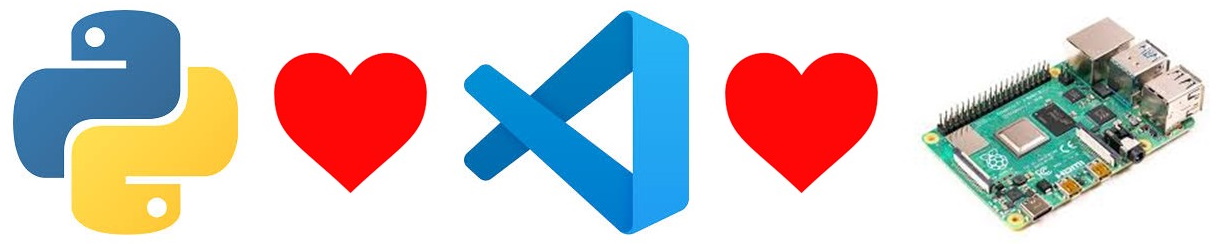
PyLab 1：Raspberry Pi，调试Python物联网应用程序

在Twitter [@dglover](https://twitter.com/dglover)上关注我



| **作者** | **微软云开发者倡导者**[**Dave Glover**](https://developer.microsoft.com/en-us/advocates/dave-glover?WT.mc_id=pycon-blog-dglover) |
| --- | --- |
| 平台 | Linux，macOS，Windows，Raspbian Buster |
| 服务 | [Azure物联网中心](https://docs.microsoft.com/en-us/azure/iot-central/?WT.mc_id=pycon-blog-dglover) |
| 工具 | [Visual Studio代码](https://code.visualstudio.com/?WT.mc_id=pycon-blog-dglover) |
| 硬件 | [Raspberry Pi](https://www.raspberrypi.org/products/raspberry-pi-4-model-b/)，[Raspberry Pi Sense HAT](https://www.raspberrypi.org/products/sense-hat/) |
| 语言 | 蟒蛇 |
| 日期 | 2019年9月 |

PDF实验指南

您可能会发现下载和遵循**调试Raspberry Pi物联网Flask App** PyLab 的PDF版本更容易。

* [英语实验室指南](https://github.com/gloveboxes/PyLab-1-Debugging-a-Python-Internet-of-Things-Application/blob/master/README.pdf)
* [简体中文实验室指南](https://github.com/gloveboxes/PyLab-1-Debugging-a-Python-Internet-of-Things-Application/blob/master/README-Simplified-Chinese.pdf)

PyLab内容

* [PyLab 1：Raspberry Pi，调试Python物联网应用程序](https://gloveboxes.github.io/PyLab-1-Debugging-a-Python-Internet-of-Things-Application/)
* [PyLab 2：Raspberry Pi，Azure物联网中心和Docker容器调试](https://gloveboxes.github.io/PyLab-2-Python-Azure-IoT-Central-and-Docker-Container-Debugging/)

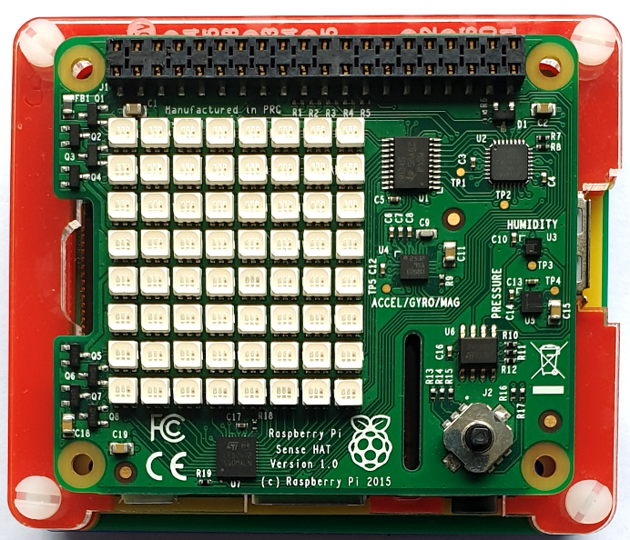
介绍

在本动手实验中，您将学习如何使用[Visual Studio Code](https://code.visualstudio.com/?WT.mc_id=pycon-blog-dglover)和[Remote SSH](https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh&WT.mc_id=pycon-blog-dglover)扩展在Raspberry Pi上创建和调试Python Web应用程序。Web应用程序将从连接到Raspberry Pi的传感器读取温度，湿度和气压遥测。

PyLab设置

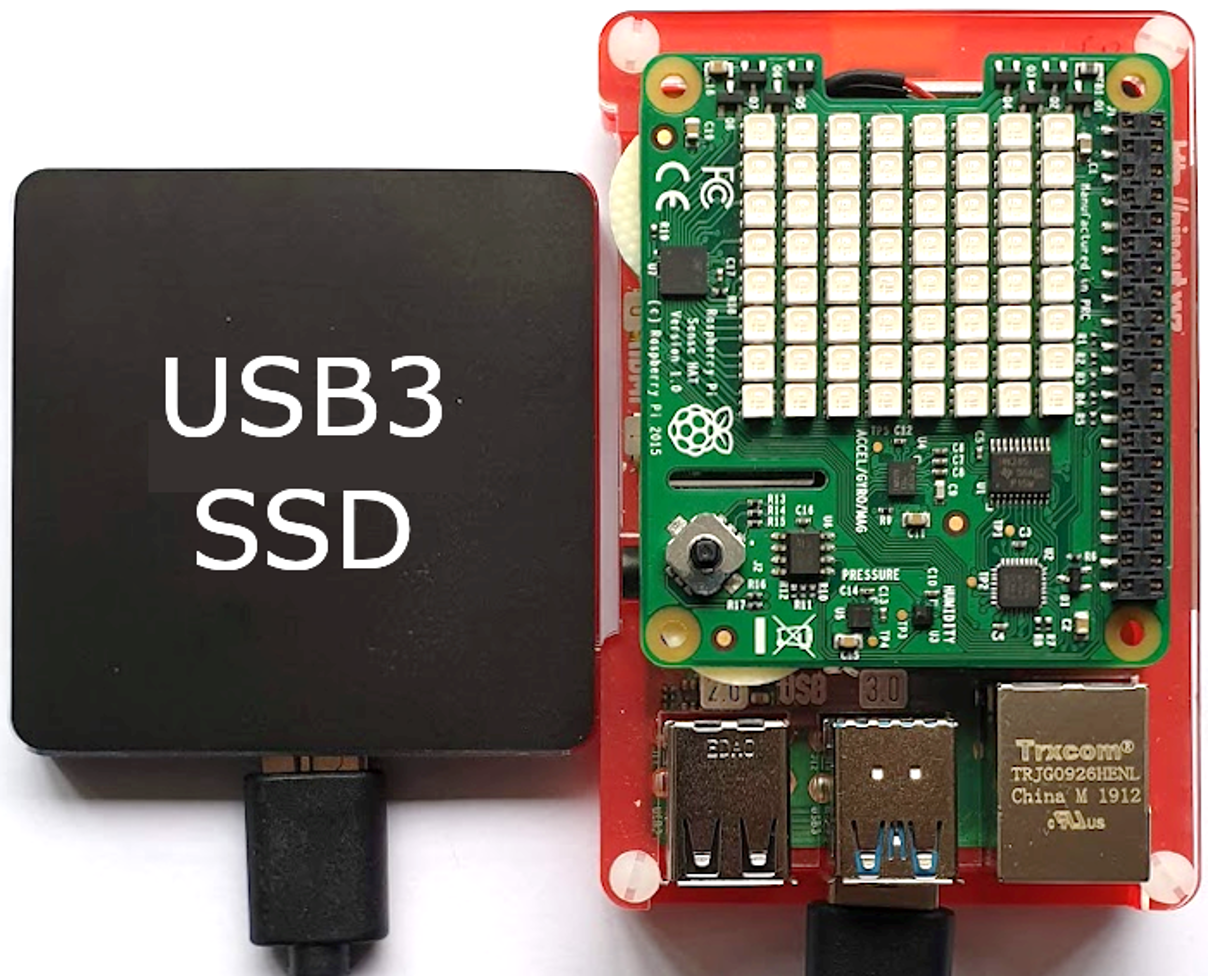
* [单用户设置](https://github.com/gloveboxes/PyLab-0-Raspberry-Pi-Set-Up/tree/master/setup-singleuser/README.md)

