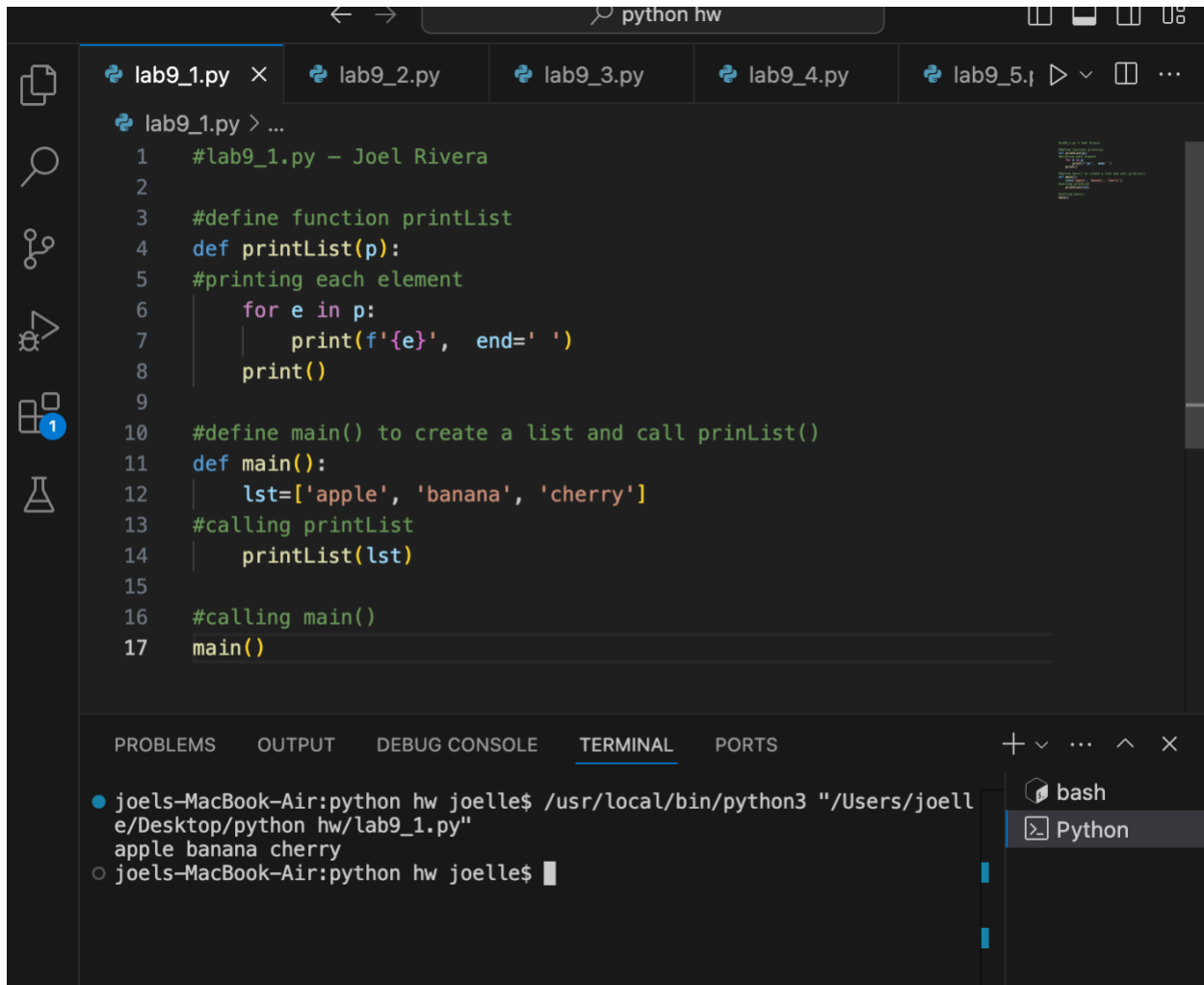


Joel Rivera

Et-574 F2F4L

Lab 9

11/7/2024



The image shows a code editor window with a dark theme. The top bar shows a search icon and the text "python hw". Below the top bar, there are tabs for "lab9_1.py", "lab9_2.py", "lab9_3.py", "lab9_4.py", and "lab9_5.py". The "lab9_1.py" tab is active, showing the following code:

```
1 #lab9_1.py - Joel Rivera
2
3 #define function printList
4 def printList(p):
5     #printing each element
6     for e in p:
7         print(f'{e}', end=' ')
8     print()
9
10 #define main() to create a list and call prinList()
11 def main():
12     lst=['apple', 'banana', 'cherry']
13     #calling printList
14     printList(lst)
15
16 #calling main()
17 main()
```

