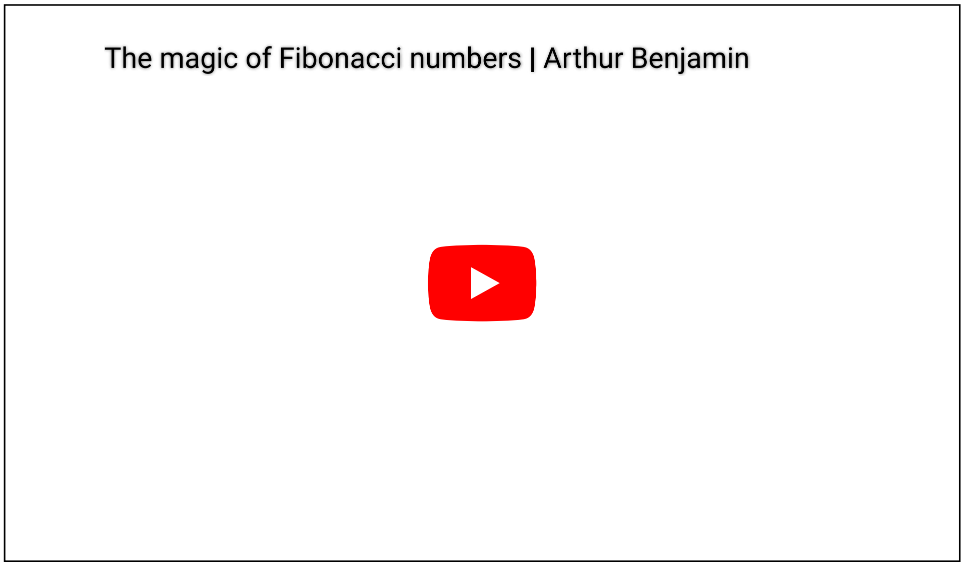
# Fibonacci Numbers

HackerRank

**Objective**
In this challenge, we learn about using the Fibonacci Function.

**Resources**
Here's a helpful video on the topic:




*The Fibonacci Series*

The Fibonacci sequence begins with $0$ and $1$. These are the first and second terms, respectively. After this, every element is the sum of the preceding elements:

```
Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)
```

**Task**
Given the starter code, complete the Fibonacci function to return the $N^{th}$ term.

We start counting from *Fibonacci*$(1) = 0$. This might differ from some other notations that treats *Fibonacci*$(0) = 0$.

The overall equation is:

```
             = 0 , n = 1
Fibonacci(n) = 1 , n = 2
               Fibonacci(n-1) + Fibonacci(n-2)  , n > 2
```

**Input Format**

One line of input, the integer $N$.

**Constraints**

$0 < N <= 40$

#### Output Format

Output one integer, the $N^{th}$ Fibonacci number.

#### Sample Input

```
3
```

#### Sample Output

```
1
```

#### Function Prototype

The starter code is provided for Scala. The code for accepting the input and displaying the output is provided. You will be provided the input parameter $N$, and you need to return the $N^{th}$ Fibonacci term.

#### Sample Input and Output Values for the Fibonacci Series

```
fibonacci(3) = (0+1) = 1
fibonacci(4) = (1+1) = 2
fibonacci(5) = (1+2) = 3
```

#### Requirements

Simple test cases can be cleared with a purely recursive function exponentially. To clear the more challenging test cases without violating the principles of functional programming, you might benefit from learning about the accumulator technique.