**S5\_\_Callback Hoisting Parameter in Function**

* **Closure**:- function that references variable in the outer scope from its inner scope.
* **Hoisting :-** Hoisting is the default behaviour of moving all the declarations at the top of the scope before code execution. no matter where functions and variables are declared, they are moved to the top of their scope regardless of whether their scope is global or local. it allows us to call functions before even writing them in our code. Not possible for function expression.
* **Callback**:- it’s a function passed as argument to another function. It can run after another function has finished. This will allow to call another function. U have to wait for previous function to get complete.
* **Parameters in Function : -** parameter are the names which is included in the function parenthesis block. And at the time of calling function we have to pass the argument. While inserting parameter no need to give data type. its not perform type checking too while passing argument. Dosnt check number of argument recevd.
  + **Default Parameter :-** If a function is called with **missing arguments** (less than declared), the missing values are set to undefined. We can define default parameter too.
  + **Rest parameter:-** way to handle function parameter, allowing us to more easily handle various input as parameters in a function (…) used to assign value in an array. With the help of a rest parameter a function can be called with any number of arguments, no matter how it was defined. It’s the feature of ES6.
  + **Param Destructuring:-** destructuring keys from object. Its used with the object to fetch value as key and make it variable.

**Program**

//Callback

let names = ["Keshav", "Kanchana", "Sahiba", "Imran", "Aniket", "Pritam", "Soham", "Ritu Raj"];

const displayName = (names,dispalyWorldCity  /\*Parameter\*/) => {

    console.log('Printing the Names');

    for (var i = 0; i < names.length; i++) {

        process.stdout.write(names[i] + "  ");

    }

    dispalyWorldCity();  //diffrent argument

}

// displayName(names, displayCities);  Not Possible Hoisting

let city = ["Goa", "Mumbai", "Dubai", "Uk", "Briglab"];

const displayCities = () => {

    console.log("\nPrinting Cities: ");

    for (var i = 0; i < city.length; i++) {

        process.stdout.write(city[i] + "  ");

    }

}

displayName(names,displayCities /\*Argument\*/);  //Callback

const functionReturns = functionReturnFunction(); //Hoisting

console.log(functionReturns);

// console.log(functionReturnFunction());

// //Function Return Function Closure

function functionReturnFunction() {

    function addition() {

        var a = 100, b = 20, c = 40;

        var d = a + b + c;

        return d;

    }

    return addition();

}

// //Parameters

function multiplication(no1, no2 = 6, ...no6 /\*Defult Value no2\*/) {

    no4 = no1 \* no2;

    console.log();

    console.log("value of No1 "+ no1 + "\n value of Number 2 "+no2);

    console.log(`Rest Para Value: ${no6}`)

    return no4;

}

// QQ]] How to calculate rest parameter?

const multiplications = multiplication(2, 22, 33, 44, 55, 66);

console.log(multiplications);

// //Param Destructurig

const personNames =

    { firstName: "Imran", gender: "M", age: "100", }

function printDetails({firstName,gender,age}){

    console.log(firstName);

    console.log(gender);

    console.log(age);

}

printDetails(personNames);