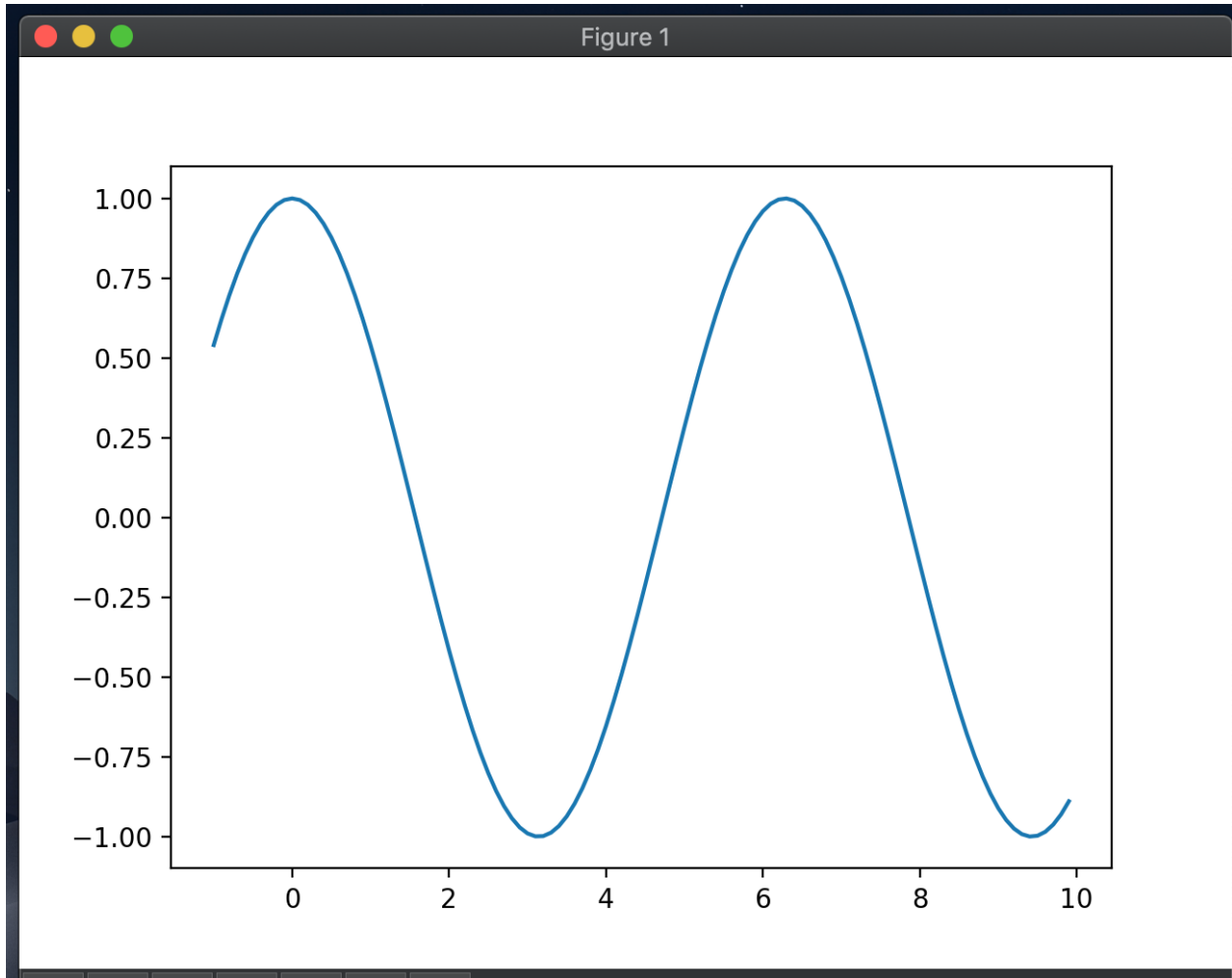Raul Rodriguez

```python
1    '''Exercise 1
2    Create a Numpy array with values from -1 to +10 and a step of 0.1. Plot the cos trigonometric
3    function, using the np.cos() built-in function'''
4    from matplotlib import pyplot as plt
5    import numpy as np
6    x=np.arange(-1,10,.1)
7    y=np.cos(x)
8    plt.plot(x,y)
9    plt.show()
```
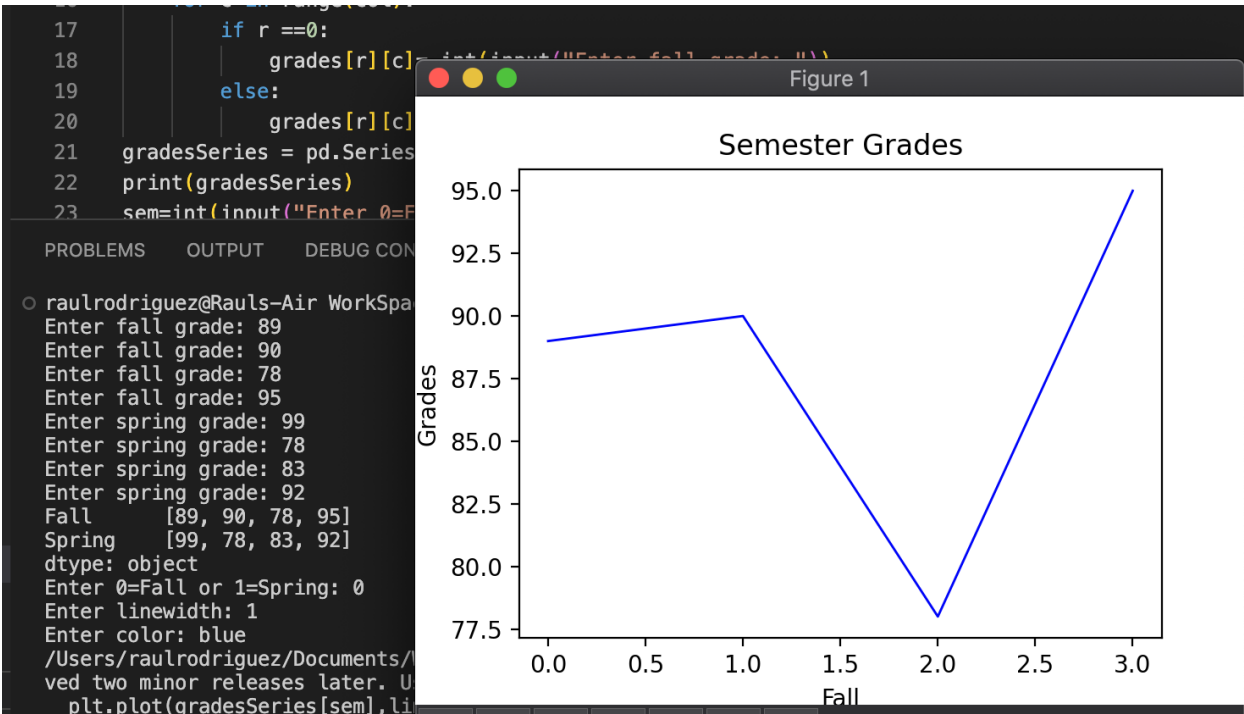

Figure 1

```python
1    '''Exercise 2
2    Similarly to Ex.1, create a Numpy array with 120 elements and values that range from -1 to
3    +10. Plot the tan trigonometric function using the np.tan() built-in function'''
4    from matplotlib import pyplot as plt
5    import numpy as np
6    x=np.linspace(-1, 10, num=120)
7    y=np.tan(x)
8    plt.plot(x,y)
9    plt.show()
```
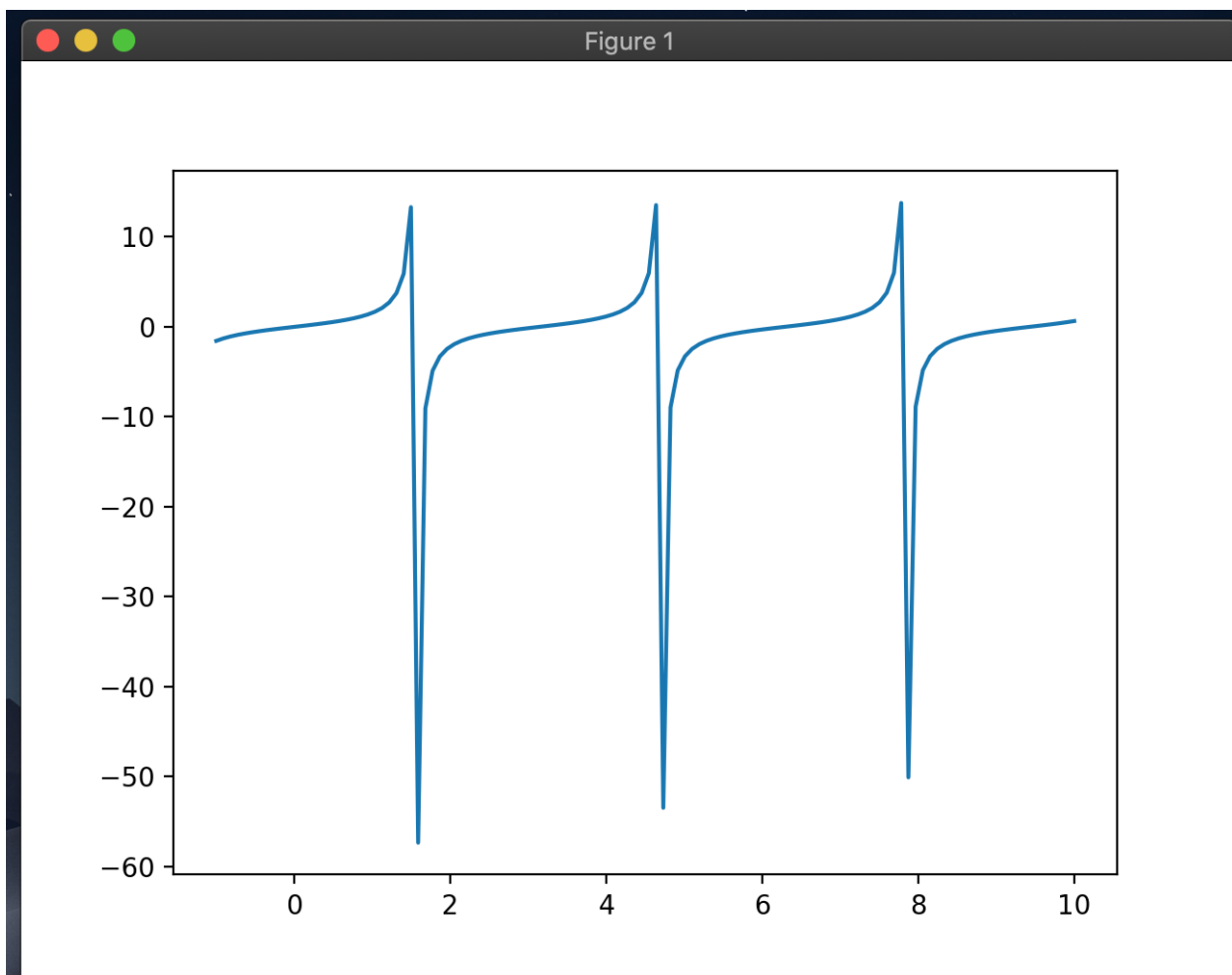
```python
'''Exercise 3
Create a 2 x4 Numpy array filled with zeros. Using a nested for loop enter integer grade
values to the elements of the array (assume the first row is Fall Semester and the second row
is Spring Semester). Create a Series using a dictionary. For keys use 'Fall' and 'Spring' and for
values use the first and second rows from Numpy array, respectively (use slicing to get the proper
rows). Ask user which semester they wish to plot (e.g., Fall), what linewidth and color they wish
to use (the color should be case insensitive). Add title, xlabel, ylabel to your plot'''
