**Recursion**

Recursion is a function that calls itself unit it doesn’t.

We must have a return statement once we break out of the recursion function.

In recursion, we need to have a base case that will stop the function from running.

If there is no base case, the function runs infinitely.The function openGiftBox() runs until we reach our base case.

Here the base case = isBall.

function openGiftBox() {

    if (isBall) return ball;    // This is our BASE CASE

    openGiftBox(); // This is our RECURSIVE CASE

}

This function will run forever because 1 will never be less than 2.

function openGiftBox() {

    if (1 > 2) return 1;

    openGiftBox();

}

This function will run forever because there is no return statement therefore we go to the next line of code which is the recursion function.

function openGiftBox() {

    if (isBall) console.log('Hello');

    openGiftBox();

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Stack Overflow**

Occurs when a recursive function runs infinitely.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Call Stack**

We run funcOne().

funcOne() runs funcTwo() before running it’s console.log(‘One’).

funcTwo() runs funcThree() before running it’s console.log(‘Two’).

funcThree() console.log(‘Three’), then funcTwo() console.log(‘Two’), then funcOne() console.log(‘One’).

The output would be:

Three

Two

One

function funcThree() {

    console.log('Three')

}

function funcTwo() {

    funcThree()

    console.log('Two')

}

function funcOne() {

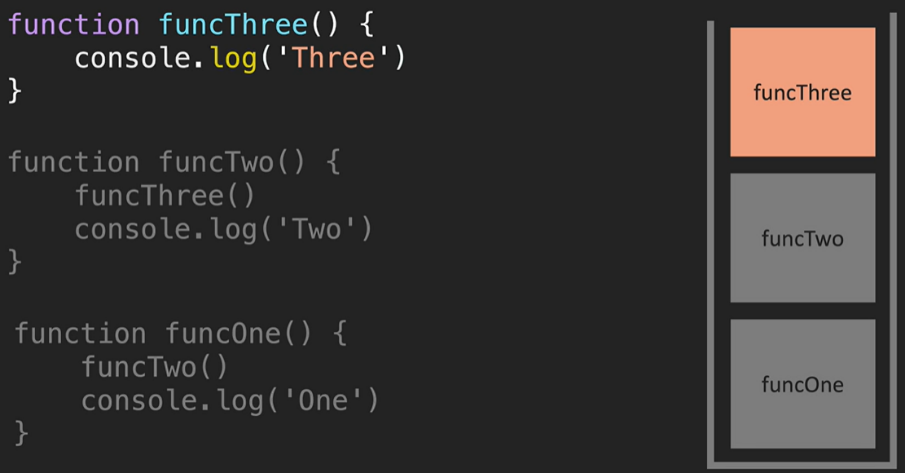
    funcTwo()

    console.log('One')

}

funcOne()

This is the call stack for these 3 functions.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Factorial**

4! – this means 4 factorial 🡪 4 \* 3 \* 2 \* 1

3! 🡪 3 \* 2 \* 1

2! 🡪 2 \* 1

1! 🡪 1

4! = 4 \* 3!

3! = 3 \* 2!

2! = 2 \* 1!

1! = 1 --- 1 is our base case.

function factorial(n) {

    if (n === 1) return 1

    return n \* factorial(n - 1)

}

factorial(4)

