Solution Review 3: Find nth Fibonacci Number

This lesson gives a detailed review of the challenge in the previous lesson.

We'll cover the following

Solution:

```
fn fibonacci(term: i32) -> i32 {
    match term {
        0 =>  0,
        1 =>  1,
        _ => fibonacci(term-1) + fibonacci(term-2),
    }
}
fn main(){
    println!("fibonacci(4)={}",fibonacci(4));
}
```

Explanation

A recursive function, `fibonacci`, takes a **parameter** `term` **of type i32** and **returns** an integer of type `i32`, i.e., the nth term of the Fibonacci number.

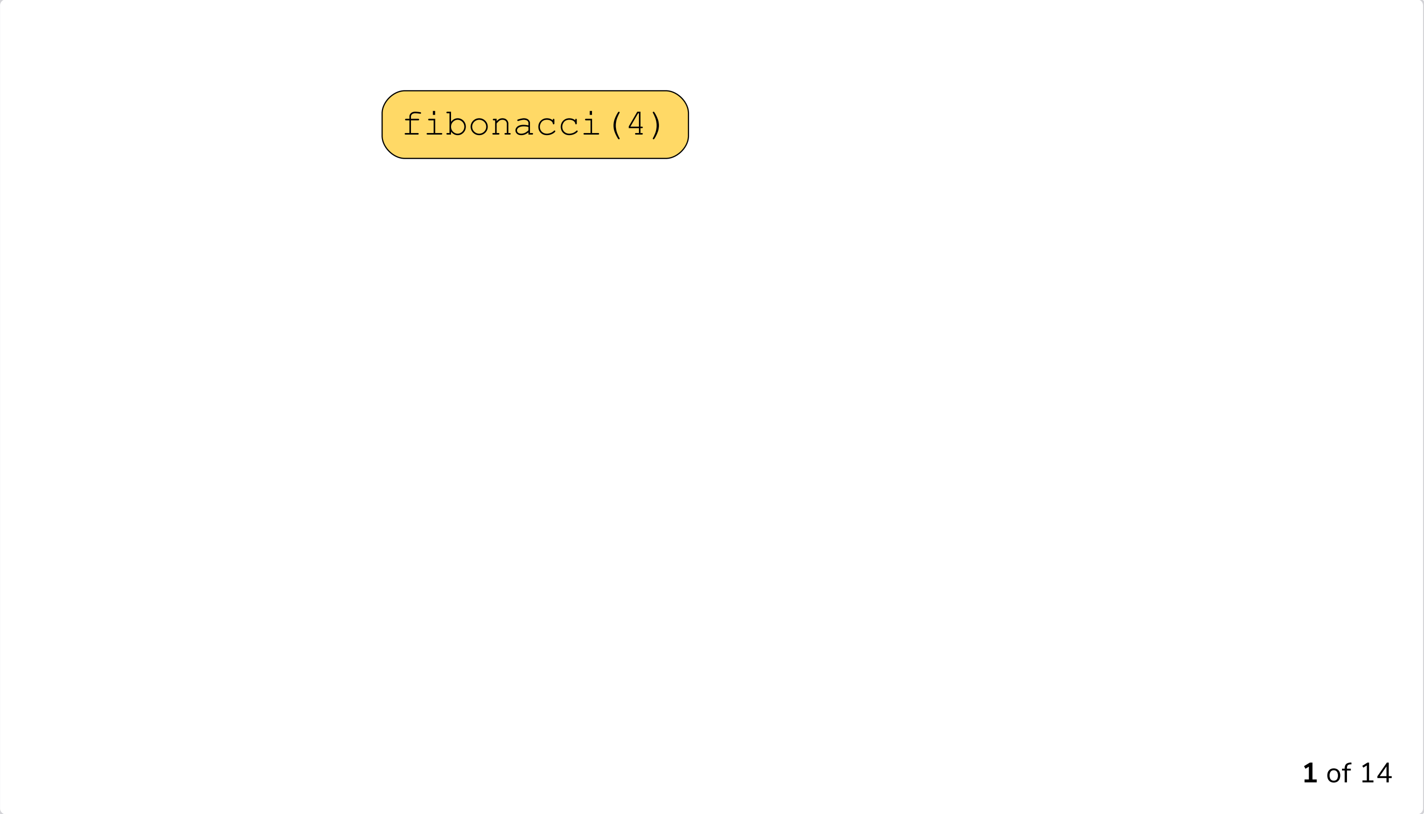
The recursive function has two parts: the base case, and the recursive case.

