

## \* Data structure in Python:

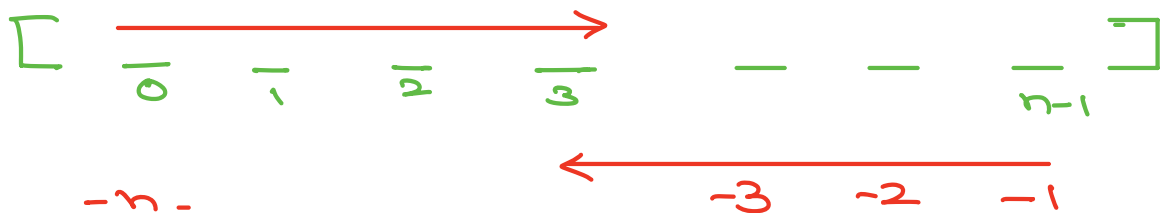
- ① List
- ② Set
- ③ Dictionary
- ④ Tuple

### List

- ① List is heterogeneous
- ② Lists are dynamic
- ③ Lists are mutable

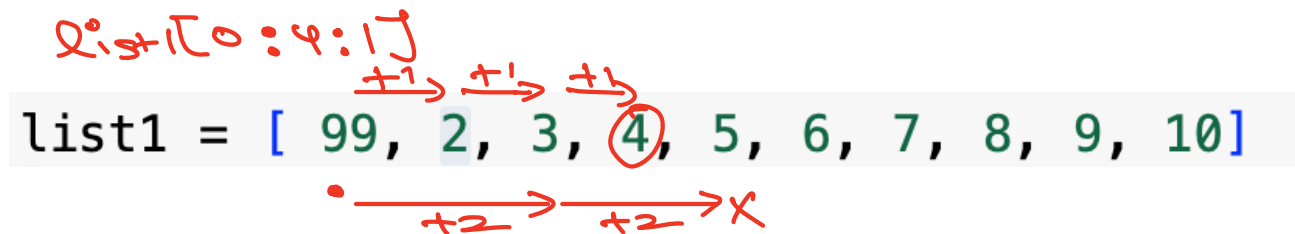
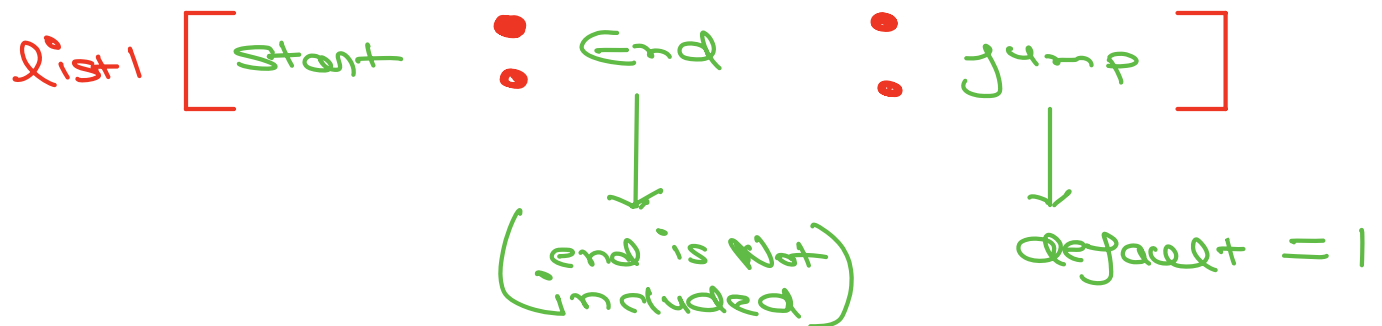
### Indexing

length =  $n$



# Slicing

③ Access Set of Elements

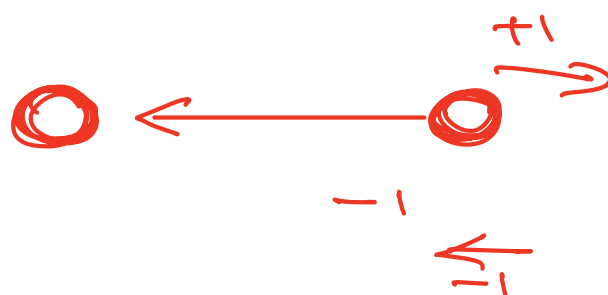


list1[0:4:2]

