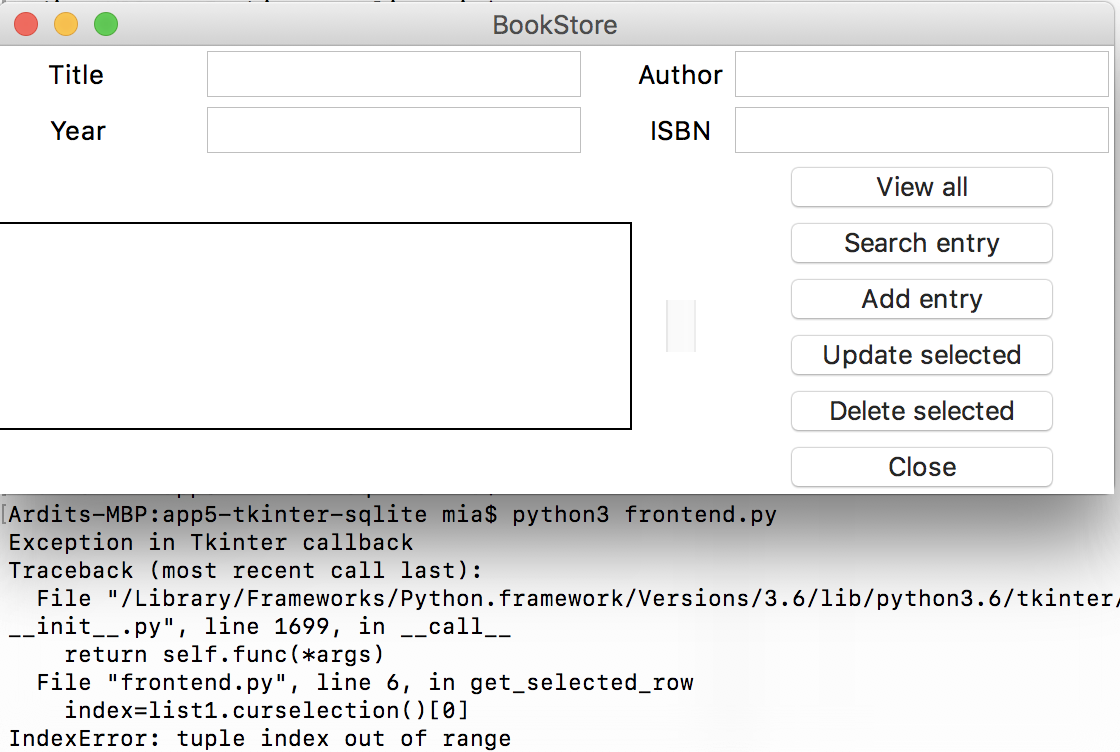
Fixing the Bug (Practice)

Exercise

If you haven't already noticed, the program has a bug. When the listbox is empty and the user clicks the listbox, an *IndexError* is generated in the terminal:



Why does this error happen?

Well, everything starts with the user clicking on the listbox. Clicking the listbox executes the following code:

list1.bind('<<ListboxSelect>>',get\_selected\_row)

That code calls the get\_selected\_row  function:

1. def get\_selected\_row(event):
2. global selected\_tuple
3. index=list1.curselection()[0]
4. selected\_tuple=list1.get(index)
5. e1.delete(0,END)
6. e1.insert(END,selected\_tuple[1])
7. e2.delete(0,END)
8. e2.insert(END,selected\_tuple[2])
9. e3.delete(0,END)
10. e3.insert(END,selected\_tuple[3])
11. e4.delete(0,END)
12. e4.insert(END,selected\_tuple[4])

Since the listbox is empty,  list1.curselection()  will be an empty list with no items. Trying to access the first item on the list with [0]  in line 3 will throw an error, because there is no first item in the list.

Try to fix that bug. The next lecture contains the solution.

Solution

Solution

1. def get\_selected\_row(event):
2. try:
3. global selected\_tuple
4. index=list1.curselection()[0]
5. selected\_tuple=list1.get(index)
6. e1.delete(0,END)
7. e1.insert(END,selected\_tuple[1])
8. e2.delete(0,END)
9. e2.insert(END,selected\_tuple[2])
10. e3.delete(0,END)
11. e3.insert(END,selected\_tuple[3])
12. e4.delete(0,END)
13. e4.insert(END,selected\_tuple[4])
14. except IndexError:
15. pass

Explanation

The error was fixed by simply implementing a try  and except  block. When the get\_selected\_row  function is called, Python will execute the indented block under try . If there is an *IndexError*, none of the lines under try  will be executed; the line under except  will be executed, which is pass. The pass  statement means "do nothing". Therefore the function will do nothing when there's an empty listbox.