**FUNCTIONS WITH OUTPUTS**

def my\_function():

res = 3 \* 2

return res

Let’s consider converting the entered names into title case.

def format\_name(f\_name, l\_name):

formatted\_f\_name = f\_name.title()

formatted\_l\_name = l\_name.title()

return f”{formatted\_f\_name} { formatted\_l\_name}

print(format\_name(“AngeLA”, “YU”)

---- o/p: Angela Yu

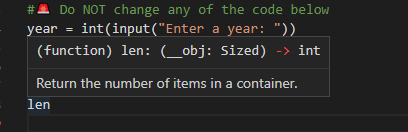
**Multiple return values:**

Return statement implies the end of the functions. Anything written after the return statement will never be executed.

We can have multiple return statements in a function. We can also have an empty return statement in a function.

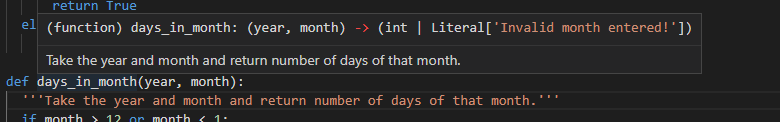
**Docstrings:**

These are a way for us to create and build the documentation.

e.g.:  here, Return the number of items in a container is a docstring.

--- Docstrings have to go at the **first line** after the declaration.

--- It is in between 3 quotation marks.



--- We can write as many lines as we want.

--- We can, also use this as a multiline comment.

**Print vs Return:**

**While Loops, Flags & Recursion:**

#addition

def add(n1, n2):

    return n1 + n2

#subtraction

def subtract(n1, n2):

    return n1 - n2

#multiplication

def multiply(n1, n2):

    return n1 \* n2

#division

def divide(n1, n2):

    return n1 / n2

operations = {

    "+" : add,

    "-" : subtract,

    "\*" : multiply,

    "/" : divide,

}

def calculator():

    num1 = int(input("What is the first number?"))

    for operators in operations:

            print(operators)

    flag = True

    while flag:

        num2 = int(input("What is the next number?"))

        operation\_symbol = input("Pick an operation from the line above: ")

        calculation\_function = operations[operation\_symbol]

        answer = calculation\_function(num1, num2)

        print("{} {} {} = {}".format(num1, operation\_symbol, num2, answer))

        continue\_ = input("Type 'y' to continue calculating with the {} or type 'n' to start a new calculation ".format(answer))

        if(continue\_ == 'y'):

            num1 = answer

        else:

            flag = False

            calculator()

calculator()