# CPSC 250 - Functions (Methods)

```
def my_function(.....):
    return .....
```

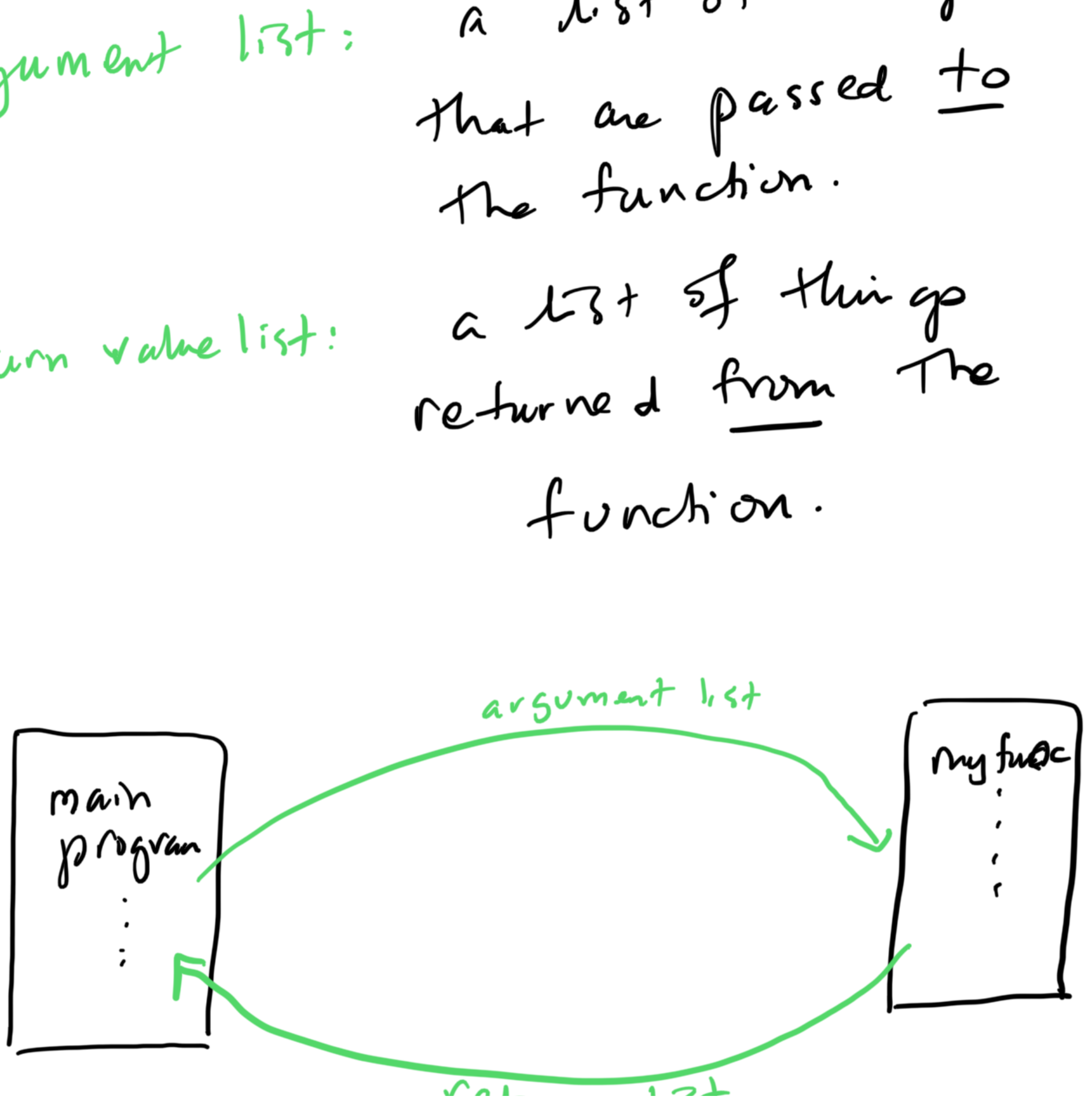
name

argument list

return value list

argument list: a list of things that are passed to the function.

return value list: a list of things returned from the function.



