

Fibonacci Series

The Fibonacci series is a sequence of numbers where each number is the sum of the two preceding ones

It usually starts with 0 and 1.

Formula:

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

with

$$F(0) = 0, \quad F(1) = 1$$

Example:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Two Ways to Find Fibonacci Numbers

1. Recursive Method

- The function calls itself to calculate smaller Fibonacci numbers.

2. Non-Recursive (Iterative) Method

- Uses a loop instead of function calls.

Difference Between Recursive and Non-Recursive Fibonacci

Feature	Recursive	Non-Recursive (Iterative)
Definition	Function calls itself to solve smaller subproblems.	Uses a loop to compute directly.
Time Complexity	$O(2^n)$ — very slow for large n	$O(n)$ — much faster
Space Complexity	$O(n)$ due to function call stack	$O(1)$ — uses only a few variables

Ease of Understanding	Easier to understand conceptually	Easier to implement efficiently
Speed	Slow	Fast
Example Output for n=5	5	5

code

Recursive method

```
def fib_recursive(n):
```

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

```
        return n
```

```
    else:
```

```
        return fib_recursive(n-1) + fib_recursive(n-2)
```

Non-recursive (Iterative) method

```
def fib_iterative(n):
```

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

```
        return n
```

```
    a, b = 0, 1
```

```
    for i in range(2, n+1):
```

```
        c = a + b
```

```
        a, b = b, c
```

```
    return b
```

Test both functions

```
n = int(input("Enter n: "))
```

```
print("Recursive Fibonacci:", fib_recursive(n))
```

```
print("Non-recursive Fibonacci:", fib_iterative(n))
```