此自动设置安装所需的库，Docker，并构建实验室docker镜像。



* [多用户设置](https://github.com/gloveboxes/PyLab-0-Raspberry-Pi-Set-Up/tree/master/setup-multiuser/README.md)

多用户设置允许每个Raspberry Pi 4 4GB最多20个用户/学生。需要USB3 SSD驱动器来支持此数量用户的磁盘IO要求。安装脚本安装实验室内容和Docker。构建实验室Docker镜像，并设置所有用户。



软件安装



这个动手实验室使用Visual Studio Code。Visual Studio Code是一个代码编辑器，是[GitHub上](https://github.com/microsoft/vscode)最受欢迎的**开源**项目之一。它可以在Linux，macOS和Windows上运行。

安装Visual Studio代码

1. **安装**[**Visual Studio代码**](https://code.visualstudio.com/Download?WT.mc_id=pycon-blog-dglover)

Visual Studio代码扩展

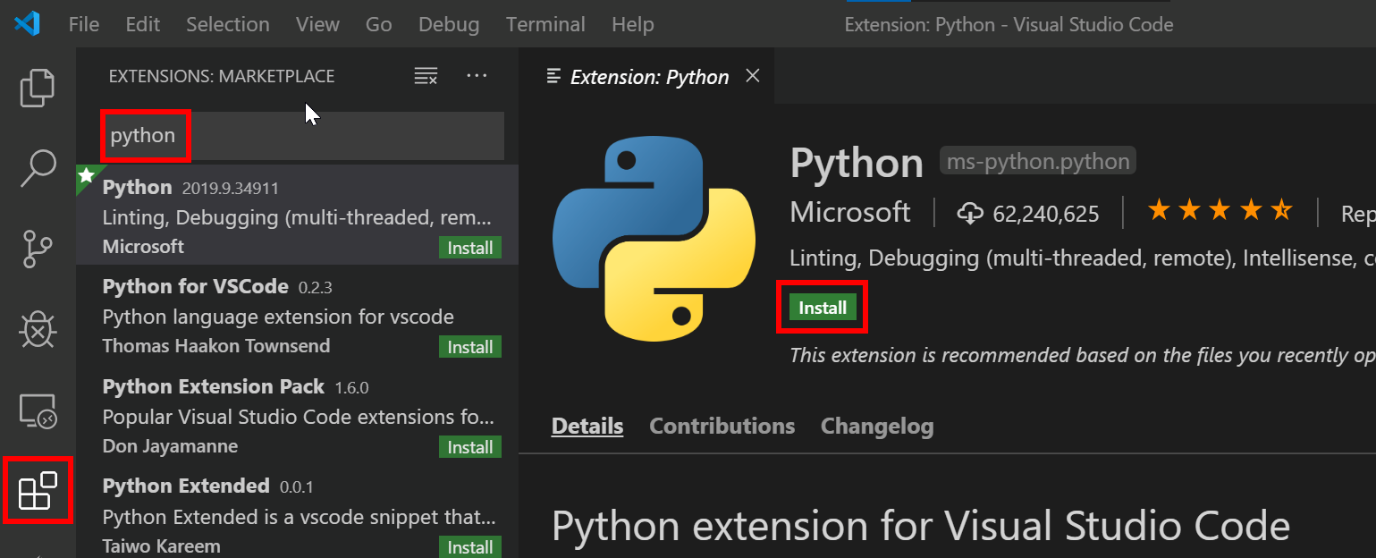
Visual Studio Code包含开箱即用的功能只是一个开始。VS代码扩展允许您为安装添加语言，调试器和工具，以支持您的开发工作流程。

浏览扩展程序

您可以从Visual Studio Code中搜索和安装扩展。从Visual Studio代码主菜单中打开Extensions视图，选择**View** > **Extensions**或单击Visual Studio Code一侧**活动栏中**的Extensions图标。



这将显示[VS Code Marketplace](https://marketplace.visualstudio.com/VSCode?WT.mc_id=pycon-blog-dglover)上最受欢迎的VS Code扩展的列表。



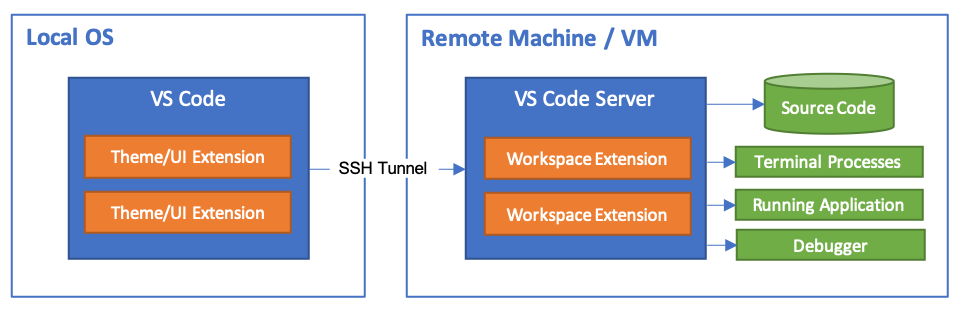
安装Python和远程SSH扩展

搜索并安装Microsoft发布的以下两个Visual Studio代码扩展。

1. [**蟒蛇**](https://marketplace.visualstudio.com/items?itemName=ms-python.python&WT.mc_id=pycon-blog-dglover)
2. [**远程 - SSH**](https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh&WT.mc_id=pycon-blog-dglover)

远程SSH开发

Visual Studio Code Remote - SSH扩展允许您使用正在运行的SSH服务器在任何远程计算机，虚拟机或容器上打开远程文件夹，并充分利用Visual Studio Code。