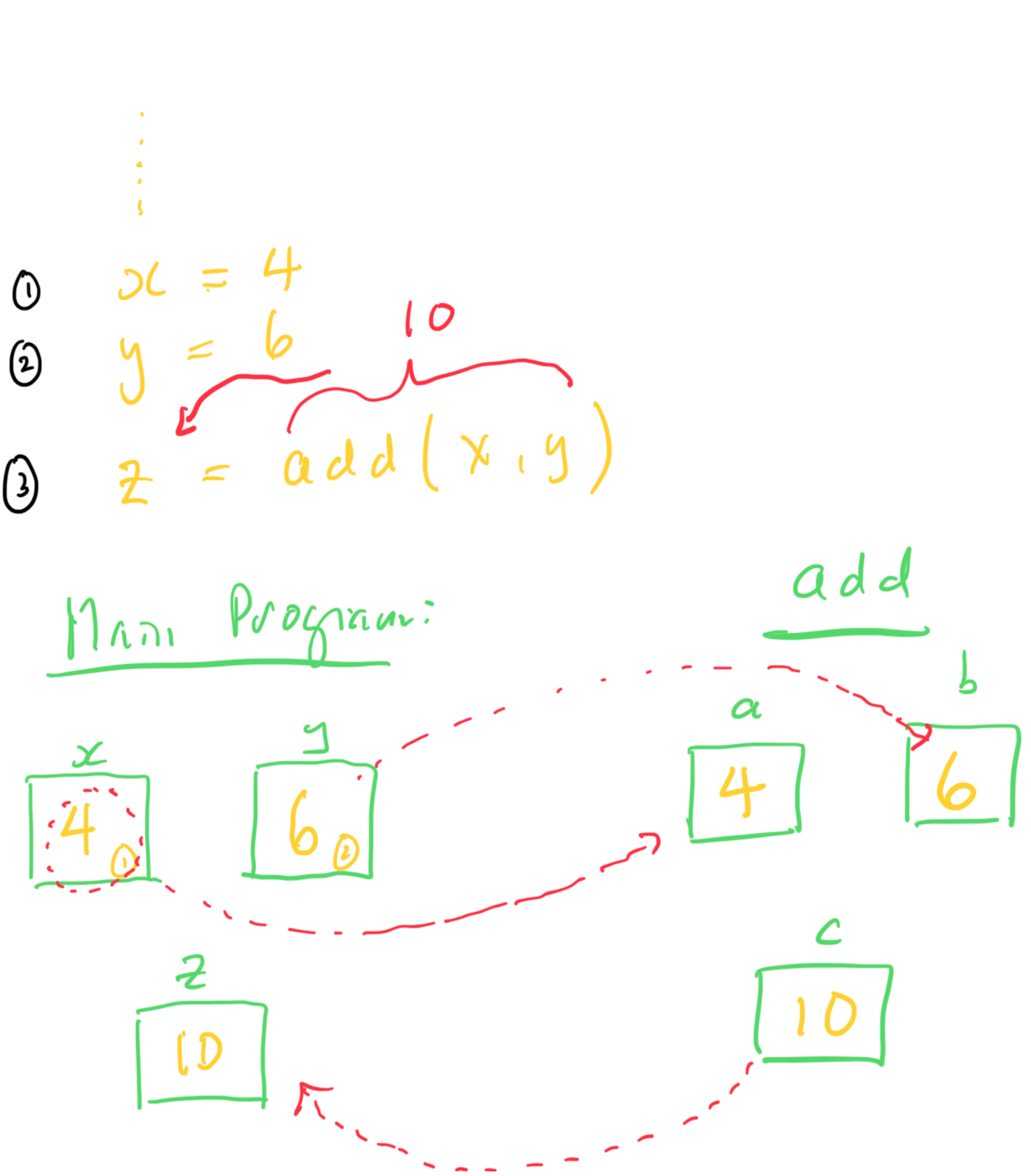
return list

## Example 1:

```
def add ( a , b ):
    c = a + b
    return c
```

```
    .
    .
    .
①   x = 4
②   y = 6
③   z = add ( x , y )
```

10



Super incredibly important ideas:

① a, b are <u>copies</u> of x, y !!!!!!! They are different places in memory !!

② changing a, b in the function does <u>not</u> change x, y in the main program !!

This is called <u>PASS BY VALUE</u> !!

<u>Big Hint</u>: Use different variable names in the main program and the function !!

<u>Main</u>

x, y, z

<u>Function</u>

a, b, c

This will reinforce to the reader, and to you, that the variables in the function are different than in the main program :)

main program :)

## static typing (C, C++, Java, Go, ...)

A note on dynamic typing (or duck typing) in Python:

The add function above would work equally well on strings or floats!!

