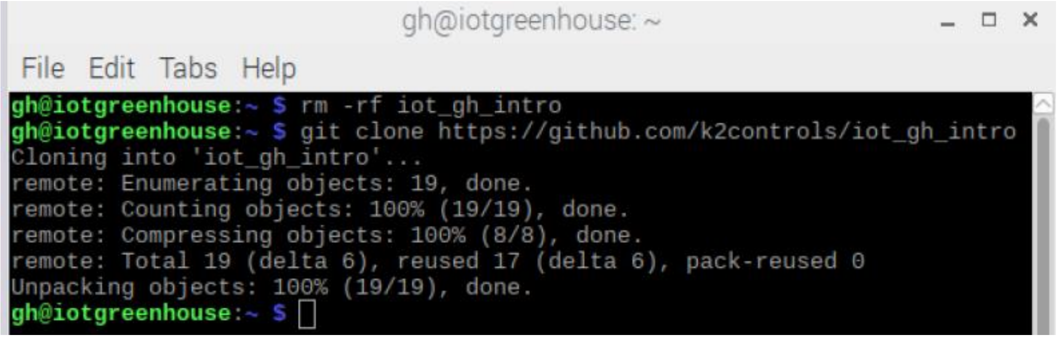
9. Enter the following command at the terminal prompt to create the initial Python code file for this activity.

```
git clone https://github.com/k2controls/iot_gh_intro
```



```
gh@iotgreenhouse:~ $ rm -rf iot_gh_intro
gh@iotgreenhouse:~ $ git clone https://github.com/k2controls/iot_gh_intro
```

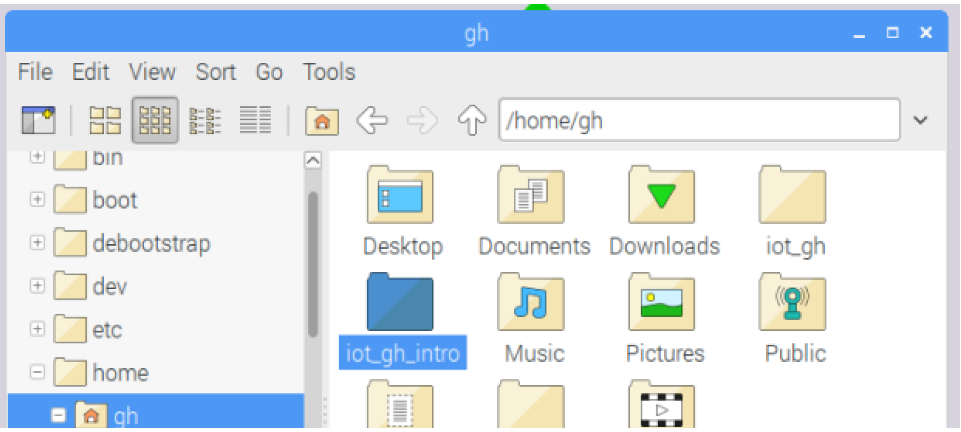
10. The required **iot_gh_intro** folder and file are created.



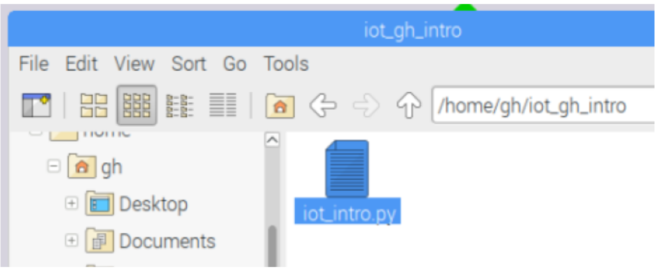
```
gh@iotgreenhouse:~ $ rm -rf iot_gh_intro
gh@iotgreenhouse:~ $ git clone https://github.com/k2controls/iot_gh_intro
Cloning into 'iot_gh_intro'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 19 (delta 6), reused 17 (delta 6), pack-reused 0
Unpacking objects: 100% (19/19), done.
gh@iotgreenhouse:~ $
```