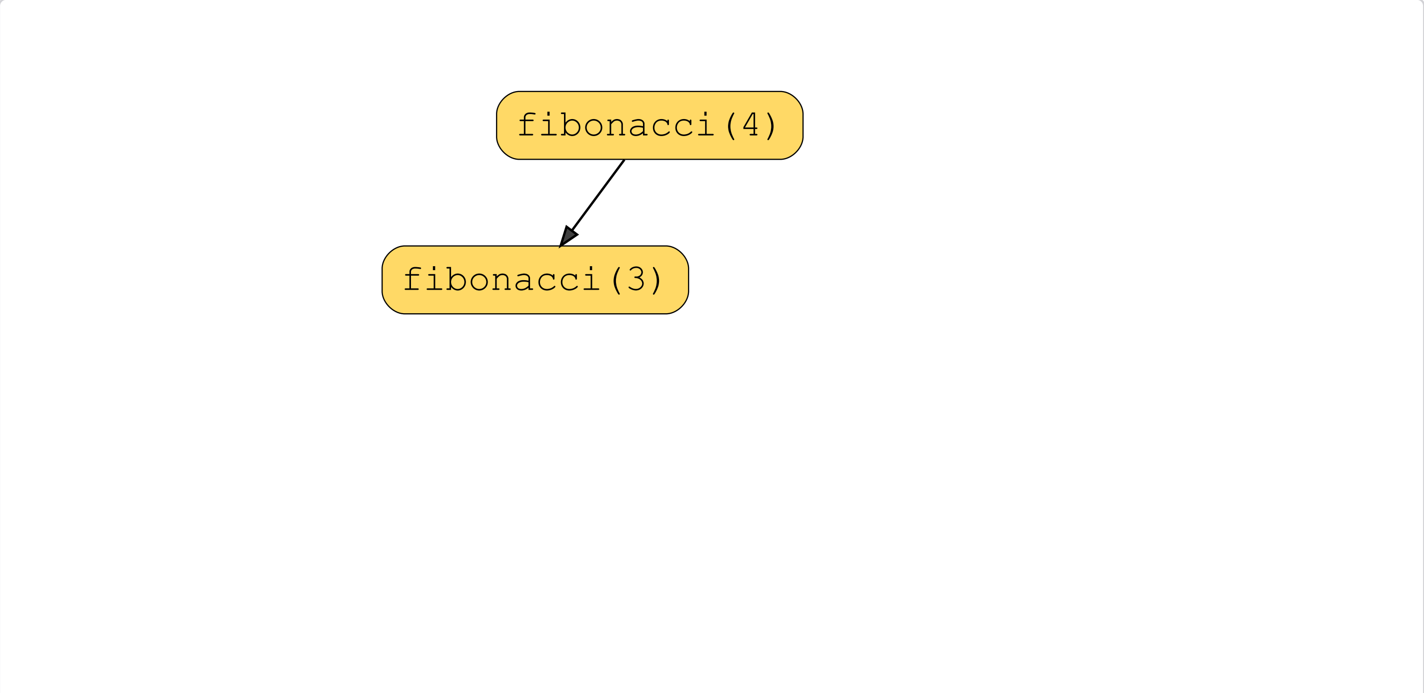
- **base case**
  - `match` takes the `term` and if it matches with 0 it returns 0 and if it matches with 1 it returns 1
- **recursive case**
  - It decrements the value of the `term` by 1. A recursive call is made with argument `term - 1` or `term - 2` and that the function execution can't proceed until the recursive calls return. Now the answer before the `+` operator is calculated.

- It then decrements the value of the `term` by 2. A recursive call is made with argument `term - 1` or `term - 2` and that the function execution can't proceed until the recursive calls return. Now the answer after the `+` operator is calculated.
- It simply adds the two values and returns the result.

