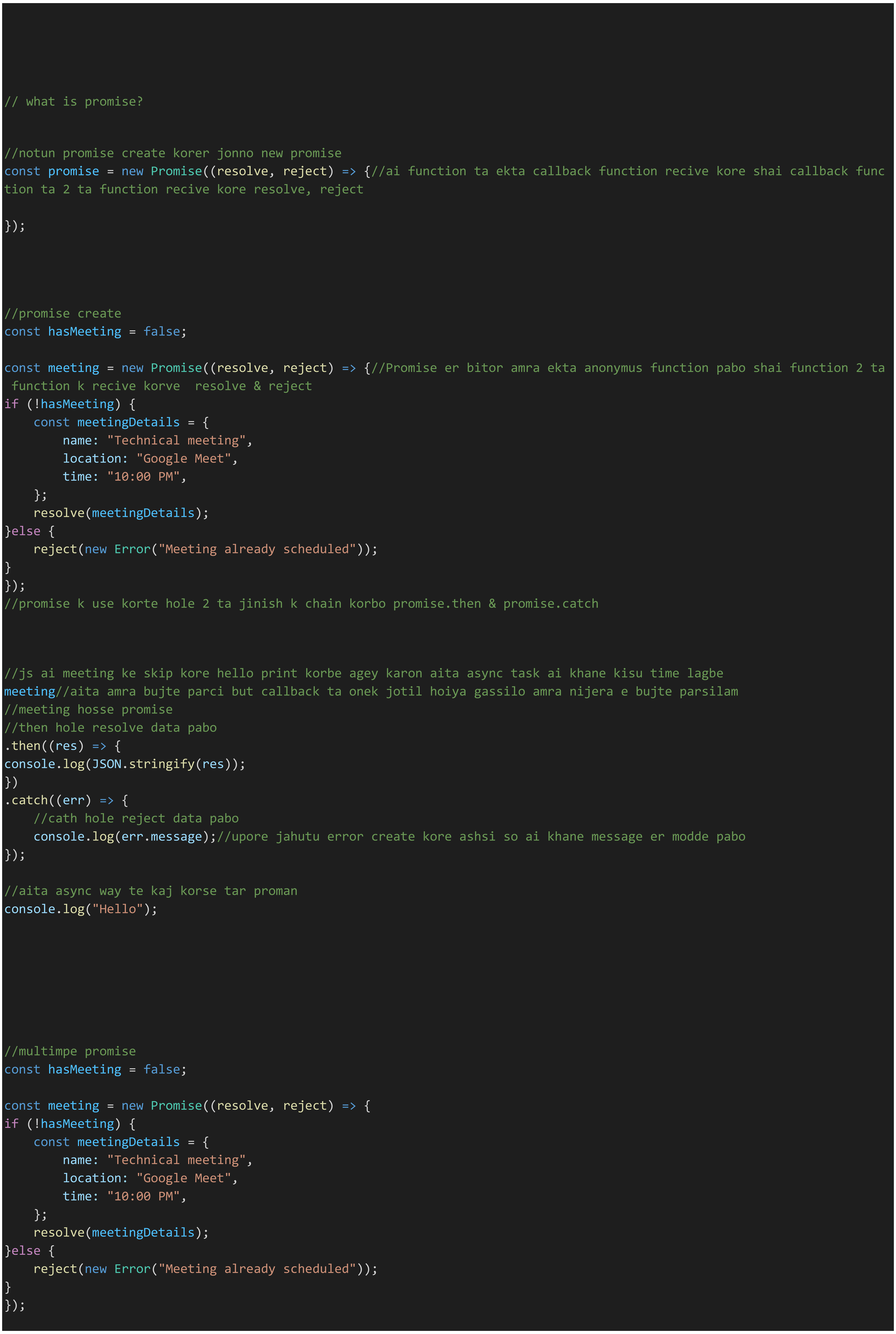
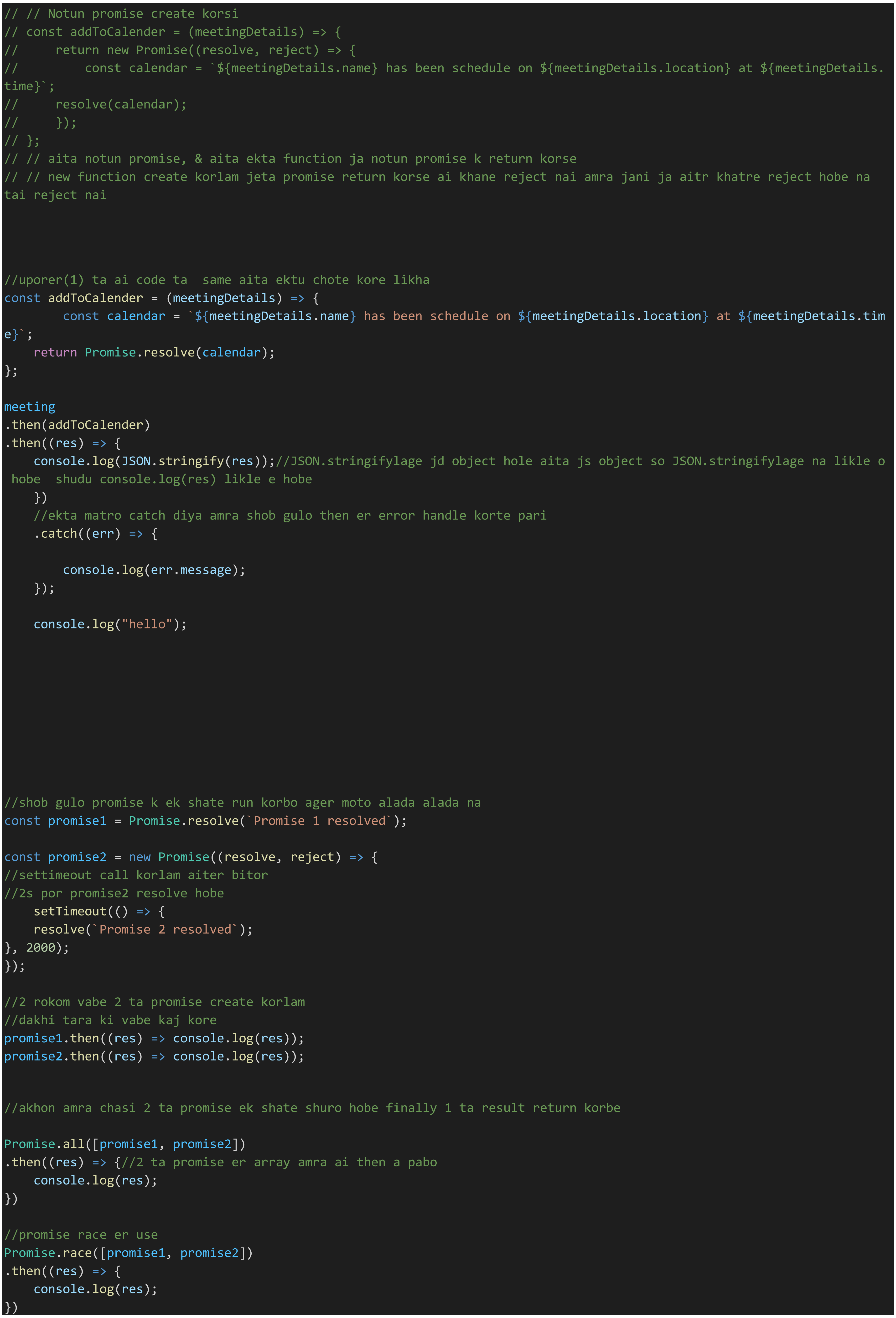
# Asynchronous js callback promise async-await

```
// Asynchronous JavaScript - callbacks, promises and async-await
//async task shob shomoi promise return kore shai promise er shate try or catch kore amra promise execution korte peri, shob cheya valo upai a async task korte amra async-await baboher korte pari

// synchronous Example
const processOrder = (customer) => {
    console.log(`Processing order for customer 1`);

    var currentTime = new Date().getTime();
    while (currentTime + 3000 >= new Date().getTime());//ai line a 3 second block hoiya thakbe

    console.log(`order processed for customer 1`);
};

console.log(`take order for customer 1`);
processOrder();
console.log(`completed order for customer 1`);


// Asynchronous Example
const processOrder = (customer) => {
    console.log(`Processing order for customer 1`);

    // setTimeout(() => {}, 3000);
    // setTimeout js er ekta buildin function eta Asynchronous function akhon er block hobe na
    setTimeout(() => {
        console.log(`Cooking completed`);//last a print hobe


    }, 3000);
    console.log(`order processed for customer 1`);
};

console.log(`take order for customer 1`);
processOrder();
console.log(`completed order for customer 1`);



// Asynchronous callback Example
const takeOrder = (customer, callback) => {//callback pass korer jonno callback recive korte hobe
    console.log(`take order for ${customer}`);
    callback(customer);//callback pass kore dibe
}

const processOrder = (customer, callback) => {
        console.log(`Processing order for ${customer}`);

        setTimeout(() => {
            console.log(`Cooking completed`);
            console.log(`order processed for ${customer}
            `);
            callback(customer);
        }, 3000);

    };

    const completeOrder = (customer) => {
        console.log(`Complete order for ${customer}`);
    }
    //tackOrder k call korlam parameter pass korsi customer & callback pass korsi
