

Lab 9 Summary

Jhonatan Parada

ET574

```
labs > Jhonatan_LAB9 > lab9_1.py > ...
1 # lab9_1.py - Jhonatan Parada
2
3 def printList(p):
4     if not isinstance(p, list): return
5     print(*p)
6
7 def main():
8     lst = ['apple', 'banana', 'cherry ']
9     printList([lst])
10
11 main()
```

```
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_1.py
['apple', 'banana', 'cherry ']
@jhonatanparada499 →/workspaces/ET574 (main) $
```

Ln 1, Col 30 Spaces: 4 UTF-8

```
labs > Jhonatan_LAB9 > lab9_2.py > ...
1 # lab9_2.py - Jhonatan Parada
2
3 def nameFormat(first, middle, last):
4     print(f'{first} {middle[0]}. {last}'.title())
5
6 def main():
7     nameFormat('john', 'stu', 'smith')
8     nameFormat(last='Kennedy', first='john',
9               middle='fitzgerald')
10
11 main()
```

```
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_2.py
John S. Smith
John F. Kennedy
@jhonatanparada499 →/workspaces/ET574 (main) $
```

Ln 1, Col 30 Spaces: 4 UTF-8

labs > Jhonatan_LAB9 > lab9_3.py > main

```
1 # lab9_3.py - Jhonatan Parada
2
3 def nameFormat(last, first, m=None):
4     if m: return f'{last}, {first}, {m[0]}'.title()
5     return f'{last}, {first}'.title()
6
7 def main():
8     name_1 = nameFormat(first='james', last='bond')
9     name_2 = nameFormat(first='henry', m='indiana',
10                        last='jones')
11
12     print(name_1, name_2, sep='\n')
13
14 main()
```

• @jhonatanparada499 →/workspaces/ETS74 (main) \$ python labs/Jhonatan_LAB9/lab9_3.py

Bond, James

Jones, Henry, I.

○ @jhonatanparada499 →/workspaces/ETS74 (main) \$

despaces: vigilant space carnival

main*

Ln 11, Col 5 Spaces: 4 UTF-8

labs > Jhonatan_LAB9 > lab9_4.py > ...

```
1 # lab9_4.py - Jhonatan Parada
2
3 def printNames(*names):
4     print(*names)
5
6 def main():
7     printNames(
8         'Ann', 'Bianca', 'Coco',
9         'Dora', 'Emily'
10    )
11
12 main()
```

• @jhonatanparada499 →/workspaces/ETS74 (main) \$ python labs/Jhonatan_LAB9/lab9_4.py

Ann Bianca Coco Dora Emily

○ @jhonatanparada499 →/workspaces/ETS74 (main) \$

despaces: vigilant space carnival

main*

Ln 1, Col 1 Spaces: 4 UTF-8

```

labs > Jhonatan_LAB9 > lab9_5.py > ...
1 # lab9_5.py - Jhonatan Parada
2 from random import randint
3
4 def average(*grades):
5     return sum(grades) / len(grades)
6
7 def main():
8     x= randint(-100,-1)
9     y = randint(0,1)
10    z = randint(1, 100)
11
12    stat_grades = (95, 87, 83, 74)
13    rand_grades = (x, y, z)
14
15    # average(*rand_grades) = average(x, y, z)
16    # average(rand_grades) = average((x, y, z))
17
18    print(
19        f'Average of {str(stat_grades)[1:-1]}:',
20        f'{average(*stat_grades):.2f}',
21        sep=' '
22    )
23
24    print(
25        f'Average of any three random numbers,',
26        f'{str(rand_grades)[1:-1]}:',
27        f'{average(*rand_grades):.2f}',
28        sep=' '
29    )
30
31    main()

```

```

@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_5.py
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -6, 0, 69: 21.00
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_5.py
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -83, 0, 10: -24.33
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_5.py
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -64, 1, 6: -19.00
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_5.py
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -40, 0, 23: -5.67
@jhonatanparada499 →/workspaces/ET574 (main) $ python labs/Jhonatan_LAB9/lab9_5.py
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -59, 0, 80: 7.00
@jhonatanparada499 →/workspaces/ET574 (main) $

```

2. On this lab, I learned about Arbitrary values and how to simplify functions by using return statements. I almost made a mistake in lab9_9.py when I tried to use a for statement to print the averages of stat_grades and rand_grades. While taking a look, I noticed that the structure of both print functions was similar, the only difference was the grade variable and the initial string, if the initial string was the same for both, then it could be simplified further.

When calling a function: