 //a promise a special javascript object that represents the eventual completion(or failure )

        //of an asynchronous operation .

        //it allows to handle asynchronous tasks more efficiently than callbacks .

        //three states:

        //1)pending : the initial state,the promise is still waiting for the operation to finish

        //2)fulfilled : the operation completed successfully.

        //3)Rejected: the operation failed.

        //exp:

        //1)to create a promise -new promise()

        //Two parameters-

        //resolve -- to mark the promise as successfully

        //reject --mark it as failed

        //2)handling the result:

        //then()method is called when promise is fullfilled.

        //It will receives the result from resolve()

        //catch() method is called if the promise is rejected it receives the error message from reject().

        //example:

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

            let success =true;

            if(success){

                resolve("the operation is successfull");

            }else{

                reject("the operation failed")

            }

        });

        mypromise

        .then(result=>{

            console.log(result);

        });

        mypromise

        .catch(error=>{

            console.log(error);

        });

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

        resolve(5);

        //initial resolve value is 5

    });

    promise.then((result)=>{

        console.log(result);

        return result \* 2

        //return a new value

    })

    .then((result)=>{

        console.log(result);

        return result + 3

        //return a new value

    })

    .then((result)=>{

        console.log(result);

    })

    .catch(error=>{

        console.log("error", error)

    })

     //ex: promise with setTimeout

     //asynchronous task using setTimeout, and the promise will resolve after a certain amount of time

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

        resolve(5);

    })

    pro.then(result=>{

        console.log(result)

        setTimeout((pro)=>{

            console.log("success")

        },3000)

    })