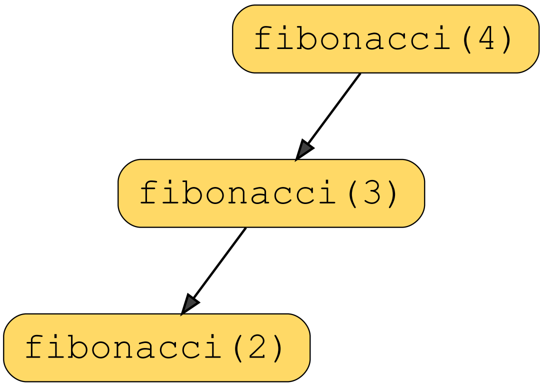
The following illustration explains the code through a recursion tree:

> **Note:** This is a binary recursion since the function makes two recursive calls to itself when invoked.

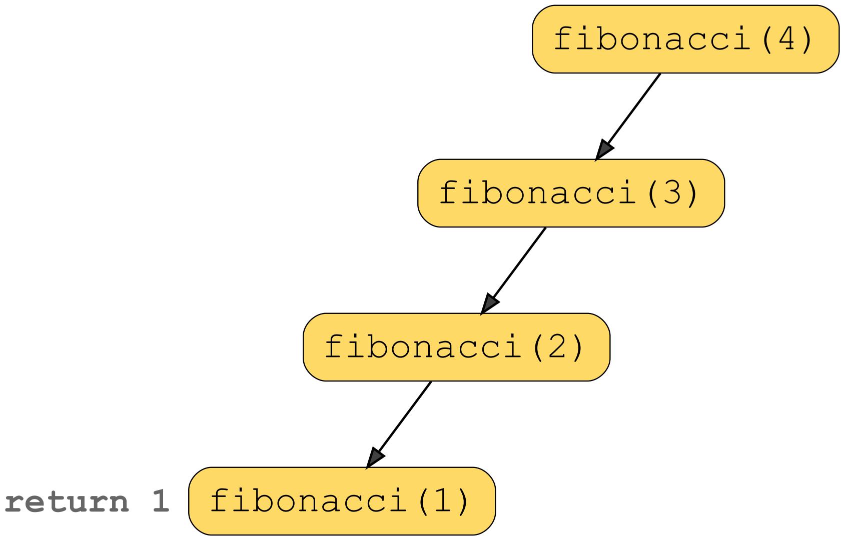




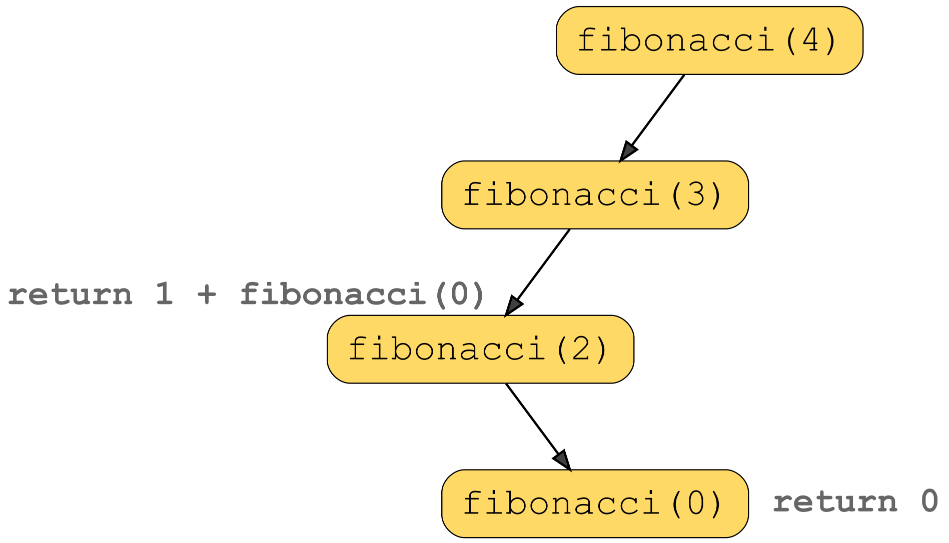
2 of 14



3 of 14



4 of 14



fibonacci(4)

fibonacci(3)