99, 3

0 1 2 3 4 5 6 7 8 9  
 list1 = [ 99, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]  
 ←  
 -5 -4 -3 -2 -1

list1[-1:-5:-1]



[1, 8] → [10]  
 7 8 9 10  
 ↓  
 9

[5] [7] [9]  
 ← 100  
 -2

list [-1:-5:-1]



# Tuples

③ Immutable Lists

# Dictionary

- ① Key : Value Pairs
- ② Mutable
- ③ Indexing in case of dictionary is done using 'Key'
- ④ All the keys must be 'immutable' while value can be either mutable or 'immutable'
- ⑤ Key will always be Unique

① Functions

② Sets

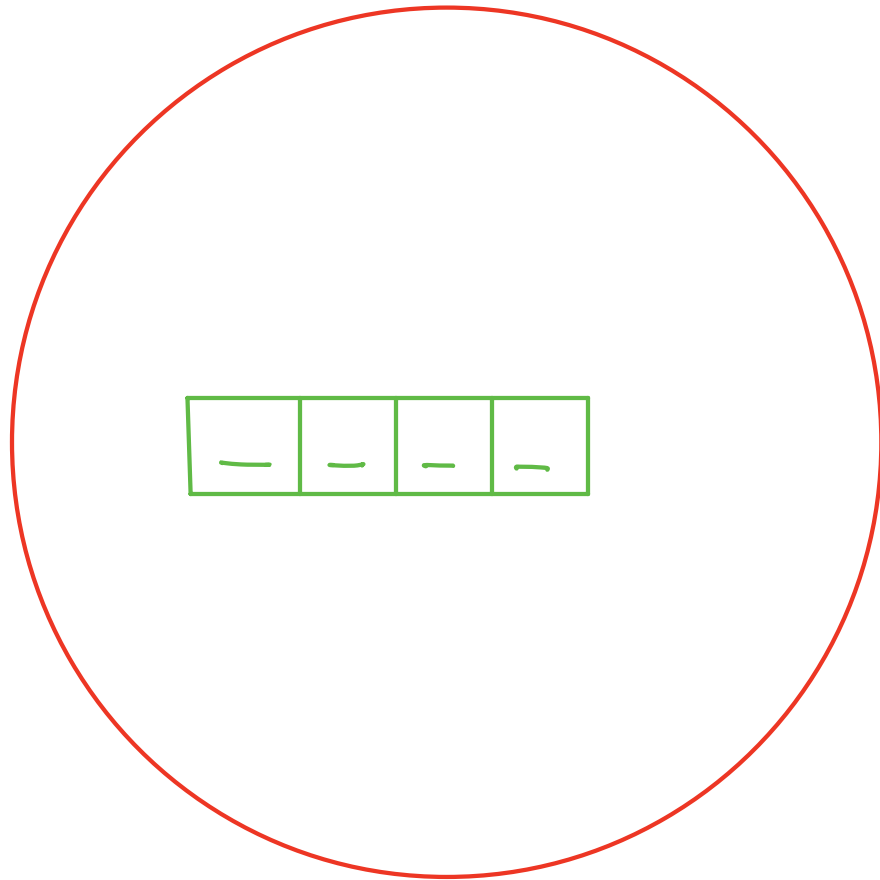
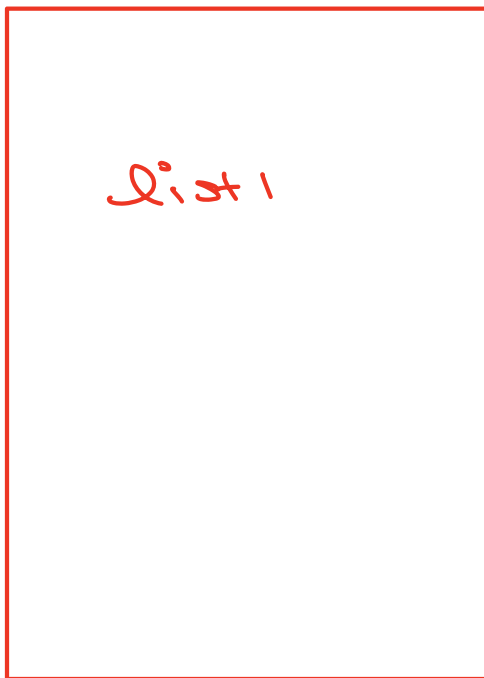
③ Comprehension