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
grades = np.zeros((2,4), dtype=int)
arrShape= grades.shape
rows=arrShape[0]
col=arrShape[1]
for r in range(rows):
    for c in range(col):
        if r ==0:
            grades[r][c]= int(input("Enter fall grade: "))
        else:
            grades[r][c]= int(input("Enter spring grade: "))
gradesSeries = pd.Series({'Fall':grades[0,:],'Spring':grades[1,:]})
print(gradesSeries)
sem=int(input("Enter 0=Fall or 1=Spring: "))
lw=int(input("Enter linewidth: "))
c=input("Enter color: ")
plt.plot(gradesSeries[sem],linewidth=lw,color=c.lower())
if sem==0:
    semName="Fall"
elif sem==1:
    semName="Spring"
plt.xlabel(semName)
plt.ylabel("Grades")
plt.title("Semester Grades")
plt.show()
```

```python
        if r ==0:
            grades[r][c]
    else:
            grades[r][c]
gradesSeries = pd.Series
print(gradesSeries)
sem=int(input("Enter 0=F
```

PROBLEMS   OUTPUT   DEBUG CON

```
raulrodriguez@Rauls-Air WorkSpa
Enter fall grade: 89
Enter fall grade: 90
Enter fall grade: 78
Enter fall grade: 95
Enter spring grade: 99
Enter spring grade: 78
Enter spring grade: 83
Enter spring grade: 92
Fall      [89, 90, 78, 95]
Spring    [99, 78, 83, 92]
dtype: object
Enter 0=Fall or 1=Spring: 0
Enter linewidth: 1
Enter color: blue
/Users/raulrodriguez/Documents/
ved two minor releases later. U