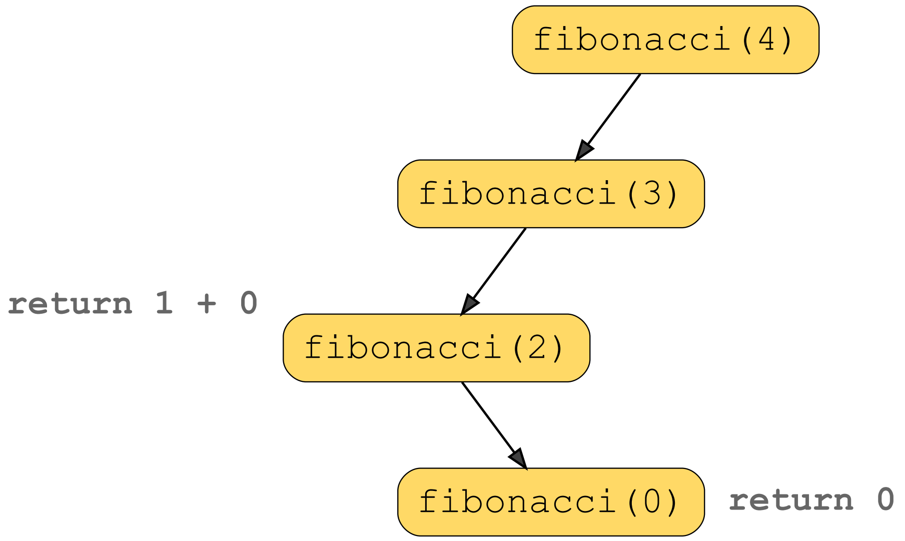
return 1 + fibonacci(0)

fibonacci(2)

fibonacci(0)

return 0

5 of 14



fibonacci(4)

fibonacci(3)

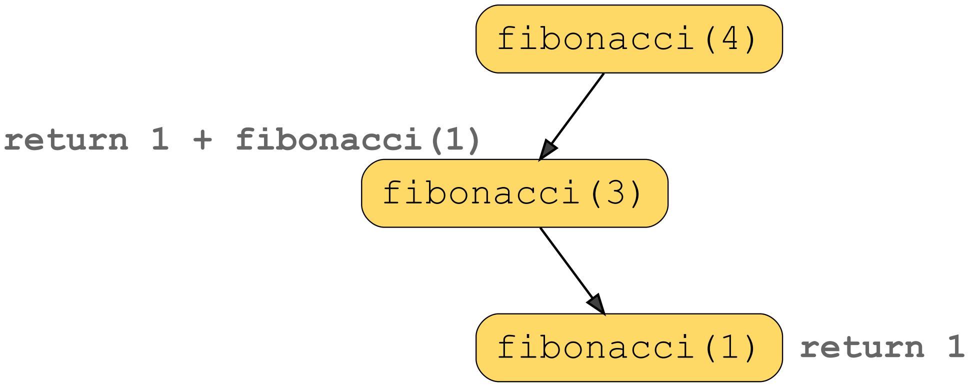
return 1 + 0

fibonacci(2)

fibonacci(0)

return 0

6 of 14



fibonacci(4)

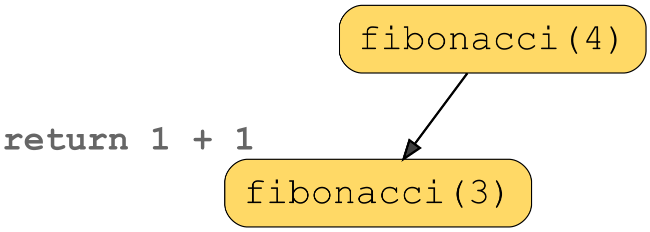
return 1 + fibonacci(1)

fibonacci(3)

fibonacci(1)

return 1

7 of 14

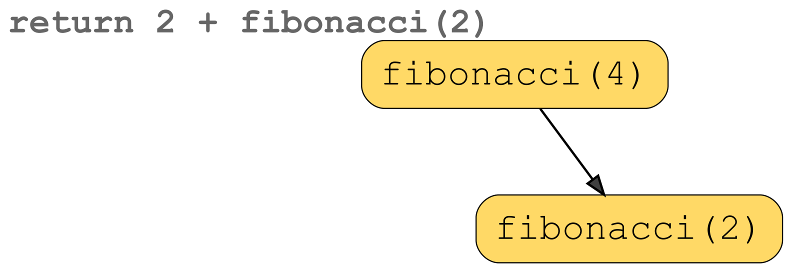


fibonacci(4)

return 1 + 1

fibonacci(3)

