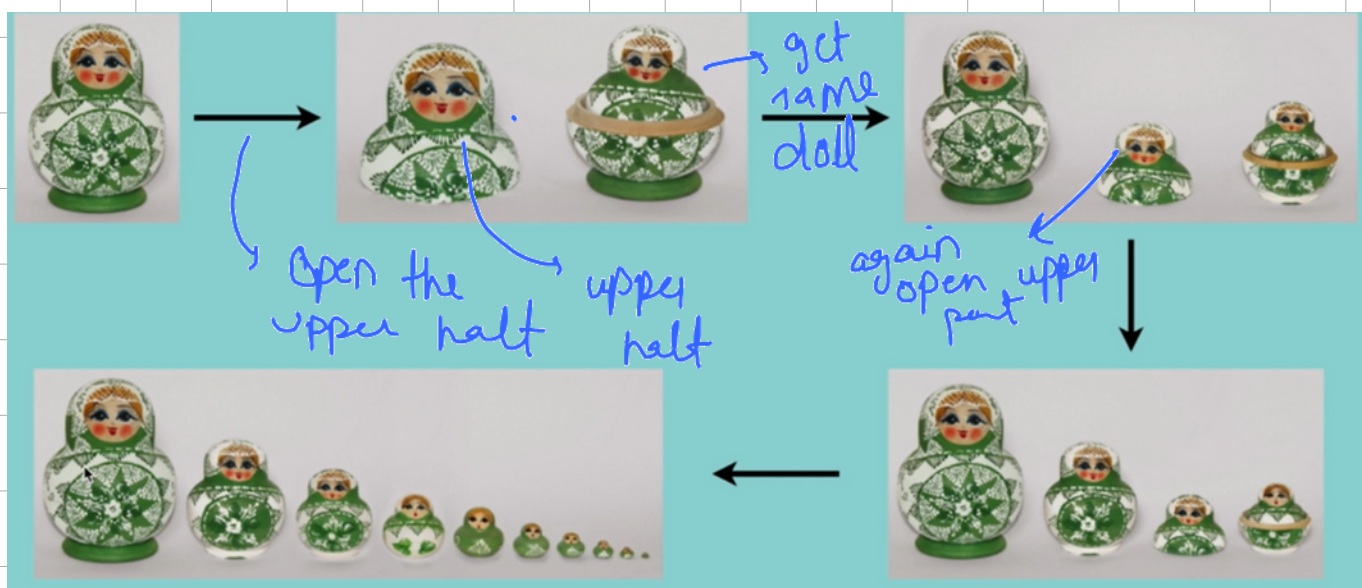


## ② Recursion

(i) What is Recursion?

↳ A way of solving any problem where function calls itself.  
↳ we solve problem by solving small instance of the problem.

e.g Russian Doll - There are same dolls inside a doll. You keep on opening it and you will get a same doll but smaller in size. (until the last one)



(ii) Properties of Recursion:-

① Performing same operation with different inputs.  
↳ like in above russian dolls, every time we had different sized doll.

② Based condition is used to stop recursion.  
↳ like A russian doll can have 10 smaller russian dolls in it.

So → code for Russian Dolls will be like;

```
def open_russian_doll(doll):  
    if doll == 1:  
        print('All dolls are open')  
    else:  
        open_russian_doll(doll - 1)
```

we give largest doll here

otherwise

keep on opening the next doll.

condition  
↳ so doll will be set to 1 when last doll will come.



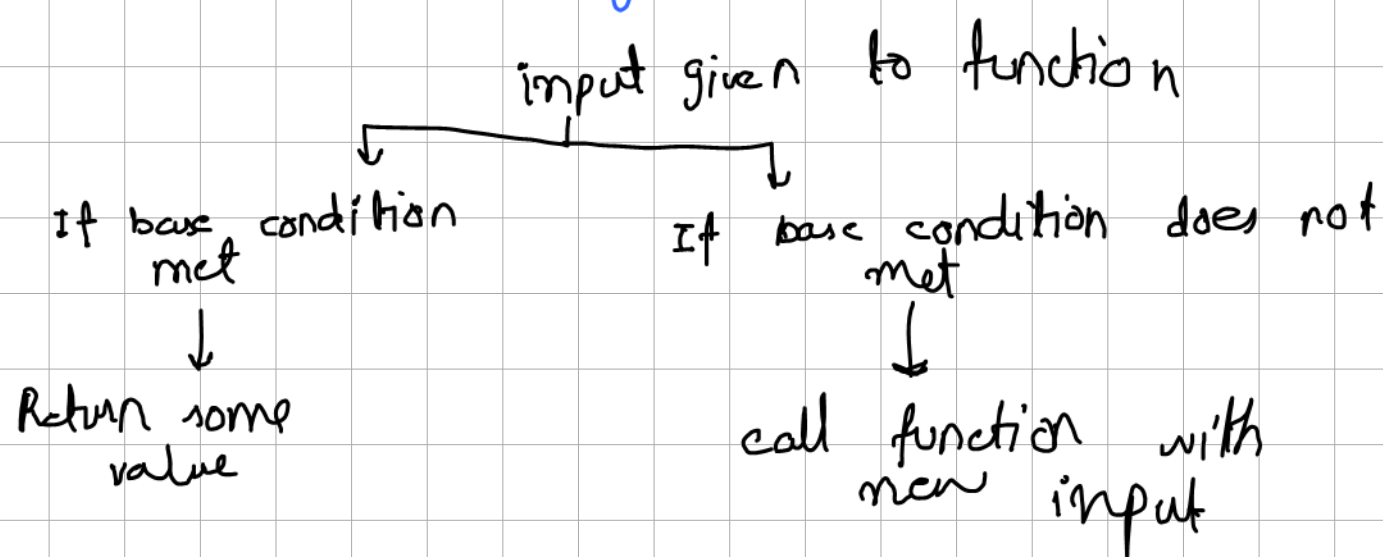
Produced with a Trial Version of PDF Annotator - www.PDFAnnotator.com

## Why Recursion is important?

- (i) You break the problem into smaller pieces & solve each piece and that makes the job easy.
- (ii) Recursion has huge usage in Data structures like Trees & Graphs.
- (iii) Also it is part of many Algorithms.

## How Recursion works?

- (1) Function calls itself
- (2) Exit this calling loop when met the condition.



(i) Before Recursion lets see how function works:  
consider following code:

```
def first_function():
    second_function()
    print('I am first function.')

def second_function():
    print('I am second function.')

first_function()
```

So what happens in stack is,

computer memory

STACK

second function  
first function

when second function is executed, comes back to first function

STACK

first function

STACK  
NULL

when first function is executed

