**Recursion**

A function that refers to itself inside of the function.

function foo() {

    foo();

}

foo();

Recursion is good for tasks that are repetitive.

**Recursive functions have 2 paths:**

1. The recursive function
2. The base case

**Recursion Rules:**

1. Identify the base case
2. Identify the recursive case
3. Get closer and closer and return when needed. Usually, you have 2 returns meaning we have to return the base case and the recursive case.

Diagram

Description automatically generated

**Base Case**

Condition to meet to exit out of recursive function.

let counter = 0;

function foo() {

    if (counter > 3) { // this is our base case

        return 'done';

    }

    counter++

    foo();

}

foo();

This function will return undefined.

let counter = 0;

function foo() {

    if (counter > 3) { // this is our base case

        return 'done';

    }

    counter++

    return foo(); // this will return ‘done’

}

foo();

This function will return ‘done’ because of the return foo();