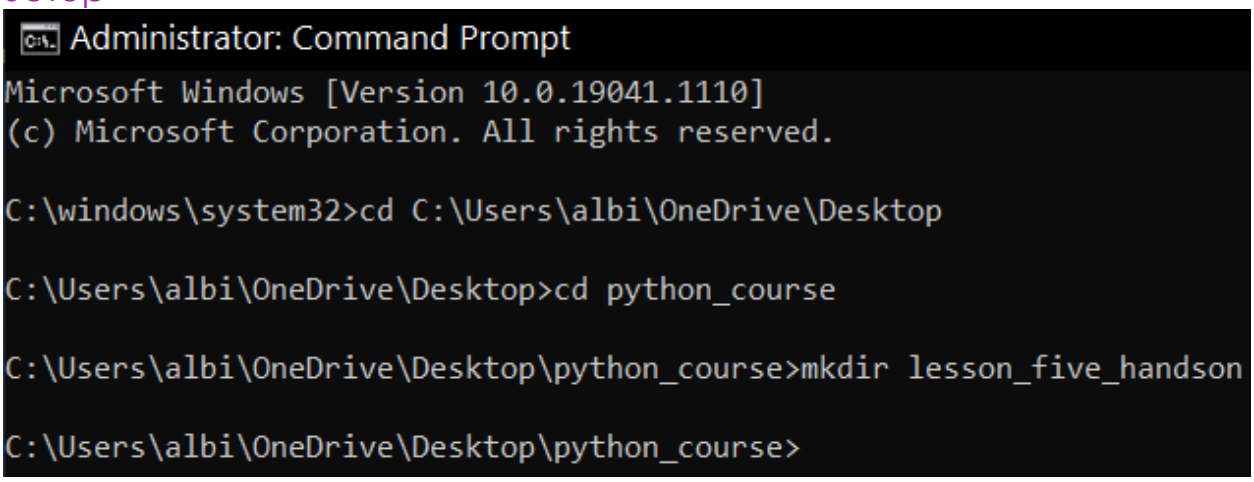
Lesson 5 Hands-On

Alberta “Albi” Kovatcheva

Setup



Lesson 5 Hands-On

Alberta "Albi" Kovatcheva

Part 1

1. Create three functions that each accept three parameters.

- The first function should be named sum_function and should return the sum of all numbers (add them all together)
- The second function should be named product_function and should return the product of all numbers (multiply them all together)
- The third function should be named average_function and should return the average of all numbers HINT: The average is the sum divided by the number of items.

2. Print out the result of calling each function. For example:

```
print(sum_function(1, 2, 3))
```

Should print:

```
6
```

Python Commands:

```
def sum_function(x, y, z):
    """ Returns the sum of all input numbers. """
    return x + y + z
def product_function(x, y, z):
    """ Returns the product of all input numbers. """
    return (x * y * z)
def average_function(x, y, z):
    """ Returns the average of all input numbers. """
    return (x + y + z)/3

print(sum_function(1,2,3))
print(product_function(4,5,6))
print(average_function(7,8,9))
```

Results:

Lesson 5 Hands-On

Alberta “Albi” Kovatcheva

```
>>> def sum_function(x, y, z):
...     """ Returns the sum of all input numbers. """
...     return x + y + z
...
>>> def product_function(x, y, z):
...     """ Returns the product of all input numbers. """
...     return (x * y * z)
...
>>> def average_function(x, y, z):
...     """ Returns the average of all input numbers. """
...     return (x + y + z)/3
...
>>> print(sum_function(1,2,3))
6
>>> print(product_function(4,5,6))
120
>>> print(average_function(7,8,9))
8.0
```

Lesson 5 Hands-On

Alberta "Albi" Kovatcheva

Part 2

1. Create three lambda functions that do the same thing as the functions in step 1. Assign each lambda function the following variables:
   add_numbers
   multiply_numbers
   average_numbers
2. Print and call the above functions.

Python Commands:

```
add_numbers = lambda x, y, z: x + y + z
multiply_numbers = lambda x, y, z: x * y * z
average_numbers = lambda x, y, z: (x+y+z)/3
print(add_numbers(1,2,3))
print(multiply_numbers(4,5,6))
print(average_numbers(7,8,9))
```

Results:

```
>>> add_numbers = lambda x, y, z: x + y + z
>>> multiply_numbers = lambda x, y, z: x * y * z
>>> average_numbers = lambda x, y, z: (x+y+z)/3
>>> print(add_numbers(1,2,3))
6
>>> print(multiply_numbers(4,5,6))
120
>>> print(average_numbers(7,8,9))
8.0
```

Lesson 5 Hands-On

Alberta "Albi" Kovatcheva

Part 3

1. Creating three separate lists named the following: list_one, list_two, list_three
2. Add the following numbers in to their respective lists:
   numbers 4, 6, 88, and 24 should go within list_one
   numbers 17, 34, 9, and 5 should go within list_two
   numbers 63, 20, 98, and 4 should go within list_three
3. Create one lambda function named average_maker that takes in three numbers and finds the average.
4. Use map to compute the average of each set of values at each index. This will produce a new list of the four average calculations.
   The variable name for this calculation should be map_results
   You will be using each of the lists within the map function.
5. Print out the end result of using map.
   Hint! You will need to use list()
6. The final output should be as shown below:

```
[28.0, 20.0, 65.0, 11.0]
```

Python Commands:

```
list_one = [4, 6, 88, 24]
list_two = [17, 34, 9, 5]
list_three = [63, 20, 98, 4]
average_maker = lambda x, y, z: (x+y+z)/3
map_results = map(average_maker, list_one, list_two, list_three)
list(map_results)
print(list(map_results))
```

Results:

```
>>> list_one = [4, 6, 88, 24]
>>> list_two = [17, 34, 9, 5]
>>> list_three = [63, 20, 98, 4]
>>> average_maker = lambda x, y, z: (x+y+z)/3
>>> map_results = map(average_maker, list_one, list_two, list_three)
>>> list(map_results)
[28.0, 20.0, 65.0, 11.0]
>>> print(list(map_results))
[]
```