8 of 14

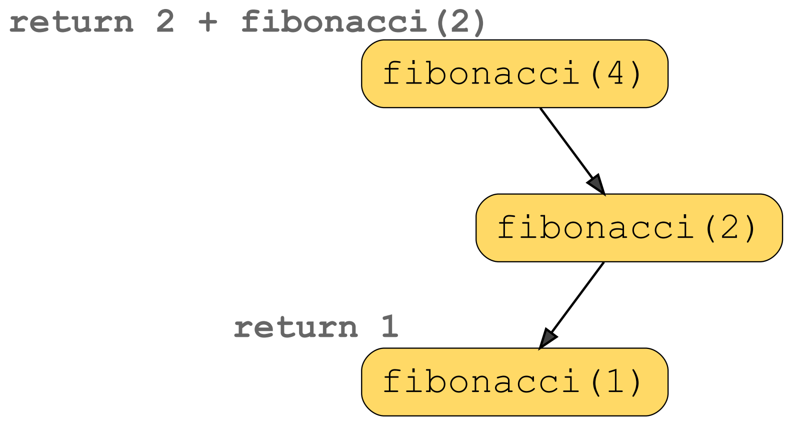


return 2 + fibonacci(2)

fibonacci(4)

fibonacci(2)

9 of 14



return 2 + fibonacci(2)

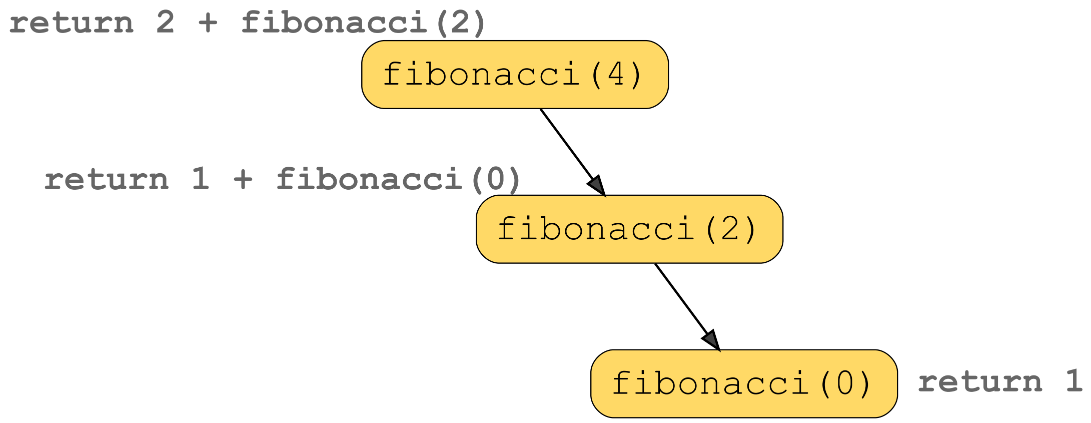
fibonacci(4)

fibonacci(2)

return 1

fibonacci(1)

10 of 14



return 2 + fibonacci(2)

fibonacci(4)

