**CALL BACK HELL**

getData(function(x){

console.log(x);

getMoreData(x, function(y){

console.log(y);

getSomeMoreData(y, function(z){

console.log(z);

});

});

});

Ref: <https://blog.bitsrc.io/understanding-promises-in-javascript-c5248de9ff8f>

micro task :-

--------------

1. take a long time.

2. Event loop gives higher priority to MicroTask Queue.

2. Eg Promises, Process.nextTick, I/O operation

Macro Task:

-----------------

1. setTimeout, setInterval,

For example the code shown below is

console.log("Start");

setTimeout(function() {

console.log("Timeout");

}, 0);

Promise.resolve().then(function() {

console.log("Promise"); // microTask!

});

console.log("End");

The out put is start --> End --> Promise --> Timeout

**queueMicrotask :-**

It convert the synchronous task into asynchronous

**What is the difference between promise and async await?**

The promise involves chaining . then and . catch methods, whereas Async Await uses a try-catch block that looks more like synchronous code.