PYTHON

```
lambda x: x * 2
```

LAMBDA FUNCTIONS

Sebastiano Gazzola
@sebastianogazzola

FOLLOW ME ON LINKEDIN
FOR THE NEXT TIP!

IN PYTHON, A LAMBDA FUNCTION IS A SMALL ANONYMOUS FUNCTION THAT CAN TAKE ANY NUMBER OF ARGUMENTS, BUT CAN ONLY HAVE ONE EXPRESSION.

LAMBDA FUNCTIONS ARE TYPICALLY USED WHEN A SIMPLE FUNCTION IS NEEDED FOR A SHORT PERIOD OF TIME AND IT DOESN'T MAKE SENSE TO DEFINE A SEPARATE FUNCTION.

HERE'S THE BASIC SYNTAX FOR DEFINING A LAMBDA FUNCTION:

```
lambda arguments: expression
```

```
lambda arguments: expression
```

LET'S BREAK DOWN WHAT EACH PART OF THIS SYNTAX MEANS:

LAMBDA IS THE KEYWORD USED TO DEFINE A LAMBDA FUNCTION.

ARGUMENTS IS A COMMA-SEPARATED LIST OF ARGUMENTS THAT THE FUNCTION TAKES.

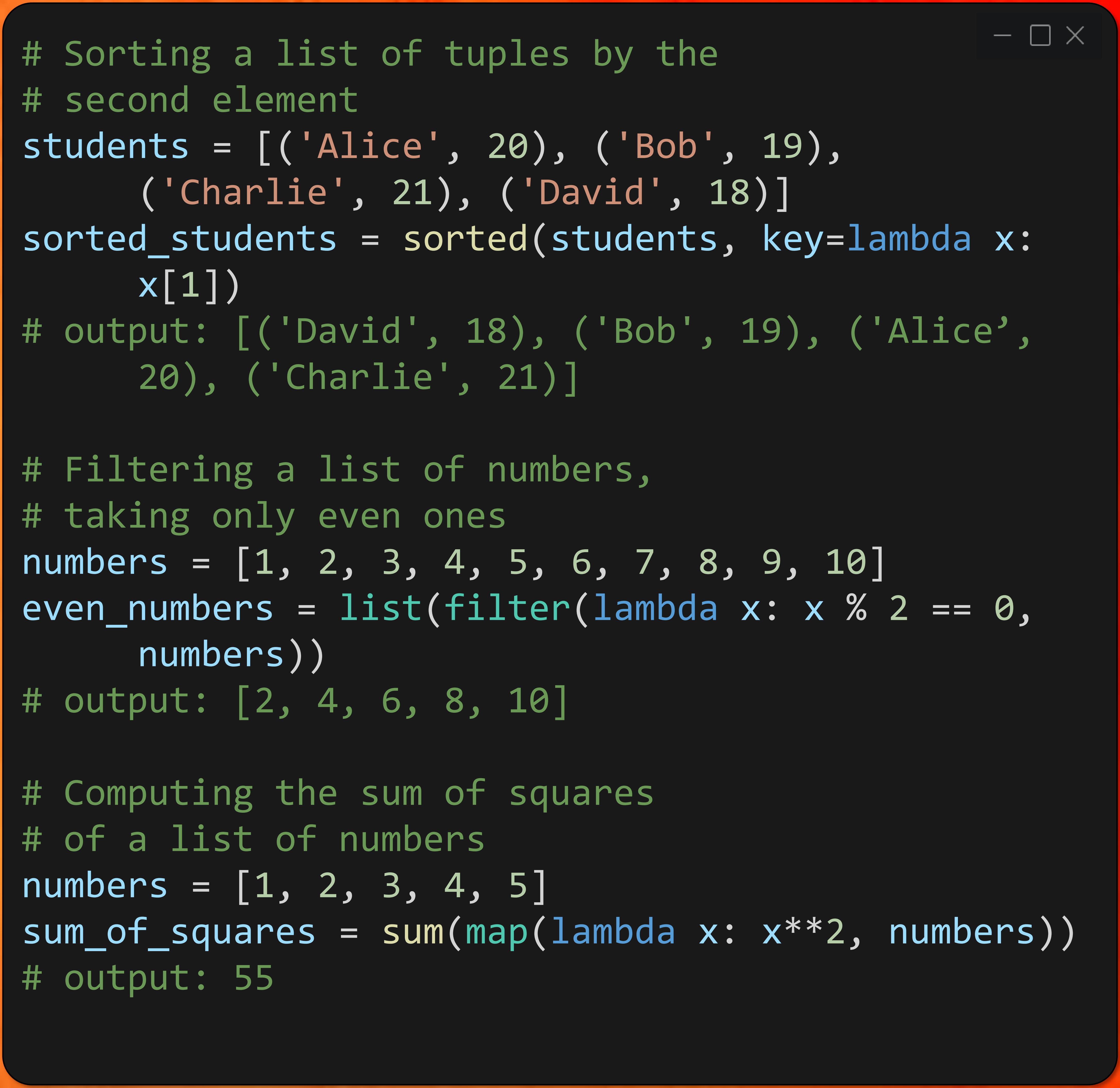
: SEPARATES THE ARGUMENT LIST FROM THE EXPRESSION THAT DEFINES THE FUNCTION.

