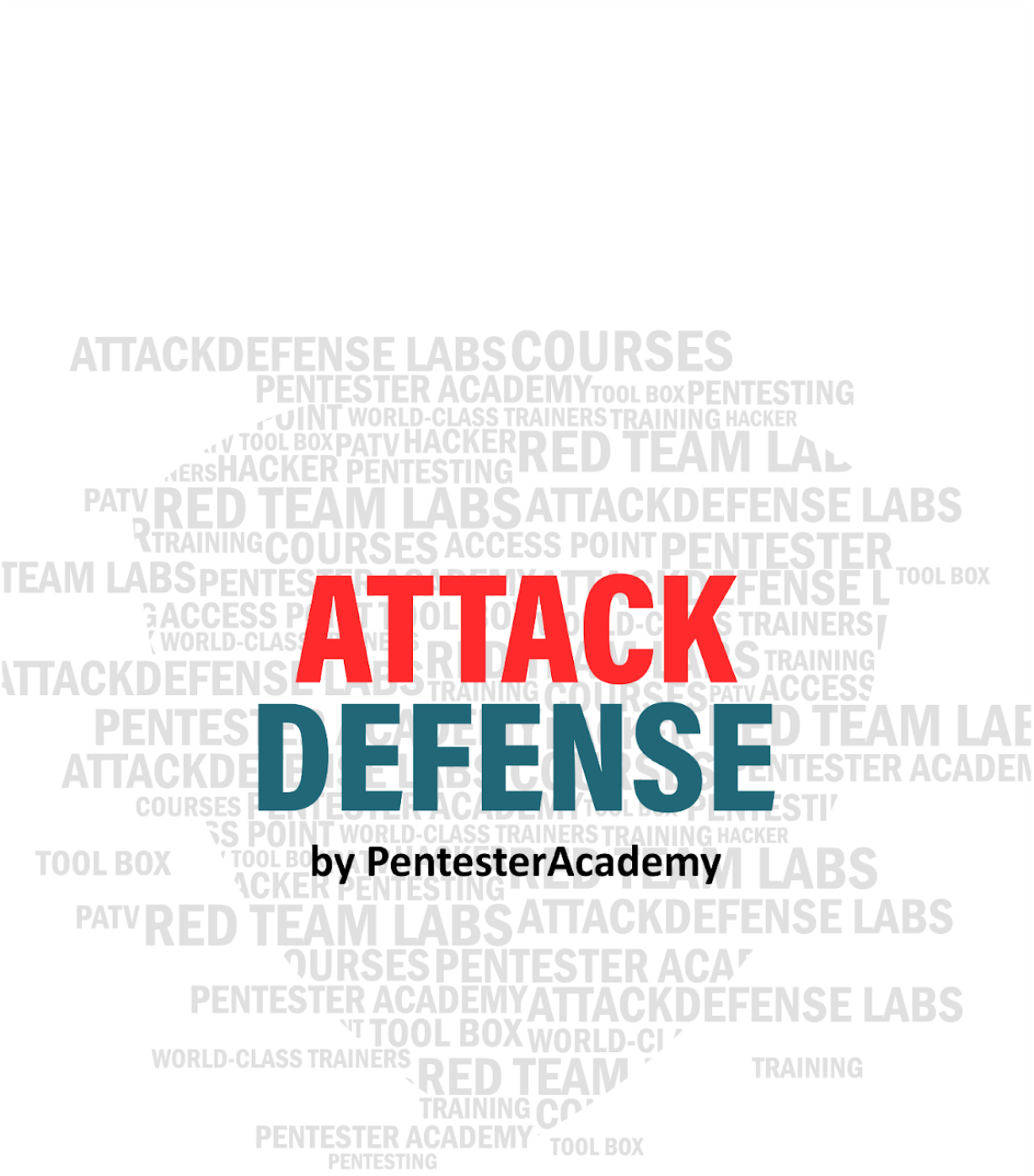

ATTACK
DEFENSE
by PentesterAcademy

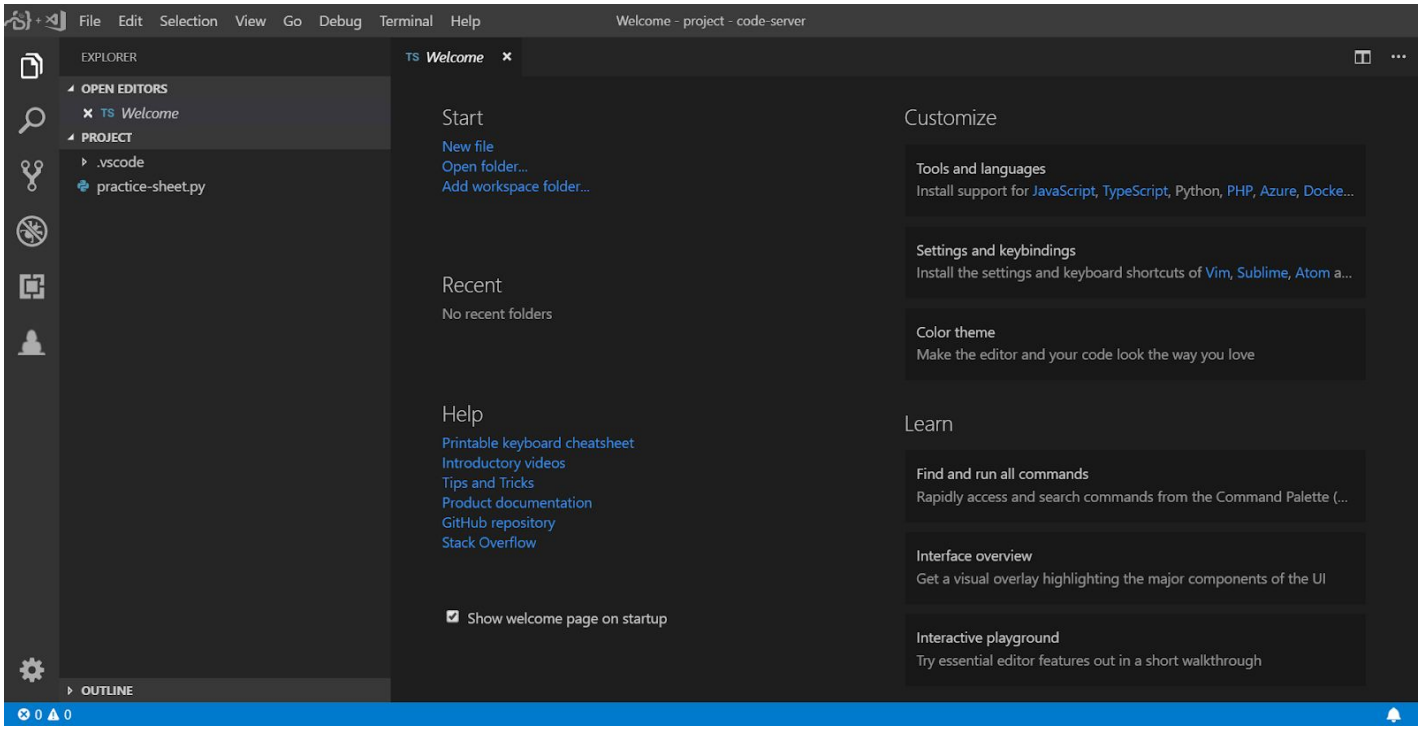
Name	Python IDE Warmup Lab
URL	https://attackdefense.com/challengedetails?cid=1209
Type	Offensive Python : Debugging

