**Callback Hell**

**--Saksham Khandelwal**

Callback hell refers to the situation where **callbacks** are nested within other **callbacks** several levels deep, potentially making it difficult to understand and maintain the code.

To explain callback hell, let’s try to implement a simple and trivial scenario of a beef cooking process.

To cook beef, there are multiple steps taken one after another, namely:

1. Get beef
2. Slice beef
3. Cook beef
4. Serve beef

let getBeef = (user, nextStep) => {

    console.log(`Retrieving beef from ${user}`);

    nextStep("raw beef");

};

let sliceBeef = (rawBeef, nextStep) => {

    console.log(`Slicing ${rawBeef}`);

    nextStep("sliced beef");

};

let cookBeef = (slicedBeef, nextStep) => {

    console.log(`Cooking ${slicedBeef}`);

    nextStep("cooked beef");

};

let serveBeef = (cookedBeef, nextStep) => {

    console.log(`Putting ${cookedBeef} on the plate`);

    nextStep("Beef on plate");

};

The 4 functions above are very simple functions with the second parameter nextStep being a callback function taking in a string representing the result of the step.

E.g. sliceBeef() would lead to nextStep(“sliced beef”), “sliced beef” represents the result of the sliceBeef() step.

With the 4 functions above, let’s chain them up so that one function is executed after another. Let’s do it one by one so that we can visualise how the callback hell is being formed.

Step 1: Get beef from user Saksham

getBeef("Saksham", (rawBeef) => {

    //Do something with raw beef here

});

Step 2: Slice the raw beef

getBeef("Saksham", (rawBeef) => {

    sliceBeef(rawBeef, (slicedBeef) => {

        //Do something with raw beef here

    });

});

Step 3: Cook the sliced beef

getBeef("Saksham", (rawBeef) => {

    sliceBeef(rawBeef, (slicedBeef) => {

        cookBeef(slicedBeef, (cookedBeef) => {

            //Do something with raw beef here

        });

    });

});

Step 4: Serve the cooked beef

getBeef("Saksham", (rawBeef) => {

    sliceBeef(rawBeef, (slicedBeef) => {

        cookBeef(slicedBeef, (cookedBeef) => {

            serveBeef(cookedBeef, (servedBeef) => {

                console.log(servedBeef);

            });

        });

    });

});

We have finished the beef-cooking process and successfully created a callback hell. The callback of serveBeef() is nested within the callback of cookBeef() which is nested within the callback of sliceBeef() which is nested within the callback of getBeef().