FUNCTIONS :

/\*

Functions...

there are different ways of writing functions one is simple given below.

function dance() {

  console.log("Dancing...");

}

second one:

let funny = function(){}

tu iss mein hm nay variable banaya orr uss ko function bana diyaa

orr uss function ko name nai diyaa qk jo variable ka name haii wohiii function ka name hoga.

tu ab agar hm iss ko call karna chahtay hain tu uss variable k name say uss ko call karengay

like

funny(); tu ye call ho jaye ga.

Function Declarations:

function dancy(){}  agar hm iss tarah function banayeingay tu iss ko (function statement) kahengay

let abb = function(){} agar mein function ko kisi variable mein save kar lun tu uss ko

                        hm (function expression) kahtay hain...

Fat Arrow Function:  let name = ()=>{}  (this is called fat arrow function)

Parameters:

kahin b jab hm functions banatay hain or agar un mein bracket mein hm

agar kuch valued dey dey tu un ko parameters kahtay hain jaisay k hm

nay addition walay function mein likhay hain... like

function add(v1,v2){} here v1 and v2 are parameters.

when we call it like

add(1,3);  here 1 and 3 are arguments.

Default,Rest and Spread Parameter:

function add(a1,a2){

    console.log(a1 + a2);

}

add();

tu hm nay ooper function mein parameter maangay hain magar jab hm nay function

ko call kiyaa hai tu yahan prr hm hm nay koii argument nai diyaaa...

tu ab in mein jab hm nay koii value nahi bhejiiii tu ye ab khud say undefined

le lega... tu jab ye undefined le lega.. tu for eg hm nay aik function banaya haii

jiss mein hm 2 numbers ko add kar rahay hain, magar hum nay arguments mein koii

value dii hiii nai tu ye undefined le lega tu jab ye values ko add karay ga tu ye

undefined iss mein hogii tu ye humein aik error dega NaN jiss ka matlab hai k not a number

matlab k aap nay 2 values dii hain jo k undefined hain matlab k NaN hain jo add ho hi nai

saktiii...

tu ab iss say bachnay k liyee hm default parameter ka use krtay hain k matlab

hm pahlay say hii parameter ko value dey detay hain jaisay k

function add(v1=0, v2=0){}

tu matlab k agar v1 mein hm nay koii value nai dii tu ye by default 0 value lelega

undefined nahi lega qk hmnay bydefault 0 value dii hai...

ye value hm kuch b rakh sktay hain jaisay hum chahein...

so this is called Default Parameters.

Rest Parameter:

For eg ham nay aik function banaya jiss mein hm nay bohat saray arguments banaye hon tu uss k liyee humein utnay hii parameters b bananay parengay. For eg hm nay aik function banaya orr uss mein hm nay 10 arguments pass kiyee tu ab uss k liye tu chahiyee k utnay yani k 10 parameters b mojood hon uss function mein tu ab ye thoda mushkill ho jata hai agar hm aisa karein… tu iss say bachnay k liyee hm ye kar sktay hain k paremeter mei naik Array ko pass kar dein jiss mein hm jitney b values hon yani arguments tu hm un mein pass kar sktay hain… so this is called rest parameter

Eg :

function abcd(...val) {

  console.log(val);

}

abcd(1, 2, 3, 4, 5, 6, 7, 8, 9);

hm nay … lagayein orr parameter ka name val rakh diya.

Orr phirr jab hm function ko call kar rahay thay tu hm apnii marzi k mutabik jitney chahein utnay arguments pass kar sktay hain

So this is what called a rest operator...

Diff bw Rest And Spread Parameter:

Agar ye jo 3 dots hain (…) agar ye function k parameter space mein lagein matlab k parameter mein use hon tu tub hm iss ko rest operator kahengay orr agr wo arrays and objects mein lagein tu wo spread operators hain…

Hm iss tarah b kar sktay hain k adhay values ko kuch variable mein safe kar lein orr aadhay ko … yani k rest operator ka use karein jaisy k for eg.

function onee(a, b, c, ...val) {

  console.log(a, b, c, val);

}

onee(1, 2, 3, 758, 8, 8239, 27);

tu ab ye jo 1 hain ye a mein chala jayega, 2 b mein chala jayega, 3 c mein chala jayega orr jo baki jitni b hm values put karein wo rest operator mein chali jayengiii yani ...val mein

Return Values And Early Return :

Return ka matlab k function koi value wapis bhej raha hai..