EXPRESSION IS A SINGLE EXPRESSION THAT THE FUNCTION RETURNS.

Sebastiano Gazzola
@sebastianogazzola

FOLLOW ME ON LINKEDIN
FOR THE NEXT TIP!

LAMBDA FUNCTIONS IN PYTHON - SOME EXAMPLES

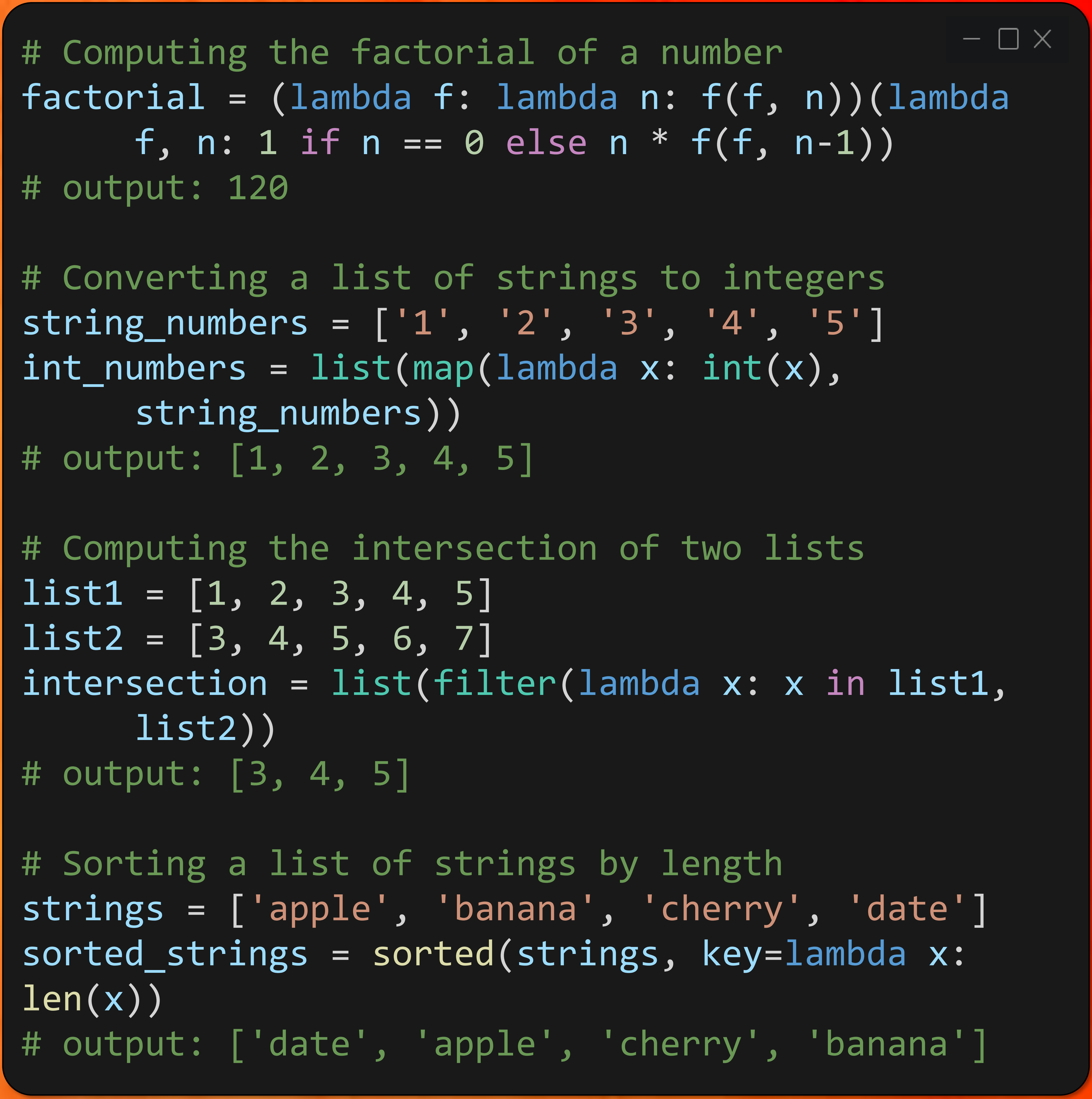


```
# Sorting a list of tuples by the
# second element
students = [('Alice', 20), ('Bob', 19),
     ('Charlie', 21), ('David', 18)]
sorted_students = sorted(students, key=lambda x:
     x[1])
# output: [('David', 18), ('Bob', 19), ('Alice’,
     20), ('Charlie', 21)]

# Filtering a list of numbers,
# taking only even ones
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_numbers = list(filter(lambda x: x % 2 == 0,
     numbers))
# output: [2, 4, 6, 8, 10]

# Computing the sum of squares
# of a list of numbers
numbers = [1, 2, 3, 4, 5]
sum_of_squares = sum(map(lambda x: x**2, numbers))
# output: 55
```

