26/7/25

Day 15

Loops in Python

For Loop in python :

1) Items

2) Range Values 1 → 10
                 100 → 0

Array index → (0 → n)

for {variable_name} in {collection}:

{collection} : List, Set, Range.

Range() → it is fn

for ex:- 1 to 10 in range.
it will generate

(1, 2, 3, 4 ... 10)

Ex:-

```
for name in names:

for student in students:

for n in range(1,10):
```

Date :...../...../.....
Page No. :.....

def :-

The for loop in python is used for iterating over a sequence (such as a list, tuple, dictionary, set, or string).

-> python's for loop is more like a "for-each" loop in other languages.

Syntax Explanation :-

A for loop in python iterates over items of any sequence (like a list, tuple, string / dictionary) in the order they appear.

General Syntax

for variable in iterable :

Variable :- A name representing the current element in the iteration

Iterable :- Any object capable of returning its elements one at a time

Ex :- list, string, tuple, dictionary, set, range.

=> The loop executes the block once for each item in the iterable.

Basic Syntax :-

```
for element in iterable :
```

Understanding range() Function

The range() function is commonly used with for loops to generate a sequence of no.'s.

-> It can take one, two, / three arguments:

Syntax :-

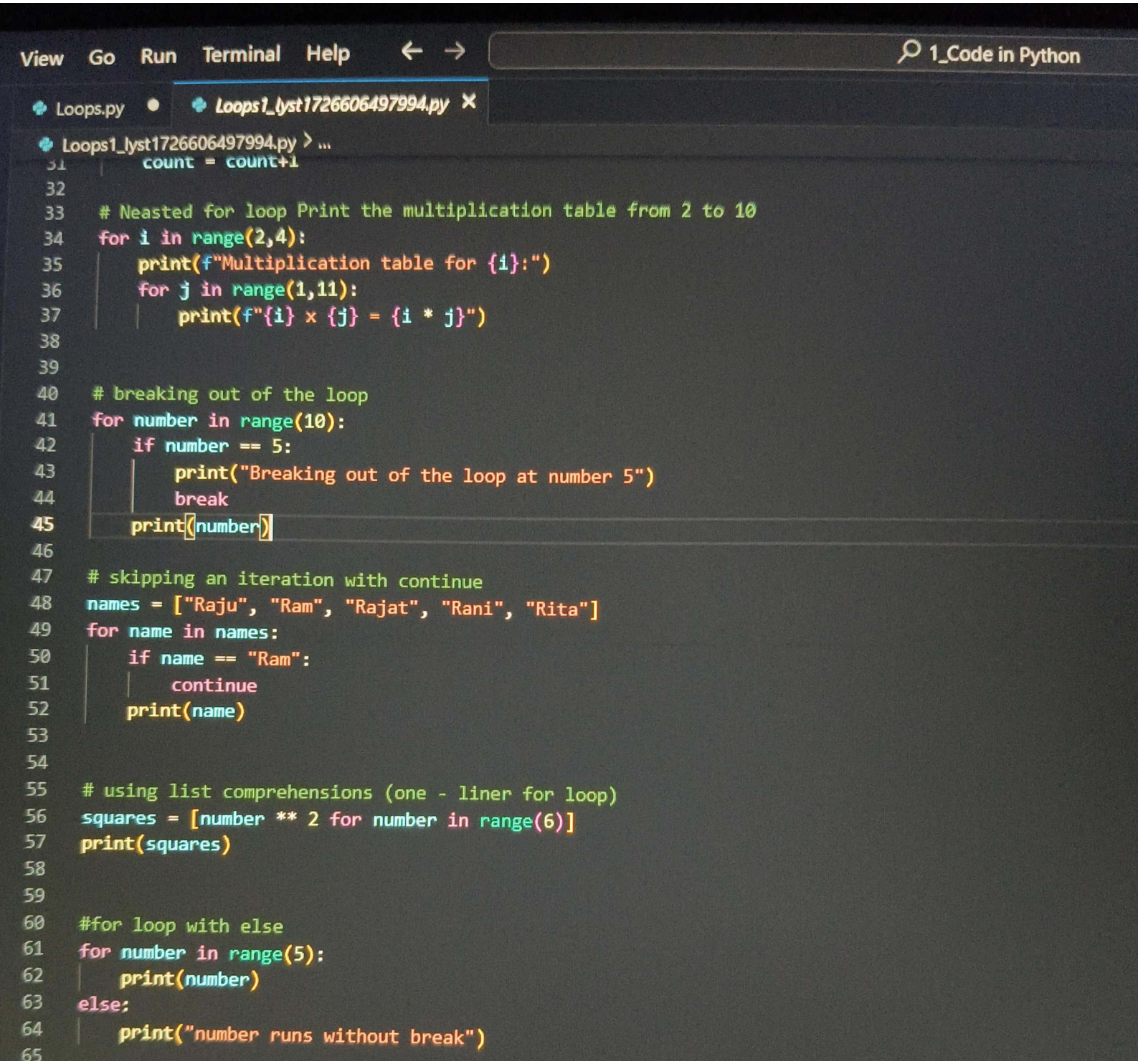
```
range ( stop)          # 0 to stop - 1
range (start, stop)    # start to stop - 1
range (start, stop, step)
                       # start to stop - 1 with increment/
                         decrement of step.
```