Jaisay k for eg hm nay koi function banaya

Function abc (){

Return 10;

}

Tu matlab k hm jab k iss function ko call karengay tu ye hymein 10 dega.

Tu ab agar mein iss ko call karun

Abc(); tu ye line humein 10 degi and agar hm iss ko agar chahein tu iss ko kisi variable mein safe b kr sktay hain like

Let abbbb = abc();

Console.log(abbbb);

So this is called Return statement.

First Class Functions :

Java script mein first class functions exist kartay hain jiss ka matlab hai k wo functions jin ko value ki tarah treat kiyaa jaasakay.

Matlab k hm un ko kisi variable mein safe kar sakein, ya kisi function mein pass kar sakein tu in ko kahtay hain first class functions.

Matlab k hm un ko bilkul hii koi values ki tarah treat kiyaa kar sktay hain.

Tu matlab k hm javascript mein koi variable jab hm declare kartay hain like eg

Let vall = 12;

Tu ab ye vall name hai orr iss ki value 12 hai tu javascript mein hmm jiss tarah simple hm nay variable declare kiyaa orr value dii use tarah uss mein hm koii b function b uss mein de sktay hain jaisay k for eg.

Let checking = function() {}

Tu ab ye checking aik variable haiii orr wo jo value except kar raha hai wo function hai…

Tu bsss jahan prrr koi value hm de sktay hainnn kisi variable ko use jagah agar hm chahhein tu function b de sktay hain…..

Qk javascript mein function b value ki tarah ki treat hota hai

Hm iss tarah b kr sktay hain k jab hm koii function banatay hainn orr uss mein koi valued detay hain uss k parameter mein accept karwatay hain. Tu ham ye b kr sktay hain k for eg hm nay aik function banaya add name ka orr uss mein val1,val2 do variables diye jo k 2 number lengay. tu ab jab hm uss function ko call karengay tu uss k arguments mein jo b 2 values ko add karna chahtay hain un ko put karengay. Tu ab hm nay jaisay 2 values put kii use tarah agar hm chahtay hain k koi function ko arguments mei dein tu hm wo b kr sktay hain jaisay k given below

Function abby(val){ }

Abby(function(){});

Tu ab ham nay function ko call kiyaa tu uss ki jo val ki value haii hm nay uss ki jagah aik orr function put kar diyaa haiii tu hm ye b krrr sktay hainnn java script mein…. Qk javascript mein function ko b aik value ki tarah treat kiyaa jata hain..........

Function abcd(val){

val();

}

Abcd(function(){

Console.log(“print something”);

});

Tu isss ki story ye hai hm nay abcd function ko banaya uss mein jo b val mein aye uss ko call kiyaa. Tu jab hm nay abcd ko call kiyaa uee mein parameter mein hm nay aik function ko call kiyaa jiss mein hm console.log print karwaya tu wo val mein chalaaa gaya orr wo val hm nay chala diyaa function mein tu matlab k hm nay function chala diyaa

Tu first class function jo hain ye aik concept hain jo k ye concept hai k function ko hm value ki tarah treat kr sktay hain....

Higher OrderFunctions:

Ye b aik concept haiii jo k kahta hai k wo function jo k return karay aik orr function ya accept karay aik another function apnay parameter mein.

Dono mein say agar koi aik b ho jaye tu ye higher order function kahlaye ga.

Tu agar hm example lein tu

Function abcddd(val){

}

Abcddd(Function(){

})

Tu ye ab aik higher order function hai.qk hm nay neechay iss function ko call kiyaa orr uss k parameter jo hai wo aik function accept kar raha hai val ki value mein tu hm iss ko kahengay aik higher order function.

Function ttyyyy(){

Return check(){};

}

Ttyyy()();

Ye b aik higher order function kahlaye ga qk iss nay koi value tu accept naii kii magar iss nay aik or function ko return kiyaaaa hai

Tu jab hm iss ko call karengay tu 2 dafa () () lagayengay

Aik bracket () ye jo lagaya hai ye pahlay function chalanay k liyee haii jo k hofy function hai or ye jo doosra lagaya hai bracket () ye uss k liyee lagaya haii jo k return horaha haiii function.

