

RECURSION

IN JAVA PROGRAMMING



overview

A recursive function is a function that calls itself until a “base case” is true, and execution stops.

Base Case:
It is to make sure that the function will terminate, or unexpected behavior is expected.

code

```
public static int fibonacci(int number) {
    if (number == 1 || number == 2) {
        return 1;
    }
    return fibonacci(number-1) +
           fibonacci(number-2);
}
```

Base case

Function being called again by itself

Quote

To understand recursion,
one must first understand
recursion.

Stephen Hawking

details

When a recursive function is called, a context gets placed on the execution *stack*—a data structure that operates on a "Last In, First Out" basis. An item is "pushed" onto a *stack* to add to it and "popped" out of the *stack* to remove from it.

When the last item on the *stack* finishes execution, that context generates a return value that gets passed down as a return value to the next item, then that execution context is popped off the *stack*.

fun fact

Recursion comes directly from the mathematical concept of induction which proof by **Plato** , 370 BC.



FAKULTAS
ILMU
KOMPUTER

Lecturer :
Dr. Fariz Darari.
TA : LA

Made by:

Audrey Annisa 1806146871
M. Jilham Luthfi 1806141340
Syanne Limarwan 1806191282

References:
<https://www.freepik.com/>
<https://www.geeksforgeeks.org/recursive-functions/>
<https://medium.freecodecamp.org/recursion-is-not-hard-858a48830d83>
<https://www.quora.com/Who-invented-or-has-used-for-the-first-time-recursion-in-a-program-Who-invented-the-main-programming-paradigms-imperative-functional-etc>