    takeOrder('customer 1', (customer) => {
        processOrder(customer, (customer) => {
            completeOrder(customer);
        })

    });
//ai nested callback k dur korer jonno promise
    console.log(`Hello`);//aita Processing order for customer er por print korbe karon er por async function baki gula sync function
```



```
// what is promise?


//notun promise create korer jonno new promise
const promise = new Promise((resolve, reject) => {//ai function ta ekta callback function recive kore shai callback function ta 2 ta function recive kore resolve, reject

});




//promise create
const hasMeeting = false;

const meeting = new Promise((resolve, reject) => {//Promise er bitor amra ekta anonymus function pabo shai function 2 ta function k recive korve  resolve & reject
if (!hasMeeting) {
    const meetingDetails = {
        name: "Technical meeting",
        location: "Google Meet",
        time: "10:00 PM",
    };
    resolve(meetingDetails);
}else {
    reject(new Error("Meeting already scheduled"));
}
});
//promise k use korte hole 2 ta jinish k chain korbo promise.then & promise.catch



//js ai meeting ke skip kore hello print korbe agey karon aita async task ai khane kisu time lagbe
meeting//aita amra bujte parci but callback ta onek jotil hoiya gassilo amra nijera e bujte parsilam
//meeting hosse promise
//then hole resolve data pabo
.then((res) => {
console.log(JSON.stringify(res));
})
.catch((err) => {
    //cath hole reject data pabo
    console.log(err.message);//upore jahutu error create kore ashsi so ai khane message er modde pabo
});

//aita async way te kaj korse tar proman
console.log("Hello");







//multimpe promise
const hasMeeting = false;

const meeting = new Promise((resolve, reject) => {
if (!hasMeeting) {
    const meetingDetails = {
        name: "Technical meeting",
        location: "Google Meet",
        time: "10:00 PM",
    };
    resolve(meetingDetails);
}else {
    reject(new Error("Meeting already scheduled"));
}
});
```



```
// // Notun promise create korsi