④ String

Pending

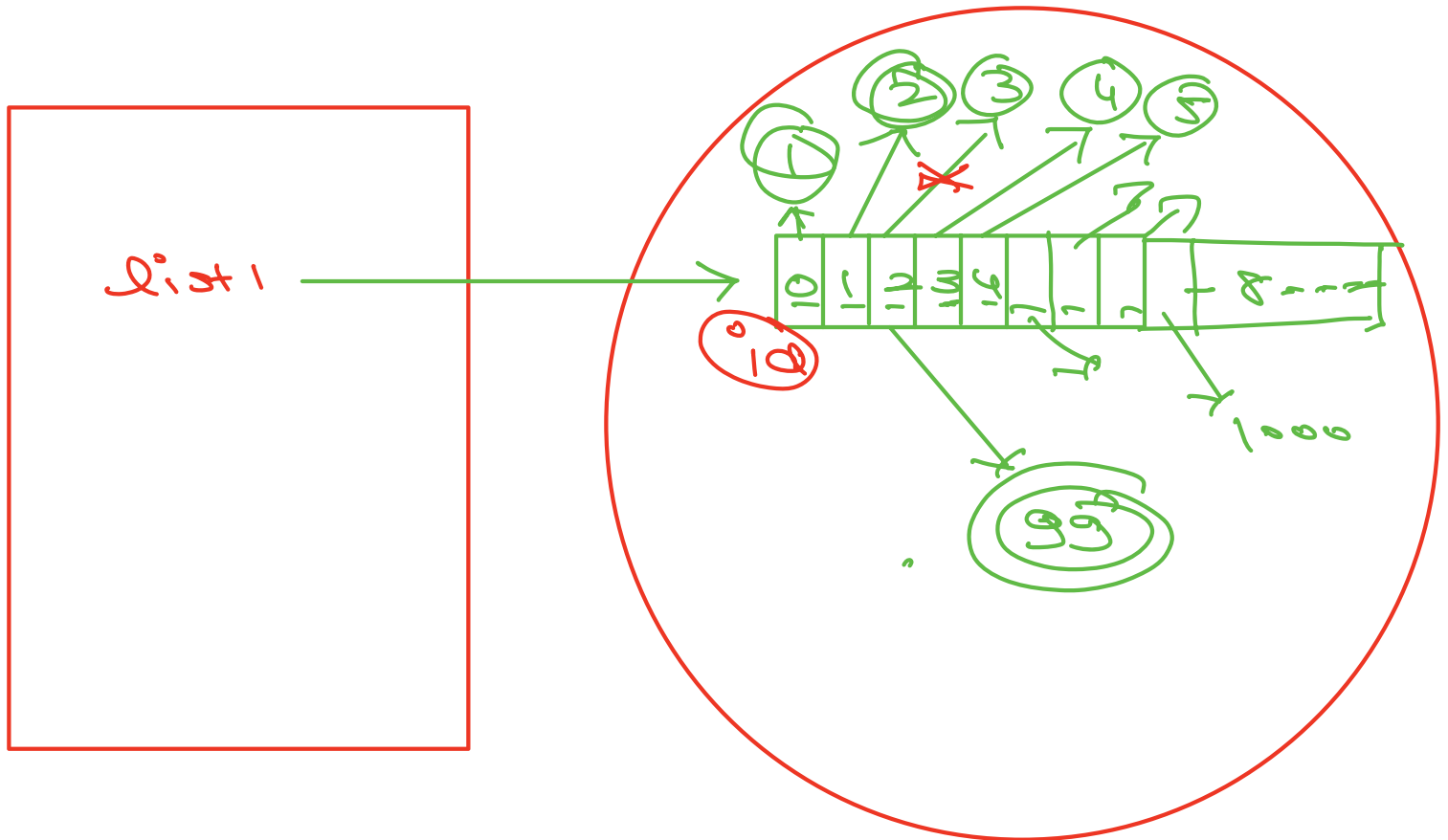
# Memory Allocation of List

List = [ ]



④, ⑧, 16, 32, 44, - - - -

①  $list1 = [1, 2, 3, 4, 5] \rightarrow 8 \text{ blocks}$