Pure vs Impure Functions :

Aisa function jo k bahir ki value ko na badlay so that is called pure function

And aisa function jo bahir ki value ko change karday yani k badal dy tu un ko hm impure functions kahengay....

Jaisay for eg

Let aaa = 10;

Function chkk(){

Console.log(“chhhhh”);

}

Tu ye jo function hai ye siraf log ko print kar raha hai ye jo ooper ka variable hai aaa iss ki value ko koi change naii kar raha tu iss ko hm kahengay pure function. Siraf ye nai koi b matlab bahir ki koi value ko change nai kr rha

//

// this is impure function because if we see when this function run it will

// increment the value of bb to 11 from 10 so it change the value outside from the

// function to this is impure function.

let bb = 10;

function hofy3(){

    bb++;

}

Closures and lexical scoping :

Closures jo hain ye b aik concept hai iss ka matlab k aisa function jo k return karay aik aur function and wo return honay wala function jo hai wo always use karay ga parent function ka koi na koi variable tu tub hm uss ko kahengay closures.

Jaisay for eg:

function hhhh() {

  let aby = 10;

  return function () {

    console.log(aby);

  };

}

Tu matlab k kahin na kahin prr ye jo return honay wala function hai iss mein kahi na kahi parent function k kisi value ka koi zikar hona hiiii chahiyee... jaisay k hm nay return honay walay function mein jo parent function mein variable declare thaa uss ko print karwa diyaa tu iss tarah kisi na kisi tareekay say return honay walay function mein parent k variable ka koi zikar hona hii chahiyeee....

So this is called Closures..

Lexical Scoping :

Lexical scoping b aik concept haii k aik variable ka scope kitnay tk haii function k andar matlab k eg given below.

function abcdef(){

    let age = 5;

    function ghij(){

        let birth = 10;

        function klmn(){

            let salary = 100;

        }

    }

}

Hm nay aik function banaya abcdef uss k andar aik variable banaya age ka tu ab ye age variable ko hm kahin prr b access kar sktay hain in teeno functions k andar magar in k bahir naii tu ab ye age variable ka scope jo hai ye saray functions tuk hai matlab sab mein access ho skta hai magar iss k bahir naiii.

Ab hm agar ghij function ki baat karein. Uss mein aik variable hm nay banaya haii birth name ka, tu ab uss ki jo range haii wo ghij function mein b haii or klmn function mein b haii qk jo klmn function hai wo ghij k andar haiii magar jo birth variable haii hm uss ko abcdef function mein access nai kar sktay qk ye uss ki range say bahir haii..

Iss k baad klmn function mein hm nay variable declare kiyaa hai salary tu ab hm uss ko siraf isse function mein access kar sktay hain qk iss ki range iss tak haiii.. hm iss ko iss say ooper k 2 functions mein access nai kr sktay...

So this is all called Lexical Scoping.

IIFE(immediate invoked functions expresssions):

Bss simply ye haiii aik function banaao orr uss ko name mat do orr jab bana lo tu uss pooray function ko () say wrap kar do orr direct uss k sth hii uss ko call kar do, iss ko kahtay IIFE. Like eg given below

(function(){

Console.log(“immediately incovked…”)

})();

this is called IIFE. K aik function haii uss ka koi name nai hai orr uss poray function ko hm nay () say wrap kr diyaa hai orr sath mein hii direct uss ko call b kar diyaa hai this is called IIFE

iss function ko dobara call karnay ki zaroorat naii hogiii qk iss ko direct uss k sth hii hm nay call kiyaa huaa haii......

IIFE say hm secret variables ya jo b ho cheezain bana saktay hain qk jo b cheezain hm IIFE k andar hm define kar lein wo bahir access nai ho sktii…

Private variables etc hm bana sktay hain using IIFE

Hoisting differences bw declaration expression:

Basically jo hoisting hm pahlay b starting mein study kr chukay hain jaisay k variables k sth hota hai hai k hmm variable ko declare karnay say pahlay hii uss ko use kar patay hon.. jaisay k hm nay aik variable banaya 10th line pr magar hm uss ko variable ko use kar parahay hain 5th line pr aisa jo hai wo baki languages mein allowed nai hota magar only javascript

tu aisa kaam jo hai wo functions k sth b hota hai k hm nay uss ko declare neechay kiyaa huaa haiii orr hm uss ko use us say ooper b kr sktay hain jaisy k like.. for eg

hoistcheck();

function hoistcheck() {

  console.log("hoist check...");