Raspberry Pi硬件

如果您正在参加研讨会，那么您可以使用共享的网络连接Raspberry Pi。您也可以使用自己的网络连接Raspberry Pi进行动手实验。

您将需要实验室指导员提供以下信息。

1. Raspberry Pi 的**网络IP地址**
2. 您指定的**登录名**和**密码**。

使用私钥/公钥进行SSH身份验证



为[SSH](https://en.wikipedia.org/wiki/Secure_Shell)身份验证设置公钥/私钥对是一种安全，快速的方法，可以从您的计算机进行身份验证到Raspberry Pi。建议用于此动手实验。

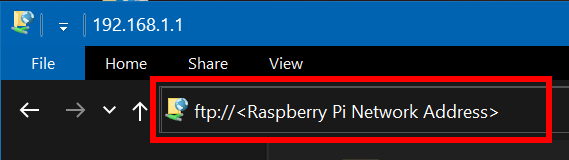
SSH为Windows用户设置

SSH实用程序将指导您完成为Visual Studio Code和Raspberry Pi设置安全SSH通道的过程。

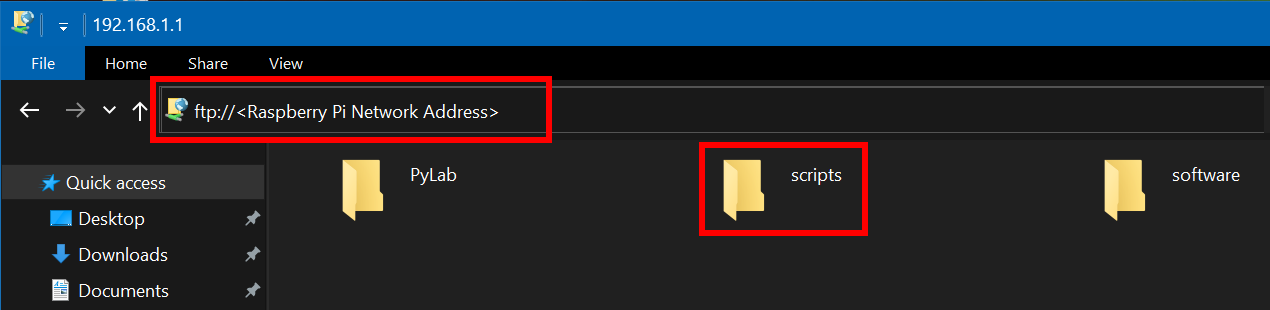
系统将提示您：

* Raspberry Pi网络IP地址，
* Raspberry Pi登录名和密码

1. 从**Windows文件资源管理器中**，打开**ftp：// <Raspberry Pi地址>**



1. 将**脚本**目录复制到**桌面**



1. 打开复制到**桌面**的**脚本**文件夹
2. 双击**windows-setup-ssh.cmd**

SSH为Linux和macOS用户设置

SSH实用程序将指导您完成为Visual Studio Code和Raspberry Pi设置安全SSH通道的过程

系统将提示您：

* Raspberry Pi网络IP地址，
* Raspberry Pi登录名和密码

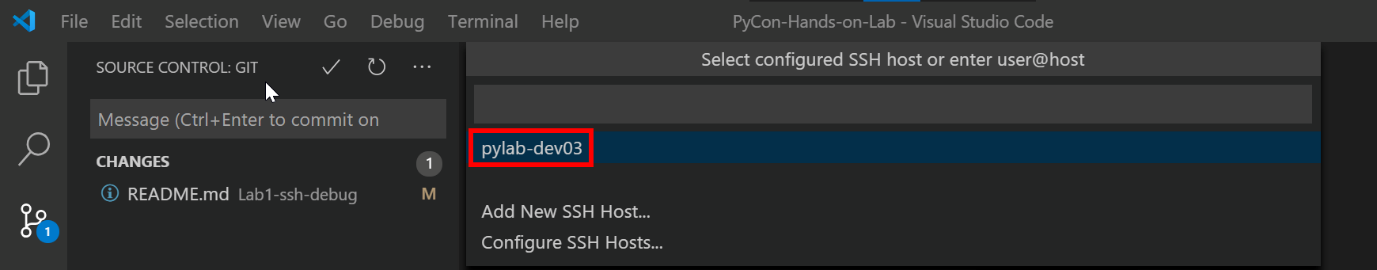
1. 打开终端窗口
2. 复制并粘贴以下命令，然后按**ENTER键**

read -p "Enter the Raspberry Pi Address: " pyurl **&&** \

curl ftp://$pyurl/scripts/ssh-setup.sh | bash

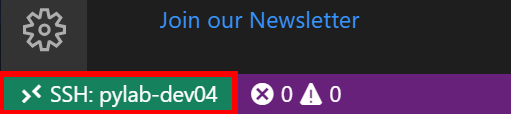
启动远程SSH连接

1. **启动Visual Studio代码**
2. 按**F1**打开命令选项板，键入**ssh connect**并选择**Remote-SSH：Connect to Host**
3. 选择**pylab-devnn**配置



1. 检查远程SSH是否已连接。

连接需要一些时间，然后Visual Studio Code左下角的SSH Status将更改为**>< SSH：pylab-devnn**。devnn是你的Raspberry Pi登录名。

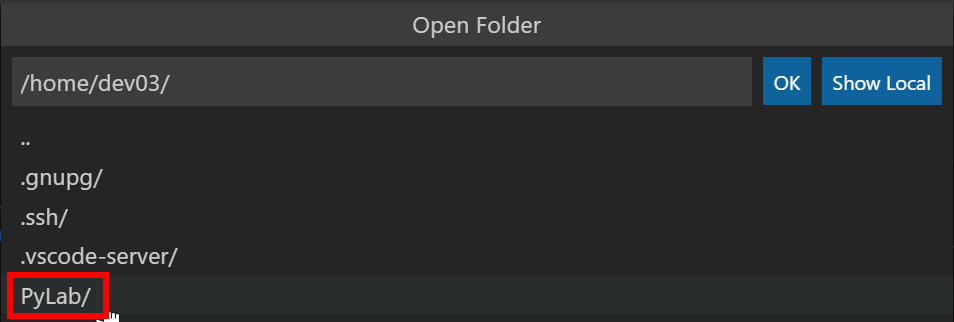


打开PyLab 1 Python调试项目

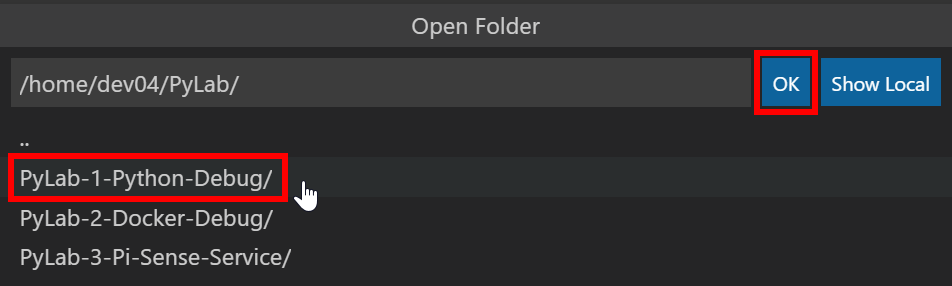
Python Flask Web应用程序

在本实验中，我们将启动并调试[Flask](https://www.fullstackpython.com/flask.html)应用程序，该应用程序读取连接到Raspberry Pi的传感器。Flask是一个流行的Python Web框架，功能强大，但对初学者也很容易。