Ex:-

```
for number in range (3):
    print (number)     # O/P  0, 1, 2
```

Loops1_lyst1726606497994.py > ...

```python
# for loop normal counting starting from 0 to 9
for i in range(10):
    print(i)

# item in values
values = range(10)
for item in values:
    print(item)


#  for loop(start, stop, step) for reverse counting
for index in range(10, 0, -1):
    print(index)

# for loop(start, stop, step) for 1 + 2 = 3, 3+2 = 5, 5+2 = 7, etc.
for index in range(1, 10, 2):
    print(index)

# for name in names also prints the name in uppercase
names = ["Raju", "Ram", "Rajat", "Rani", "Rita"]

for name in names:
    print(name)
    print(name.upper())


# counting using while loop
count = 0
while(count <= 10):
    print(count)
    count = count+1
```


View Go Run Terminal Help
1_Code in Python
Loops.py
Loops1_lyst1726606497994.py
Loops1_lyst1726606497994.py > ...
count = count+1
# Neasted for loop Print the multiplication table from 2 to 10
for i in range(2,4):
print(f"Multiplication table for {i}:")
for j in range(1,11):
print(f"{i} x {j} = {i * j}")
# breaking out of the loop
for number in range(10):
if number == 5:
print("Breaking out of the loop at number 5")
break
print(number)
# skipping an iteration with continue
names = ["Raju", "Ram", "Rajat", "Rani", "Rita"]
for name in names:
if name == "Ram":
continue
print(name)
# using list comprehensions (one - liner for loop)
squares = [number ** 2 for number in range(6)]
print(squares)
#for loop with else
for number in range(5):
print(number)
else:
print("number runs without break")

Loops.py ● Loops1_lyst1726606497994.py ×

Loops1_lyst1726606497994.py > ...

```
58
59
60  #for loop with else
61  for number in range(5):
62      print(number)
63  else:
64      print("number runs without break")
65
66
67  #for loop with else with break
68  for number in range(5):
69      print(number)
70      if number == 3:
71          print("3 is found so break")
72          break
73  else:
74      print("number runs without break")
75
76  #for loop with else with break with list
77  numbers = [1,2,3,4,5,6,]
78  for number in numbers:
79      print(number)
80      if number == 3:
81          print("3 is found so break")
82          break
83  else:
84      print("3 is not found")
```