11. Close the terminal window.

12. Open the File Manager window using the taskbar. Open the new **iot_gh_intro** folder.



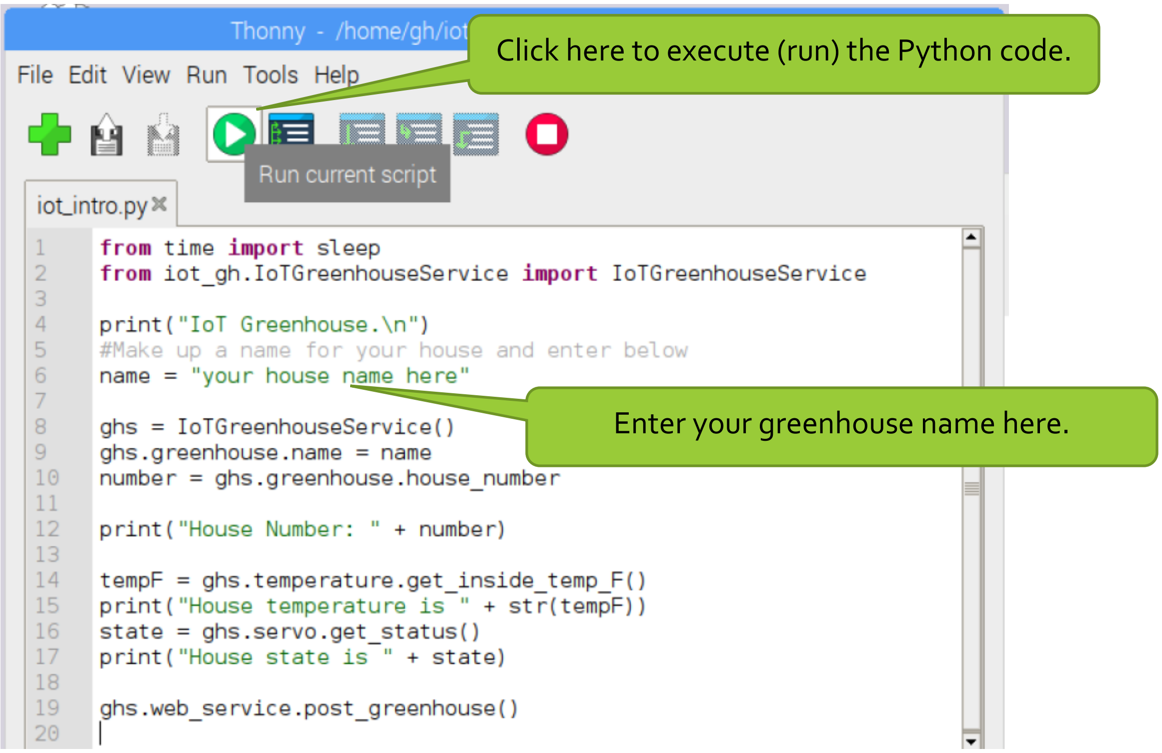
13. The **iot_gh_intro** folder contains a single **iot_intro.py** file that is used in this activity.



Step 4: Run Python code.

14. Double click on the **iot_intro.py** file to open in the Thonny editor.

15. Decide on a name for your greenhouse and enter it on line 6 of the code. Be sure your text includes quotation marks. This specifies the text as a valid string of characters. The text color should be green.



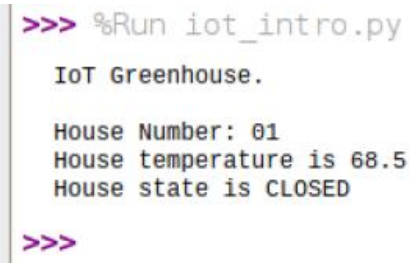
Click here to execute (run) the Python code.

