PYTHON PROGRAMMING
GATE DA/DSA

Agenda:

- What is a function? ~ Why do we need it?
- How to declare a function? Syntax in Python.
- Calling a function.
- Passing Arguments to the function
- Return Statement.

Functions: * A function is a reusable block of code that performs a specific task.

* It takes a input, processes it and produces the output.

* Indentation should be taken Care.

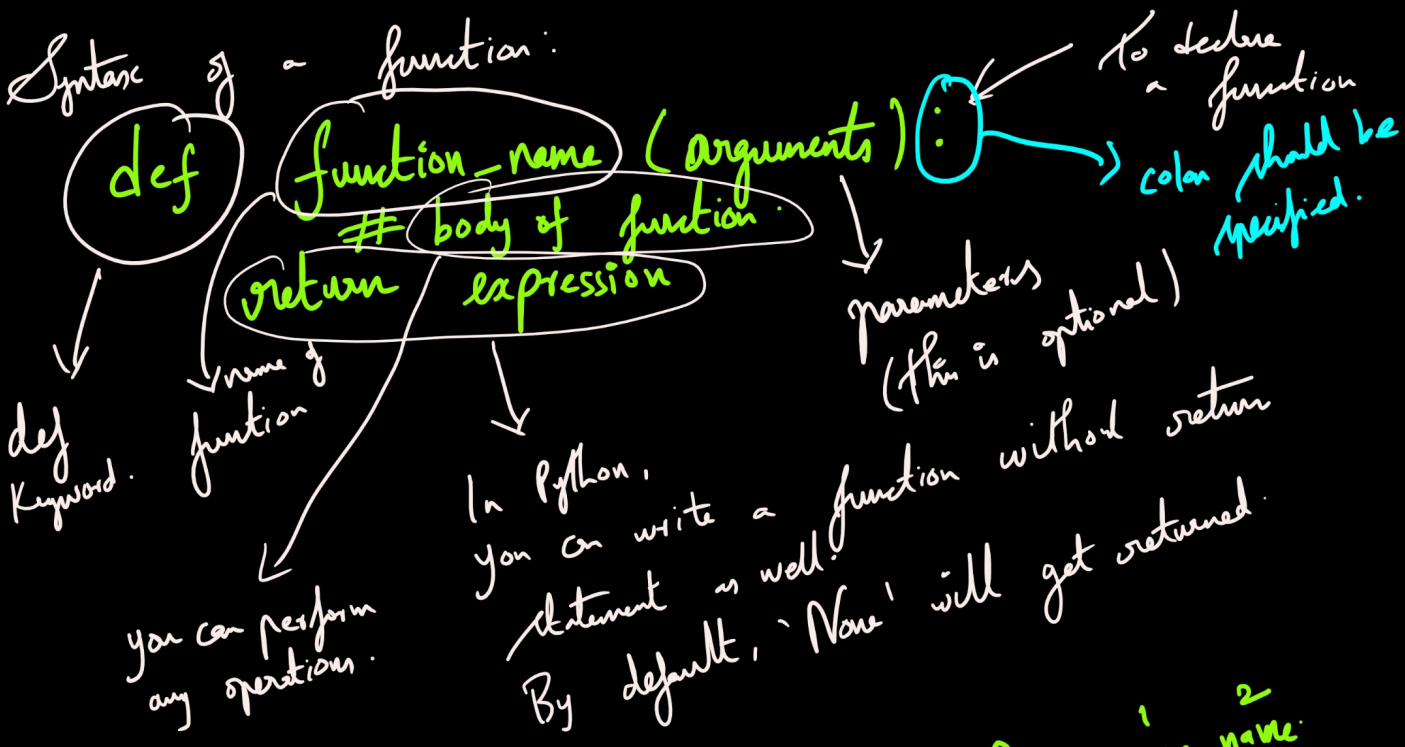
Benefits:

