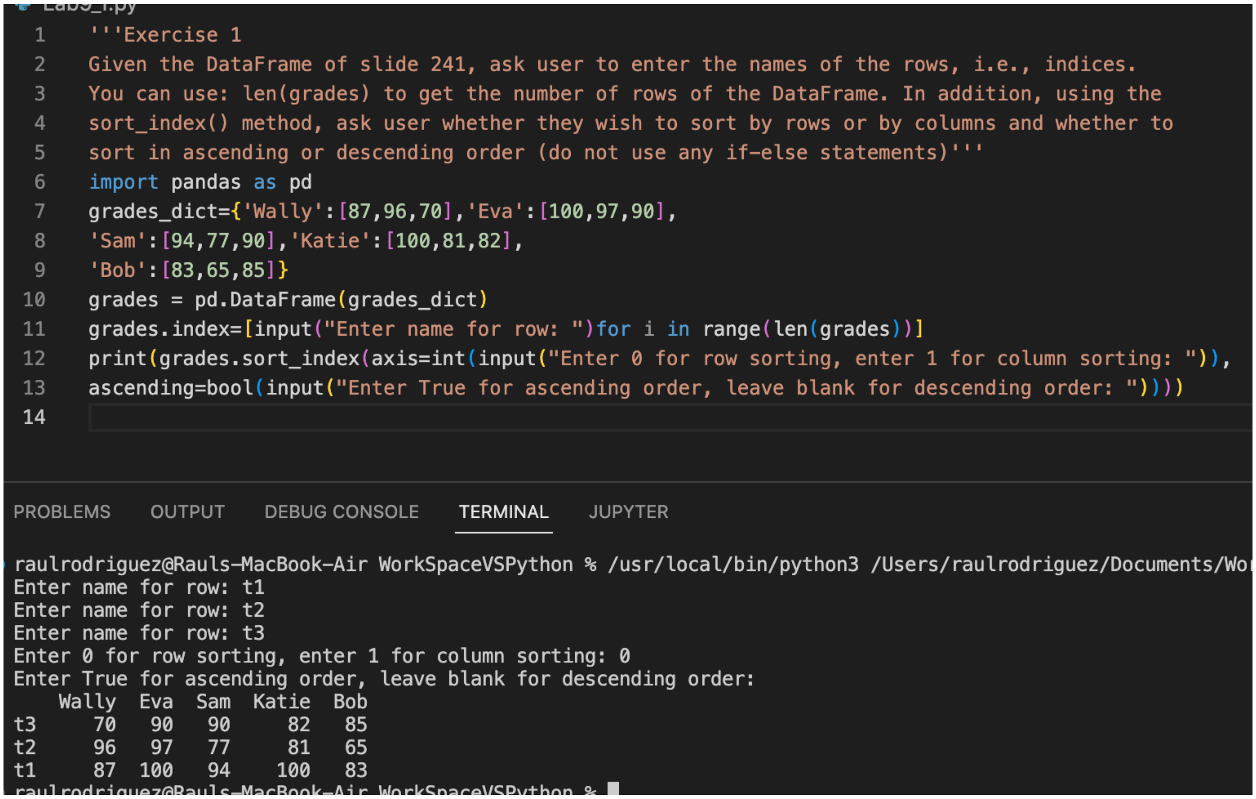
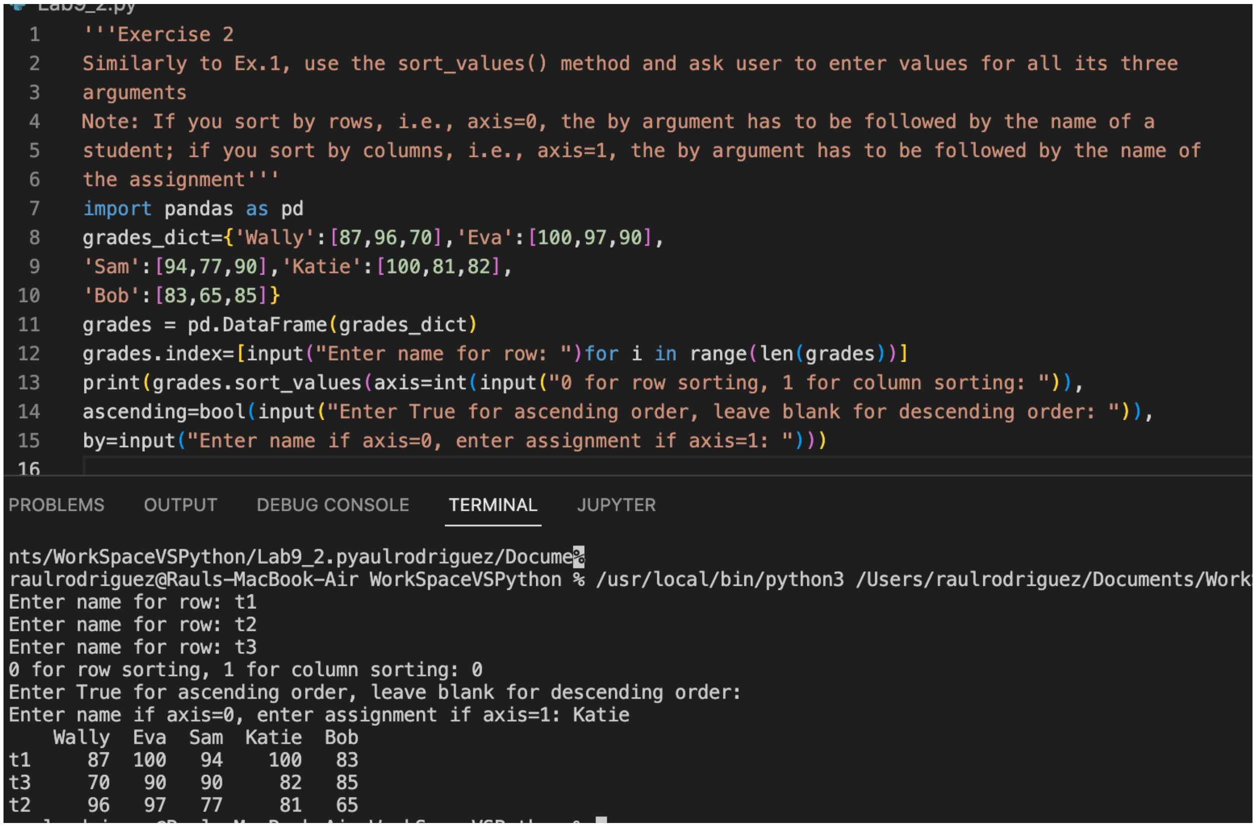
Raul Rodriguz



```python
'''Exercise 1
Given the DataFrame of slide 241, ask user to enter the names of the rows, i.e., indices.
You can use: len(grades) to get the number of rows of the DataFrame. In addition, using the
sort_index() method, ask user whether they wish to sort by rows or by columns and whether to
sort in ascending or descending order (do not use any if-else statements)'''
import pandas as pd
grades_dict={'Wally':[87,96,70],'Eva':[100,97,90],
'Sam':[94,77,90],'Katie':[100,81,82],
'Bob':[83,65,85]}
grades = pd.DataFrame(grades_dict)
grades.index=[input("Enter name for row: ")for i in range(len(grades))]
print(grades.sort_index(axis=int(input("Enter 0 for row sorting, enter 1 for column sorting: ")),
ascending=bool(input("Enter True for ascending order, leave blank for descending order: "))))

