Recursion

when a Function calls itself, that function is called a recursive function.

Any problem that can be solved using iteration, that can be programmed using Recursive approach as well.

Applications of Recursion

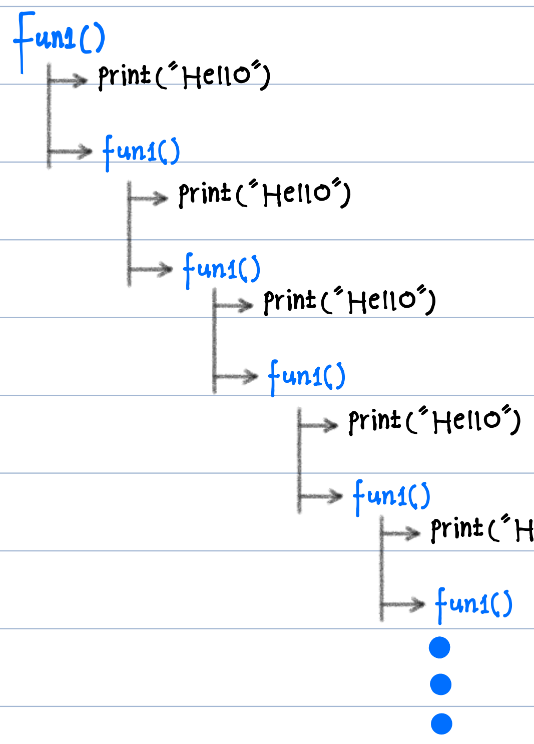
① Divide and conquer (Binary search, Quick sort)
② Dynamic Programming

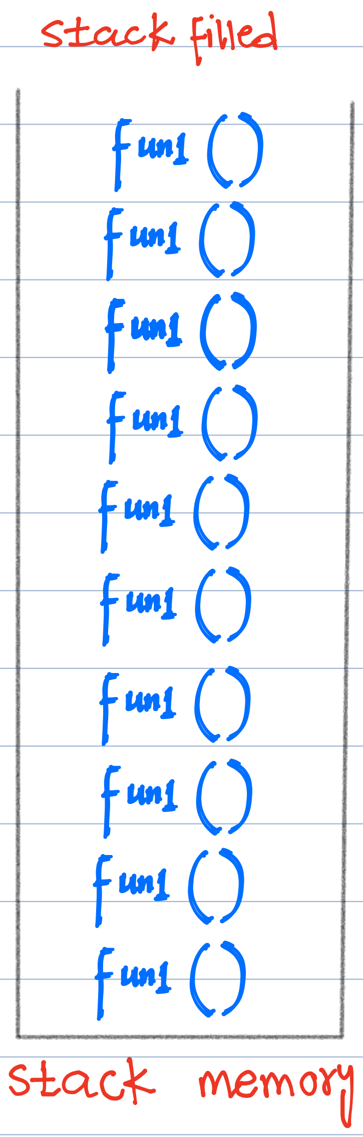
which one is better, solving recursively or iteratively ?

There are problems that can be solved using recursion efficiently and those are recursive in nature.
For instance, if answer to a problem can be achieved by repeatedly by dividing that problem into smaller problems, it is best to use recursive approach there.

How to solve a problem using recursion?

```
def    fun1 ():
           print ("Hello")
           fun1 ()
```





function calls are stored in stack memory and that memory is limited; if a function is called beyond certain The stack memory will full and cannot accept more calls.

it can be observed that if a function continously calls itself, the stack memory will be filled and maximum recursion error will happen.

How to handle maximum recursive calls?

Base case: The main idea behind recursion is to reduce a larger problem into smaller problems and build the answer for that larger problem from these smaller problem; The point where a smaller problem cannot be divided into smaller problem is known as the base case.

This will be point where your function cannot be executed further, either it will return an answer or just return from the function.

Base case should be added as a condition and handled explicitly inside the function.

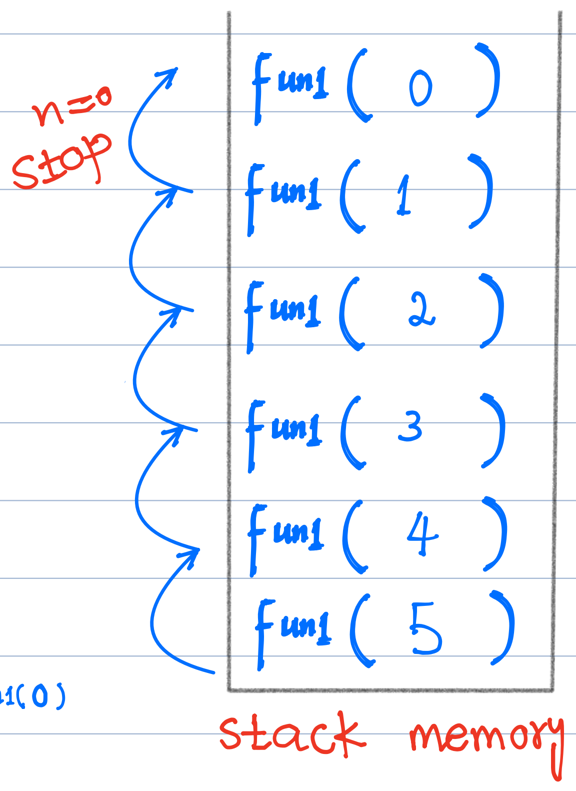
Now, lets define a base for previous function.

```
def  fun1 (n):
     if  n<=0:
          return
     Print ("Hello")
     fun1 (n-1)
```

// n<=0 is our base case; at every call the value of n is decreasing by 1 until it reaches 0 and stops.

n=5

```
fun1( 5 )
 |--> print("Hello")
 |--> fun1( 5-1 )
       |--> print("Hello")
       |--> fun1(4-1)
             |--> print("Hello")
             |--> fun1(3-1)
                   |--> print("Hello")
                   |--> fun1( 2-1)
                         |--> print("Hello")
                         |--> fun1( 1 - 1 )=fun1(0)
                                    ↓
```



when we reach to fun(0)
n=0, the function will stop
calling itself and returns.

A recursive function's body

```
def function(Parameter):

    • • • • • • • • • • • • •
      Base case
    • • • • • • • • • • • • •

    • • • • • • • • • • • • •
    Recursive call ( such that the Parameter
    inside the function approaches the base we
    defined earlier)
    • • • • • • • • • • • • •

    • • • • • • • • • • • • •
```