Objects:

Log array and objects mein confuse hotay hain tu hm ye kah sktay hain k jaisay humein city mein saaray trees chaahiyee tu ye tu bohat ho saktay hain tu hm array use karengay. Agar hm kahein k city mein saaray dogs tu ye tu bohat hain tu hm array use karengay. Humein agar kisi aik cheez k baray mein aik asy zyada information chahiyee tu hm objects ka use karengay like aik person hai sharma uss ka name,uss ka address,age,father name, etc tu ab ye saari info aik banday ki yani k aik entitiy ki haii tuphirr hm object ka use karengay.

If we take eg of object so given below

let obj = {

  name: "ali",

  age: 23,

  coding: "C++",

  car: "nothing",

};

So this is how we make object.

If we want to access any thing we can use like

Obj.name or obj.age

We can also access things in objects using square bracket like

Obj[“name”]

Nesting and deep access :

For eg hm nay aik ye object banaya haii jo k bohat deep hai matlab aik key k value k andar orr objects diyee huee hain

let userProfile = {

  id: 101,

  username: "maaz\_khan",

  email: "maaz@example.com",

  isVerified: true,

  age: 22,

  address: {

    street: "123 Main Street",

    city: "Karachi",

    postalCode: "74000",

    country: "Pakistan",

    geo: {

      lat: 24.8607,

      lng: 67.0011,

    },

  },

  hobbies: ["coding", "football", "gaming", "traveling"],

  skills: {

    programming: ["JavaScript", "Python", "C++"],

    design: ["Figma", "Photoshop"],

    softSkills: ["communication", "problem-solving", "teamwork"],

  },

  education: [

    {

      degree: "Matric",

      year: 2018,

      institute: "ABC School",

    },

    {

      degree: "Intermediate",

      year: 2020,

      institute: "XYZ College",

    },

    {

      degree: "BS Computer Science",

      year: 2024,

      institute: "DEF University",

    },

  ],

  socialMedia: {

    twitter: "@maaz",

    github: "maazkhan",

    linkedin: "linkedin.com/in/maaz",

  },

  posts: [

    {

      id: 1,

      title: "Learning JavaScript",

      content: "Today I practiced objects and arrays!",

      likes: 120,

      comments: [

        { user: "Ali", text: "Great job!" },

        { user: "Sara", text: "Keep going 🚀" },

      ],

    },

    {

      id: 2,

      title: "MERN Stack Journey",

      content: "Started learning Node.js and Express!",

      likes: 85,

      comments: [],

    },

  ],

  isOnline: false,

  lastLogin: new Date("2025-08-10T14:30:00"),

  settings: {

    theme: "dark",

    notifications: {

      email: true,

      sms: false,

      push: true,

    },

    language: "English",

  },

};

Tu ab for eg humein access krna hai userProfile mein kuch tu humein

userProfile.address.geo.lat;

tu ye jaye ga userprofile object mein phirr uss k andar address object mein phirr uss k andar geo mein orr wahan say lat ko access karay ag.

Tu basically nesting hai aik k andar aaik orr and so on. Or jo deep access hai wo aik key k andar aik key value, and so on and un ko access karna tu iss ko deep access kahtay hain.

Object Destructuring:

Tu jiss tarah hm nay array mein de structuring ki thi use tarah hm objects mein b karengay magar wo array tha tu hm nay [] lagaye thay yahan hm {} lagayengay qk ye object hai.

Tu for eg hm bar bar aik lambi line likh rahay hain jaisay

userProfile.address.get.lat tu agar ye line hmuein zyada baar likhni hoti tue ye mushkill ho jata tu hm ye iss ko destructure kr sktay hain matlab k aik orr new variable mein un ko save kr sktay hain. Like

let {lat,lng} = userProfile.address.geo;

yahan prr jo names hm nay use kiyee hain wo exact wohi honay chahiyee jo ooper object mein hain. Ab yahan prr matlab humein lat and lng dono ko lena hai tu hm yee iss tarah karengay.

Tu ab iss ka matlab haii k userprofile mein address mein geo mein jaao orr uss mein jo b lat lng mein hai uss ko {lat,lng} mein save kr do. Tu abb humein future mein bar bar nahi likhna parega ye poori line bs lat ko access krnay k liye direct lat likh sktay hain.