```

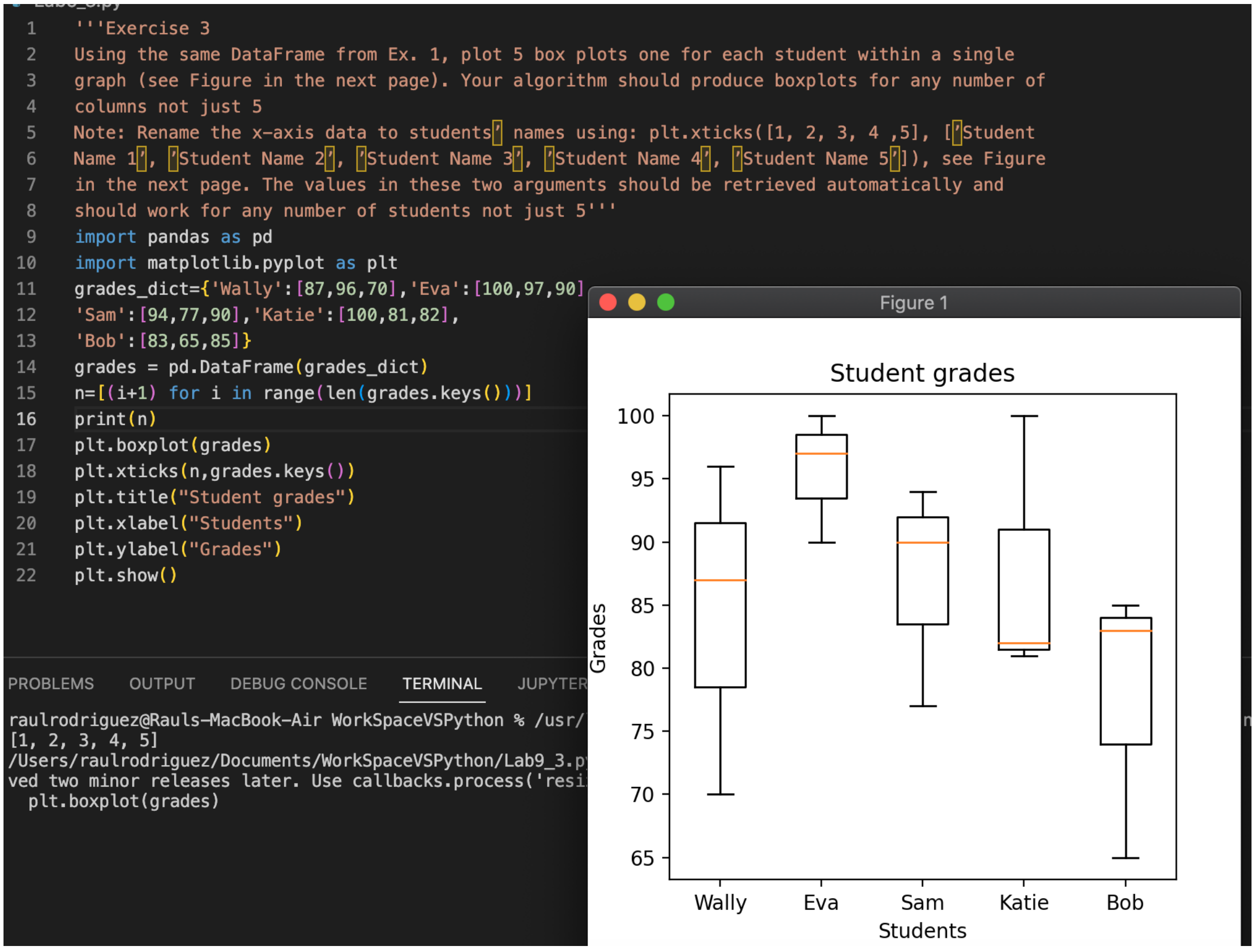
```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/Wo
Enter name for row: t1
Enter name for row: t2
Enter name for row: t3
Enter 0 for row sorting, enter 1 for column sorting: 0
Enter True for ascending order, leave blank for descending order:
    Wally  Eva  Sam  Katie  Bob
t3     70   90   90     82   85
t2     96   97   77     81   65
t1     87  100   94    100   83
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```



```python
'''Exercise 2
Similarly to Ex.1, use the sort_values() method and ask user to enter values for all its three
arguments
Note: If you sort by rows, i.e., axis=0, the by argument has to be followed by the name of a
student; if you sort by columns, i.e., axis=1, the by argument has to be followed by the name of
the assignment'''
import pandas as pd
grades_dict={'Wally':[87,96,70],'Eva':[100,97,90],
'Sam':[94,77,90],'Katie':[100,81,82],
'Bob':[83,65,85]}
grades = pd.DataFrame(grades_dict)
grades.index=[input("Enter name for row: ")for i in range(len(grades))]
print(grades.sort_values(axis=int(input("0 for row sorting, 1 for column sorting: ")),
ascending=bool(input("Enter True for ascending order, leave blank for descending order: ")),
by=input("Enter name if axis=0, enter assignment if axis=1: ")))

```

```
nts/WorkSpaceVSPython/Lab9_2.pyaulrodriguez/Docume%
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/Work
Enter name for row: t1
Enter name for row: t2
Enter name for row: t3
0 for row sorting, 1 for column sorting: 0
Enter True for ascending order, leave blank for descending order:
Enter name if axis=0, enter assignment if axis=1: Katie
    Wally  Eva  Sam  Katie  Bob
t1     87  100   94    100   83
t3     70   90   90     82   85
t2     96   97   77     81   65
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```



```python
'''Exercise 3
Using the same DataFrame from Ex. 1, plot 5 box plots one for each student within a single
graph (see Figure in the next page). Your algorithm should produce boxplots for any number of
columns not just 5
Note: Rename the x-axis data to students’ names using: plt.xticks([1, 2, 3, 4 ,5], [’Student
Name 1’, ’Student Name 2’, ’Student Name 3’, ’Student Name 4’, ’Student Name 5’]), see Figure
in the next page. The values in these two arguments should be retrieved automatically and
should work for any number of students not just 5'''
import pandas as pd
import matplotlib.pyplot as plt
grades_dict={'Wally':[87,96,70],'Eva':[100,97,90]
'Sam':[94,77,90],'Katie':[100,81,82],
'Bob':[83,65,85]}
grades = pd.DataFrame(grades_dict)
n=[(i+1) for i in range(len(grades.keys()))]
print(n)
plt.boxplot(grades)
plt.xticks(n,grades.keys())
plt.title("Student grades")
plt.xlabel("Students")
plt.ylabel("Grades")
plt.show()
```

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/
[1, 2, 3, 4, 5]
/Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab9_3.p
ved two minor releases later. Use callbacks.process('resi
  plt.boxplot(grades)
```