Recursion

→ Concept in progmmig

→ Fuction calls itself to solve a problem

→ A problem can be solved by dividing into sub-problems

→ Why & when

① Problem can be divided into sub problems.

② Base case.

→ ③ Tree, Graph traversal.

→ ④ Backtracking.

---

① Base condition

→ there is a simple solution of the problem that can be directly solved.

② Recursive call

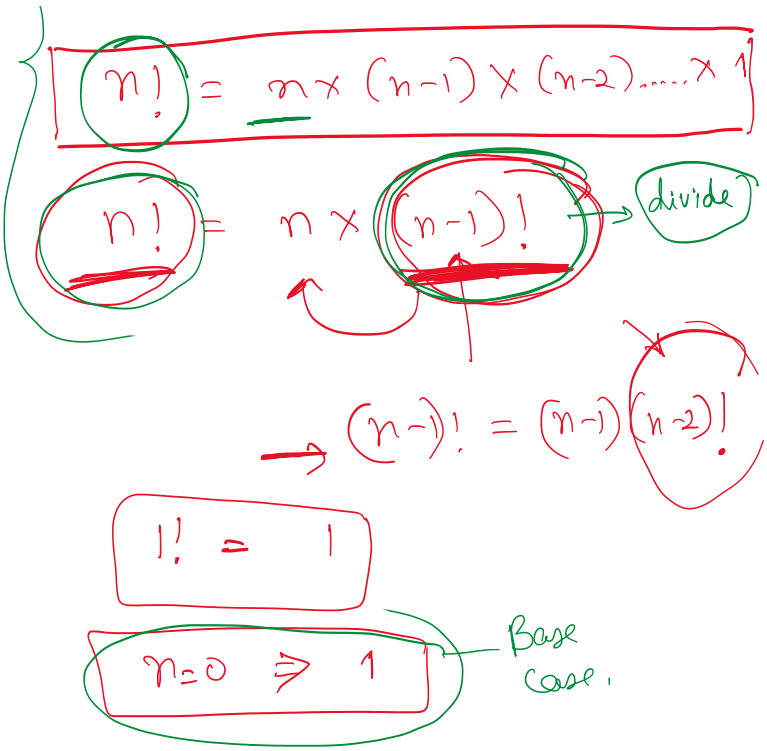
→ we should be able to divide the problem.

③ Combine the results.

Example :- Factorial of a number

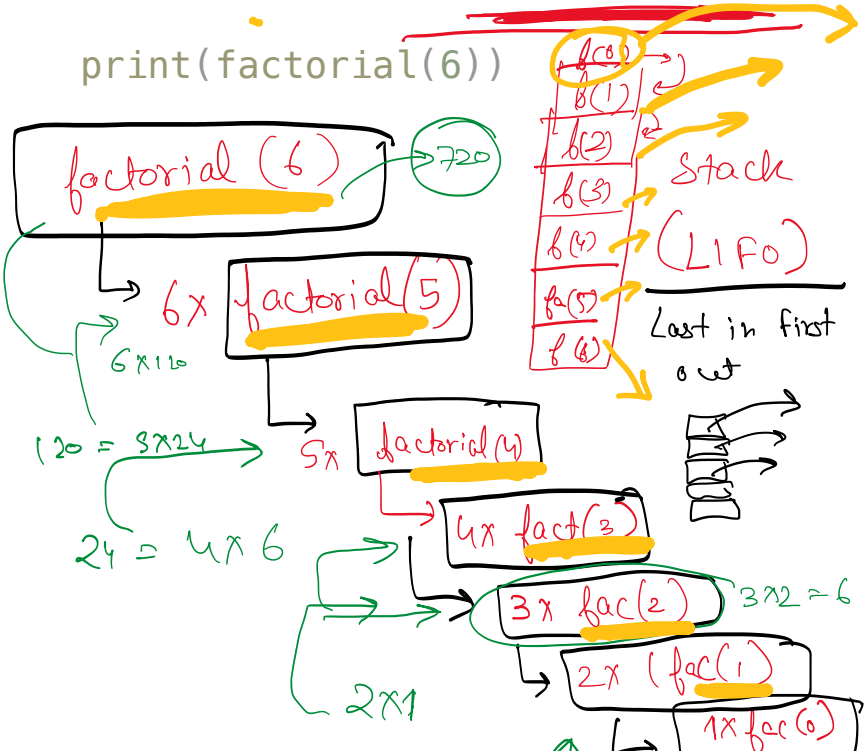
Eg $5! = 120$

$= 5 \times 4 \times 3 \times 2 \times 1$




```python
def factorial(n):
    # base condition
    if n==0:
        return 1
    else:
        return n*factorial(n-1)

print(factorial(6))
```