Enter your greenhouse name here.

```
1   from time import sleep
2   from iot_gh.IoTGreenhouseService import IoTGreenhouseService
3
4   print("IoT Greenhouse.\n")
5   #Make up a name for your house and enter below
6   name = "your house name here"
7
8   ghs = IoTGreenhouseService()
9   ghs.greenhouse.name = name
10  number = ghs.greenhouse.house_number
11
12  print("House Number: " + number)
13
14  tempF = ghs.temperature.get_inside_temp_F()
15  print("House temperature is " + str(tempF))
16  state = ghs.servo.get_status()
17  print("House state is " + state)
18
19  ghs.web_service.post_greenhouse()
20
```

16. Click on the green arrow on the Thonny taskbar to run the introductory Python code.

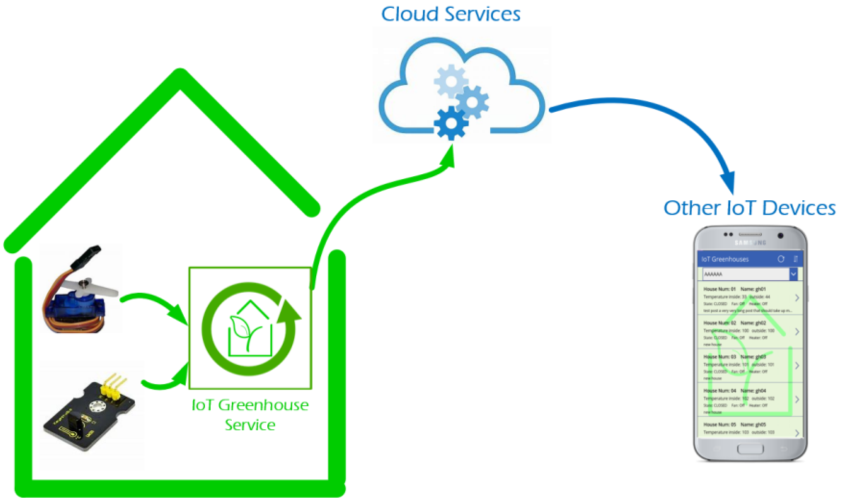
17. The results are shown below the editor window and should be similar to those shown below.



```
>>> %Run iot_intro.py

  IoT Greenhouse.

  House Number: 01
  House temperature is 68.5
  House state is CLOSED

>>>
```

Step 5: What Happened?

18. When the program ran, an instance of the IoT Greenhouse Service was created. This service read the temperature from the sensor and checked the status of the servo motor. An update was posted to the cloud service enabling other IoT Devices access to the data.



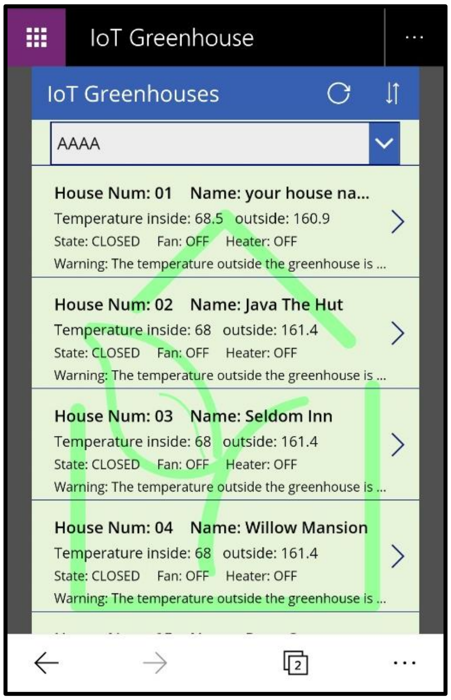
19. To view your greenhouse data enter the following URL into your mobile phone or computer browser. Enter the credentials below to access the app.

    URL: `http://bit.ly/iot_gh_app`

    User: **`gh@k2controls.net`**

    Password: **`IOT_greenhouse`**

20. A screen capture of the mobile app is below. Use the drop down control at the top to select your IoT Greenhouse group identifier. This filters the list so that only your greenhouse data are visible.