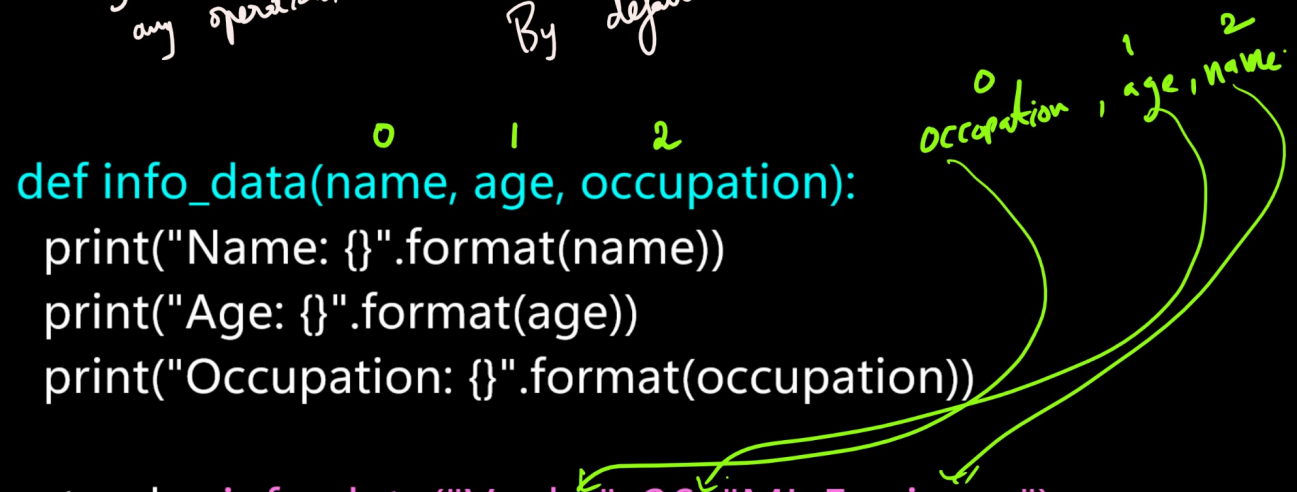
Increase code Re-usability

Increase Code Readability

Creating a Function:

A function is defined using "def" Keyword.





In positional arguments → Ordering of inputs while calling a function matters.

In Keyword arguments → Order of arguments/inputs doesn't matter.

Default arguments → should come after non-default arguments.

Positional only Arguments:

```
def func_name (x, y, /):
    # body of function
```

Keyword arguments are not accepted.

indicates func_name accepts positional only arguments

~~func_name (x=10, y=5)~~

func_name (10, 5) ✓

x, y, /

Keyword Only Arguments:

```
def func_name (*, x, y):
    # body of function.
```

~~func_name (10, 5)~~

func_name (x=10, y=5) ✓

Pass Variables to functions:

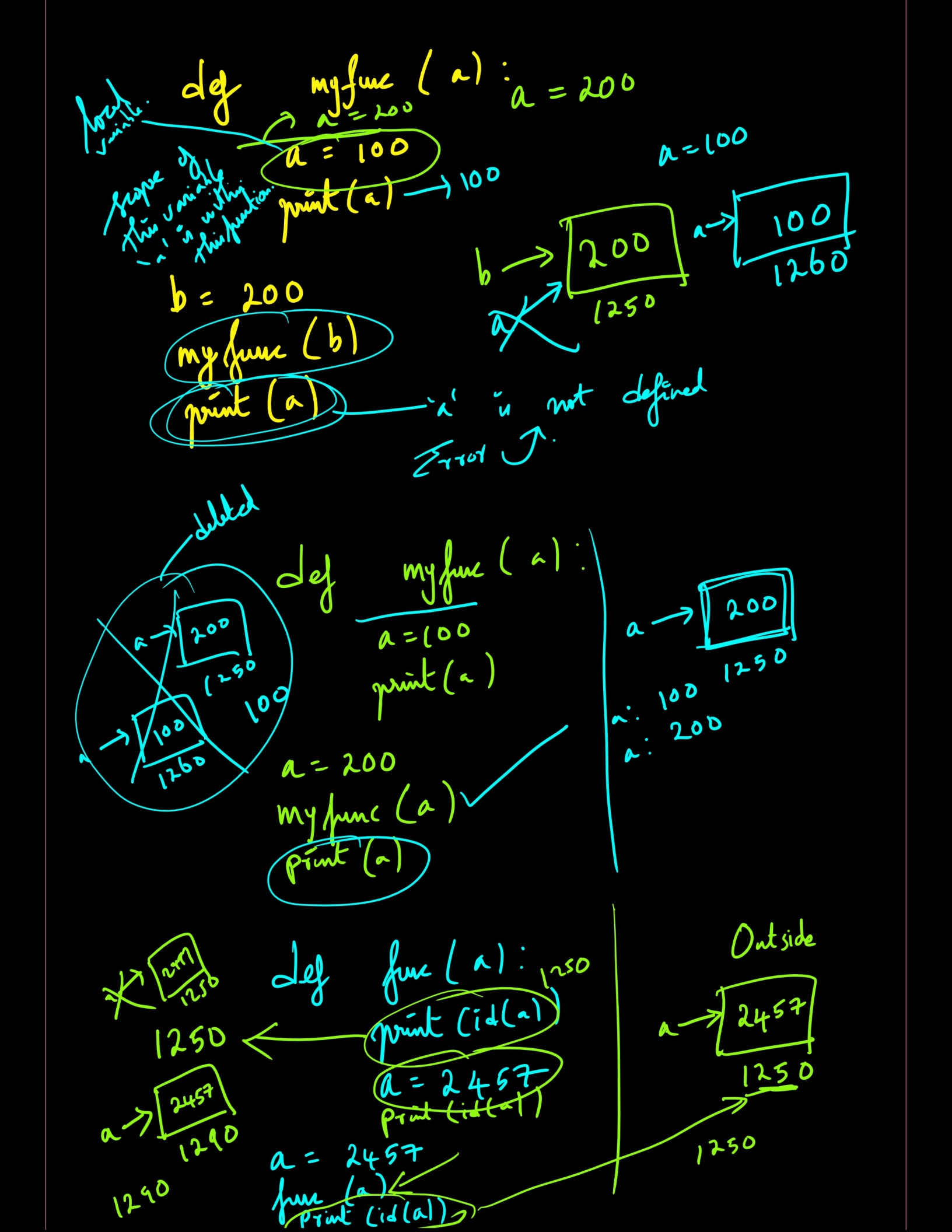
```
def func_name (a):
```

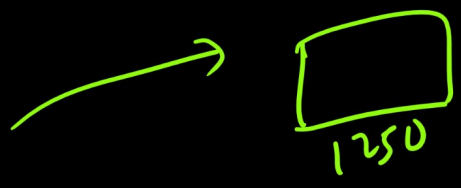
← even if changes in function

Pass

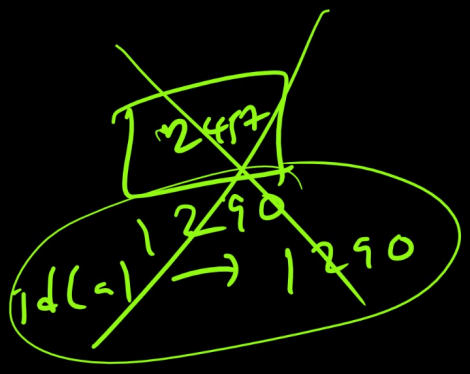
immutable datatype → func_name (a) → mutable

outside function change is reflected.


def myfunc (a):
a = 200
a = 100
print (a) → 100
local variable
Scope of this variable 'a' is within this function
b = 200
myfunc (b)
print (a)
'a' is not defined
Error
a = 100
b → 200
1250
a → 100
1260
deleted
a → 200
1250
100
1260
a: 100
a: 200
a = 200
myfunc (a)
print (a)
2457
1250
1250
a → 2457
1290
1290
def func (a):
print (id(a))
a = 2457
print (id(a))
a = 2457
func (a)
print (id(a))
Outside
a → 2457
1250
1250



a = 22457



b → (10, 20, 30)

func(a)