Sebastiano Gazzola
@sebastianogazzola

FOLLOW ME ON LINKEDIN
FOR THE NEXT TIP!

LAMBDA FUNCTIONS IN PYTHON - SOME EXAMPLES

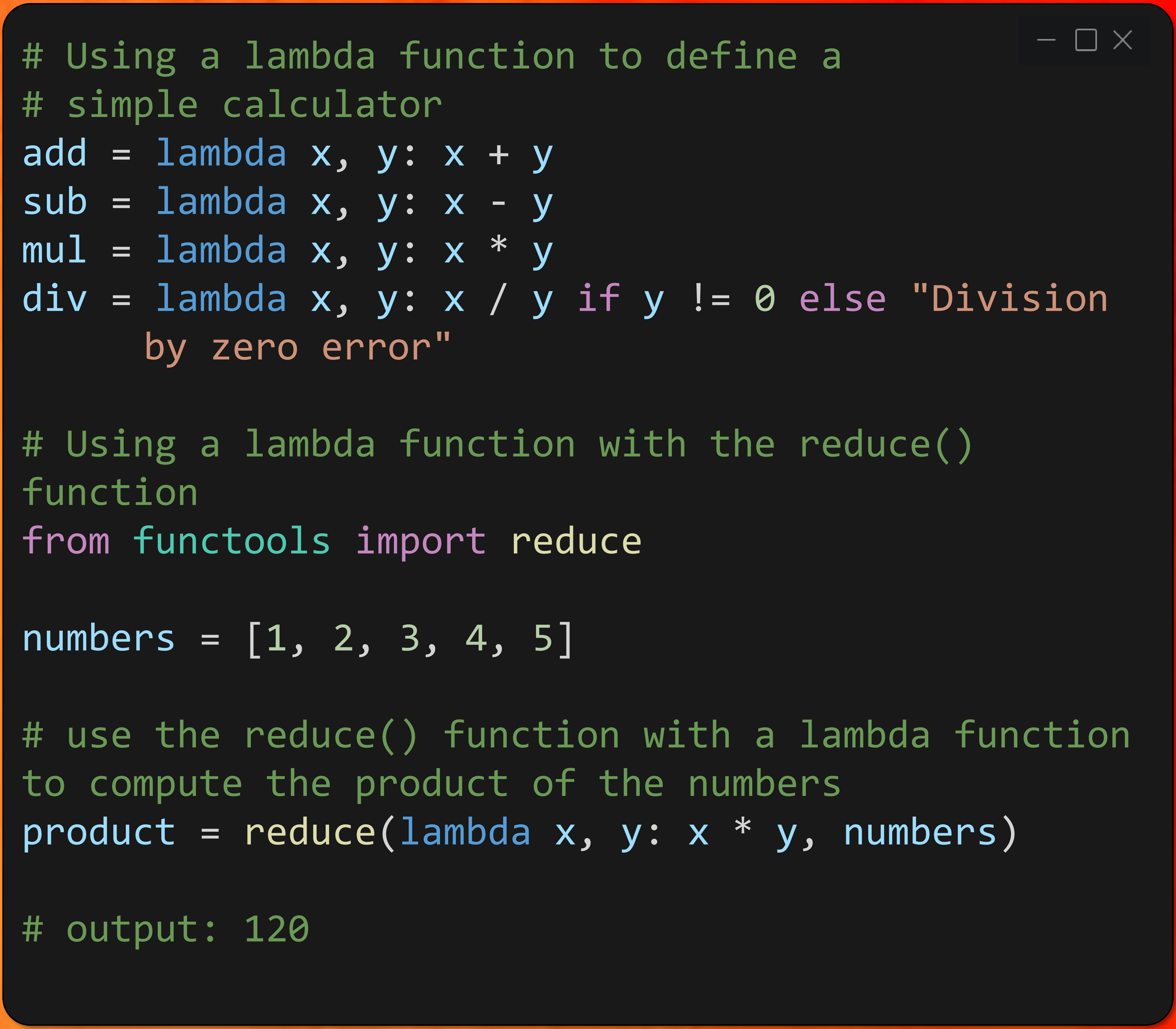


```
# Computing the factorial of a number
factorial = (lambda f: lambda n: f(f, n))(lambda
      f, n: 1 if n == 0 else n * f(f, n-1))
# output: 120

# Converting a list of strings to integers
string_numbers = ['1', '2', '3', '4', '5']
int_numbers = list(map(lambda x: int(x),
      string_numbers))
# output: [1, 2, 3, 4, 5]

# Computing the intersection of two lists
list1 = [1, 2, 3, 4, 5]
list2 = [3, 4, 5, 6, 7]
intersection = list(filter(lambda x: x in list1,
      list2))
# output: [3, 4, 5]

# Sorting a list of strings by length
strings = ['apple', 'banana', 'cherry', 'date']
sorted_strings = sorted(strings, key=lambda x:
len(x))
# output: ['date', 'apple', 'cherry', 'banana']
```

Sebastiano Gazzola
@sebastianogazzola

FOLLOW ME ON LINKEDIN
FOR THE NEXT TIP!

LAMBDA FUNCTIONS IN PYTHON - SOME EXAMPLES



```python
# Using a lambda function to define a
# simple calculator
add = lambda x, y: x + y
sub = lambda x, y: x - y
mul = lambda x, y: x * y
div = lambda x, y: x / y if y != 0 else "Division
     by zero error"

# Using a lambda function with the reduce()
function
from functools import reduce

numbers = [1, 2, 3, 4, 5]

# use the reduce() function with a lambda function
to compute the product of the numbers
product = reduce(lambda x, y: x * y, numbers)

# output: 120
```

Sebastiano Gazzola
@sebastianogazzola

FOLLOW ME ON LINKEDIN
FOR THE NEXT TIP!