

Self-test: Exceptions

1. The following (somewhat inefficient) function counts the number of lines in a file. However, if the file is not found, it creates a run-time error and the program terminates, because a `FileNotFoundError` is raised. Use a try block in order to handle this exception by just printing a line that says: "File could not be found".

```
def length_file(filename):  
    with open(filename) as infile:  
        return len(infile.readlines())
```

This is the current behavior:

```
>>> length_file("ttt")  
Traceback (most recent call last):  
  File "<pyshell#0>", line 1, in <module>  
    length_file("ttt")  
  File "/Users/thomasschwarz/Documents/My website/Classes/Python_Big_Data/Module  
22/selftest.py", line 3, in length_file  
    with open(filename) as infile:  
FileNotFoundError: [Errno 2] No such file or directory: 'ttt'  
>>> length_file("selftest.py")  
4  
|
```

This is how it should behave:

```
RESTART: /Users/thomasschwarz/Documents/My website/Classes/Python_Big_Data/Modu  
le22/selftest.py  
>>> length_file("selftest.py")  
7  
>>> length_file("ttt")  
File does not exist, you stupid person  
|
```

Solution:

```
def length_file(filename):  
    try:  
        with open(filename) as infile:  
            return len(infile.readlines())  
    except FileNotFoundError:  
        print("File does not exist, you stupid person")
```