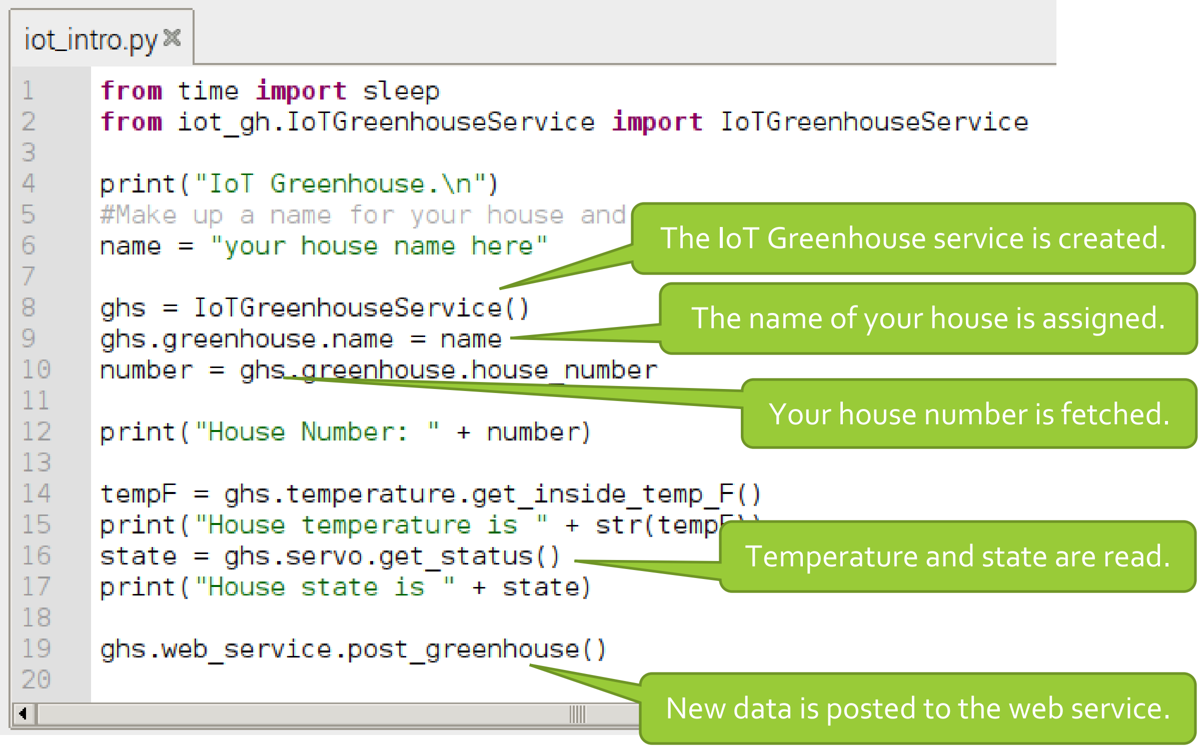
House Num: 01 Name: your house na...
Temperature inside: 68.5 outside: 160.9
State: CLOSED Fan: OFF Heater: OFF
Warning: The temperature outside the greenhouse is ...

House Num: 02 Name: Java The Hut
Temperature inside: 68 outside: 161.4
State: CLOSED Fan: OFF Heater: OFF
Warning: The temperature outside the greenhouse is ...

House Num: 03 Name: Seldom Inn
Temperature inside: 68 outside: 161.4
State: CLOSED Fan: OFF Heater: OFF
Warning: The temperature outside the greenhouse is ...

House Num: 04 Name: Willow Mansion
Temperature inside: 68 outside: 161.4
State: CLOSED Fan: OFF Heater: OFF
Warning: The temperature outside the greenhouse is ...

21. Review the program listing below to identify lines of code responsible for greenhouse functions.



```
from time import sleep
from iot_gh.IoTGreenhouseService import IoTGreenhouseService

print("IoT Greenhouse.\n")
#Make up a name for your house and
name = "your house name here"

ghs = IoTGreenhouseService()
ghs.greenhouse.name = name
number = ghs.greenhouse.house_number

print("House Number: " + number)

tempF = ghs.temperature.get_inside_temp_F()
print("House temperature is " + str(tempF))
state = ghs.servo.get_status()
print("House state is " + state)

ghs.web_service.post_greenhouse()

```

The IoT Greenhouse service is created.

The name of your house is assigned.