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

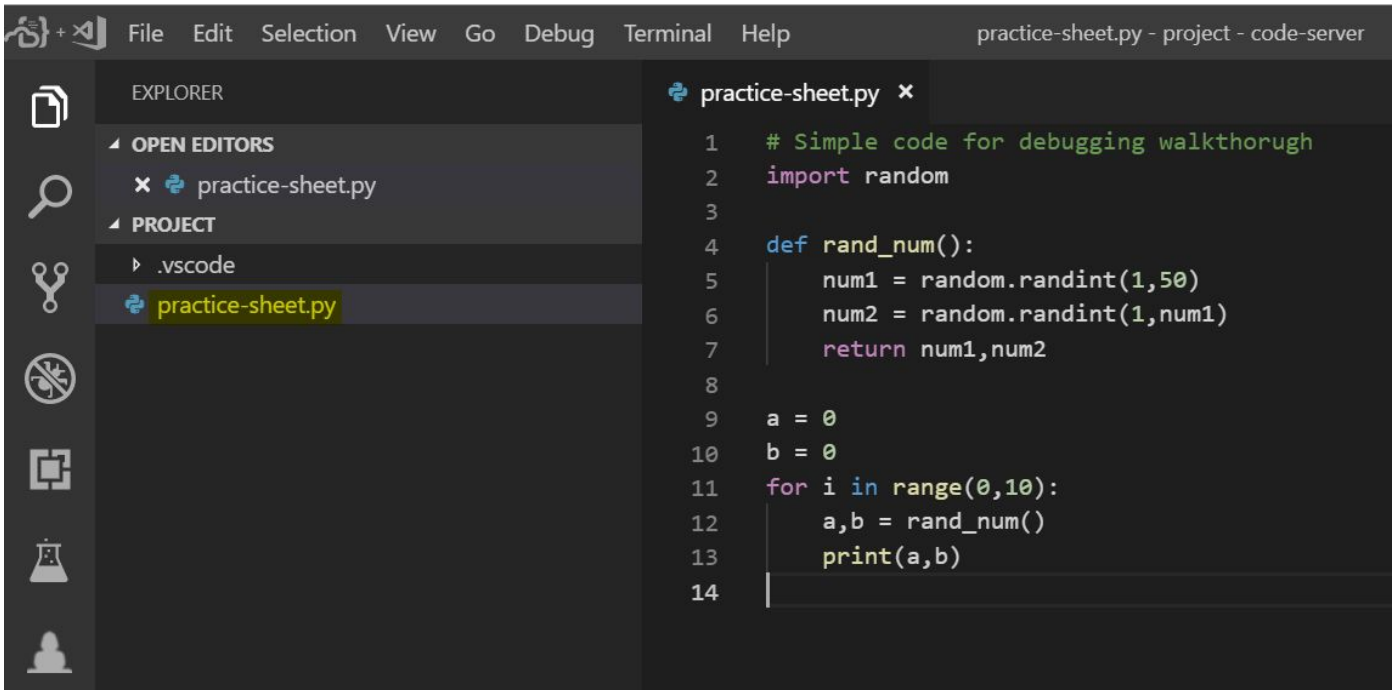
Objective: Use IDE features to interact and debug the Python script.

Solution:

Landing Page:

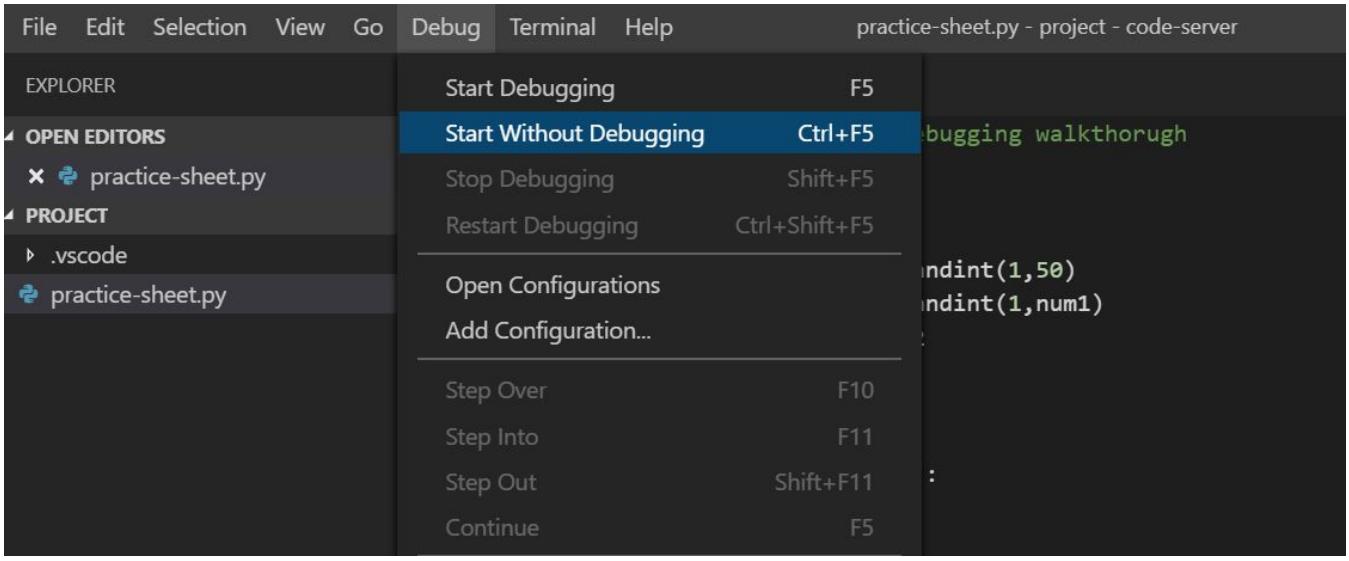


Step 1: Select “practice-sheet.py” from the Project Explorer.

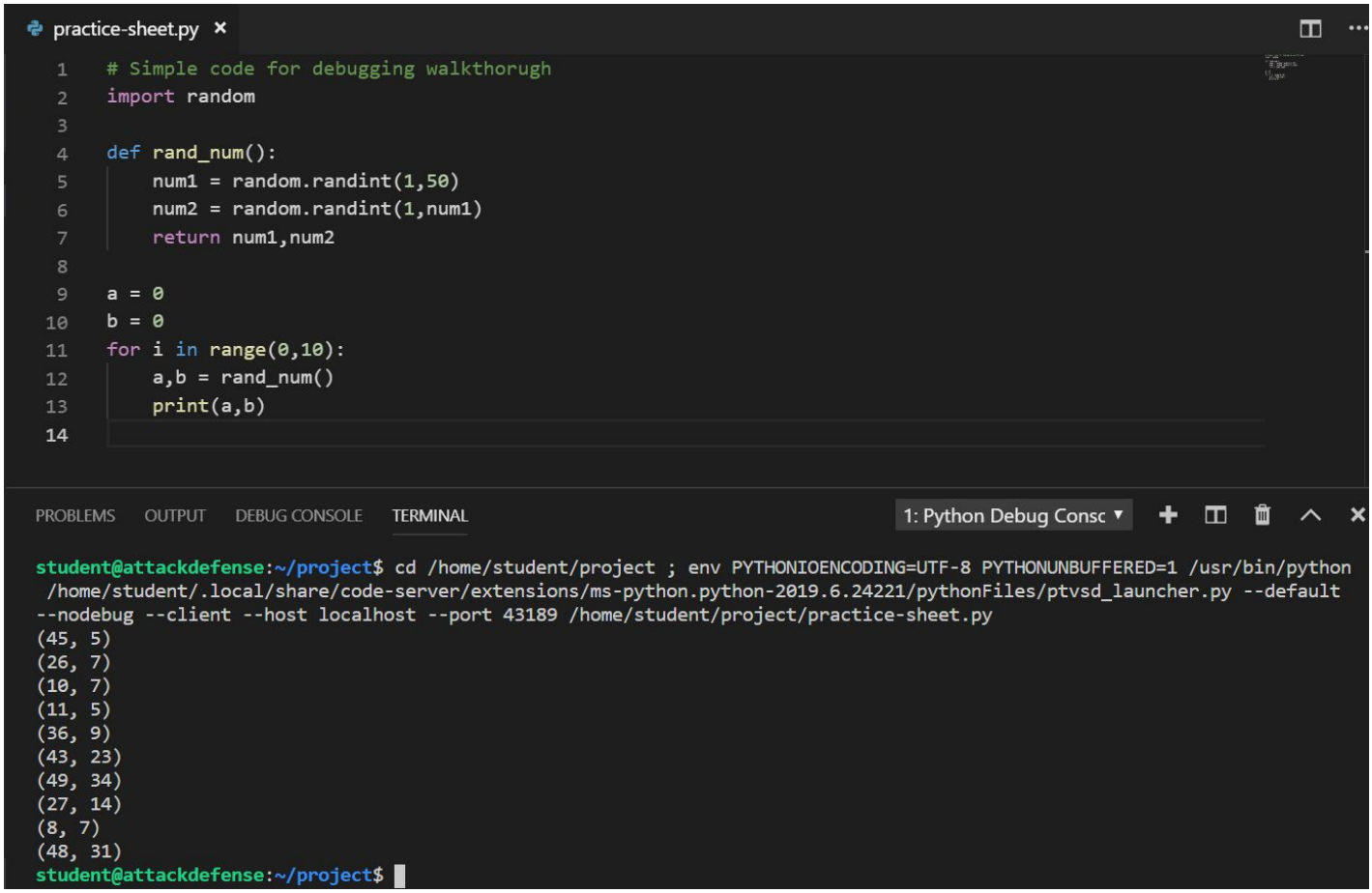


The python script will print 10 pairs of random numbers.

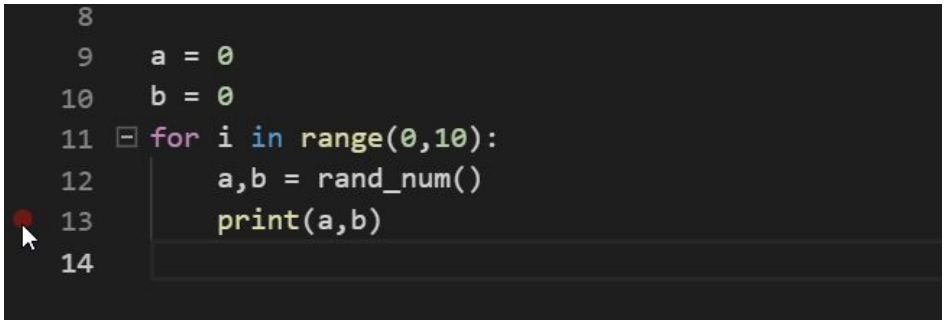
Step 2: Navigate to Debug Menu and click on “Start Without Debugging option”.



