Debugging 3: Using a Debugger

SPD 2.3

Agenda

- Learning Outcomes
- Warm-Up: Read & Discuss
- TT: The VSCode Debugger
- Debugging Terminology
- Debugging Lab

Learning Outcomes

By the end of today, you should be able to…

1. Identify the functionality of the VSCode debugger.
2. Use the VSCode debugging functions to step through a program and identify bugs.

Warm-Up: Read & Discuss

Read & Discuss (10 minutes)

Read this article on 6 ways to improve your debugging. Then, in a group of 3, answer the following questions:

1. What is one technique from the article that you already use well?
2. What is one technique from the article that you could use more effectively?

The VSCode Debugger

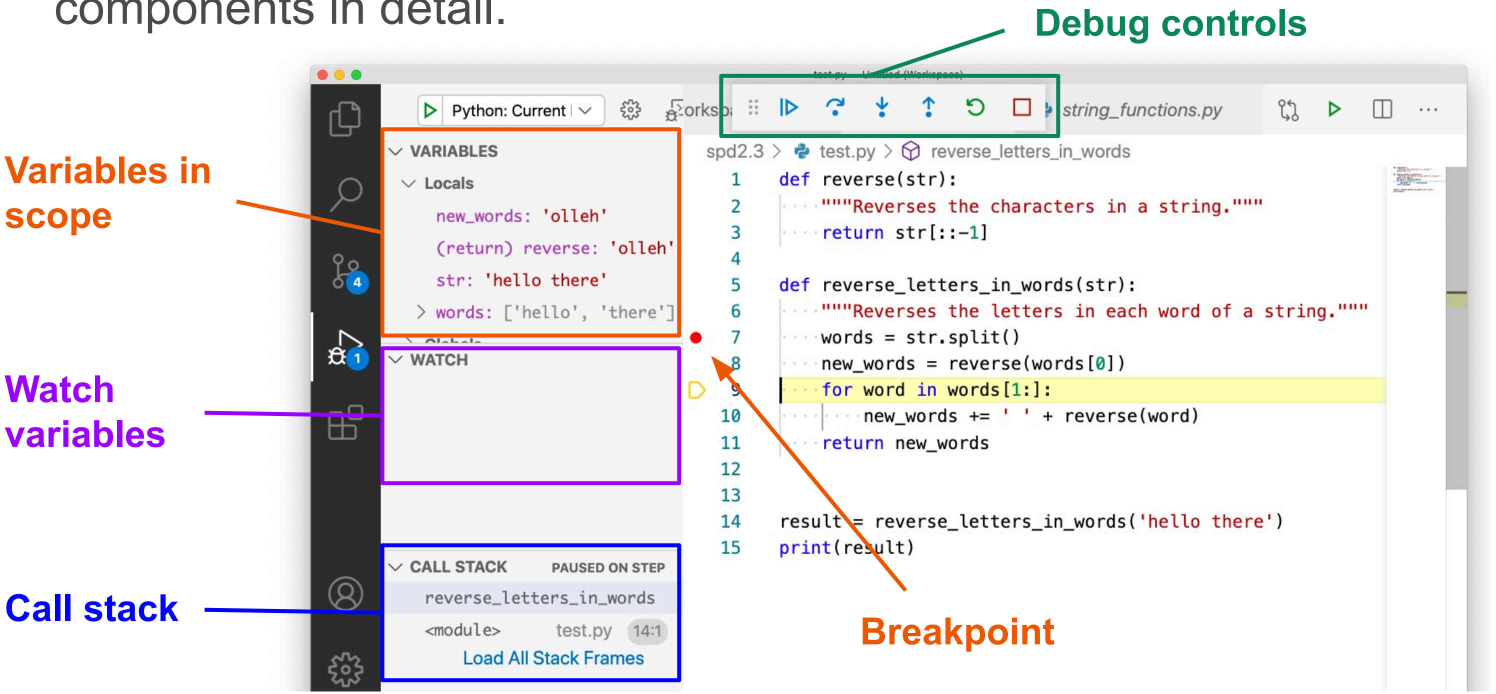
The Debugger

Using print statements to debug is kind of like using a small flashlight to "illuminate" your code: It gives you **one data point** that you can use to check assumptions.

Using a debugger is like using a floodlight - you can see the **entire state of the program** at once.