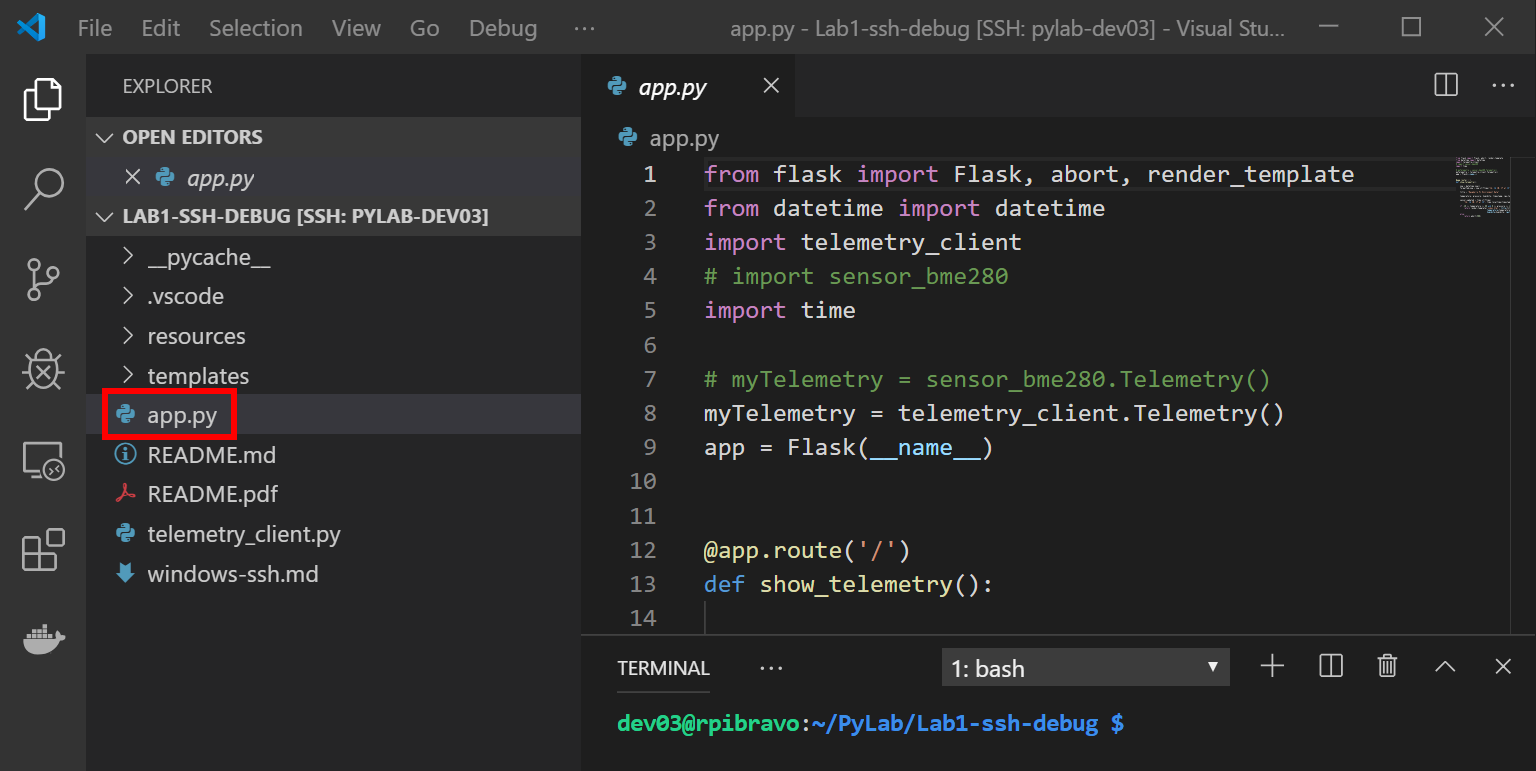
1. 从Visual Studio代码主菜单：**文件** > **打开文件夹**
2. 选择**PyLab**目录



1. 接下来选择**PyLab-1-Python-Debug**目录



1. 单击“ **确定”**以打开目录
2. 在**资源管理器**栏中，打开**app.py**文件并查看内容



花一点时间来查看Python Flask Web应用程序。

应用的.py

**from** flask **import** Flask, abort, render\_template

**from** datetime **import** datetime

**import** telemetry\_client

*# import sensor\_bme280*

**import** time

*# myTelemetry = sensor\_bme280.Telemetry()*

myTelemetry **=** telemetry\_client**.**Telemetry()

app **=** Flask(\_\_name\_\_)

**@app.route**('/')

**def** **show\_telemetry**():

now **=** datetime**.**now()

formatted\_now **=** now**.**strftime("%A, %d %B, %Y at %X")

title **=** "Raspberry Pi Environment Data"

temperature, pressure, humidity, timestamp, cpu\_temperature **=** myTelemetry**.**measure()

sensor\_updated **=** time**.**strftime(

"%A, %d %B, %Y at %X", time**.**localtime(timestamp))

**if** **-**40 **<=** temperature **<=** 60 **and** 0 **<=** pressure **<=** 1500 **and** 0 **<=** humidity **<=** 100:

**return** render\_template('index.html', title**=**title,

temperature**=**temperature, pressure**=**pressure,

humidity**=**humidity, cputemperature**=**cpu\_temperature)

**else**:

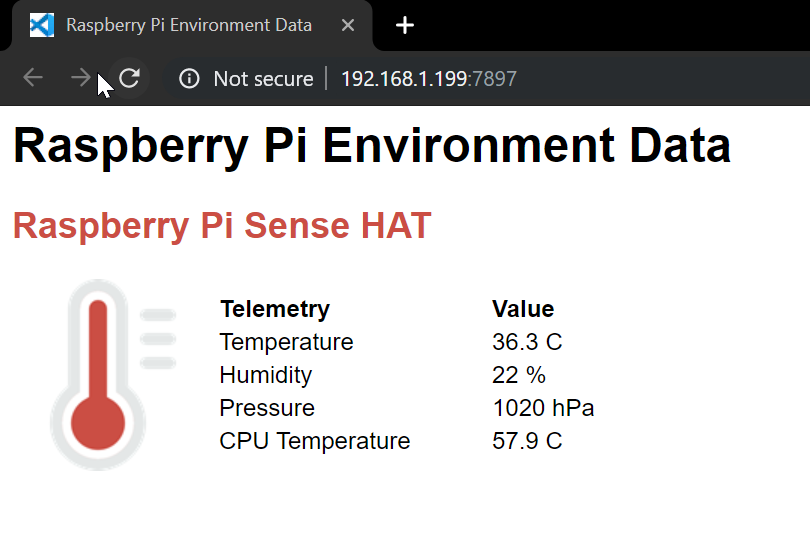
**return** abort(500)