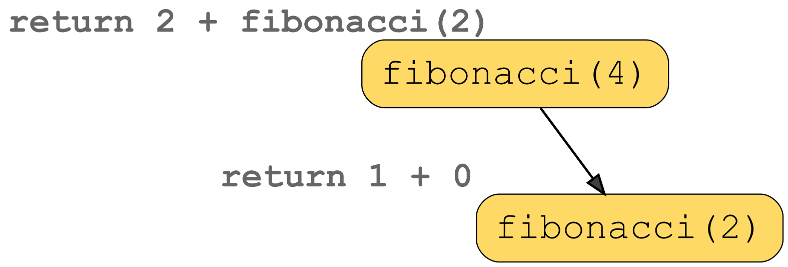
return 1 + fibonacci(0)

fibonacci(2)

fibonacci(0)

return 1

11 of 14



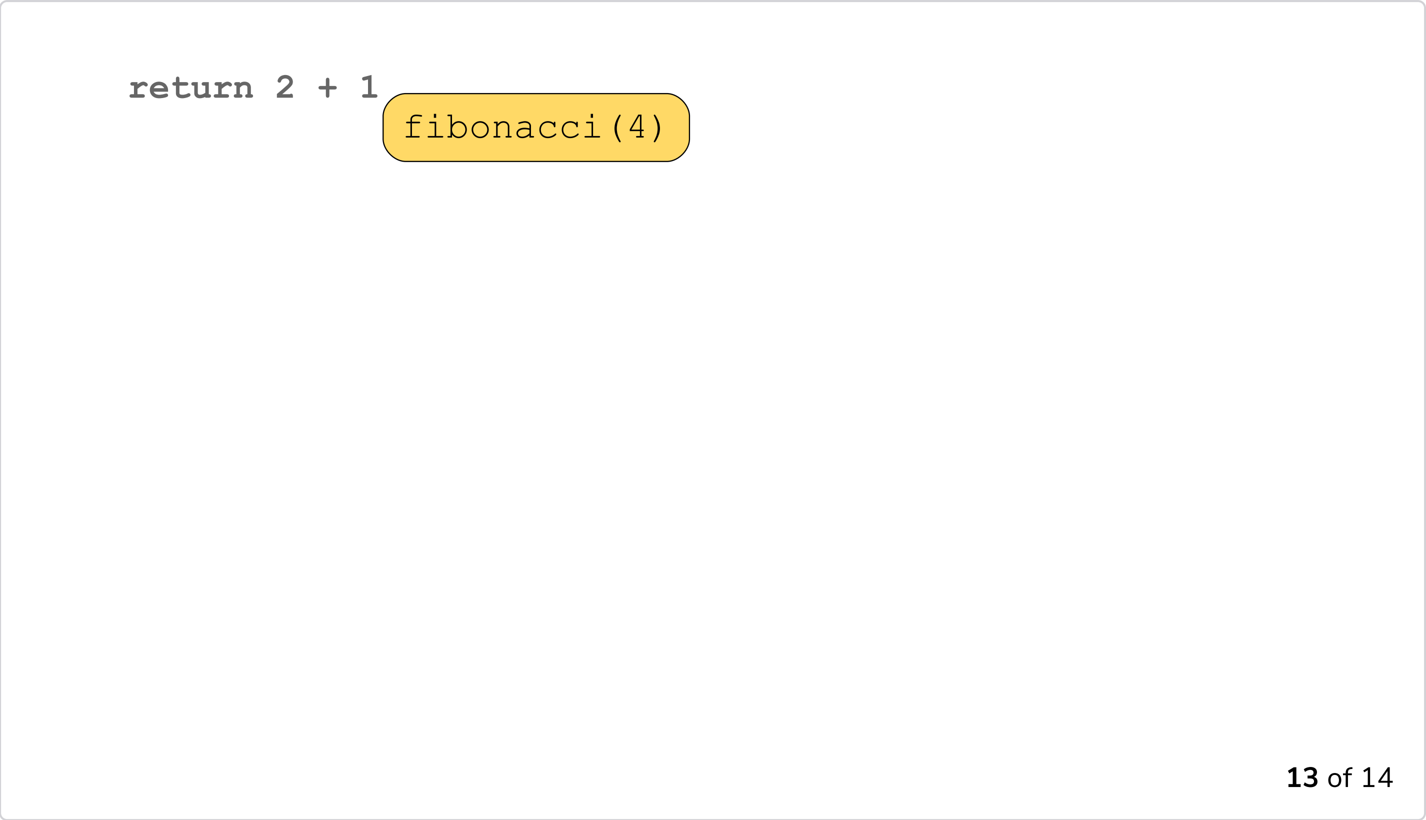
return 2 + fibonacci(2)

fibonacci(4)