Looping : for-in, Object.keys, Object.entries:

for (let keyy in userProfile) {

  console.log(keyy);

}

Tu iss mein b ussse tarah har baar aik aik value yani k key aati rahegi ye jo hum nay keyy lagaya hai ye simple aik variable hai ye kuch b ho skta hai . magar wo jo last mein hai ye object ka name hai ye same hona chahiyee…

Tu ab agar hm iss keyy ko print karein tu ye saari keys de dega jo b userProfile object mein hain.

Tu if for eg humein print krni hai key ki values b tu values access krnay ka koi option nai diyaa tu yahan prr ab jo hm square bracket use kray thay object say value nikalnay k liyee tu wo ab yahan prr use hoga. Like

for(let mky in userProfile){

    console.log(userProfile[mky]);

}

Tu ab ye kia kr raha hai k mky variable mein tu har bar aik aik key ati rahti hai object ki. Tu ab for eg hm nay keys ki values b nikalni hain tu ab hm ye kr sktay hain k userProfile k baad [ ] mein mky likh lengay tu jab har baar mky mein key change hoti rahi gii tu har baar humein iss key ki value milti rahegi…

Orr agar hum nay key orr value dono sath hii print krni tu hm iss tarah b kr sktay hain

for (let bb in obj) {

  console.log(bb, obj[bb]); // for printing key and value both.

}

Tu ye har baar jab key aye gii tu uss ksath obj[bb] mein ye uss ki value b print kr dega.

Object.keys: tu jab b humein kisi object ki sari keys ko lena ho orr un keys ko kisi array mein dalna ho tu hm object.keys ka use kr sktay hain

Object.keys(userProfile);

Tu ye humein saray keys de dega userProfile object ki orr un ko doosray array mein save kr lega

Tu is ka ans hoga

['id', 'username', 'email', 'isVerified', 'age', 'address', 'hobbies', 'skills', 'education', 'socialMedia', 'posts', 'isOnline', 'lastLogin', 'settings']

Object.Entries: tu wo jo object.keys jo tha wo jo humara object hota hai un ki sari keys ko utha leta haii orr aik aur array mein uss ko dal deta hai.

Agar hm hm object.entries ki baat karein tu ye iss tarah karta hai k jitney b elements hotay hain yani keys and un ki value dono ye sab say pahlay un saro ka aik array bana leta hai phirr jo b key value pair hota hai jitney b hotay hain un sab k alag alag arrays banata hain matlab k array k andar arrays.

Object.entries(obj);

Iss ka answer iss tarah hoga.

[Array(2), Array(2), Array(2), Array(2), Array(2)]

0: (2) [‘name’: ‘ali’]

1: (2) [‘age: 23]

2: (2) [‘coding: ‘c++]

3: (2) [‘car: ‘nothing]

4: (2) [‘male: true]

Ye aik whole array hai orr uss k andar orr arrays hain

Array or arrays bana hai.

Copying Objects :

Using Spread Operator

obj = {...userProfile};

this … are the spread operator and due to this … all the things inside the userProfile object will be copied to obj object.

Object.Assign() : ye b copying krnay k kaam hi ata hai magar ab koi iss ko zyada use nai krta

Tu agar hm iss ko use karein tu iss tarah karengay

let myoo = Object.assign({}, myobj);

tu ye kiaa karega k hm aik empty {} iss mein rakh lengay orr jiss ko copy krwana hai uss ka name rakh lengay tu ye uss obj ko copy kr lega.

Orr agar hm chahtay hain k uss doosray object ko copy tu karo hi karo sath sath mein hm kisi orr cheez ko add b krna chahtay hain tu phir hm iss empty {} ka use kr sktay hain like

let myoo = Object.assign({earning: “not”}, myobj);

tu ye myobj ko copy krnay k sath sath ye aik new key value ko b add kr lega earning : not iss ko add kr dega myoo mein.

Deep Cloning:

Jab hm spread operator k through kisi object ko copy krtay hain tu cheezain copy ho jatii hain wo tu koi msla nai hai.. magr for eg humaray object mein kuch nested objects b hain orr hm uss object ko copy krna chahtay hain through spread operator tu wo humein aisa show karega k copy tu ho gya hai magar wo asal meincopy nai huaaa hota wo siraf uss nay reference pass kiyaa hota hai tu matlab k

let myobj = {

  id: 101,

  username: "maaz\_khan",

  email: "maaz@example.com",

  isVerified: true,

  age: 22,

  address: {

    street: "123 Main Street",

    city: "Karachi",

    postalCode: "74000",

    country: "Pakistan",

    geo: {

      lat: 24.8607,

      lng: 67.0011,

    },

  },

};

Let myobj2 = {...myobj};