$list1[2] = 99$

$list2.append(99) \dots (100)(200)$

8 slots are filled

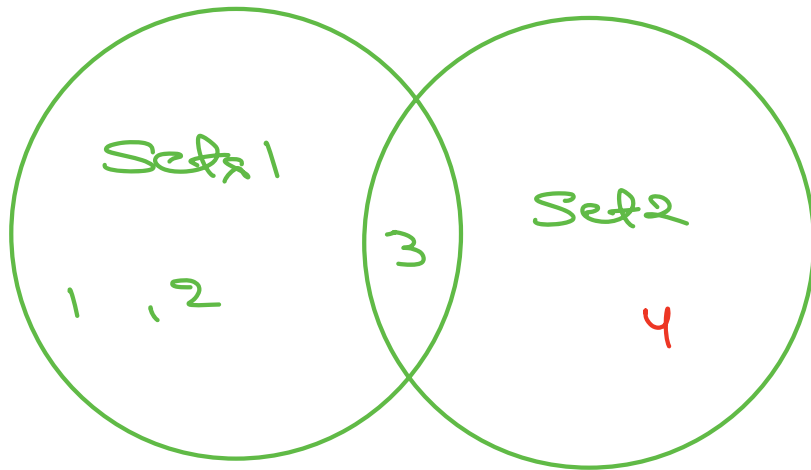
$list2.append(1000)$

# Set

① A collection of Unique Elements (keys)