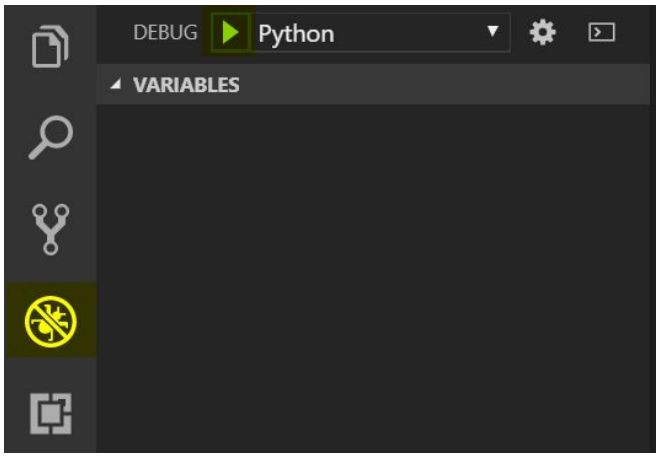
The python script will be executed and the output will be displayed on the integrated terminal.



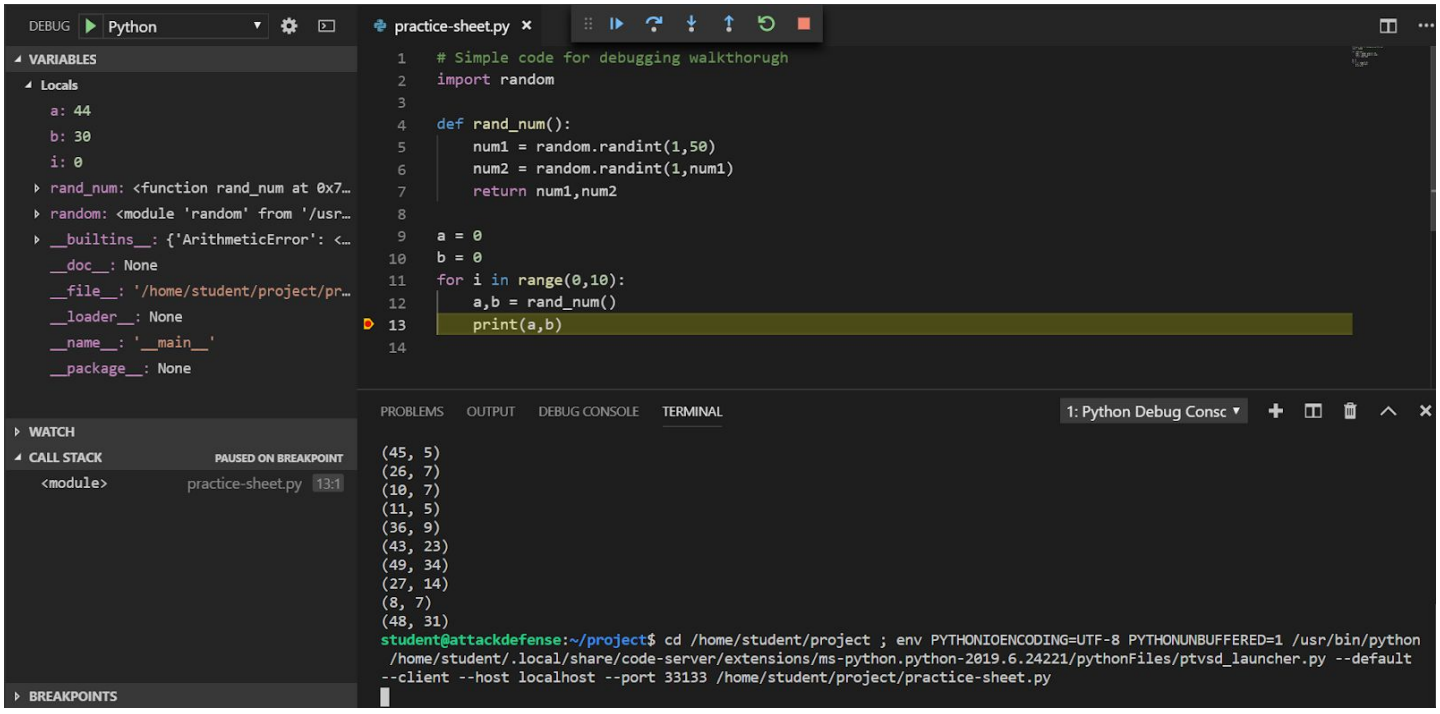
Step 3: Set a breakpoint on line 13. To set the breakpoint, click on the light red icon which appears upon hovering over space at the left side of the line numbers.



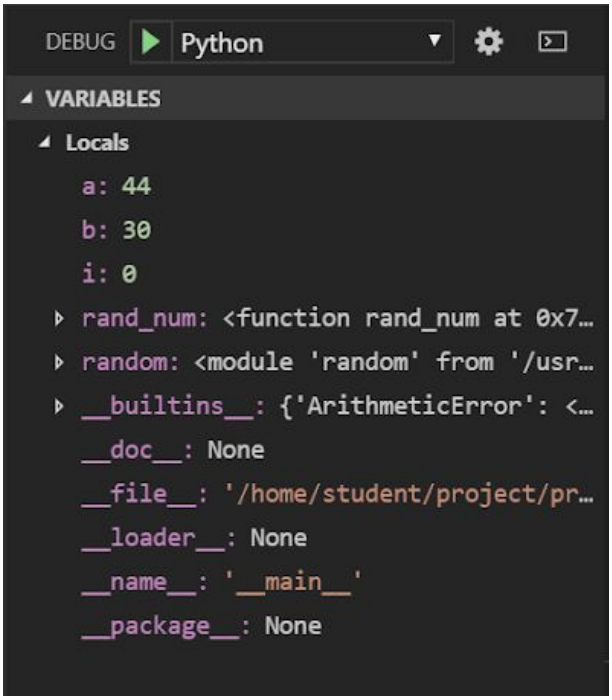
Step 4: Click on the Debug icon on the activity bar and click on Start Debugging button.