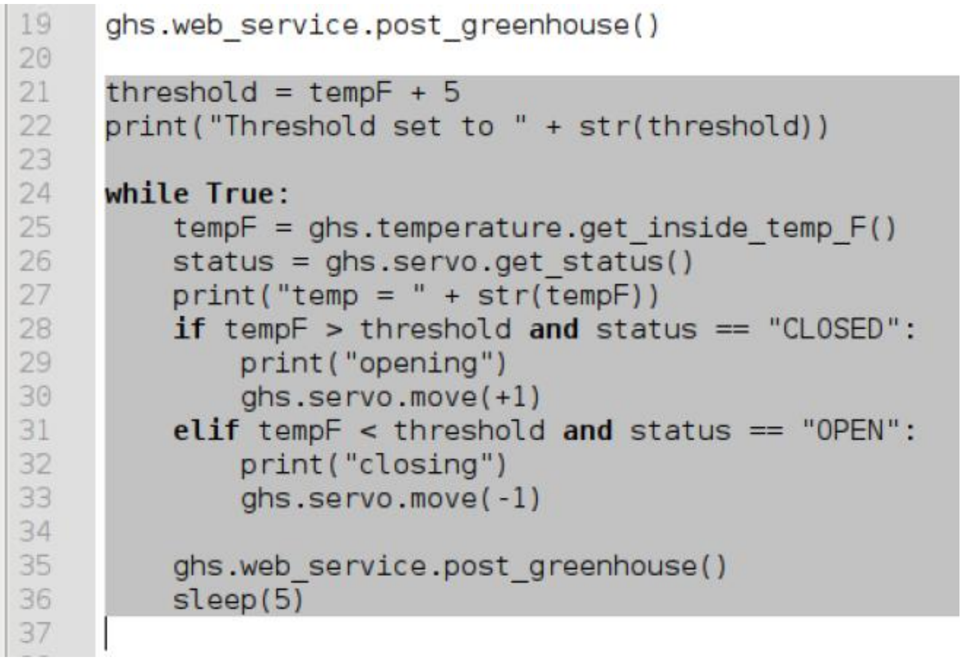
Your house number is fetched.

Temperature and state are read.

New data is posted to the web service.

Step 6: Write Code to Control the IoT Greenhouse.

22. Click into the Thonny code window and move to the bottom. Add the following code at **line 21** to control the greenhouse louvers. **Watch your CaPiTaLiZaTion…it matters!** Spaces too! Keep the code in-line. The Tab key is recommended.



```
19  ghs.web_service.post_greenhouse()
20
21  threshold = tempF + 5
22  print("Threshold set to " + str(threshold))
23
24  while True:
25      tempF = ghs.temperature.get_inside_temp_F()
26      status = ghs.servo.get_status()
27      print("temp = " + str(tempF))
28      if tempF > threshold and status == "CLOSED":
29          print("opening")
30          ghs.servo.move(+1)
31      elif tempF < threshold and status == "OPEN":
32          print("closing")
33          ghs.servo.move(-1)
34
35      ghs.web_service.post_greenhouse()
36      sleep(5)
37
```

23. Can you read Python code?

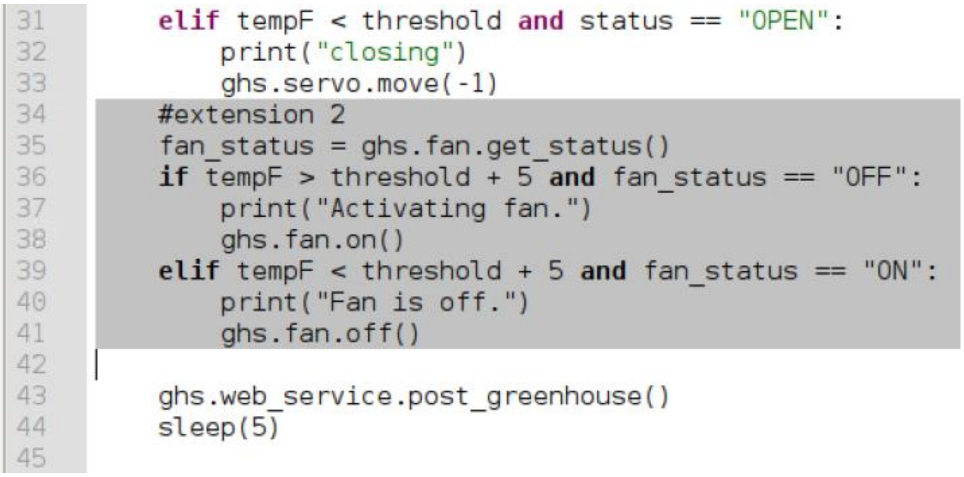
Step 7: Run the Code and Test!