启动Python Flask App

1. 按**F5**启动Python Flask应用程序。
2. 在Visual Studio代码**终端窗口中**，单击**在http：// ...** web链接上运行。

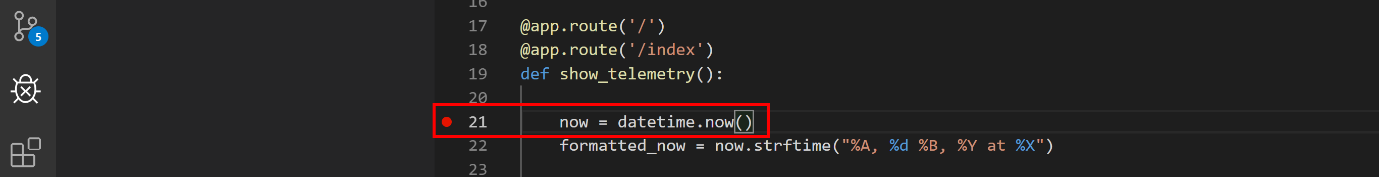


1. 这将启动您的桌面Web浏览器。
   * Flask应用程序将读取连接Raspberry Pi 的**传感器**的温度，气压，湿度，并在Web浏览器中显示结果。

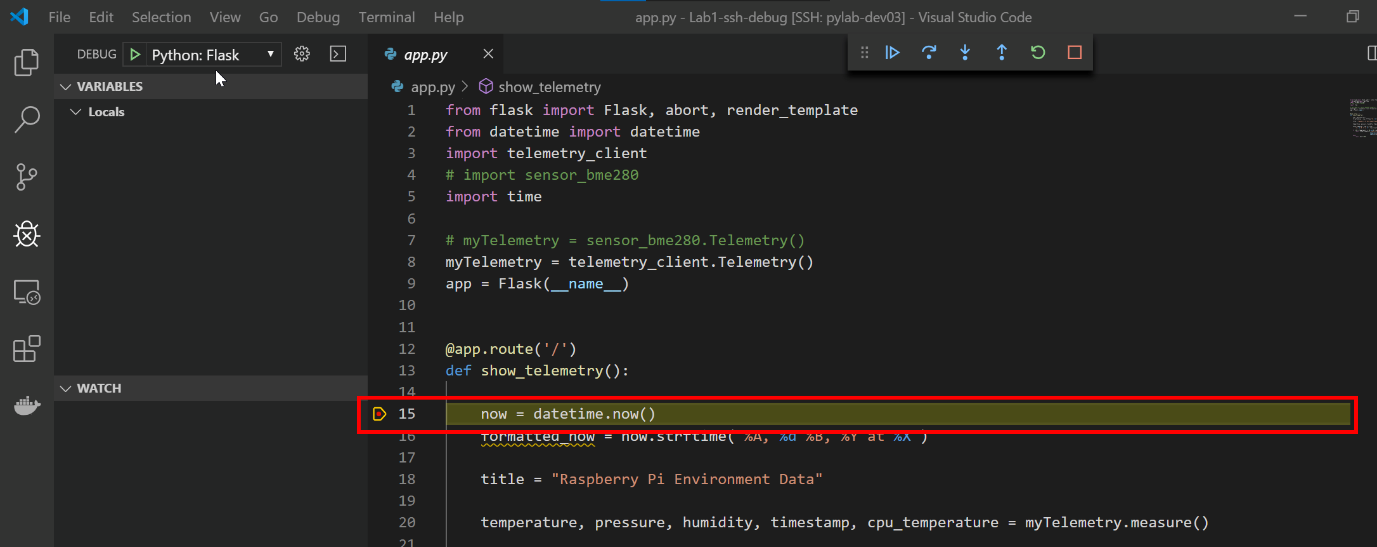


使用断点进行调试

1. 切换回Visual Studio代码并确保**app.py**文件已打开。
2. 将光标放在**现在**读取的行**= datetime.now（）**
3. 按**F9**设置断点。线上将出现一个红点，表示已设置断点。



1. 切换回**Web浏览器**并单击“ **刷新”**。由于调试器已在您设置的断点处停止，因此网页**不会响应**。
2. 切换回**Visual Studio代码**。您将看到代码已在**断点**处停止运行。



调试器工具栏选项

调试会话启动时，“ **调试”工具栏**将显示在编辑器窗口的顶部。

调试工具栏（如下所示）将出现在Visual Studio Code中。它有以下选项：

调试操作

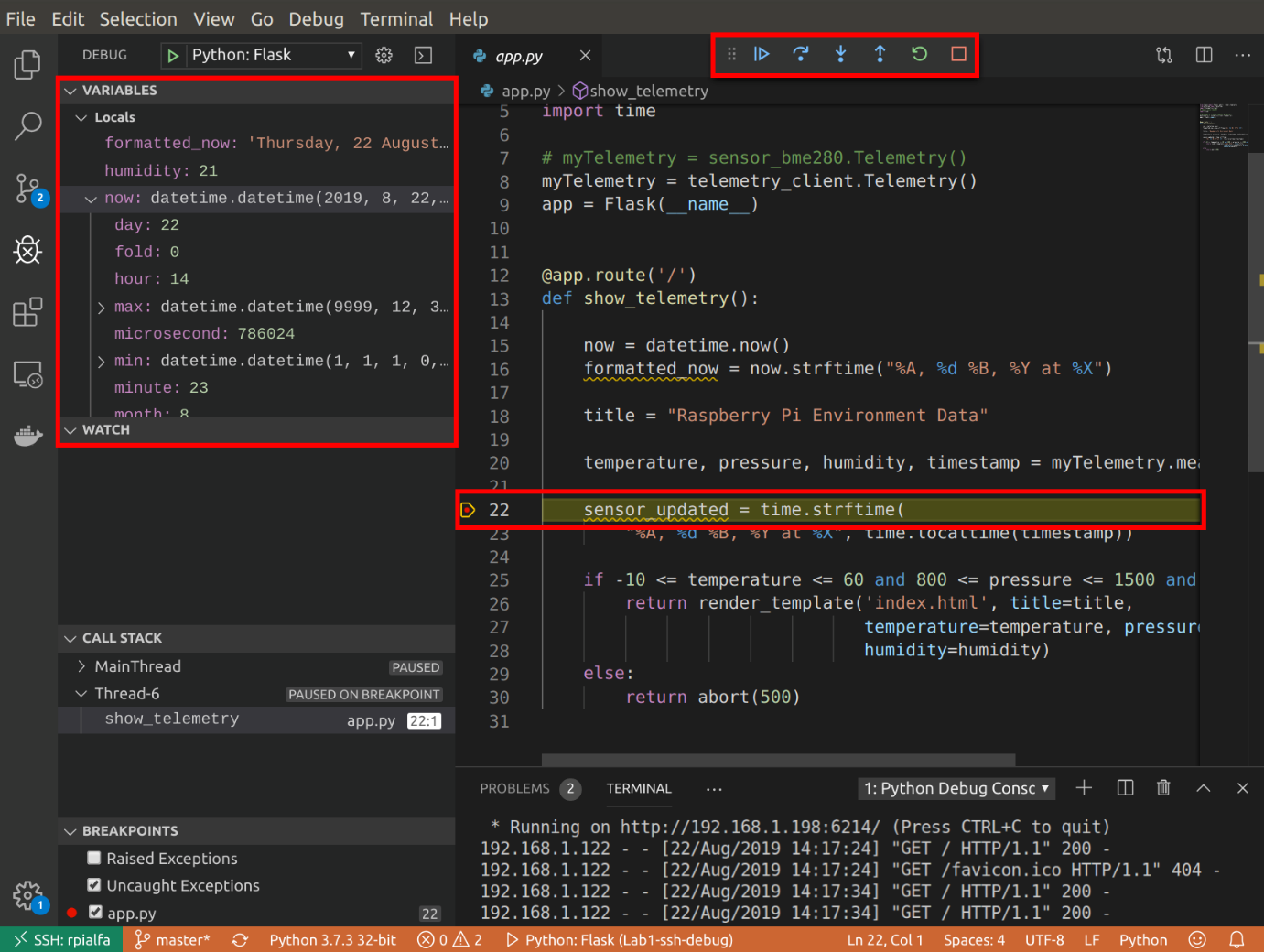
1. 暂停（或继续，F5），
2. 跳过（F10）
3. 步入（F11），
4. 走出去（Shift + F11），
5. 重启（Ctrl + Shift + F5），
6. 并停止（Shift + F5）。