② Set 1  $\ni 1, 2, 3$

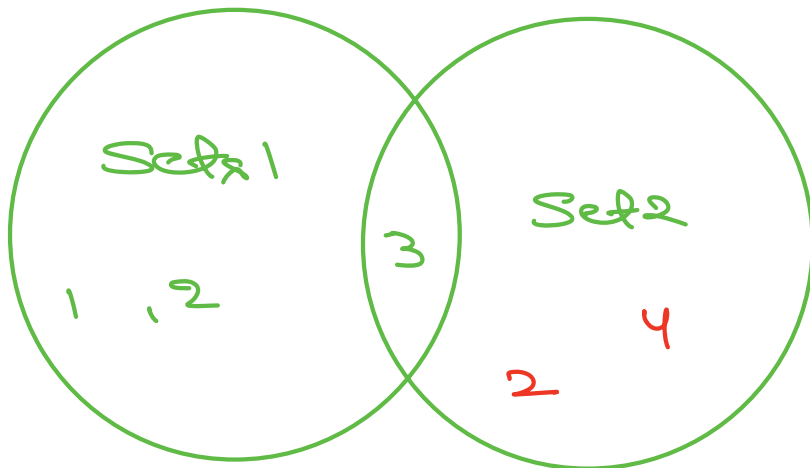
dic  $\ni$  Key



Intersection

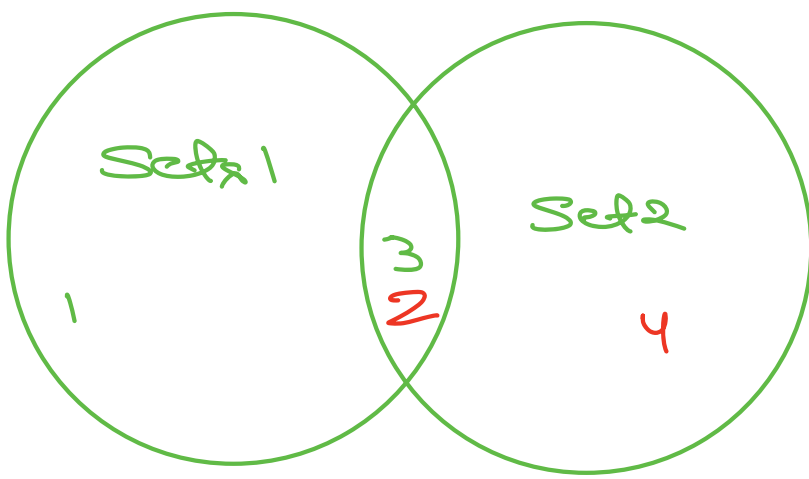
$$S_1 \cap S_2 = \{3\}$$

$$S_2 \cap S_1$$



Union

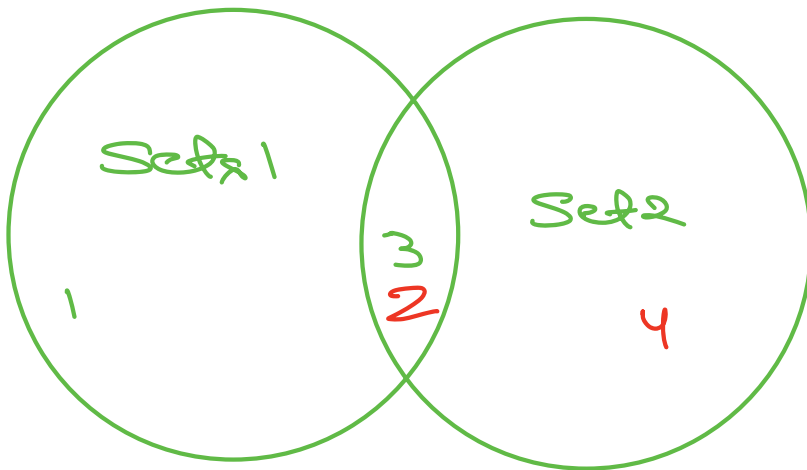
$$S_1 \cup S_2 = \{1, 2, 3, 4\} = S_2 \cup S_1$$



difference

$$S_1 - S_2 = \{1\}$$

$$S_2 - S_1 = \{4\}$$



Symmetric  
difference

$$S_1 \Delta S_2 = (S_1 \cup S_2) - (S_1 \cap S_2)$$

$$= \{1, 4\}$$



# Comprehension

→ A shorthand Notation of creating Lists, sets and Dictionary

---

## String

→ chars, words, sentences

→ Immutable

→ iterable

str1 = 'Hello'

for i in str1:

H  
e  
l  
l  
o

✓ iterable

str1[2] = 'D' ✗ mutable

① Split a String  $\rightarrow$  List

② Join  $\Rightarrow$  List  $\rightarrow$  String

## Function

a A piece of code that can be used to do repetitive tasks

def function\_name (parameter  
par1, par2)