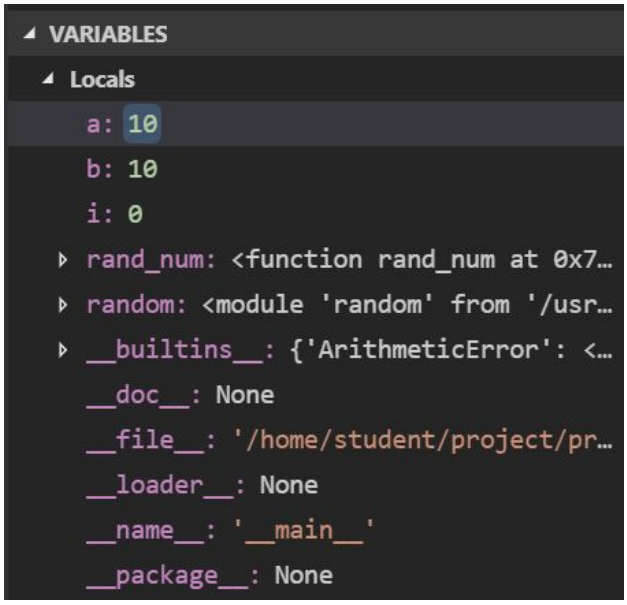
The program will run and the breakpoint will hit on line 13.



The value of variables will be listed in the Variables Section.

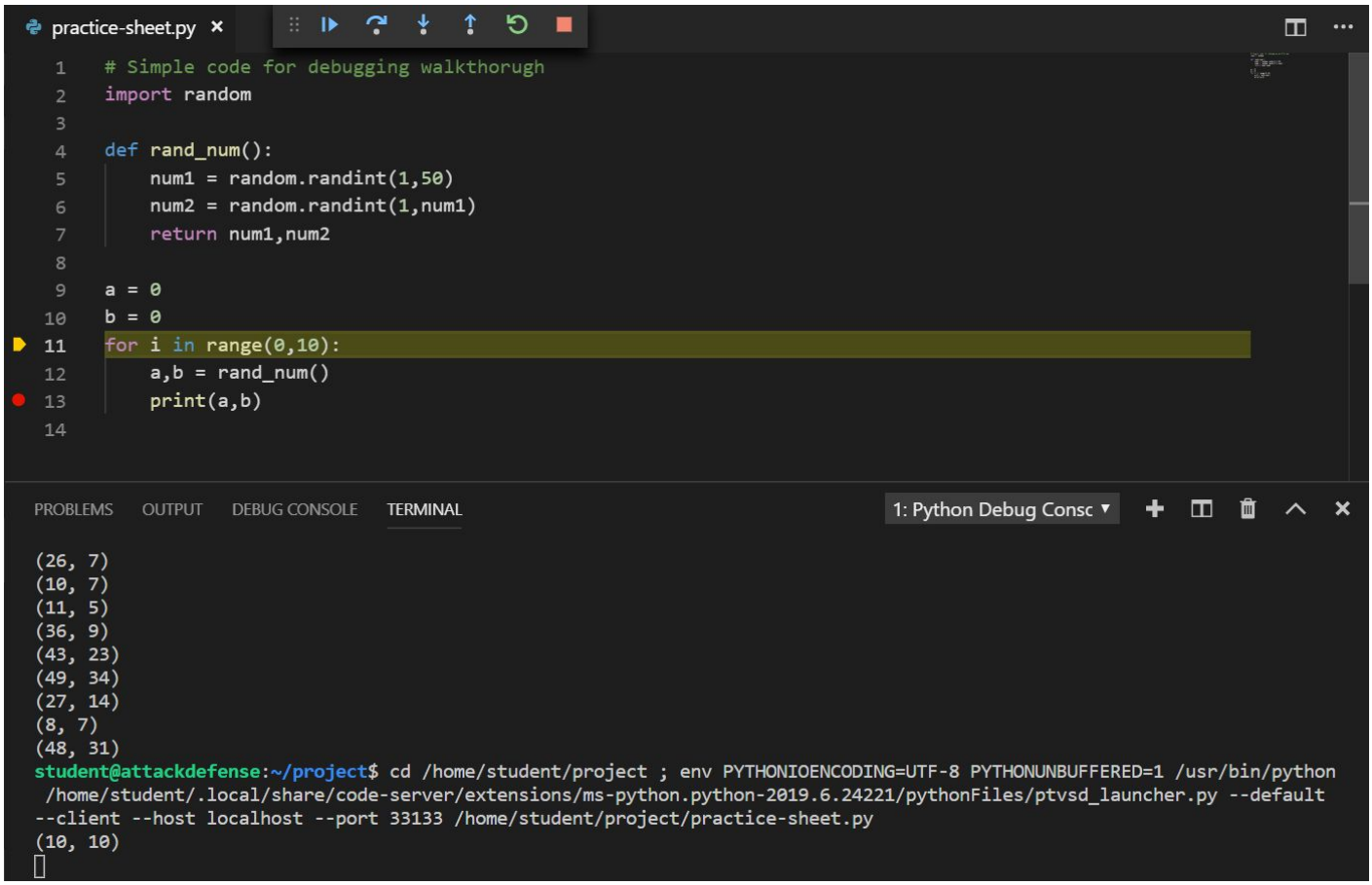


Step 5: Click on variable “a” and variable “b” in the Variables section and modify the value to 10 and 10 respectively.