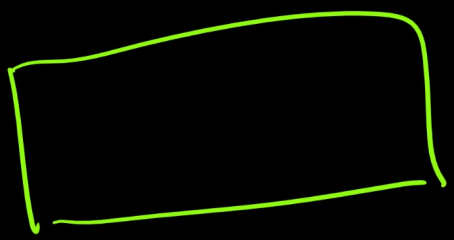
a = list(a) → [10, 20, 30]

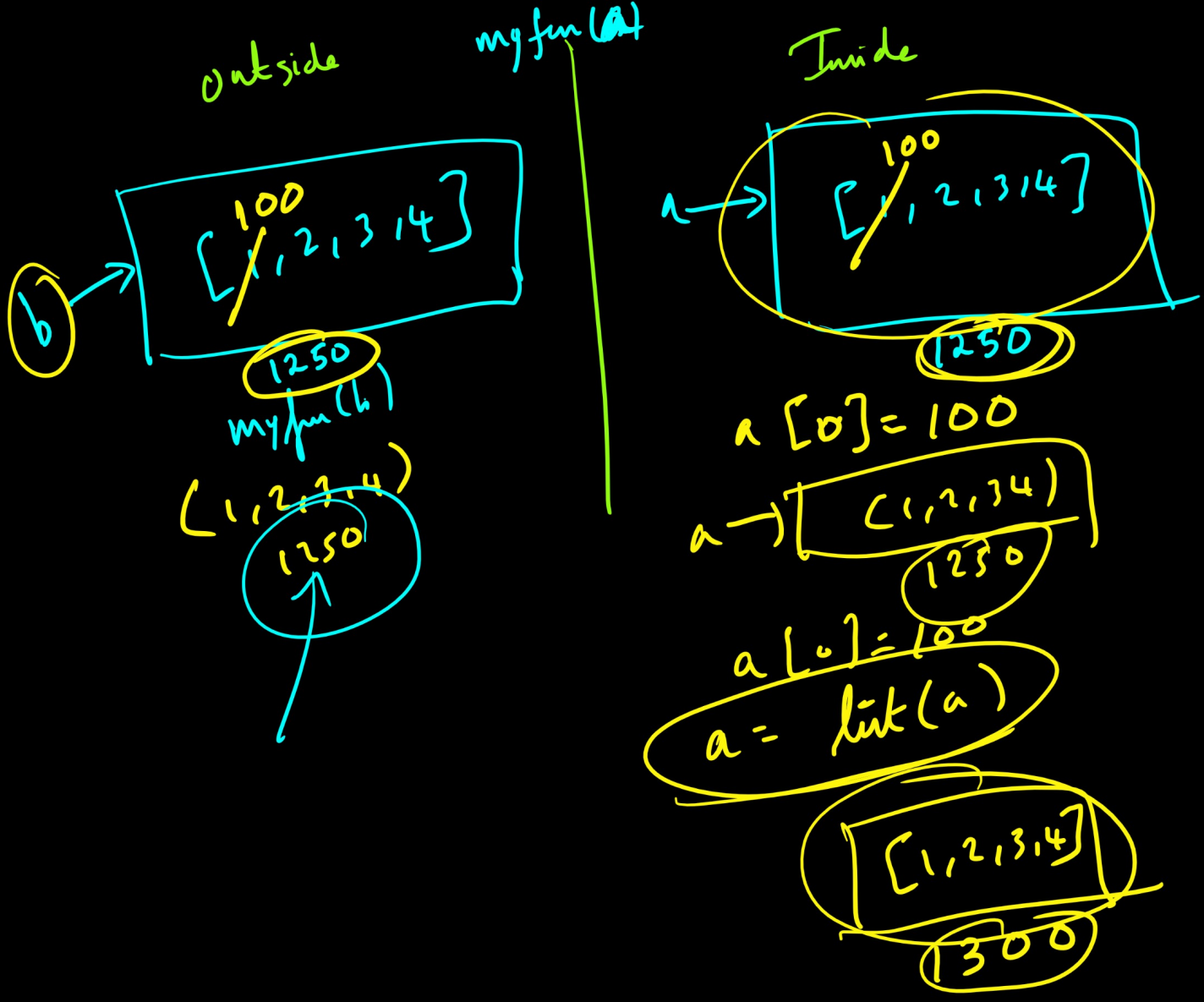
(100, 20, 30)

a[0] = 100

a = [100, 20, 30]

return ~~tuple~~(a)