}

tu jaisa k hm dhaik sktay hain k hm nay hoistcheck ka aik function banaya hai jiss mein hm nay siraf console kiaa hai tu jaisa k hm dhaik sktay hain k hm nay iss function ko declare kiyaa neechay haii magar hm nay is ko iss say ooper call kiaa haii orr ye run ho raha hai.. so this is called Hoisting

tu jo hoisting jo haii wo function mein tu kaam krti hai magar agar hm iss ko function say function expression bana dein yani k function ko aik variable mein save kr lein orr phirr uss ko function declare honay say pahlay uss ko call krna chahein tu phirrr ye error dega... for eg.

Abcd();

Let abcd = function(){

Console.log(“hoiiisssttt check…”);

}

Tu ab hm agar iss tarah iss mein karengay tu ye error dega.

function dance() {

  console.log("Dancing...");

}

dance();

console.log("check...");

let funny = function () {

  console.log("funny function...");

};

console.log("last.");

let name = () => {

  console.log("name function");

};

function animaldance(animal) {

  console.log(`${animal} Dance Kar Raha Hai`);

}

animaldance("cow");

function additions(num1, num2) {

  console.log(num1 + num2);

}

additions(40, 20);

discount calculator

(Higher Order Function Question)

function discountCalculator(discount) {

  return function (price) {

    return price - price \*(discount/100);

  };

}

let discounter = discountCalculator(10);

discounter(200);

iss nay ye kiyaa hai hai pahlay hm nay aik function banaya jiss ka name hai discountCalculator. uss k parameter mein hm nay aik variable diyaa hai discount k name say. Wo aik aur function ko return kr raha hai uss main parameter mein price accept kiii hai orr wo humein return kar raha hai wo wali price orr uss say – kar raha hai 10% discount.

So now when we run the discount calculator orr uss ko doosray variable discounter mein save kr liyaa hai tu ye jo discountCalculator(10) chalega tu ye opper wala function run hoga orr wo new function ko return karega tu iss ki jagah prr function ajaye ga.. tu ye jo discounter variable hai ye ab aik function haii… tu iss mein hm value in rs dy sktay hain

function county(){

    let count = 0;

    return function(){

        count++;

        return count;

    }

}

county();

let cc = county();

console.log(cc());

tu iss mein hm nay aik function banaya hai county k name say. Iss mein aik variable hai count=0 k name say. Ye function aik aur function ko return kr raha hai orr jo return honay wala function haii uss mein count ki value increment ho rai hai orr wohi count ki value ko return b kr raha hai..

ab jab hm nay iss counter function ko call kiyaa tu iss nay aik function ko return kiaaa hai , hm nay aik variable banaya orr uss mein safe kr liyaa haiii tu wo variable ab aik function ki tarah treat karay ga. Like

let cc = county();

TU jab b hm cc() ko chalayengay tu ye wo wala jo function return huaa hai counter mein uss function ko chalaye ga. Jo k count ki value ko barha dega. Tu count ki value jo jaye gi 1 qk uss ko hm nay return kiaa huaa hai. Now tu ab jab hm nay console.log mein cc() ko print kiyaa tu ye 1 print karega. Agar hm dobara console.log(cc()); tu ye increment karega count ki value tu ye 2 ho jaye gi. Tu iss tarah jitnii bar hm ye lagayengay tu ye increment krta rahega. Iss mein yaad krnay kii baat ye hai k yahan prr jo count variable hai ye apni value ko yaad kr paraha hai matlab k jab hm agliii bar console.log kr rahay hain tu ye apni value ko 0 prr nai le kr jata wo wohi value prr rahta hai..

Iss ki waja ye haiii k hm nay jb cc() ko call kiyaa haii tu iss nay tu siraf wo jo return huaa thaa function wo wala run kiyaa haii tu isss wja say iss ki value wohii uss ko yaad rahti hai matlab 0 tk wapis nai jati

Agar hmm from first county() ko chalayengay tu phirr ye jo count ki value hai wo 0 say start hogii qk ye from the start hiii ye function run ho jaye ga jiss ki waja say wo count ki value 0 say dobara start ho jaye gi… like eg given below.

Let dd = county():

Console.log(dd());

Tu jo dd() hai iss ka apna aik count variable haiii orr jo cc() haii is ka apna aik count variable hai