Step 6: Select “Step Over” debug action to execute the current instruction.

The value of “a” and “b” variables will be printed on the integrated terminal.

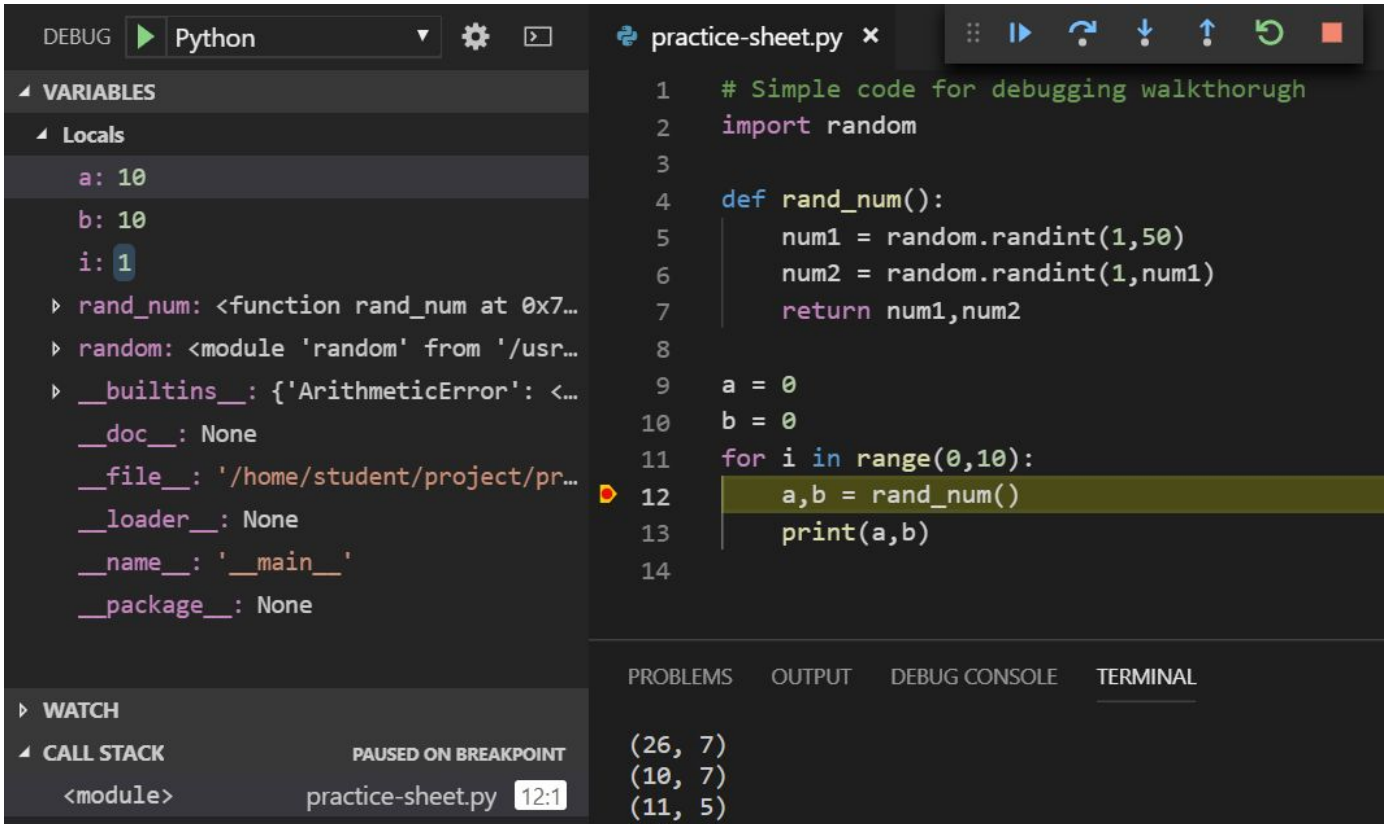


Step 7: Remove the breakpoint from line 13 and set a breakpoint on line 12.

```
 9   a = 0
10   b = 0
11   for i in range(0,10):
12       a,b = rand_num()
13       print(a,b)
14
```

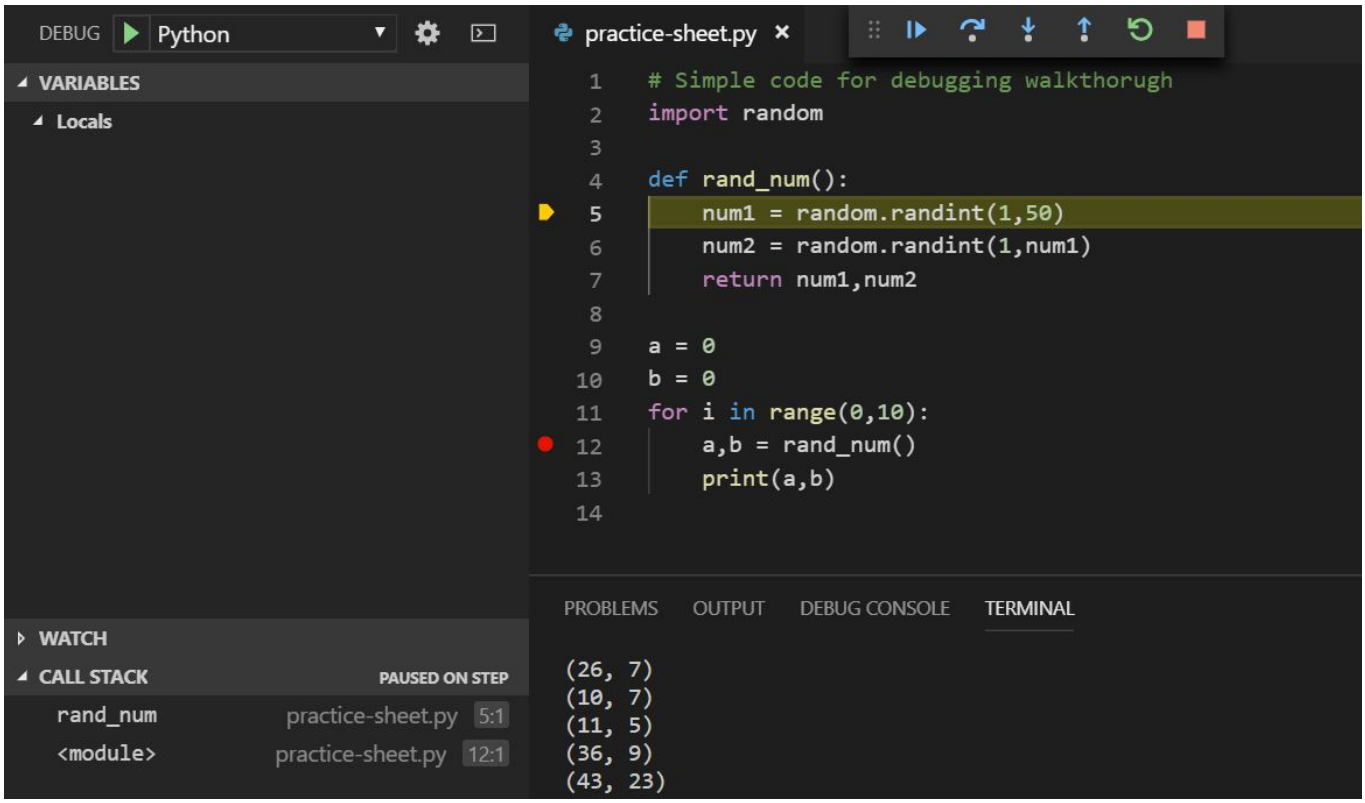
Step 8: Select “Step Over” or “Continue” debug action to hit the breakpoint.