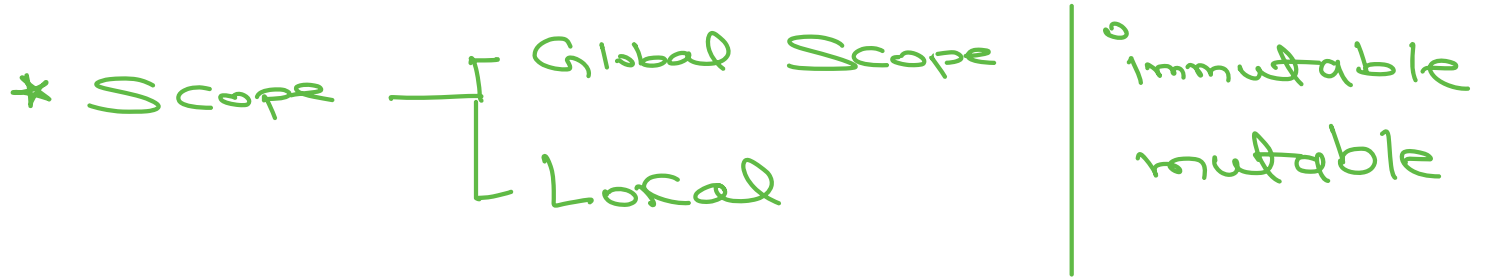
---

---

---

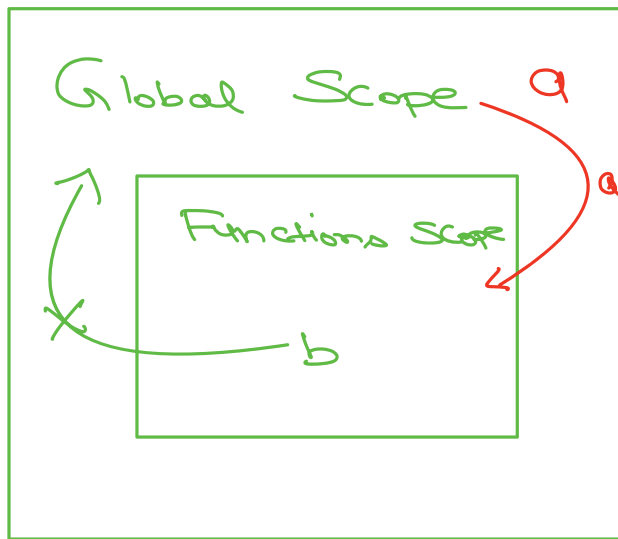
print (value)  
return Value

① = function\_name (val1, val2)  
Arguments



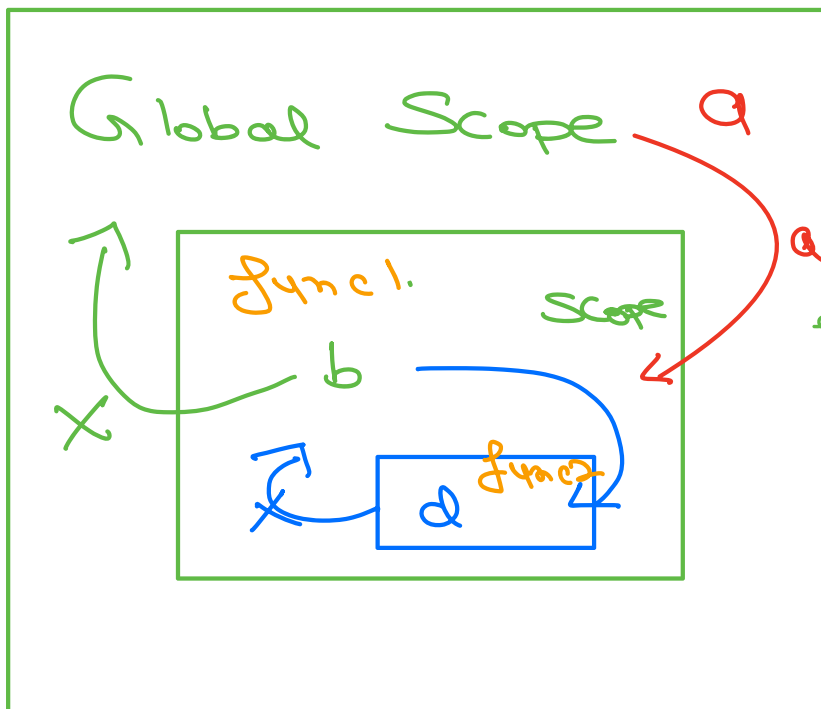
Resume @ 10:28

Accessibility