Assignments > Loops.py > ...

```python
""" Python Questions """

# 1. Write a Python program using a for loop to print numbers from 1 to 10.
for number in range(1,11):
    print(number)

#2. Write a Python function using a while loop to count down from 5 to 1.
count = 5
while count >= 1:
    print(count)
    count -= 1

#3. Write a Python function that asks for user input until the word 'exit' is entered.
#ChatGPT
def read_until_exit():

    while True:
        text = input("Enter something (type 'exit' to quit): ")
        if text.lower() == "exit":

            break

        print(f"You typed: {text}")
read_until_exit()

```

Run Terminal Help 1_Code in Python

Loops.py Loops1_lyst1726606497994.py

ssignments > Loops.py > ...

```
25
26  #4. Create a Python function using a for loop and range() to print even numbers from 2 to 20.
27  for number in range(2,21,+2):
28      print(number)
29
30  #5. Write a program that iterates over a list of names and prints each with a serial number using enumerate()
31  #ChatGPT
32  names = ["Raja","Rani","Rajat","Ram"]
33  for i, name in (enumerate(names,start=1)):
34      print(f"{i}", name)
35
36  #6. Simulate a do-while loop in Python that asks for a number until the user enters a negative number.
37  #ChatGPT
38  while True:
39      number = int(input("Enter a number (negative to exit): "))
40      if number < 0:
41          print("This is a negative number. Exiting...")
42          break
43      print("Not a negative number, try again.")
44
45
```

```python

#7. Create a Python program that uses a nested loop to print the multiplication table from 1 to 5.
for i in range(1,6):
    print(f"this is {i} table")
    for j in range(1,11):
        print(f"{i} * {j} = {i * j}" )

#8. Write a Python function that breaks the loop when the input number is divisible by 7.
def break_number(number):
    while number % 7 != 0:
        print("the number is not divisible by 7")
        break
    else:
        print("number is divisible by 7")
number = int(input("enter a number: "))
break_number(number)

# 9. Write a Python program using continue to skip numbers divisible by 3 from 1 to 15.
for number in range(1,16):
    if number % 3 == 0:
        continue
    print(number)

#10. Write a Python program to iterate over two lists using zip() and print the paired elements.
names = ["raja", "rani" , "roro", "rocket"]
scores = ["10", "20", "30", "40"]

for name,score in zip(names,scores):
    print(name,score)
```

28/7/25

Day 16

Date :...../...../.....

Page No. :.....

Recursion

1) Functions

2) Stack

3) Why & where recursion is used

   a) Folder structure

   b) Family tree

4) Important techniques to write recursive code

5) Common pitfalls of recursion

6) Seeing recursion live demo

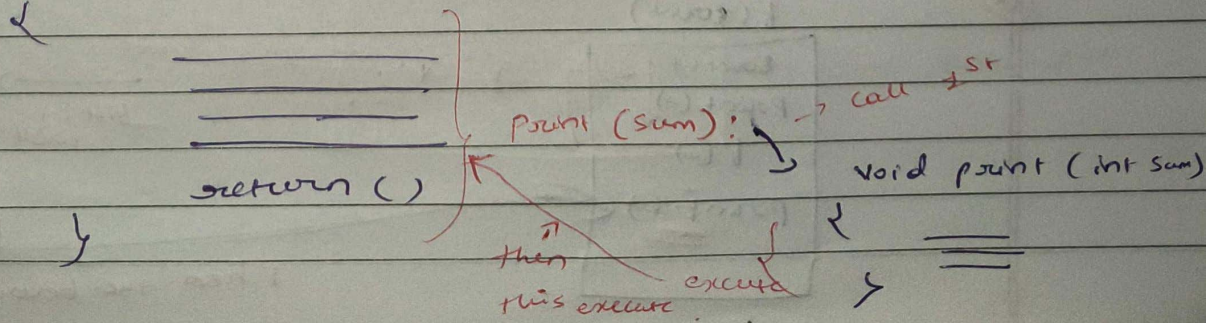
7) Recursion infinite loop / stack overflow demo / Sum