开始调试

1. 按（**F10**）或单击调试工具栏上的“ **跳过”**逐步执行代码。
2. **重复**按**F10，**直到到达读取的行，**如果-40 <=温度<= 60且0 <=压力<= 1500且0 <=湿度<= 100：**
3. 您会注意到Python变量显示在**变量窗口中**。

如果看不到变量窗口，请单击活动栏中的“ **调试** ”。



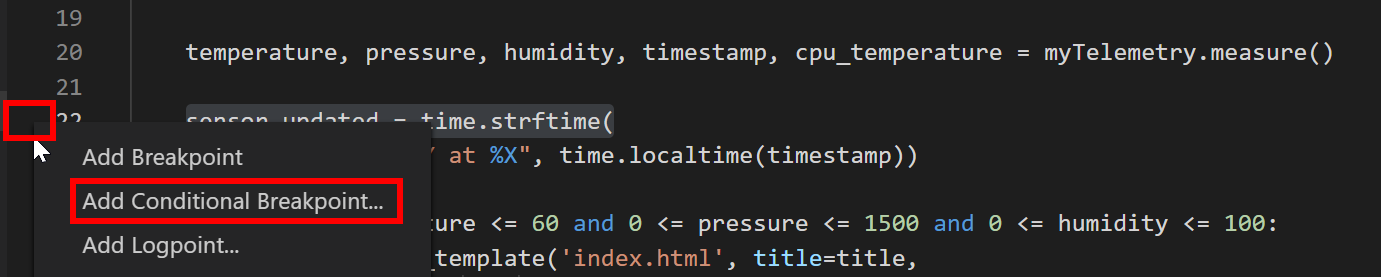


1. 尝试将**温度**变量更改为**50**。提示，**右键**单击温度变量并选择“ **设置值”**，或双击**温度**变量。
2. 按**F5**恢复Flask应用程序，然后**切换回Web浏览器**，您将看到网页上显示的温度，湿度和压力传感器数据。

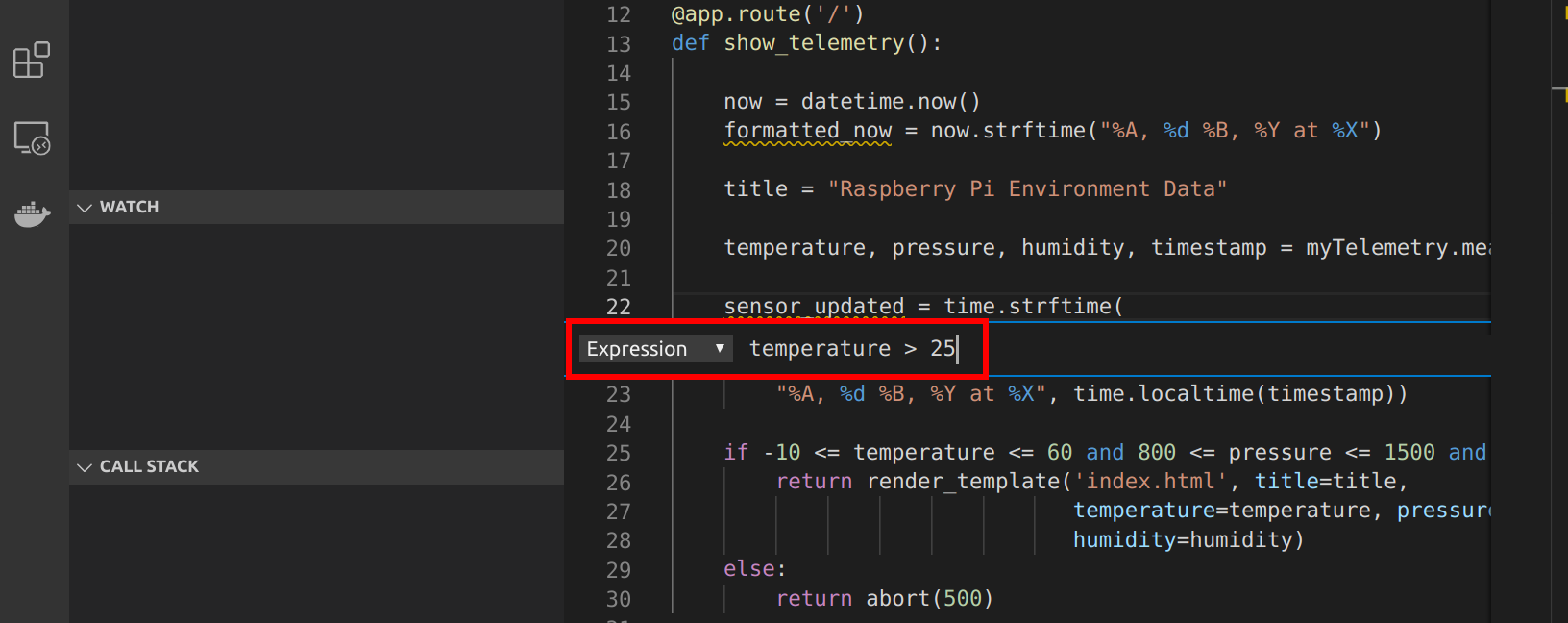
使用条件断点进行调试

尝试设置**条件**断点

1. 清除现有断点。从主菜单中选择**Debug** > **Remove all breakpoints**。
2. 确保打开**app.py**文件。
3. **右键单击**第**22**行**左侧**的边距。