24. Run the revised code using the green arrow on the task bar. Ask for help if you get errors.

25. Use the **hot air gun** and **compressed air** to change the IoT Greenhouse temperature. Watch for greeenhouse action and check the IoT Greenhouse app for your updates!

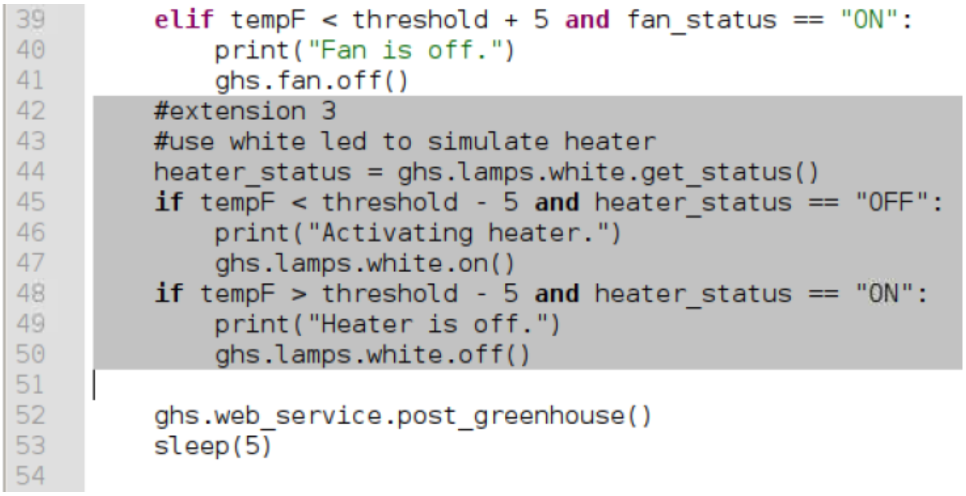
Step 8: Extend the Code

26. If time permits, insert the following Fan extension **at line 34**. Run the code and test.



```
31      elif tempF < threshold and status == "OPEN":
32          print("closing")
33          ghs.servo.move(-1)
34      #extension 2
35      fan_status = ghs.fan.get_status()
36      if tempF > threshold + 5 and fan_status == "OFF":
37          print("Activating fan.")
38          ghs.fan.on()
39      elif tempF < threshold + 5 and fan_status == "ON":
40          print("Fan is off.")
41          ghs.fan.off()
42
43      ghs.web_service.post_greenhouse()
44      sleep(5)
45
```

27. Insert the following heater extension **at line 42**. The white LED is used to simulate a heater being activated. You'll need to apply the heat. Run the code and test.



```
39      elif tempF < threshold + 5 and fan_status == "ON":
40          print("Fan is off.")
41          ghs.fan.off()
42      #extension 3
43      #use white led to simulate heater
44      heater_status = ghs.lamps.white.get_status()
45      if tempF < threshold - 5 and heater_status == "OFF":
46          print("Activating heater.")
47          ghs.lamps.white.on()
48      if tempF > threshold - 5 and heater_status == "ON":
49          print("Heater is off.")
50          ghs.lamps.white.off()
51
52      ghs.web_service.post_greenhouse()
53      sleep(5)
54
```

Step 8: Wrap-up

28. While the IoT Greenhouse is only a model to teach concepts, this technology does have applications in agriculture. Brainstorm with your group some of the ways this technology can be applied to greenhouses or agriculture in general.