Functions:

A function is reuseable block of code ,it can be called any where in the program.

You can use the same code with different arguments to get the results

Js:

Key word = function

()

Parameters=value which are declared inside the () while defining a function

Arguments: values which are passed through the parameters

function greet(){

            console.log("hello")

  }

 greet()

        function Greetings(name) {

            let message = `Hello, ${name}!`;

            console.log(message);

        }

        Greetings("Ashish");

        function showMessage(name = "Guest") {

            let message = `Hello, ${name}!`;

            console.log(message);

        }

        let name =prompt("Enter your name?")

        showMessage()

        showMessage("Ashish");

        showMessage(name)

o/p:

hello

Hello, Ashish!

Hello, Guest!

Hello, Ashish!

Hello, Trump!

Anonymous function:

A function with out name , After we create a function without name and we assign it to a variable.

Ex:

let square = function(number) {

            let n = number \* number;

            document.write(n);

 };

  let number = prompt("Enter the number:");

  let result = square(number);

Arrow Functions:

More readable and more structured

Anonymous functions(lambda functions)

Without function name but they are assigned to a variable

Let c=()=>{

Console.log(“Hi Ashish”)

}

C()

o/p:

Hi Ashish

Arrow function with parameters

let x=(x,y)=>{

            console.log(x+y)

}

x(20,30)

o/p:

50

Arrow function with one argument

Without using paranthesis

Let greet=x=>console.log(x);

greet(‘hello’)

Arrow function with out argument

let greet=()=>console.log(“hello”)

greet()

//Arrow functions as a expression

        let age=18;

        let welcome=(age<18)?

        ()=>console.log("hi Gowtham"):

        ()=>console.log("hi sai chran");

        welcome()

o/p:

hi sai chran

 //synchronous and asynchronous

        function task1(){

            console.log("Task1:Start")

        }

        function task2(){

            console.log("Task2:Start")

        }

        function task3(){

            console.log("Task:Start")

        }

        task1()

        task2()

        task3()

//asynchronomous

        //multiple tasks at a time

        console.log("hi ashish")

        setTimeout(()=>{

                console.log("Hi iam waiting")

        },5000);

        console.log("I am in lab")

 //javascript callbacks

        //A call back function is a function that is passed argument to another function;

        function greet(name, callback) {

            console.log(`Hello, ${name}!`);

            callback();

        //this calls the callback functions after greeting

        }

        function sayGoodbye(){

            console.log("Goodbye")

        }

        //call greet and pass sayGoodbyeas the callback

        greet("sam",sayGoodbye)

//promise

        //eventual completion

        //Three states

        //1)pending:still not completed --continuing

        //2)fulfilled:task completed

        //3)Rejected:The operation is failed

        let mypromise=new Promise((resolve,reject)=>{

            let success=true;

            if(success){

                resolve("task completed")

            }else{

                reject("task failed")

            }

        })