Qwiklabs Assessment: Debugging Python Scripts

**Introduction**

Imagine one of your colleagues has written a Python script that's failing to run correctly. They're asking for your help to debug it. In this lab, you'll look into why the script is crashing and apply the problem-solving steps that we've already learned to get information, find the root cause, and remediate the problem.

**You'll have 90 minutes to complete this lab.**

# **Debugging Python Scripts**

linux-instance external IP address



content\_copy

username



content\_copy

[file\_download](https://googlecoursera.qwiklabs.com/lab_instances/download_pem/15725750" \t "_blank)

[Download PEM](https://googlecoursera.qwiklabs.com/lab_instances/download_pem/15725750" \t "_blank)

[file\_download](https://googlecoursera.qwiklabs.com/lab_instances/download_ppk/15725750" \t "_blank)

[Download PPK](https://googlecoursera.qwiklabs.com/lab_instances/download_ppk/15725750" \t "_blank)

## Introduction

Imagine one of your colleagues has written a Python script that's failing to run correctly. They're asking for your help to debug it. In this lab, you'll look into why the script is crashing and apply the problem-solving steps that we've already learned to get information, find the root cause, and remediate the problem.

You'll have 90 minutes to complete this lab.

### **Start the lab**

You'll need to start the lab before you can access the materials in the virtual machine OS. To do this, click the green “Start Lab” button at the top of the screen.

**Note:** For this lab you are going to access the **Linux VM** through your **local SSH Client**, and not use the **Google Console** (**Open GCP Console** button is not available for this lab).

Start Lab

After you click the “Start Lab” button, you will see all the SSH connection details on the left-hand side of your screen. You should have a screen that looks like this:



## Accessing the virtual machine

Please find one of the three relevant options below based on your device's operating system.

**Note:** Working with Qwiklabs may be similar to the work you'd perform as an **IT Support Specialist**; you'll be interfacing with a cutting-edge technology that requires multiple steps to access, and perhaps healthy doses of patience and persistence(!). You'll also be using **SSH** to enter the labs -- a critical skill in IT Support that you’ll be able to practice through the labs.

### **Option 1: Windows Users: Connecting to your VM**

In this section, you will use the PuTTY Secure Shell (SSH) client and your VM’s External IP address to connect.

**Download your PPK key file**

You can download the VM’s private key file in the PuTTY-compatible **PPK** format from the Qwiklabs Start Lab page. Click on **Download PPK**.



**Connect to your VM using SSH and PuTTY**

1. You can download Putty from [here](https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe)
2. In the **Host Name (or IP address)** box, enter username@external\_ip\_address.

**Note:** Replace **username** and **external\_ip\_address** with values provided in the lab.



1. In the **Category** list, expand **SSH**.
2. Click **Auth** (don’t expand it).
3. In the **Private key file for authentication** box, browse to the PPK file that you downloaded and double-click it.
4. Click on the **Open** button.

**Note:** PPK file is to be imported into PuTTY tool using the Browse option available in it. It should not be opened directly but only to be used in PuTTY.



1. Click **Yes** when prompted to allow a first connection to this remote SSH server. Because you are using a key pair for authentication, you will not be prompted for a password.

**Common issues**

If PuTTY fails to connect to your Linux VM, verify that:

* You entered **<username>**@**<external ip address>** in PuTTY.
* You downloaded the fresh new PPK file for this lab from Qwiklabs.
* You are using the downloaded PPK file in PuTTY.

### **Option 2: OSX and Linux users: Connecting to your VM via SSH**

**Download your VM’s private key file.**

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



**Connect to the VM using the local Terminal application**

A **terminal** is a program which provides a **text-based interface for typing commands**. Here you will use your terminal as an SSH client to connect with lab provided Linux VM.

1. Open the Terminal application.
   * To open the terminal in Linux use the shortcut key **Ctrl+Alt+t**.
   * To open terminal in **Mac** (OSX) enter **cmd + space** and search for **terminal**.
2. Enter the following commands.

**Note:** Substitute the **path/filename for the PEM** file you downloaded, **username** and **External IP Address**.

You will most likely find the PEM file in **Downloads**. If you have not changed the download settings of your system, then the path of the PEM key will be **~/Downloads/qwikLABS-XXXXX.pem**

chmod 600 ~/Downloads/qwikLABS-XXXXX.pem

ssh -i ~/Downloads/qwikLABS-XXXXX.pem username@External Ip Address



### **Option 3: Chrome OS users: Connecting to your VM via SSH**

**Note:** Make sure you are not in **Incognito/Private mode** while launching the application.

**Download your VM’s private key file.**

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



**Connect to your VM**

1. Add Secure Shell from [here](https://chrome.google.com/webstore/detail/secure-shell-app/pnhechapfaindjhompbnflcldabbghjo) to your Chrome browser.
2. Open the Secure Shell app and click on **[New Connection]**.