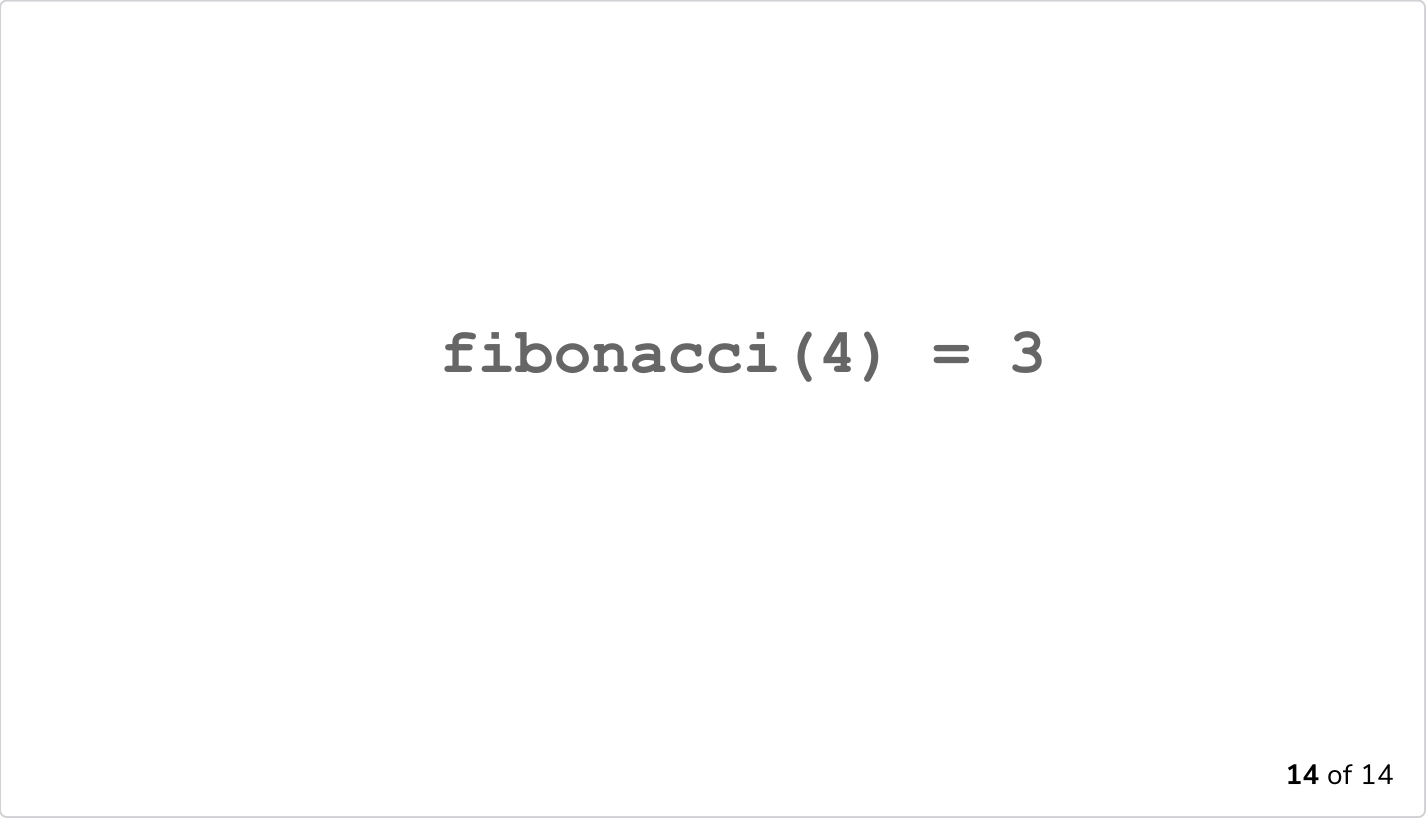
return 1 + 0

fibonacci(2)

12 of 14



```
return 2 + 1
```



```
fibonacci(4) = 3
```

Now that you have learned about Functions let’s learn about a non-primitive data type, “Strings”.