Tu iss mein ham dhaik sktay hain k nested object b hai address k name say tu agar hm iss ko copy karein spread operator say tu ye jo ooper walay simple hain ye copy ho jayengay magar ye jo address object hai iss ka siraf reference hii pass ho jaye ga. Tu ye problem hai qk agar hm ye karein k

myobj2.address.city = "Lahore";

matlab k hm nay siraf obj2 mein city ko change kr diyaaa hai tu hona ty ye chahiyee k siraf obj2 mein hi changes aye hon obj mein changes na aye ho qk ye copy huee hai orr hm iss meni changes kr rahay hain tu bss iss mein hi honi chahiyee siraf magar jab hm city ko print krtay hain tu ye dono mein changes ho gaye hoti hain jo k problem hai. Tu iss say bachnay k liyee hm deep cloning kartay hain

tu deep cloning ka matlab hai k sari cheezon ka real copy krna matlab k jo simple key value pair hain un ka b real copy krna orr jo nested things hain un ka b real copy krna(reference pass na krna)

tu main baat ye hai k jab b hm apnii object ko copy krna chahein orr dhekein k uss mein kuchh nested object things b hain tu phirr hm ...spread operator ka use nai karengay

uss k liyee hm ye karengay k aik aata JSON.stringify(obj);

tu ye kiaa krtaaa hai k jo b elements hon object mein chahay key ho uss ki value ho jo b ho wo uss ko string kr deta hai.

Orr agar hm chahein k hm dobara iss ko apni real form mein le kr ajayein tu phirr hm iss method ka use kr sktay hain JSON.parse(obj) tu ye iss ko wapis say use apni real form mein lein ky aaye ga.

Tu for eg

let ab = {

  id: 101,

  username: "maaz\_khan",

  email: "maaz@example.com",

  isVerified: true,

  age: 22,

  address: {

    street: "123 Main Street",

    city: "Karachi",

    postalCode: "74000",

    country: "Pakistan",

    geo: {

      lat: 24.8607,

      lng: 67.0011,

    },

  },

};

let aabb = JSON.parse(JSON.stringify(ab));

hm nay iss mein ye kiyaa hia k aik ab ka object banaya hai phirr uss ko deep clone krnay k liyee uss ko aabb mein copy krnay k liye pahlay iiss pooray object ko string mein change kiyaa hai using JSON.stringify phiirr uss ko wapis say apnay real form mein lanay k liyee hm nay json . parse use kiyaa hai tu iss say ye deep clone ho jaye ga matlab k agar ab hm aabb.address.city ko change karein tu jo ab.address.city hogi wo wohi same rahegi matlab k ye ab fully clone ho chukka hai.

Optional Chaining and Computed Properties:

Optional chaining ka meaning hai k for eg hm aik obj say values ko access kar rahay hain orr wo abi sab teek chal raha hai magar baad mein for eg uss mein kisi nay uss k kisi aik key ko change kr dii ho jiss ko hm access krnay ki try kr rahay hain tu ye humein error dega k ye key nai hai tu iss say bachnay k liyee hm optional chaining karty hain. Like

obj.address.city;

iss mein hm address mein city ko print krwa rahay hain orr wo saii print ho rai hai for eg agar kisi nay baad mein iss address ko addresses kar diyaa orr hm ab b obj.address.city likhengay tu ye error dega k ye address nai hai qk wo abi addresses ho chukka hai tu ab iss error say bachnay k liyee hm optional chaining kartay hain. Tu uss mein hm ye kahtay hain k

obj?.address?.city

tu iss ka matlab hai k hm iss ko ye kah rahay hain k dhaik lo obj mein k agar address ho orr address mein dhaik lo agar city name ki key ho tu humein dedo warna agar na ho tu error mat do. Tu ye siraf humein undefined de dega. Matlab k ye hm nay optional thing de dei hai. Jiss say error nai milega humein siraf undefined mil jaye ga.

Computed Properties: iss ka example ye hai k like

let role = "admin";

ab = {

  id: 101,

  username: "maaz\_khan",

  email: "maaz@example.com",

  isVerified: true,

  [role]: "Afridi",

  age: 22,

  address: {

    street: "123 Main Street",

    city: "Karachi",

    postalCode: "74000",

    country: "Pakistan",

    geo: {

      lat: 24.8607,

      lng: 67.0011,

    },

  },

};

Tu ye ab kiaa karega jahan par hm nay [role] lagaya wahan prr ye [role] ki jagah prr admin print karega orr value tu hm nay Afridi di haa tu bss simple ye hai computed properties.