1. 选择**添加条件断点...**
2. 将条件设置为**> 25**，然后按**ENTER**



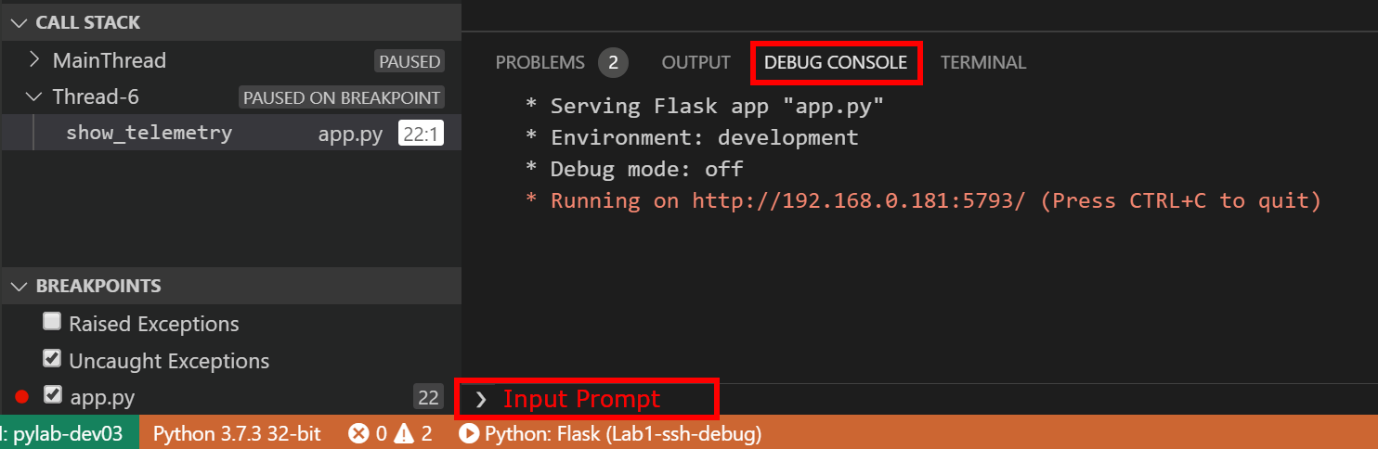
断点显示为红点，中间带有等号

1. 切换回**Web浏览器**并单击“ **刷新”**。由于调试器已在您设置的断点处停止，因此网页**不会响应**。
2. **切换**回**Visual Studio代码**，您将看到调试器已在**条件断点**处停止。
3. 按**F5**继续运行代码
4. **切换**回**Web浏览器**以查看该页面。

交互式调试控制台

Visual Studio代码**调试控制台**将允许您访问[Python REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop)（读取，评估，打印循环）。

1. **切换**回**Web浏览器**并单击“刷新”。由于调试器已停止Python代码，因此网页**不会响应**。
2. **切换**回**Visual Studio代码**
3. 代码将在您先前设置的条件断点处停止。
4. 选择Visual Studio **调试控制台**窗口。



1. 在输入提示**>中**键入以下Python代码
2. **print**(temperature)
3. 按**Enter键**以执行您键入的Python代码。
4. 尝试从输入提示符运行以下Python代码段。

temperature **=** 24

**import** random

random**.**randrange(100, 1000)

1. 按**F5**继续执行Python代码。
2. 切换回您的Web浏览器以查看更新的页面。

实验室挑战

实验室挑战1：更新Flask模板

1. 更新**templates**文件夹中的Flask **index.html**模板以显示当前日期和时间。
2. 重新运行Flask应用程序。

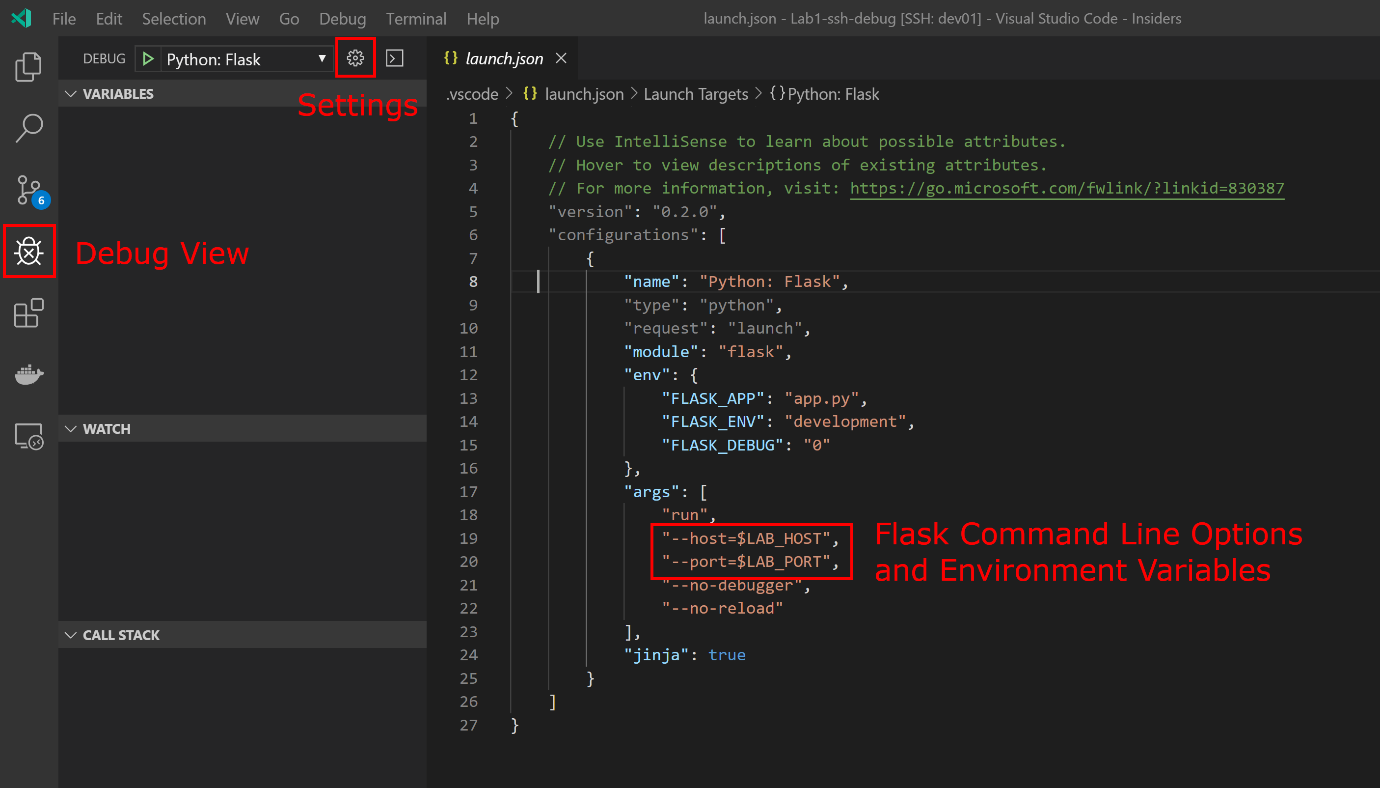
实验室挑战2：使用调试器选项进行实验

要尝试的事情：

1. 查看[Visual Studio Code Python教程](https://code.visualstudio.com/docs/python/python-tutorial?WT.mc_id=pycon-blog-dglover)
2. 查看[Python Flask教程](https://vscode-westeu.azurewebsites.net/docs/python/tutorial-flask)
3. 查看[Visual Studio代码调试教程](https://code.visualstudio.com/docs/editor/debugging?WT.mc_id=pycon-blog-dglover)

