

# Full stack web development using python

## Recursion



Saurabh Shukla (MySirG)

## Agenda

- ① What is recursion?
- ② Recursion Tree
- ③ How to approach recursive solution?

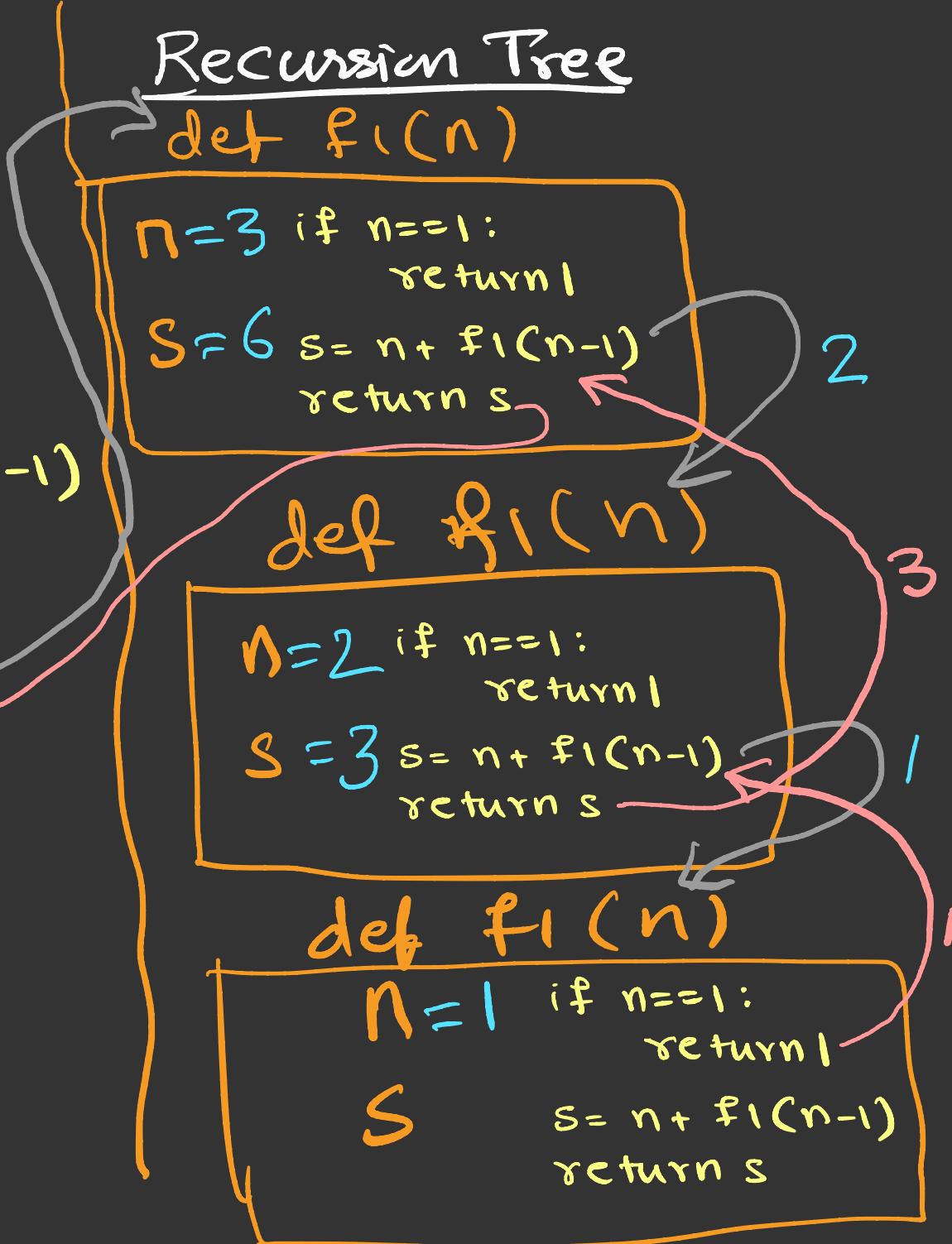
## Recursion

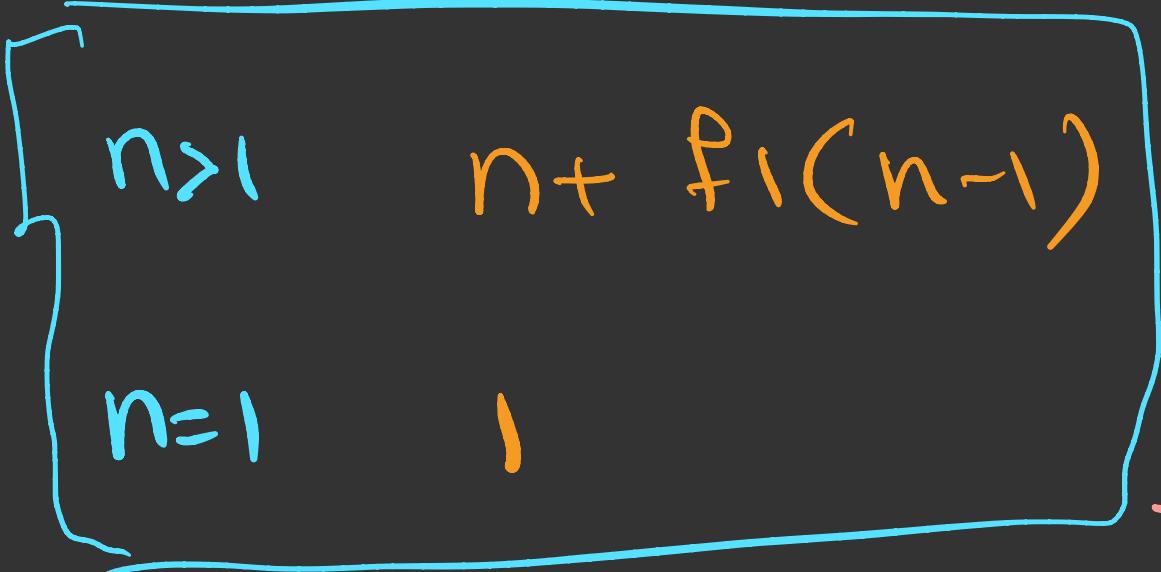
Function calling itself is called recursion.

```
def f1(n):  
    if n==1:  
        return 1  
    s = n + f1(n-1)  
    return s
```

```
r = f1(3)  
print(r)
```

6



$f_1(3)$  $\curvearrowleft 3 + f_1(2)$  $\curvearrowleft 2 + f_1(1)$ 

$$\rightarrow 5 + f_1(4)$$

$$\rightarrow 5 + 4 + f_1(3)$$

$$\rightarrow 5 + 4 + 3 + f_1(2)$$

$$\rightarrow 5 + 4 + 3 + 2 + f_1(1)$$

$$\rightarrow 5 + 4 + 3 + 2 + 1$$

- In recursion, problem is solved in terms of problem itself
- Each time recursive function call to itself for little simpler version of the original problem

## How to approach recursive function?

Write a recursive function to calculate sum of first n natural number

```
def add(n):
```

```
    if n==1:
```

```
        return 1
```

```
    return n+add(n-1)
```

①  $\text{add}(n) = 1 + 2 + 3 + \dots + n$

RC ②  $n + \text{add}(n-1) = 1 + 2 + 3 + \dots + n-1$

③  $n = -1$

Write a recursive function to calculate sum of squares of first n natural numbers.

```
def SquareSum(n): ① SquareSum(n)  $1^2 + 2^2 + 3^2 + \dots + n^2$ 
    if n==1: ② n*n + SquareSum(n-1)  $1^2 + 2^2 + 3^2 + \dots + (n-1)^2$ 
        return 1 ③ n==1
    return n*n + SquareSum(n-1)
```

Write a recursive function to print first n natural numbers.

def printN(n):

    if n > 0:

        printN(n-1)

        print(n)

    ① printN(n) 1 2 3 4 ... n

    ② printN(n-1) 1 2 3 ... n-1  
        print(n)

    ③ n == 0

printN(6)