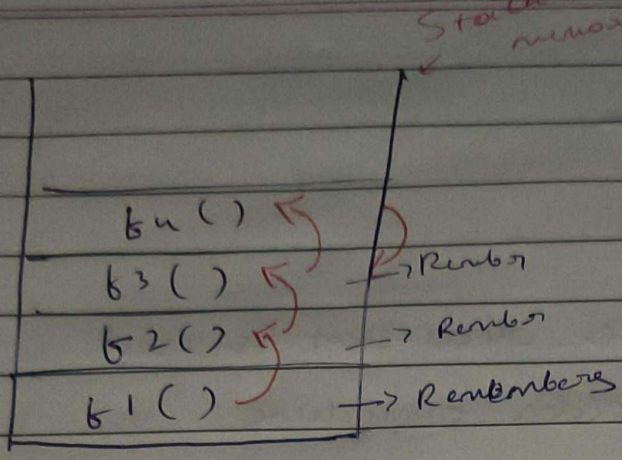
8) Code demo with different examples

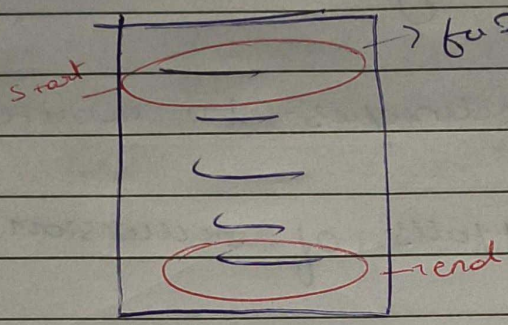
Syntax of function

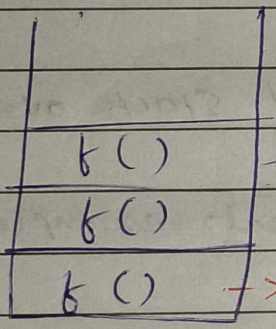
```
<return type> functionName (arguments)
```