// const addToCalender = (meetingDetails) => {
//     return new Promise((resolve, reject) => {
//         const calendar = `${meetingDetails.name} has been schedule on ${meetingDetails.location} at ${meetingDetails.time}`;
//     resolve(calendar);
//     });
// };
// // aita notun promise, & aita ekta function ja notun promise k return korse
// // new function create korlam jeta promise return korse ai khane reject nai amra jani ja aitr khatre reject hobe na tai reject nai




//uporer(1) ta ai code ta  same aita ektu chote kore likha
const addToCalender = (meetingDetails) => {
        const calendar = `${meetingDetails.name} has been schedule on ${meetingDetails.location} at ${meetingDetails.time}`;
    return Promise.resolve(calendar);
};

meeting
.then(addToCalender)
.then((res) => {
    console.log(JSON.stringify(res));//JSON.stringifylage jd object hole aita js object so JSON.stringifylage na likle o hobe  shudu console.log(res) likle e hobe
    })
    //ekta matro catch diya amra shob gulo then er error handle korte pari
    .catch((err) => {

        console.log(err.message);
    });

    console.log("hello");









//shob gulo promise k ek shate run korbo ager moto alada alada na
const promise1 = Promise.resolve(`Promise 1 resolved`);

const promise2 = new Promise((resolve, reject) => {
//settimeout call korlam aiter bitor
//2s por promise2 resolve hobe
    setTimeout(() => {
    resolve(`Promise 2 resolved`);
}, 2000);
});

//2 rokom vabe 2 ta promise create korlam
//dakhi tara ki vabe kaj kore
promise1.then((res) => console.log(res));
promise2.then((res) => console.log(res));


//akhon amra chasi 2 ta promise ek shate shuro hobe finally 1 ta result return korbe

Promise.all([promise1, promise2])
.then((res) => {//2 ta promise er array amra ai then a pabo
    console.log(res);
})

//promise race er use
Promise.race([promise1, promise2])
.then((res) => {
    console.log(res);
})
```

```
//ai promise use kora kaw larg application er khatre programmer friendy na karon aita o bujte problem hai then, then ...