The Debugger

Here is what the debugger tool looks like. We'll go over each of these components in detail.



```
new_words: 'olleh'
(return) reverse: 'olleh'
str: 'hello there'
words: ['hello', 'there']
```

```
def reverse(str):
    """Reverses the characters in a string."""
    return str[::-1]

def reverse_letters_in_words(str):
    """Reverses the letters in each word of a string."""
    words = str.split()
    new_words = reverse(words[0])
    for word in words[1:]:
        new_words += ' ' + reverse(word)
    return new_words


result = reverse_letters_in_words('hello there')
print(result)
```

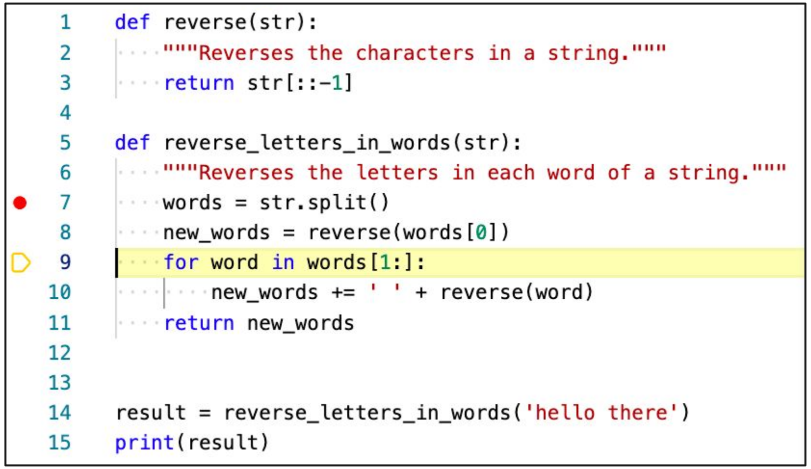
```
reverse_letters_in_words
<module>        test.py  14:1
```

Demo

Watch as your instructor demonstrates how to step through a small program using the debugger.

What is a Breakpoint?

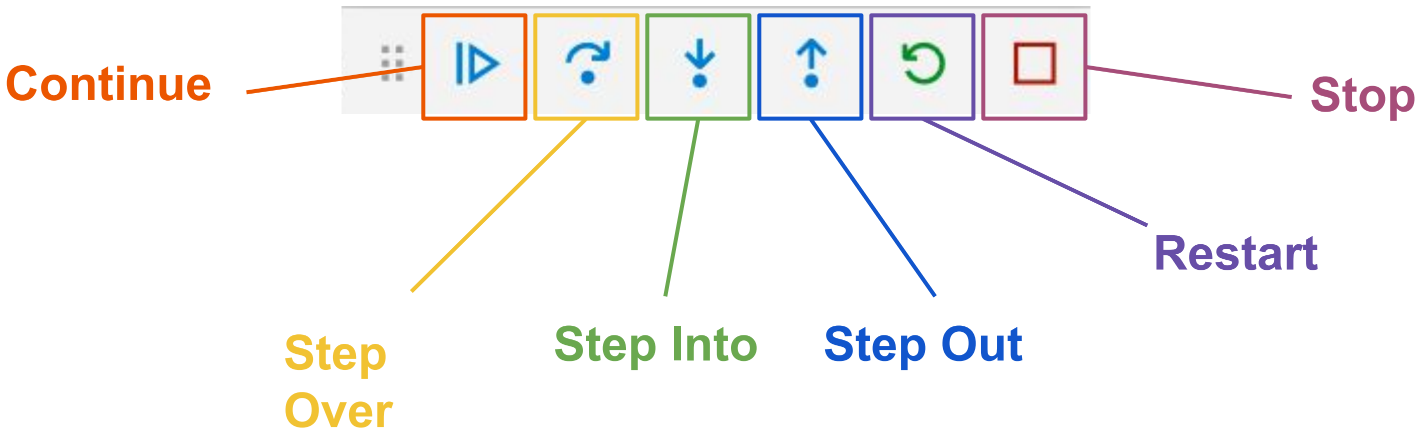
A **breakpoint** lets you specify where you want your program to pause execution. You can **set a breakpoint** by clicking the red 🔴 icon to the left of the line number.



```
   1  def reverse(str):
   2      """Reverses the characters in a string."""
   3      return str[::-1]
   4
   5  def reverse_letters_in_words(str):
   6      """Reverses the letters in each word of a string."""
●  7      words = str.split()
   8      new_words = reverse(words[0])
   9      for word in words[1:]:
  10          new_words += ' ' + reverse(word)
  11      return new_words
  12
  13
  14  result = reverse_letters_in_words('hello there')
  15  print(result)
```

The Controls

After your program reaches the breakpoint, you can use the controls to step through your code line-by-line. Let's go over these briefly.



Debugging Lab (50 minutes)

Complete the debugging lab with a partner. Make sure to use good pair programming practices! After each exercise, switch who is driver and navigator.