Func(a)

passing variable to function: Mutable datatype

↓

Changes inside function will reflect outside.

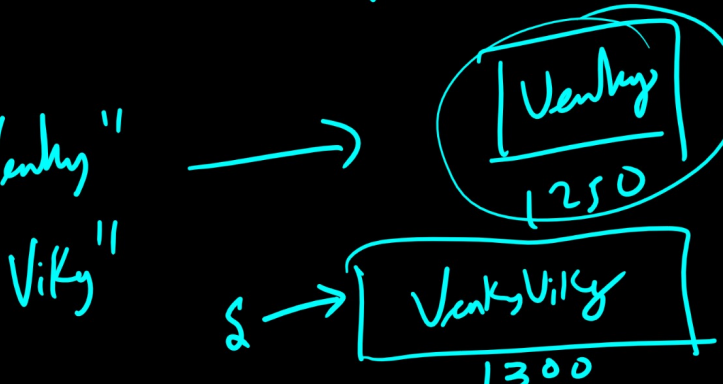
Immutable

↳ Changes won't reflect outside

Int, String, int frozenset

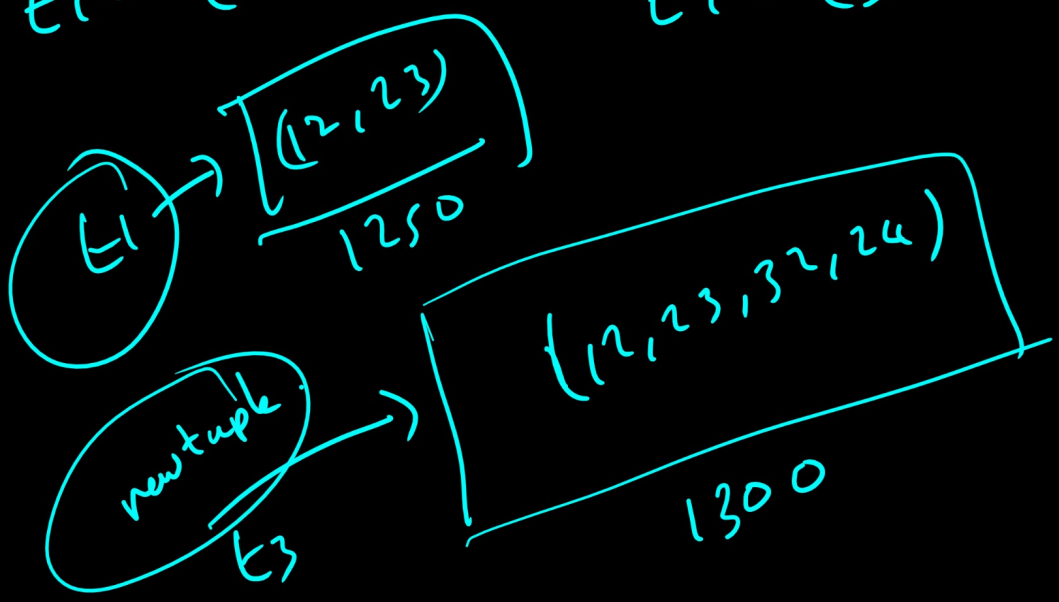
s = "Venky"

s += "Vicky"



t1 = (12, 23)

t1 + (32, 24)



function → def funcn ( ):
return

Arguments → Keyword
Positional
Combination of both

Scope of variable inside function.

How it affects outside function. → Passing (mutable & immutable)