The breakpoint will hit on line 12.

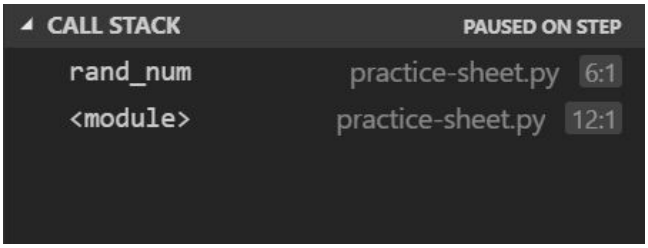


Step 9: Select “Step Into” debug action to step inside rand_num function.

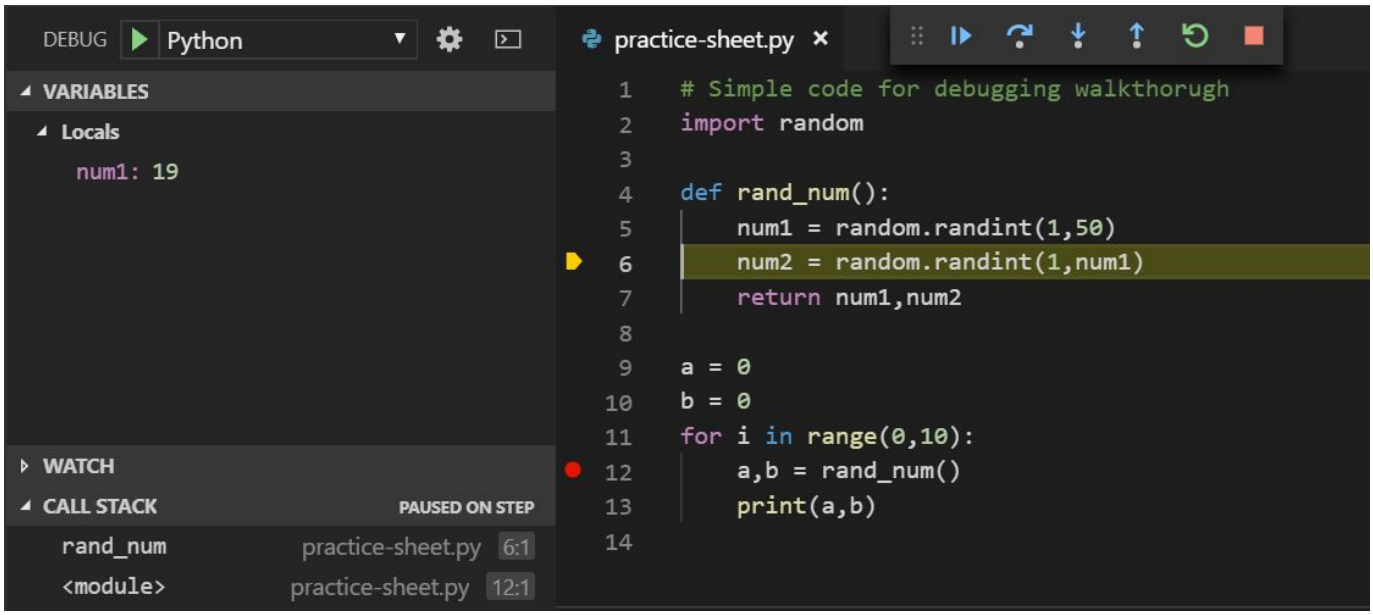
The debugger will go inside the function and line number 5 will be executed next.



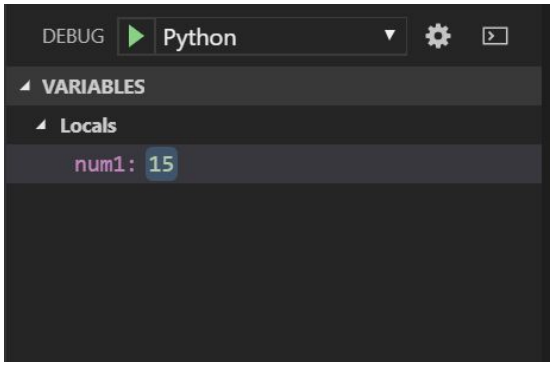
The function “rand_num” is listed in the Call Stack.