1. In the **username** section, enter the username given in the Connection Details Panel of the lab. And for the **hostname** section, enter the external IP of your VM instance that is mentioned in the Connection Details Panel of the lab.



1. In the **Identity** section, import the downloaded PEM key by clicking on the **Import…** button beside the field. Choose your PEM key and click on the **OPEN** button.

**Note:** If the key is still not available after importing it, refresh the application, and select it from the **Identity** drop-down menu.

1. Once your key is uploaded, click on the **[ENTER] Connect** button below.



1. For any prompts, type **yes** to continue.
2. You have now successfully connected to your Linux VM.

You're now ready to continue with the lab!

## Reproduce the error

The script sent by your colleague is in the **scripts** directory. Let's navigate to **scripts** directory using the following command:

cd ~/scripts

Now, use the following command to list all the files in this directory:

ls

You should now be able to see the file named **greetings.py** which was sent by your colleague.

To view the contents of the file, use the following command:

cat greetings.py

Let's update the file's permissions.

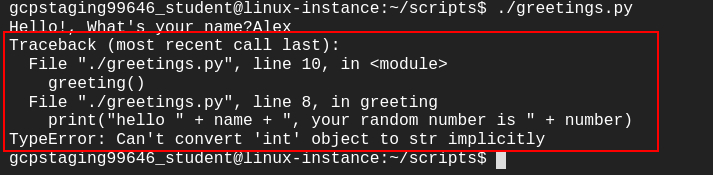
sudo chmod 777 greetings.py

Now let's reproduce the error by running the file using the following command:

./greetings.py

Enter your name at the prompt.

The output should throw an error as shown below:

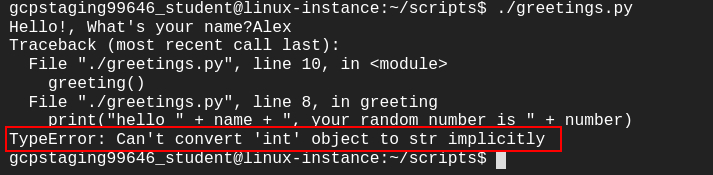


Great job! You have successfully reproduced the error.

## Find the root cause of the issue

Now that we have successfully reproduced the error, let's find its root cause.

The error message indicates that something in the code is trying to concatenate a string and an integer.



When we look at the code, we can see that there are two different data types used, string and int. The variable **name** takes string values and the variable **number** stores integer (int) values.

So, the print statement within the script concatenates both string and integer values, which is causing the error.

print("hello " + name + ", your random number is " + number)

So, we can conclude that the root cause of the issue is within the print statement, which is trying to concatenate two different data types (i.e., string and int).

## Debug the issue

In the previous section, we found the root cause of the issue, now let's debug the issue.

The print statement within the script is trying to concatenate two different data types. In Python, you can't add two different data types directly. So in this case, we can't add a string data type with an int data type. To add them, we have to turn the number into a string using str() function.

Once the integer value is converted to a string data type using str() function, the concatenate operation will work because the data types will be similar.

str() function takes in an integer as a parameter and converts it into string data type. So in our case, we will pass the variable **number** to the str() function i.e., **str(number)**.

Open the greetings.py file using nano editor.

nano greetings.py

Replace the print statement within the script with the following statement:

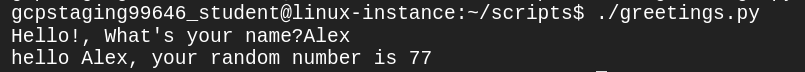
print("hello " + name + ", your random number is " + str(number))

Save the file by pressing Ctrl-o, followed by the Enter key and Ctrl-x.

Now run the file again.

./greetings.py

Enter your name for the prompt. You should now see the correct output.



Click Check my progress to verify the objective.

Debug the issue

Check my progress

## Congratulations!

You successfully debugged your colleague's python script. You reproduced the error, found its root cause, and applied the remediation to the issue. You can now close the RDP/SSH window. The lab will automatically end when the time runs out, or you can end it manually.

## End your lab