



[< Previous](#)



[Next >](#)

Activity: Debugging Exercise

[🔖 Bookmark this page](#)

Instructions

This activity provides an opportunity for you to practice what you learn in each lesson. This is an ungraded assignment. We strongly recommend that you complete the activity. Feel free to ask or answer questions in the Code talk discussion forum.

In the lesson content, we asked you to spend 5-10 minutes on your own debugging of the recursion error exercise code. Once you're finished, do the following:

1. In very general terms, use a couple of sentences to address the problem with our code. For example, give your best guess or insight on the following questions:

- What is wrong with our logic?
- Why doesn't the function stop calling itself?
- What's happening to the value of 'n' as the function gets deeper and deeper into recursion?

2. Include a copy-and-paste of your terminal debugging activity.

Sample Response

```
The function keeps calling itself and calling itself and it can't stop because it's a Tuesday today and Tuesdays are bad for recursion.
```

```
Below is my debugging log:
```

```
$ python -m pdb recursive.py
> c/users/jschilz/onedrive/py300/debugging/recursive.py(1)<module>()
-> import sys
(Pdb) ll
1 -> import sys
2
3
4 def my_fun(n):
5 if n == 2:
6 return True
7
8 return my_fun(n/2)
9
10
11 if __name__ == '__main__':
12 n = int(sys.argv[1])
13 print(my_fun(n))
(Pdb) n
> /users/jschilz/onedrive/py300/debugging/recursive.py(4)<module>()
-> def my_fun(n):
(Pdb)
```



Of course, your answer will be more insightful and your debugging log will be longer.

[< Previous](#)[Next >](#)



© 2024 University of Washington | Seattle, WA. All rights reserved.

[Help Center](#) [Contact Us](#) [Privacy](#) [Terms](#)

Built on  by RACCOONGANG 

edX, Open edX and the edX and Open edX logos are trademarks or registered trademarks of edX Inc.