.....  so ai jonno er solution async-await
//programmer hishabe amder jonno shubida hoi async kaj gulo line by line sync way te likte parly aita async-
await er maddome kora jai
function friendlyFunction(){//aita sync function output ekta string hello ashbe
return `Hello`;
}
console.log(friendlyFunction());

//1. async function nijera banabo, shamne ekta async lagalai holo async function
async function friendlyFunction(){//aita async function akhon promise return korbe output
    return `Hello`;
    }
    console.log(friendlyFunction());

//2. both 1,2 same 2 vabe e likha jai
    function friendlyFunction(){
        return Promise.resolve(`Hello`);
    }
    console.log(friendlyFunction());


    //async eka use hoi na er shate await use hoi, then er kaj tai await kore
    //await mane oppekha kori, aita async er bitor use hoi jamon setTimeout,API ai gula te use hoi, normal function a
    await use korte parbo na
    //Ager example tai async-await use kory example a dakhi…

 const hasMeeting = false;

const meeting = new Promise((resolve, reject) => {//Promise er bitor amra ekta anonymus function pabo shai function 2 ta
 function k recive korve  resolve & reject
if (!hasMeeting) {
    const meetingDetails = {
        name: "Technical meeting",
        location: "Google Meet",
        time: "10:00 PM",
    };
    resolve(meetingDetails);
}else {
    reject(new Error("Meeting already scheduled"));
}
});

 const addToCalender = (meetingDetails) => {
           const calendar = `${meetingDetails.name} has been schedule on ${meetingDetails.location} at ${meetingDetails
.time}`;
        return Promise.resolve(calendar);
    };

//ai tukor bodole async-await diya korbo…
                        // meeting
                        // .then(addToCalender)
                        // .then((res) => {
                        //     console.log(JSON.stringify(res));
                        //     })
                        //     .catch((err) => {

                        //         console.log(err.message);
                        //     });
 //await kono ekta async kajer jonno wait kore
//async-await use korbo so er jonno amder oboshi ekta function lagbe
 //3
async function myMeetig() {//async ase jahutu js jane ami ai khane wait korbo na she ai kaj ta onno ek jon k diya se
  porer line a chole gese
   const meetingDetails = await meeting;//await mane akhon js bujte parbe amak wait korte hobe
   const calendar = await addToCalender(meetingDetails);
   console.log(calendar);
}
myMeetig();
console.log(`Hello`)



//ato khon resolve deklam reject dakhi nai akhon shai tai dekbo
//nornal function a amra try, catch diya error handle kori ai kahne o tai korbo error er jonno
async function myMeetig() {
try{//(3)pura code take try er bitor diya dilan
    const meetingDetails = await meeting;//await a reject hoiya catch a chole jabe so catch print hobe// promise theke error paya fire ashle e catch print hobe
    const calendar = await addToCalender(meetingDetails);
 console.log(calendar);
}catch {
    console.log(`Something wrong`);
 }
}
myMeetig();
```