$a = 10$

def func  
 $b = 10$

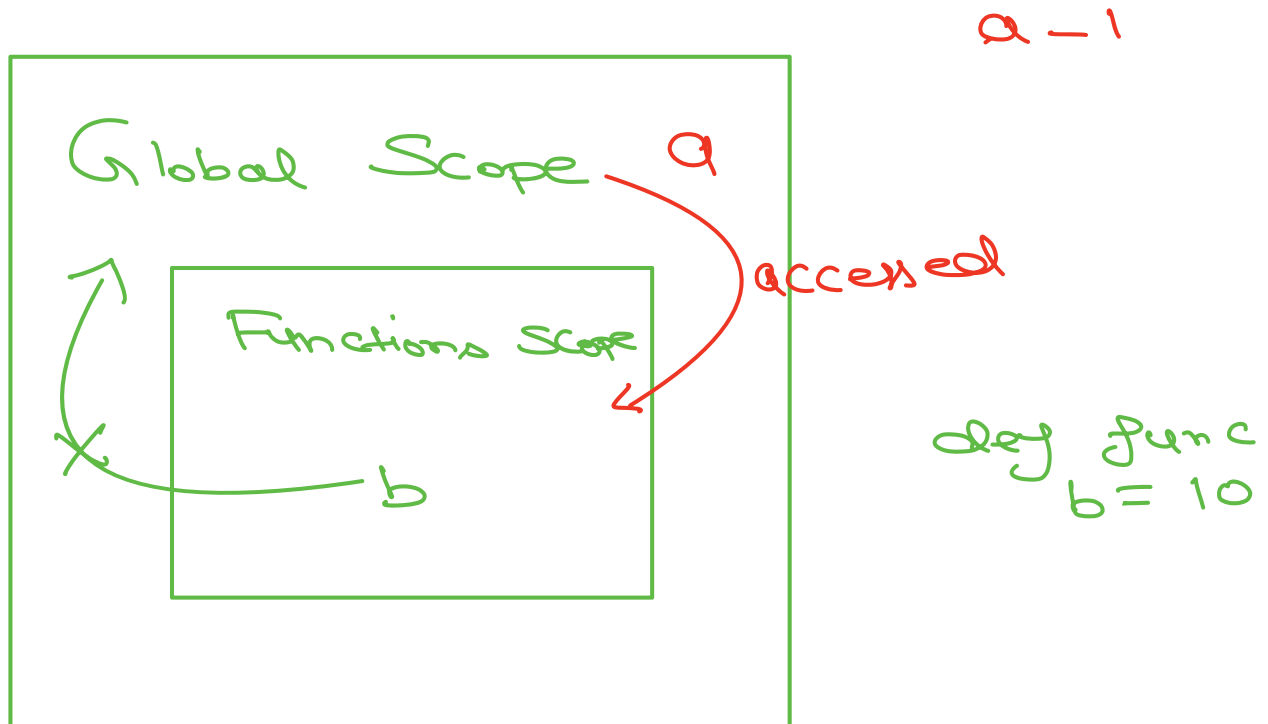


$a = 10$

def func1()

$b = 10$

def func2()  
 $a = 20$



Can we modify Global Variables

immutable

mutables

Can't modify Directly

Can modify Directly

Global keyword