Sn:- `int getSum (int num1, int num2)`



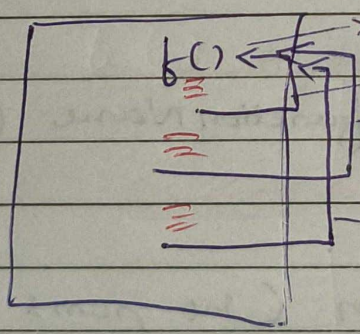




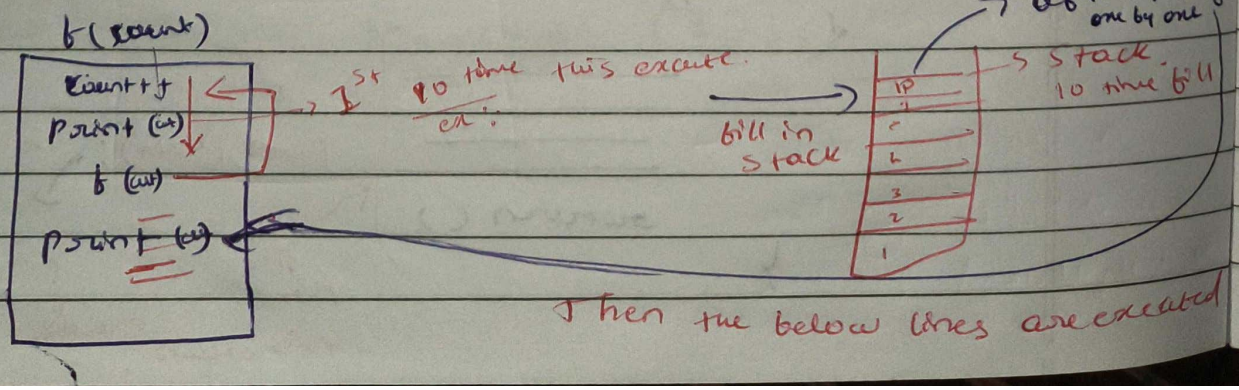


→ same function calling itself is called recursion

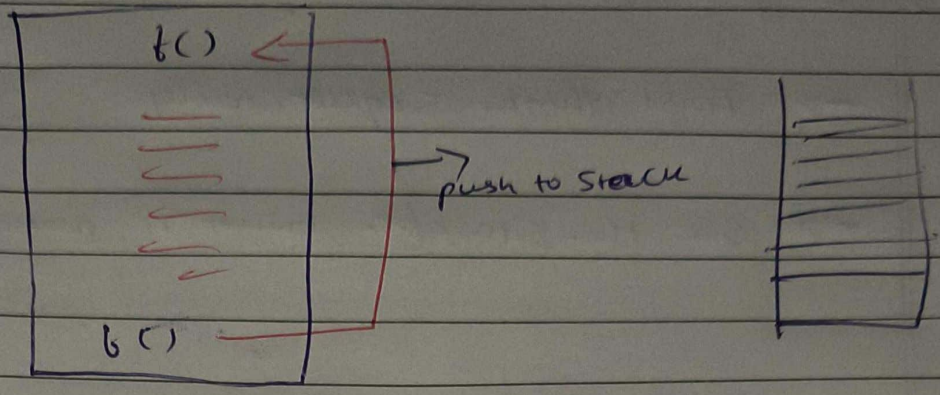
→ Remembering the state.

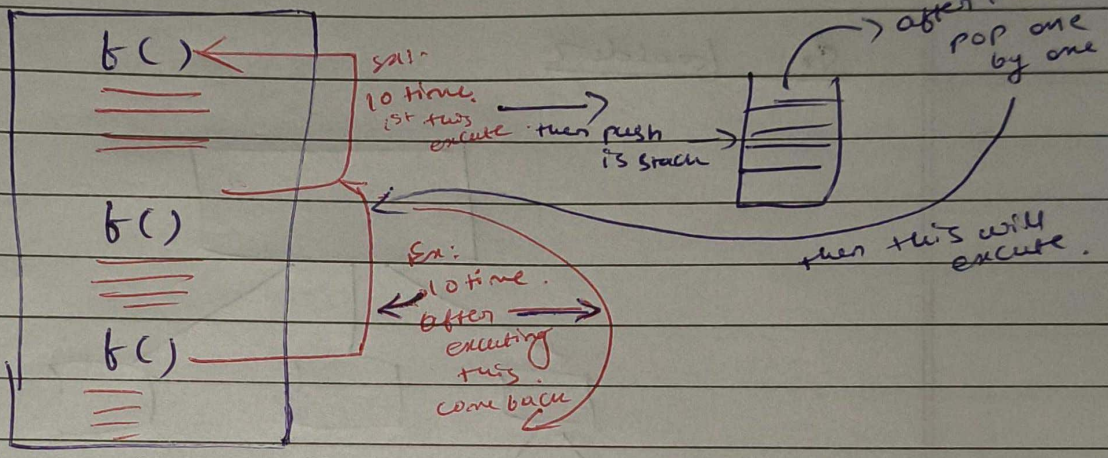


Pattern



* fu° calling in end





3) Why & where recursion is used.

a) Folder structure → Recursion is used.

↓

-Coding

- C prgm.. → Arrays
- Java → linked list.
- python

b) Family tree → Recursion example.