Python is smart enough to make a, b, and c whatever type they need to be!

```
x = "that is"
y = "so cool!"
z = add(x,y)
print(z)
```

---

Source: Me

Scope: where, do variables have a well defined means.

Ex.

employee_name = 'Bob'

def get_name():

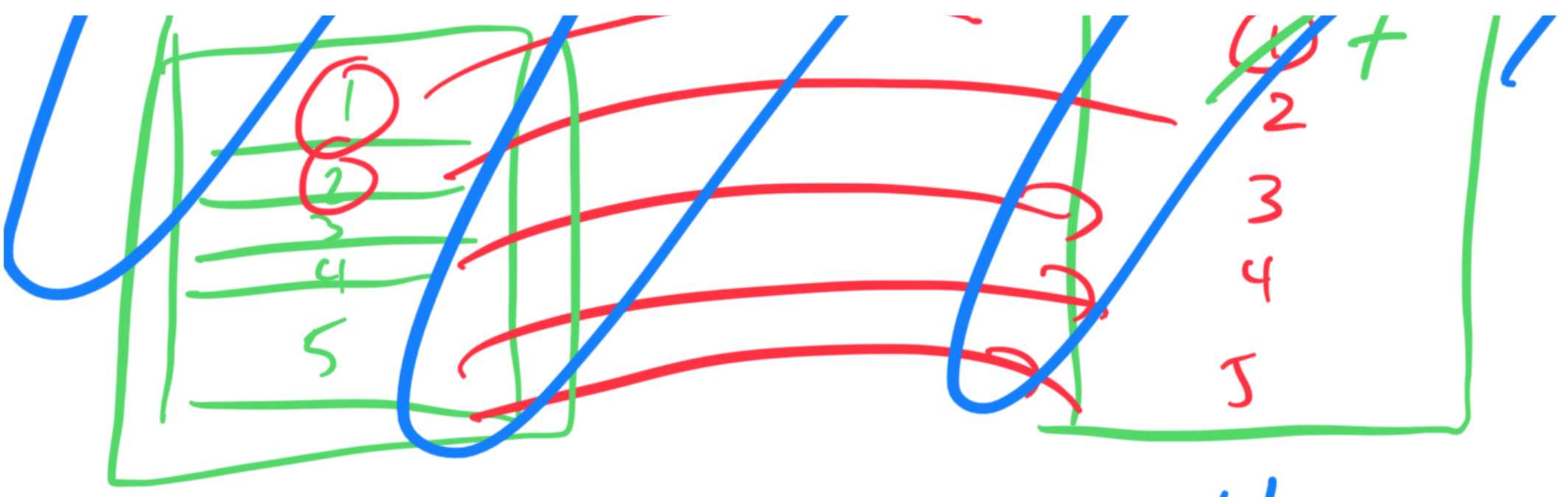
① name = input()
employee_name = name

→ get_name()

print(employee_name)

---

# Pass by Reference




Nonsense!!

Memory location

numbers

| Index | |
|---|---|
| 0 | ~~1~~ 7 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |

| | |
|---|---|
| | 0x1231 |
| | 0x1247 |
| | 0x1263 |
| | 0x1279 |
| | 0x1285 |

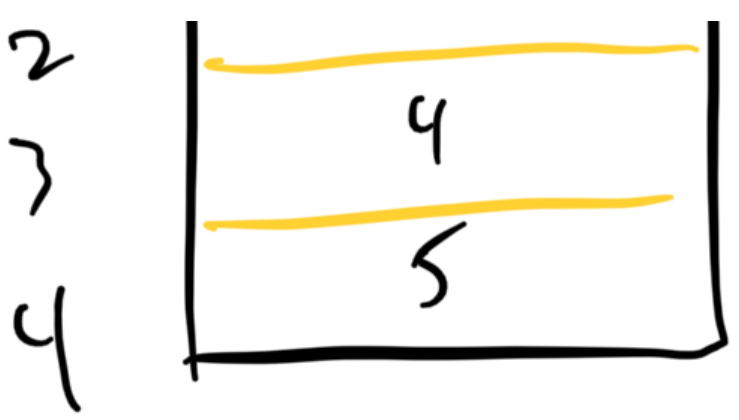
address of start of the list.

change_numbers(numbers)

Oh, it's a list

| Index | list_of_nubes. |
|---|---|
| 0 | ~~1~~ 7 |
| 1 | 2 |
| | 3 |

0x1231




0x1285