Step 10: Select “Step Over” debug action to execute the current instruction.

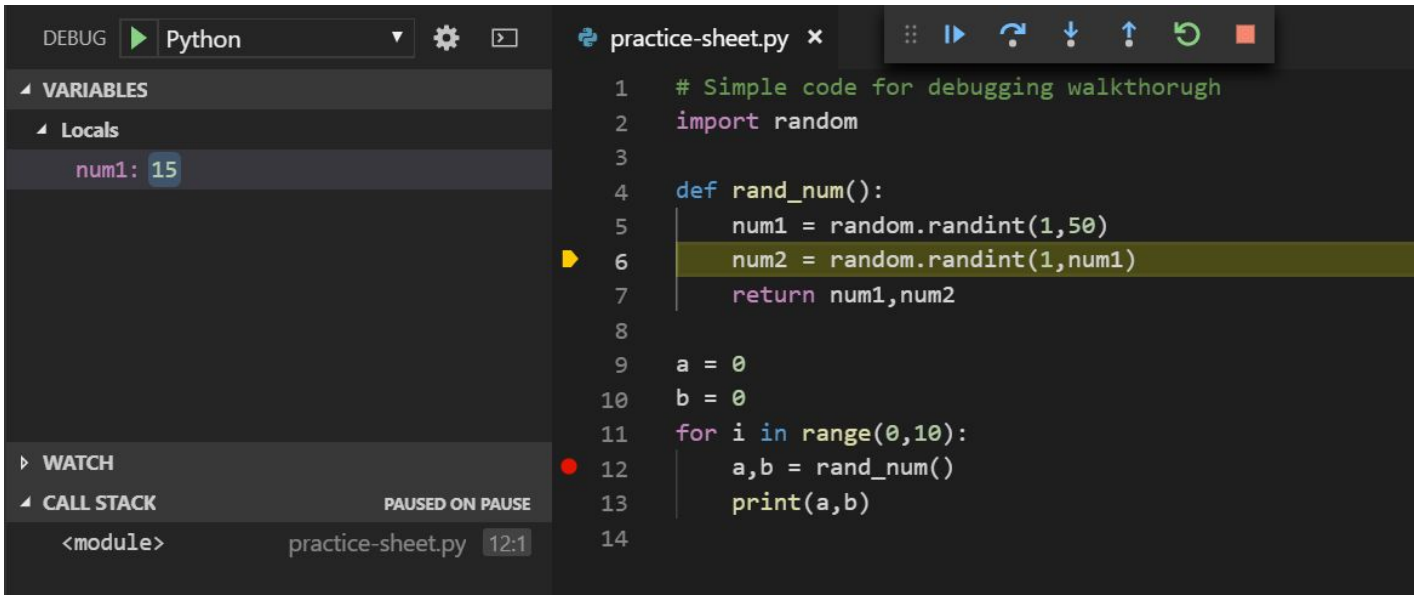


Step 11: Modify the value stored in “num1” variable to 15.

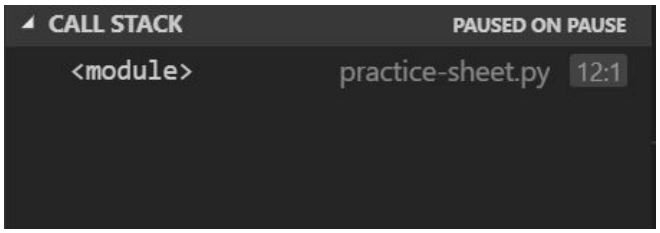


Step 12: Select “Step Out” debug action to step out of “rand_num” function.

The debugger will step out of the function and the call stack will be modified.

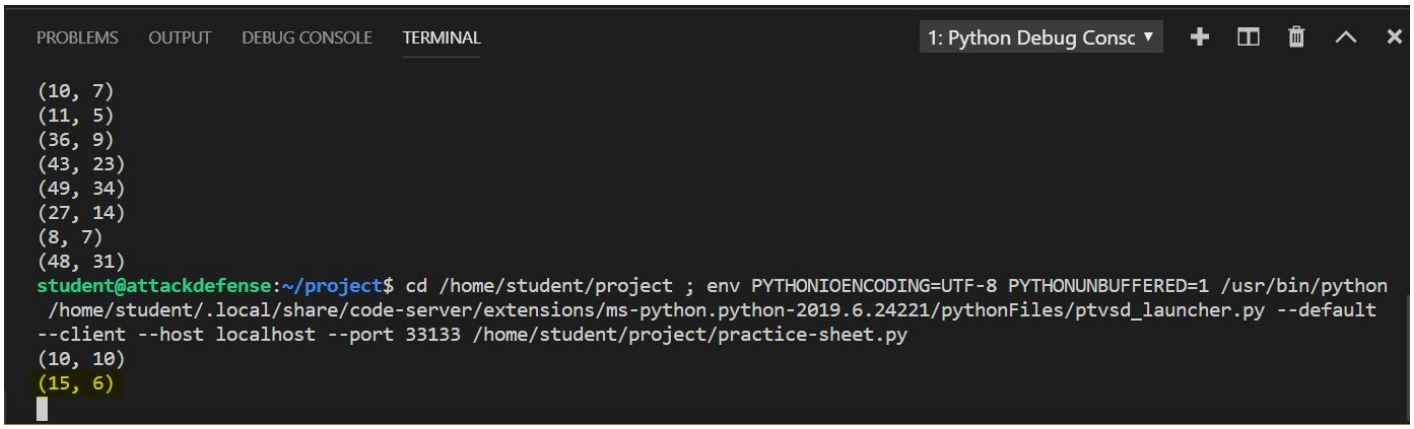


Call Stack section:



Step 13: Select the “Continue” debug action and the modified value will be printed on the integrated terminal.

Integrated Terminal:



References:

1. Visual Studio Code (https://code.visualstudio.com/)
2. VS Code Basic Editing (https://code.visualstudio.com/docs/editor/codebasics)