4) Important techniques to write recursive code

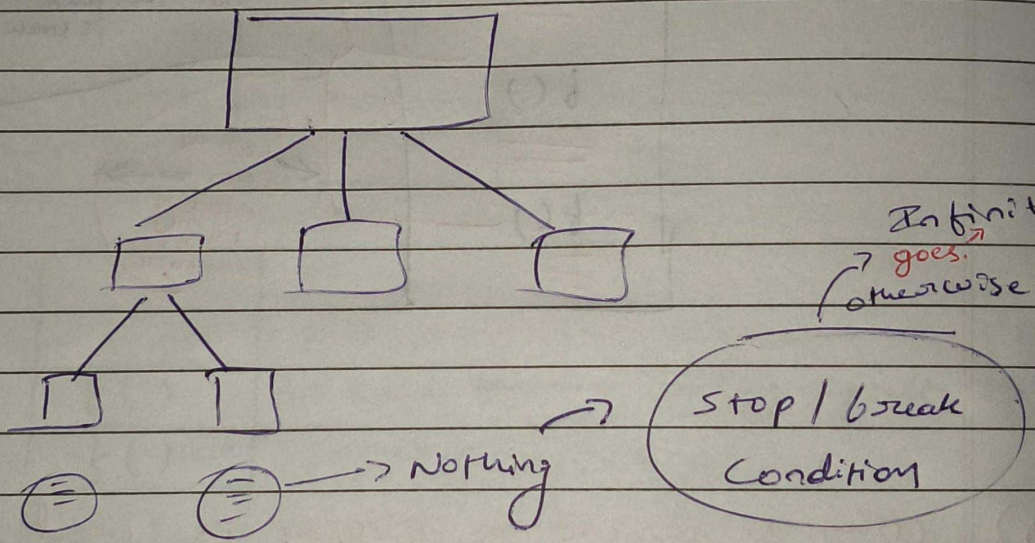
-> First think sequencially (easy / hard)

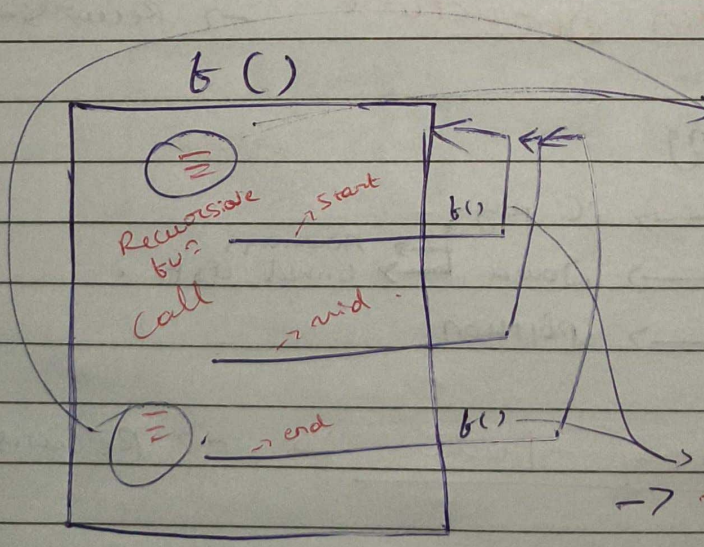
-> See the storage ~~what~~ is more / less in sequencially

5) common pitfalls of recursion :-

1) Break setting wrong.

Ex: foolder





-> Before u call the fu? what is the code u written & after that what code u written

-> The recursion calling 1 time or 2 time / multiple times

Assignments > Recursionassignment.py > ...

```python
"""Python - 5 Recursion Questions """
#1. Print numbers from 1 to N using recursion.
def print_numbers(number):

    if number == 0:
        return
    print_numbers(number - 1)
    print(number)

print_numbers(5)

#2. Calculate the sum of first N natural numbers using recursion.

def natural_number_sum(number):
    if number == 1:
        return 1
    return number + natural_number_sum( number - 1)

print(natural_number_sum (3))

#3. Find the factorial of a number using recursion.
def natural_number_sum(number):
    if number == 1:
        return 1
    return number * natural_number_sum( number - 1)

print(natural_number_sum (4))

```

```python
#4. Reverse a string using recursion.
def reverse_string(name:str):

    if len(name) == 0:
        return name
    return reverse_string(name[1:]) + name[0]

name = "Raja"
print(reverse_string(name))

#5. Check if a number is a power of 2 using recursion.
def power_two(number):

    if number < 0:
        return False
    if number == 1:
        return True
    if number % 2 != 0:
        return False
    return power_two(number // 2)

result = power_two(6)
print(result)

if result:
    print(f"power of two")
else:
    print("not power of two")
```