  plt.plot(gradesSeries[sem],li
```


Figure 1 — Semester Grades

```python
'''Exercise 4
Using the original 2 x4 Numpy array from Ex. 3 filled with Fall and Spring integer grades,
first flatten it to 1D array using deep copy. Second, create a Panda Series that consists of the
elements of the 1D array. Third, print Series and produce descriptive statistics of Series by calling
the respective built-in method'''
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
grades = np.zeros((2,4), dtype=int)
arrShape= grades.shape
rows=arrShape[0]
col=arrShape[1]
for r in range(rows):
    for c in range(col):
        if r ==0:
            grades[r][c]= int(input("Enter fall grade: "))
        else:
            grades[r][c]= int(input("Enter spring grade: "))
grades=grades.flatten()
print(grades)
grades=pd.Series(grades)
print(grades)
print(grades.describe())
```

```
raulrodriguez@Rauls-Air WorkSpaceVSPython % /usr/local/bin/
Enter fall grade: 89
Enter fall grade: 94
Enter fall grade: 92
Enter fall grade: 87
Enter spring grade: 90
Enter spring grade: 99
Enter spring grade: 79
Enter spring grade: 85
[89 94 92 87 90 99 79 85]
0    89
1    94
2    92
3    87
4    90
5    99
6    79
7    85
dtype: int64
count     8.00000
mean     89.37500
std       6.02228
min      79.00000
25%      86.50000
50%      89.50000
75%      92.50000
max      99.00000
dtype: float64
raulrodriguez@Rauls-Air WorkSpaceVSPython % 
```