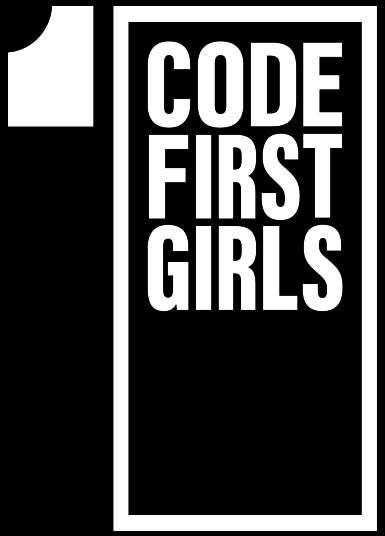
# RECURSION

## LESSON 11




NANODEGREE → ENGINEERING MODULE

# AGENDA

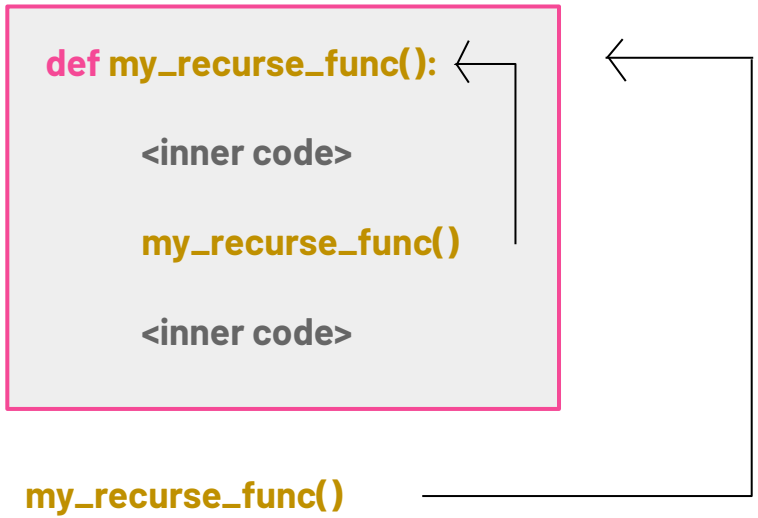
**01** Recursion

**02** Famous recursion functions

**03** Practice and coding

# RECURSION

## RECURSIVE FUNCTIONS




- **Recursion**– is a method of programming or coding a problem, in which a function calls itself one or more times in its body. Usually, it is returning the return value of this function call.

# RECURSION ADVANTAGES

1. Recursive functions make the code look clean and elegant.
2. A complex task can be broken down into simpler sub-problems using recursion.
3. Sequence generation is easier with recursion than using some nested iteration.

# RECURSION DISADVANTAGES

1. Sometimes the logic behind recursion is hard to follow through.
2. Recursive calls are expensive (inefficient) as they take up a lot of memory and time.
3. Recursive functions are hard to debug.

# FAMOUS RECURSION

## CODING EXAMPLES

| n | n! | | |
|---|---|---|---|
| 1 | **1** | 1 | 1 |
| 2 | 2 × **1** | = 2 × **1!** | = 2 |
| 3 | 3 × **2 × 1** | = 3 × **2!** | = 6 |
| 4 | 4 × **3 × 2 × 1** | = 4 × **3!** | = 24 |
| 5 | 5 × **4 × 3 × 2 × 1** | = 5 × **4!** | = 120 |
| 6 | etc | etc | |

## FACTORIAL

- The **factorial function** (symbol: **!**) says to **multiply all whole numbers** from our chosen number down to 1.

$$n! = n \times (n-1)!$$

# FAMOUS RECURSION

## CODING EXAMPLES

### The Fibonacci Sequence

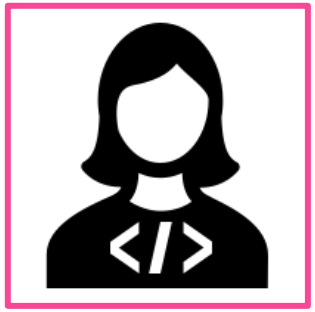
1,1,2,3,5,8,13,21,34,55,89,144,233,377…

| | |
|---|---|
| 1+1=2 | 13+21=34 |
| 1+2=3 | 21+34=55 |
| 2+3=5 | 34+55=89 |
| 3+5=8 | 55+89=144 |
| 5+8=13 | 89+144=233 |
| 8+13=21 | 144+233=377 |

## FIBONACCI

- **Fibonacci sequence** is one of the most famous formulas in mathematics. Each number in the sequence is the sum of the two numbers that precede it.

$$F_n = F_{n-1} + F_{n-2}$$



**DEMO & EXERCISES**

# RECURSION IMPLEMENTATION EXERCISES & PRACTICE

# THANK YOU!