Softy EDUCATION

&

Taki Academy
www.takiacademy.com

python™

contact@softyeducation.com

Softy EDUCATION

Day 11

python™

Content :

1. What is Recursion?
2. Recursive Example: Factorial
3. Recursive Example: Fibonacci Sequence
4. Exercises

Softy EDUCATION

01 What is Recursion?

python™

01- What is Recursion?

Recursion is a fundamental concept in computer science and programming. In essence, recursion is when a function calls itself. It's a useful tool for breaking down complex problems into simpler, more manageable tasks.

01- What is Recursion?

At its heart, recursion is the process of defining something in terms of itself. In the realm of programming, a recursive function is one that calls itself to solve a problem.

01- What is Recursion?

Base Case and Recursive Case

Two essential parts of a recursive function are:

1. **Base Case**: This stops the recursion. Without it, the function would call itself indefinitely, leading to a stack overflow.
2. **Recursive Case**: This is where the function calls itself.

Softy EDUCATION

02 Recursive Example: Factorial

python™

02 - Recursive Example: Factorial

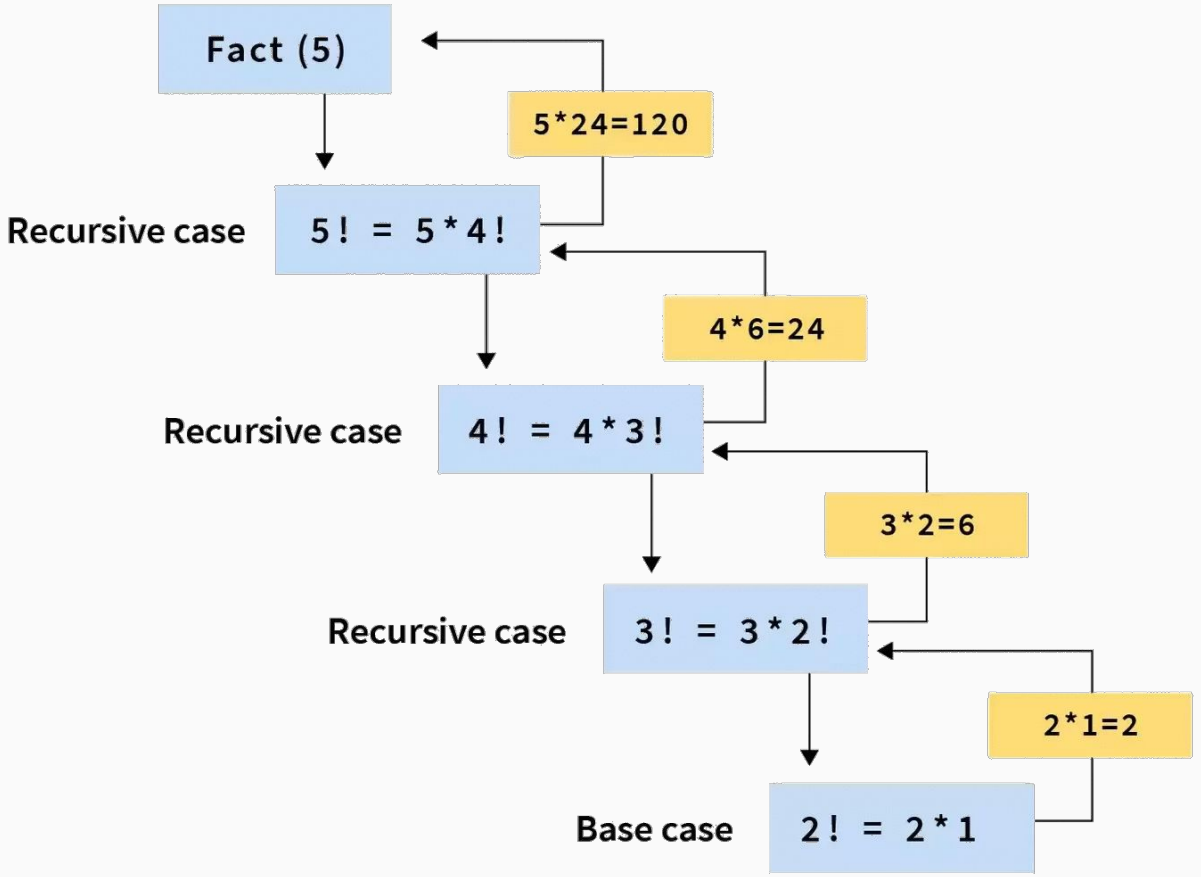
The factorial of a number n (denoted as **n!**) is the product of all positive integers less than or equal to n. For **example:**

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

In terms of recursion:

$n! = n \times (n-1)!$
$0! = 1$ (This is the base case)

02 - Recursive Example: Factorial



02 - Recursive Example: Factorial

```
factorial(5)
= 5 * factorial(4)
= 5 * 4 * factorial(3)
= 5 * 4 * 3 * factorial(2)
= 5 * 4 * 3 * 2 * factorial(1)
= 5 * 4 * 3 * 2 * 1
= 120
```

02 - Recursive Example: Factorial

Let's implement this in Python:

```python
def factorial(n):
    # Base case
    if n == 0:
        return 1
    # Recursive case
    else:
        return n * factorial(n-1)
```

Softy EDUCATION

03 Recursive Example: Fibonacci Sequence

python™

03 - Recursive Example: Fibonacci Sequence

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, typically starting with 0 and 1:

0,1,1,2,3,5,8,13,...

The sequence can be defined recursively:

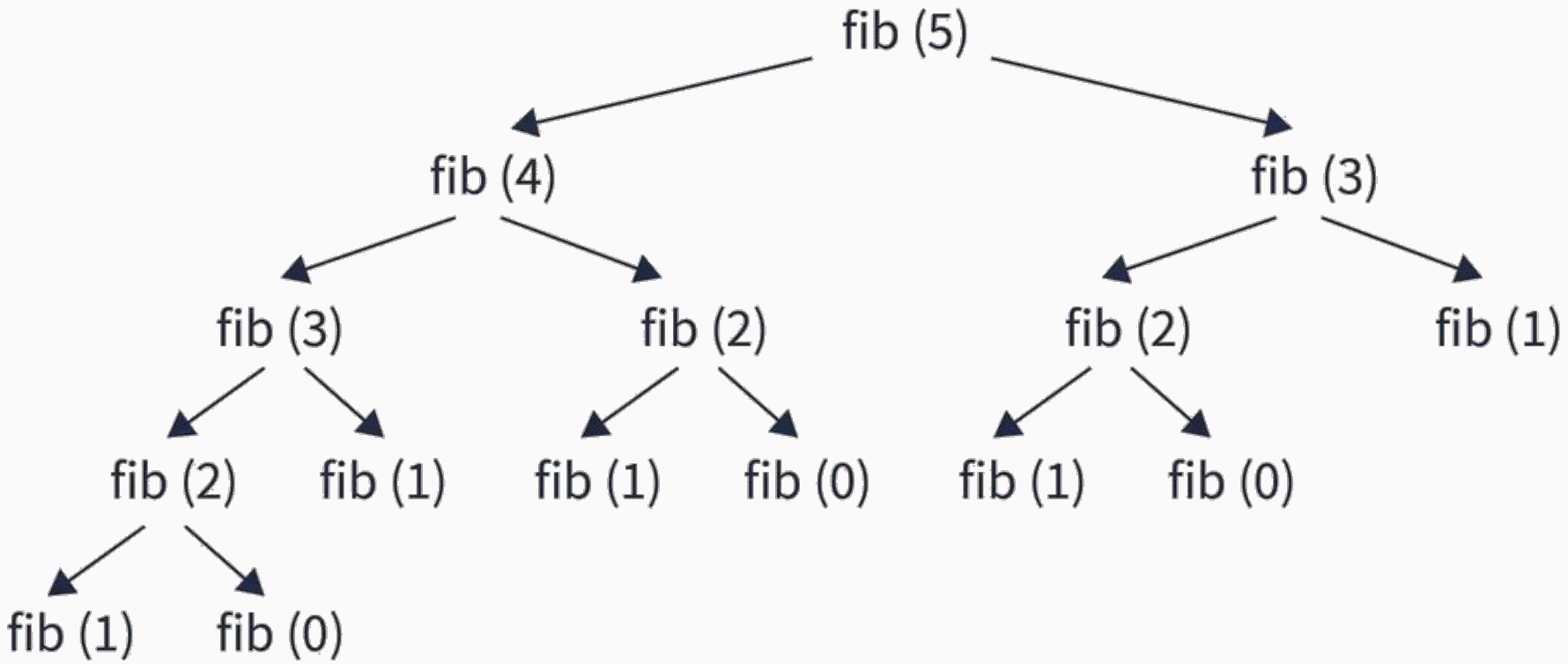
$fib(n)=fib(n-1)+fib(n-2)$
$fib(0)=0,fib(1)=1fib(0)=0,fib(1)=1$ (These are the base cases)

03 - Recursive Example: Fibonacci Sequence

Python implementation:

```python
def fibonacci(n):
    # Base cases
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive case
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

03 - Recursive Example: Fibonacci Sequence



04 Exercises

1. **Sum of Natural Numbers:**

- Write a recursive function to find the sum of natural numbers up to n.
- For example, if the input is 5, the function should return 15 because 1+2+3+4+5=151+2+3+4+5=15.

2. Count Digits:

- Write a recursive function to count the number of digits in a positive integer.
- For instance, if the input is 1054, the function should return 4.

3. Reverse a String:

- Implement a recursive function to reverse a given string.
- E.g., for the input string "python", the output should be "nohtyp".

Softy EDUCATION

Thank you

python™