

# Pair Programming exercise Functions

DSE5002, Module 3, Peter Gyorda, April 4, 2025

updated 11/13/2024

## Writing functions in Python

see

<https://docs.python.org/3/tutorial/controlflow.html#defining-functions>

## Functions

Function definitions in Python start with the keyword "def" which indicates that we are starting a function definition

After def, we state the name of the function and then the input variables

Here is a function that computes the squares of integers up to n

The input is n and there is no returned variable

The section inside the triple quotes "" is called the docstring it should explain what the function does and what the output is

In [1]: *#define the function*

```
def squares2n(n):  
    """  
    squares2n(n) prints the squares of all integers such that the square is  
    less than n, starting from 1  
    """  
    x=1  
    while(x**2<n):  
        print(x,x**2)          # notice the use of a while loop here  
        x=x+1
```

In [8]: *# now run it*

In [2]: squares2n(1500)

```
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
11 121
12 144
13 169
14 196
15 225
16 256
17 289
18 324
19 361
20 400
21 441
22 484
23 529
24 576
25 625
26 676
27 729
28 784
29 841
30 900
31 961
32 1024
33 1089
34 1156
35 1225
36 1296
37 1369
38 1444
```

In [3]: *#viewing the docstring, another way to learn about what a function does*

```
print(squares2n.__doc__)
```

```
squares2n(n) prints the squares of all integers such that the square is
less than n, starting from 1
```

Different organizations may have different protocols for what belongs in a docstring and how it should be structured

## Question/Action

Write a function in the cell below that takes in two values, a and b, and prints out the value of the smaller of the two

Include a simple doc string

use an if-else pair to do this, you may need to look up if/else in python

```
In [7]: def print_smaller(a, b):  
        """Prints the smaller of the two input values."""  
        if a < b:  
            print(a)  
        else:  
            print(b)  
        print_smaller (13,23)
```

13

## Return values

A function can return values, just as it can in R

Here, we will alter squares2n to return a list

```
In [8]: #define the function  
  
def squares2nlist(n):  
    """  
    squares2n(n) prints the squares of all integers such that the square is  
    less than n, starting from 1  
  
    returns a list of the squares  
    """  
    y=[]  
    x=1  
    while(x**2<n):  
        y.append(x**2)  
        x=x+1  
    return y
```

```
In [9]: #example call  
  
a=squares2nlist(560)  
  
a
```

```
Out[9]: [1,
        4,
        9,
        16,
        25,
        36,
        49,
        64,
        81,
        100,
        121,
        144,
        169,
        196,
        225,
        256,
        289,
        324,
        361,
        400,
        441,
        484,
        529]
```

## default values on inputs

As in R, we can define default values

```
In [10]: def powers2n(n=100, power=2):
        """
        powers2n(n,power) computes the powers of the integers less than n and returns t
        inputs are the value n to stop at and the power used
        defaults are n=100, power=2
        """
        y=[]
        x=1
        while(x**power<n):
            y.append(x**power)
            x=x+1
        return y
```

```
In [11]: a=powers2n()
        a
```

```
Out[11]: [1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
In [12]: #calling the function using input parametes in order
        a=powers2n(200,3)
```

a

Out[12]: [1, 8, 27, 64, 125]

```
In [13]: # calling the function using named parameters
a=powers2n(power=2.5)
a
```

```
Out[13]: [1.0,
          5.656854249492381,
          15.588457268119896,
          32.0,
          55.90169943749474,
          88.18163074019441]
```

For more on functions and options to control input parameters, see

<https://docs.python.org/3/tutorial/controlflow.html#defining-functions>

## Question/Action

Modify your function that takes in a and b so that a defaults to 1 and b defaults to 0

Alter the function so that it returns the smaller of the two input values

```
In [14]: def get_smaller(a=1, b=0):
          """Returns the smaller of the two input values.
          Defaults to a=1 and b=0 if nothing is provided.
          """
          if a < b:
              return a
          else:
              return b

          print(get_smaller(13, 23))
```

13

## Lambda Functions

these are simple, one line functions

They are useful when you need to pass a function or operation into another functions, say if we want to apply the same function all the values along columns of a matrix

```
In [15]: #creating a function that creates a decrementor function
          # this is a function that returns a function which decrements the input by n
          # this is the first time we have seen a function that returns a function
```

```
def make_decrementor(n):
    return lambda x: x-n
```

```
In [16]: # make a call to create the decrementor

my_decrement_by_1=make_decrementor(1)

x=10

# my_decrement_by_1 is now a function that decreases it's input by 1

my_decrement_by_1(x)
```

Out[16]: 9

## Question/Action

write a function that creates a function that multiples by n

show that it works

```
In [17]: def create_multiplier(n):
    """
    Creates and returns a new function that multiplies its input by n.

    Args:
        n: The number to multiply by.

    Returns:
        A function that takes one argument and returns the result of
        multiplying that argument by n.
    """
    def multiplier(x):
        """Multiplies its input by the value of n from the outer function."""
        return x * n
    return multiplier

# Create a function that multiplies by 5
multiply_by_5 = create_multiplier(5)

# Test the new function
print(multiply_by_5(2))    # Output: 10
```

10

## Lambda functions are helpful for passing a function into another function

this example allows use to sort by a specific entry in a list of lists

```
In [18]: customers=[(1,"Lin Ho", "Zhang"),(2,"Smith","Bob"),(3,"Fernandes","Rita")]

#sort by ID

customers.sort(key=lambda x:x[0])
customers
```

```
Out[18]: [(1, 'Lin Ho', 'Zhang'), (2, 'Smith', 'Bob'), (3, 'Fernandes', 'Rita')]
```

```
In [22]: # alter this lambda function to sort by the first name
customers = [(1, "Lin Ho", "Zhang"), (2, "Smith", "Bob"), (3, "Fernandes", "Rita")]

# Sort by first name (assuming the first word in the full_name is the first name)
customers.sort(key=lambda x: x[2].split()[0])

print(customers)

[(2, 'Smith', 'Bob'), (3, 'Fernandes', 'Rita'), (1, 'Lin Ho', 'Zhang')]
```

```
In [ ]:
```