查看调试**启动**设置

1. 在Visual Studio代码中切换到Debug视图（使用左侧活动栏）。



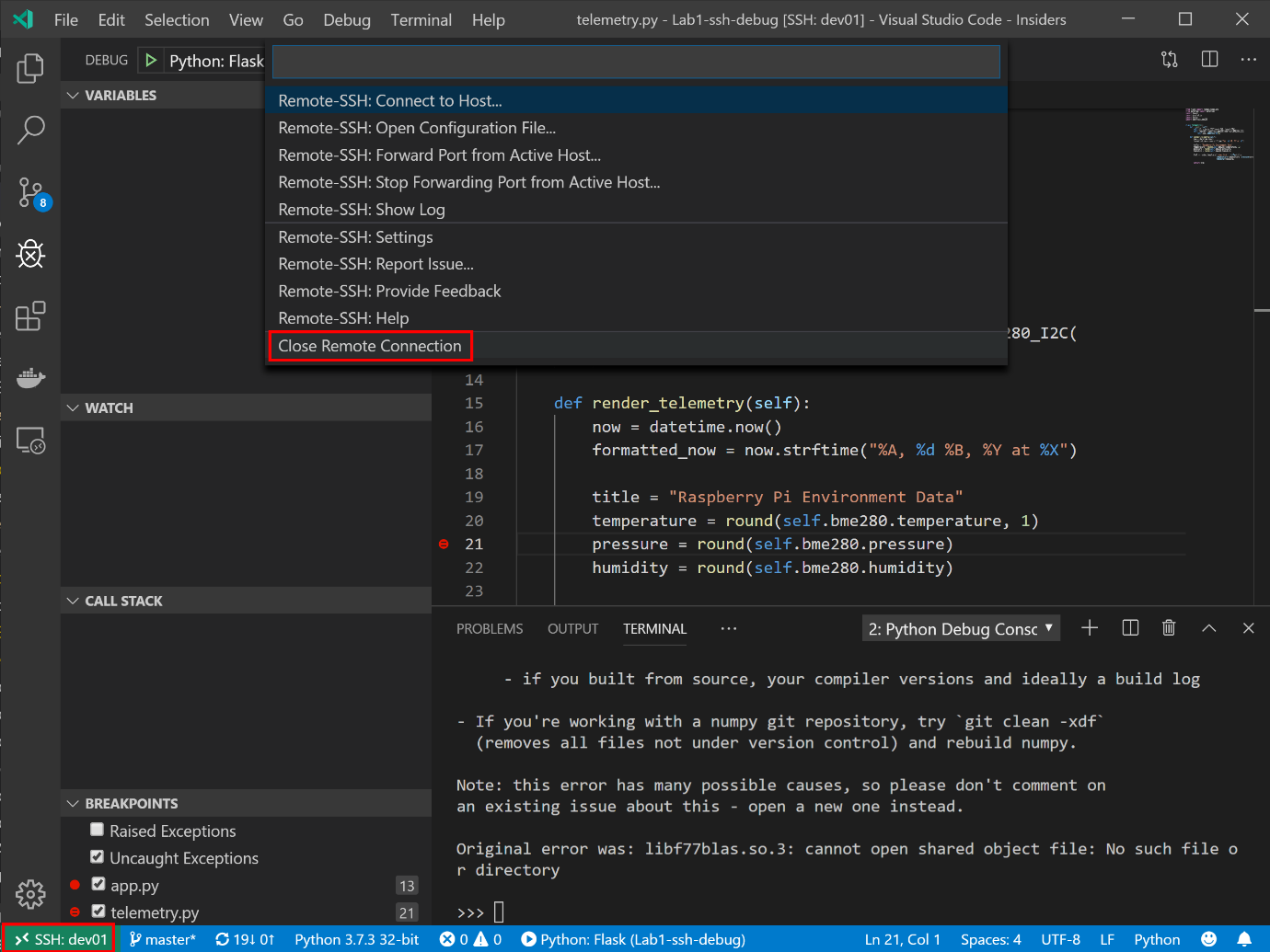
1. 单击“ **设置”**按钮，将打开**launch.json**文件。
2. 该**launch.json**文件定义了应用程序瓶将如何开始，什么[瓶命令行](https://flask.palletsprojects.com/en/1.0.x/cli/)参数在启动时通过。

launch.json文件中使用了两个环境变量。这些是**LAB\_HOST**（它是Raspberry Pi的IP地址）和**LAB\_PORT**（5000到8000之间的随机TCP / IP端口号）。这些环境变量由.bashrc脚本设置，该脚本在使用Visual Studio远程SSH连接到Raspberry Pi时运行。

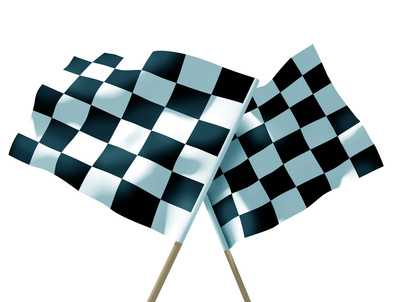
关闭远程SSH会话

从Visual Studio代码，**关闭远程连接**。

1. 单击**左下角**的**Remote SSH**按钮，然后从下拉列表中选择**Close Remote Connection**。



成品



参考

* [Visual Studio代码](https://code.visualstudio.com/?WT.mc_id=pycon-blog-dglover)
* [蟒蛇](https://www.python.org/)
* [覆盆子皮](https://www.raspberrypi.org/)
* [烧瓶](https://www.fullstackpython.com/flask.html)

故障排除SSH客户端安装

* [使用SSH进行远程开发](https://code.visualstudio.com/docs/remote/ssh?WT.mc_id=pycon-blog-dglover)
* [安装支持的SSH客户端](https://code.visualstudio.com/docs/remote/troubleshooting#_installing-a-supported-ssh-client?WT.mc_id=pycon-blog-dglover)

[**PyLab-1-Debugging-a-Python-Internet-of-Things-Application**](https://github.com/gloveboxes/PyLab-1-Debugging-a-Python-Internet-of-Things-Application)**由**[**手套箱**](https://github.com/gloveboxes)**维护。**