1*1=1

Fibonacci Series

$f(n) = f(n-1) + f(n-2)$

0	1	2	3	4	5	6	.....
0	1	1	2	3	5	8	.....

Base condition :-

$f(0) = 0$

$f(1) = 1$

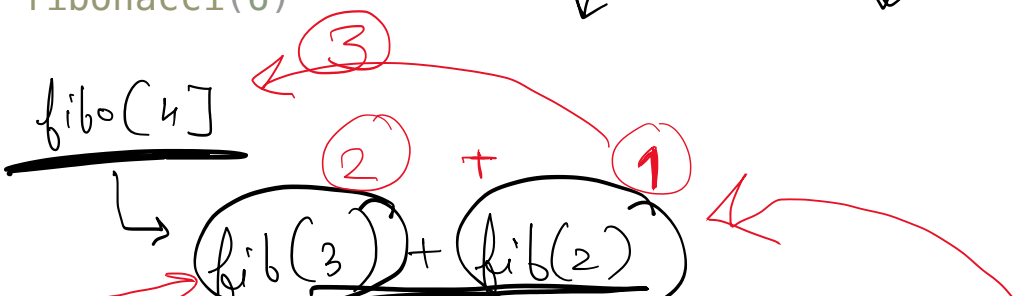
$f(n) = f(n-1) + f(n-2)$

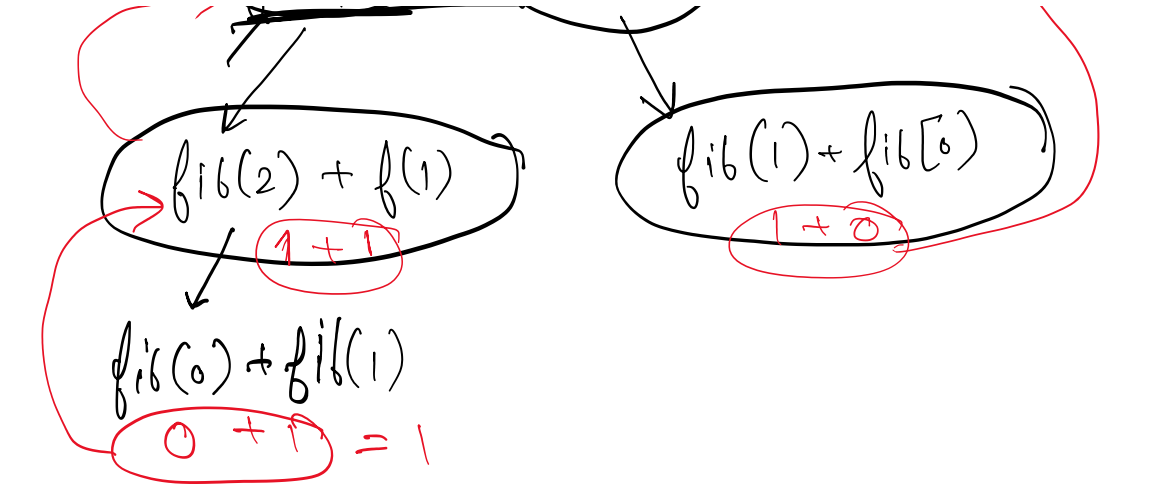
```
def fibonacci(n):   # f(n) = f(n-1) +f(n-2)
    # base condition
    if n<=1:
        return n

    else:
        return fibonacci(n-1) +fibonacci(n-2)

fibonacci(6)
```

Box




fib(2) + f(1)
1 + 1
fib(1) + fib(0)
1 + 0
fib(0) + fib(1)
0 + 1 = 1