Below the code editor, there is a terminal window. The terminal shows the command to run the script and its output:

```
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_1.py"
apple banana cherry
joels-MacBook-Air:python hw joelle$
```

On the right side of the terminal, there is a dropdown menu with "bash" and "Python" options.

```
lab9_1.py lab9_2.py x lab9_3.py lab9_4.py lab9_5.py
lab9_2.py > ...
1 #lab9_2.py - Joel Rivera
2
3 #define function nameFormat()
4 def nameFormat(first, middle, last):
5     print(f'{first.title()} {middle[0].title()}. {last.title()}')
6
7 #define function main()
8 def main():
9     #calling nameFormat() with positional arguments
10    nameFormat('john', 'stu', 'smith')
11    #calling nameFormat() with keyword arguments
12    nameFormat(last = 'kennedy', first = 'john', middle = 'fitzgerald')
13
14 #calling main()
15 main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/
python hw/lab9_2.py"
John S. Smith
John F. Kennedy
joels-MacBook-Air:python hw joelle$
```

```
python hw
lab9_1.py lab9_2.py lab9_3.py x lab9_4.py lab9_5.py
lab9_3.py > nameFormat
1 #lab9_3.py - Joel Rivera
2
3 #define nameFormat() with middle being optional
4 def nameFormat(first, last, middle=''):
5     if middle:
6         #using f string to return a string instead of a tuple
7         return f'{last.title()}, {first.title()} {middle[0].title()}.'
8     else:
9         return f'{last.title()}, {first.title()}'
10
11 #define main()
12 def main():
13     print(nameFormat(last='bond', first='james'))
14     print(nameFormat(last='jones', first='henry', middle='indiana'))
15
16 main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/
python hw/lab9_3.py"
Bond, James
Jones, Henry I.
joels-MacBook-Air:python hw joelle$
```

```
python hw

lab9_1.py lab9_2.py lab9_3.py lab9_4.py x lab9_5.py

lab9_4.py > ...
1 #lab9_4.py - Joel Rivera
2
3 #define printNames() with a parameter, '*' allows the function to accept
4 #any amount of arguments and collect them into tuple names
5 def printNames(*names):
6     for e in names:
7         print(f'{e}', end=' ')
8     print()
9
10 printNames('Ann', 'Bianca', 'Coco', 'Dora', 'Emily')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_4.py"
Ann Bianca Coco Dora Emily
joels-MacBook-Air:python hw joelle$
```

```
python hw

lab9_1.py lab9_2.py lab9_3.py lab9_4.py lab9_5.py x

lab9_5.py > ...
1 #lab9_5.py - Joel Rivera
2 from random import *
3
4 #define average() with a parameter that accepts an arbitrary number of values
5 def average(*grades):
6     #calculating average and then returning it
7     aver=sum(grades)/len(grades)
8     return aver
9
10 #define main()
11 def main():
12     #calling average()
13     print(f'Average of 95, 87, 83, 74: {average(95, 87, 83, 74):.2f}')
14     #creating random integers
15     x,y,z=randint(-100, -1), randint(0, 1), randint(1, 100)
16     print(f'Average of any three random numbers, {x} {y} {z}: {average(x, y, z):.2f}')
17
18 main()
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_5.py"
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -7 0 66: 19.67
joels-MacBook-Air:python hw joelle$
```

bash Python

6. I faced two challenges in this week's lab, the first was in question 3. I was trying to find a way to check if the user had inputted a 'middle' argument. Originally, I had written my if else statements in reverse, my old if statement checked if the user had inputted all three arguments and the output was wrong because I think the default value made the condition true always so even if there wasn't a 'middle' inputted, it was there anyway. When I made the change to only check if there was a 'middle', I then got the desired output. The second challenge I faced came from the 4th question, I had originally written a tuple inside of the function I was calling, but that wasn't the instruction. I reviewed the PowerPoint and saw that we had to use a '*' Infront of the parameter so the function is then able to accept any number of positional arguments and add those arguments into a tuple.