~\Desktop\diabetes\new.js

```
//q1 var let const me kya farq hai
// var we can chnage the varibale value
//conts we can not chnage the value and
// var are function scope varibale it can be used anywhere inside the function
//let are brace scope variable can not be used  outside the braces
//let is curly scope we can a value with in a curly braces

var kaif=12;
var kaif=11;
console.log(kaif)
// const kaif1=21;
// const kaif11=22;
// console.log(kaif)
function kaifs(){
    let a=1
    console.log("hello"+a)
}
console.log(a)
kaifs();
// 2
//type in js
// 1/ primitive type , no,null,string copy krne pe pura copy hojata hai
// 2/ reference type ,array,object, copy krne pe pura copy ni hota balki refernce value pass hojati hai
var a=[1,2,3]
var b=a;
//3 with the help of spread operator we can easy copy
var b=[...a]
a.pop()
console.log(a)
//4 funtion
// use to reduce the code work
// reduce code redundancy
// we can test function by passing different diferent value

//5 argument and parameter difference?
// argumnet are the value jo ham value dete hau function ke chlate waqt
// parameter are the variable in which argument value stored

function kaifss(a,b){
    console.log(a+b)
}
kaifss(2,3)

//6 arr questions
var bhai=[1,2,3,4]
bhai.pop() //delete the 4
console.log(bhai)
bhai.push(4)//ading value
console.log(bhai)
bhai.shift()// shure me ek value hatane ke liye
console.log(bhai)
bhai.unshift(1)
console.log(bhai)// shuru me ek value add krne ke liye

//7 Execution context is the box which contain varibale ,function and lexical environment basically execution contect can tell what the function can acces or not

```

```
//8 lexial environment -> lexical environment is chart which tell the function scope what can acces or not
//it hold scope and scope chain

//9 call back function -> call back function are the function passing as argumnet to another function best example setTimout set interval
setTimeout(function(){
    console.log("kaif")
},1000)
// the particular code is executed after the certain period of time

setInterval(function() {
    console.log("kaif")
}, 1000);
//after each and every second code will executed

//10 object are variable too object is used to hold more variables
 var shaikh={
    name:"kaif",
    age:22,
    profession:"Engineer",
    fullname:function(){
    return this.name+this.age;
    }
 }
 console.log(shaikh)// whole object
 shaikh.profession="dr"
 console.log(shaikh.profession)
 delete shaikh.name // by using delete keyword we can delete the property from the object
 console.log(shaikh.name) //print the particular name
 console.log(shaikh.fullname()) // print the function properties
 //11 this keyword is used to acces the object property in to the function in other word we can say that with the helpm of this kyewword we can acces the object properties in to the function
//12 use of new keyword with the help of new keyword we can make same type of objects
// Constructor function definition
function Js(name, work, age) {
    this.name = name;
    this.work = work;
    this.age = age;
}
//13 The new keyword in JavaScript is used to create a new object from a constructor function
// Creating an instance using the constructor function
const kaifsss = new Js("JavaScript", "frontend/backend", 22);
console.log(kaifsss);

//14 jsonstringfy() convert the object to string

//15 get and set allow to get and set object property via methord/function

//string
let c="kaif";
let d="zara";
console.log(c.toUpperCase())//convert string to upper case
console.log(c.toLowerCase())// convert string to lower case
console.log(c.concat(" ",d))
console.log(c.slice(1,3))// slice string from 1 index to 3
console.log(c.repeat(3))// string repeat 3 times
console.log(c.replace("kaif","kaifshaikh")) //replace the words
console.log(c.length)//return the lenght
```

```
console.log(c.charAt(2))// give specific char at particlar index

//16
// js is synchronus langugae
//sync and asyc
//sync matlab ek ke bad dusra hoga jab tak first execute ni hota tab tak dusra excexute ni hoga
// async matlab sare kam ek sath honge

// 17
//the comunication between main stack and side stack is managed by
//eventloop eventlop is a lopp which tellwhich satement is execute first sunc or async

//18 js is singel thread el bar me ek kam hi hoga

//19 promises promise are the improvement over call back function cleaner way to handel async operation
//it contain rej,resolve
//code divide in two parts resolve and rej
//agar resolve hoga to then() execute and rej() hoga to catch() execute
 var ans =new Promise((res,rej)=>{
    if(true){
        return res();
    }
    else{
        return rej();
    }
 })
 ans.then(function(){
    console.log("resolve hogya ")
 })

 //20 async await then se bachne ke liye aync ka use jese jo kam promises se hira jada code likhna pad rha asunc se kam likhn a padega
```