const fs=require('fs');

function kirat(cb){

    fs.readFile("a.txt","utf-8",function(err,data){

       cb(data);

    });

}

function ondone(data){

    console.log(data);

}

kirat(ondone)

const fs=require('fs');

fs.readFile("a.txt", "utf8",function(err,data){

  console.log(data);

})

console.log("hi there");

console.log("Hi!");

setTimeout(function timeout() {

console.log("Click the button!");

}, 5000);

let a=0;

for(let i=0;i<10;i++){

a=a+1;

}

console.log(a);

console.log("Welcome to loupe.");

With promise

const fs=require('fs');

function kiratReadFile(){

return new Promise(function(resolve){

fs.readFile("a.txt","utf-8",function(err,data){

resolve(data);

});

});

}

function onDone(data){

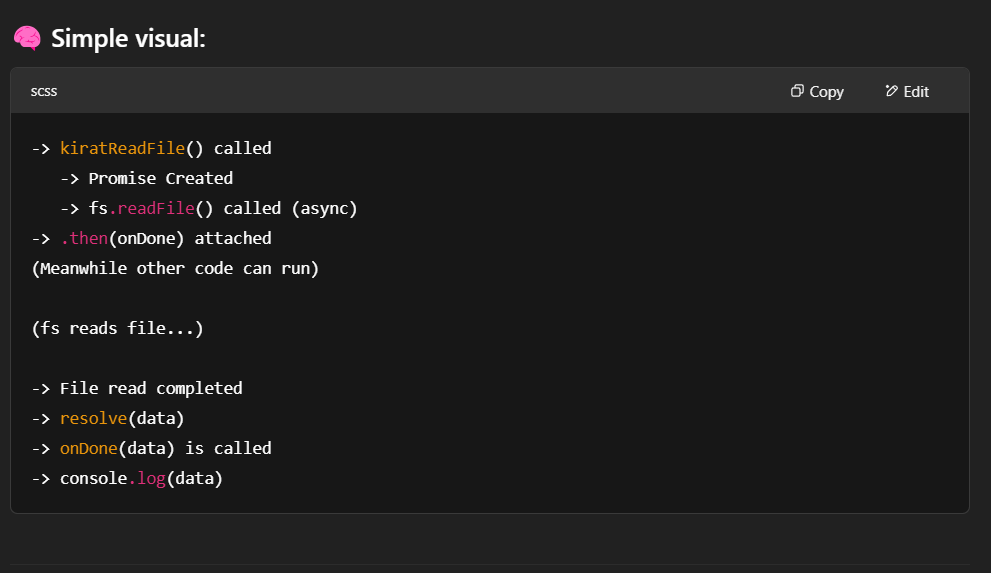
console.log(data);

} kiratReadFile().then(onDone);

Which function happens first just tell what happen after executing the functions

**Step-by-step execution:**

1. **kiratReadFile() is called.**
   * It immediately **creates a new Promise**.
   * Inside the Promise, fs.readFile("a.txt", "utf-8", function(err, data){...}) is called.
   * fs.readFile **starts reading a.txt file** asynchronously. It **does not block** the next code.
   * The Promise stays **pending** for now.
2. **Meanwhile, .then(onDone) is attached** to the Promise.
   * This means: *"when the Promise is resolved, call onDone function with the data."*
3. **After some time** (depending on file reading speed), fs.readFile **finishes** reading.
   * It calls its internal callback: function(err, data).
   * Your code resolve(data) is called inside that callback.
   * **The Promise gets resolved** with the file's contents (data).
4. **Then .then(onDone) triggers automatically**.
   * It calls the onDone(data) function.
   * onDone(data) simply does console.log(data).
   * The content of the file (a.txt) gets printed in your console.



**🔥 Important Points:**

* fs.readFile is asynchronous → It **does not block** JavaScript.
* Promise resolves **later** → .then waits for it.
* **onDone** is executed **after** file reading is finished.
* const fs=require('fs');
* function kiratReadFile(){
* console.log("inside kiratReadFile()");
* return new Promise(function(resolve){
* console.log("inside promise");
* fs.readFile("a.txt","utf-8",function(err,data){
* console.log("inside readFile()");
* resolve(data);
* });
* });
* }
* function onDone(data){
* console.log(data);
* }
* var a= kiratReadFile();
* console.log(a); // Promise { <pending> }
* a.then(onDone);

>> node main.js

inside kiratReadFile()

inside promise

Promise { <pending> }

inside readFile()

I am a superstar, don't play me with very hard

inside kiratReadFile()

inside promise

Promise { <pending> }

inside readFile()

I am a superstar, don't play me with very hard

==================================================================================

With promise

const fs=require('fs');

function kiratReadFile(cb){

    fs.readFile("a.txt","utf-8",function(err,data){

      cb(data);

    });

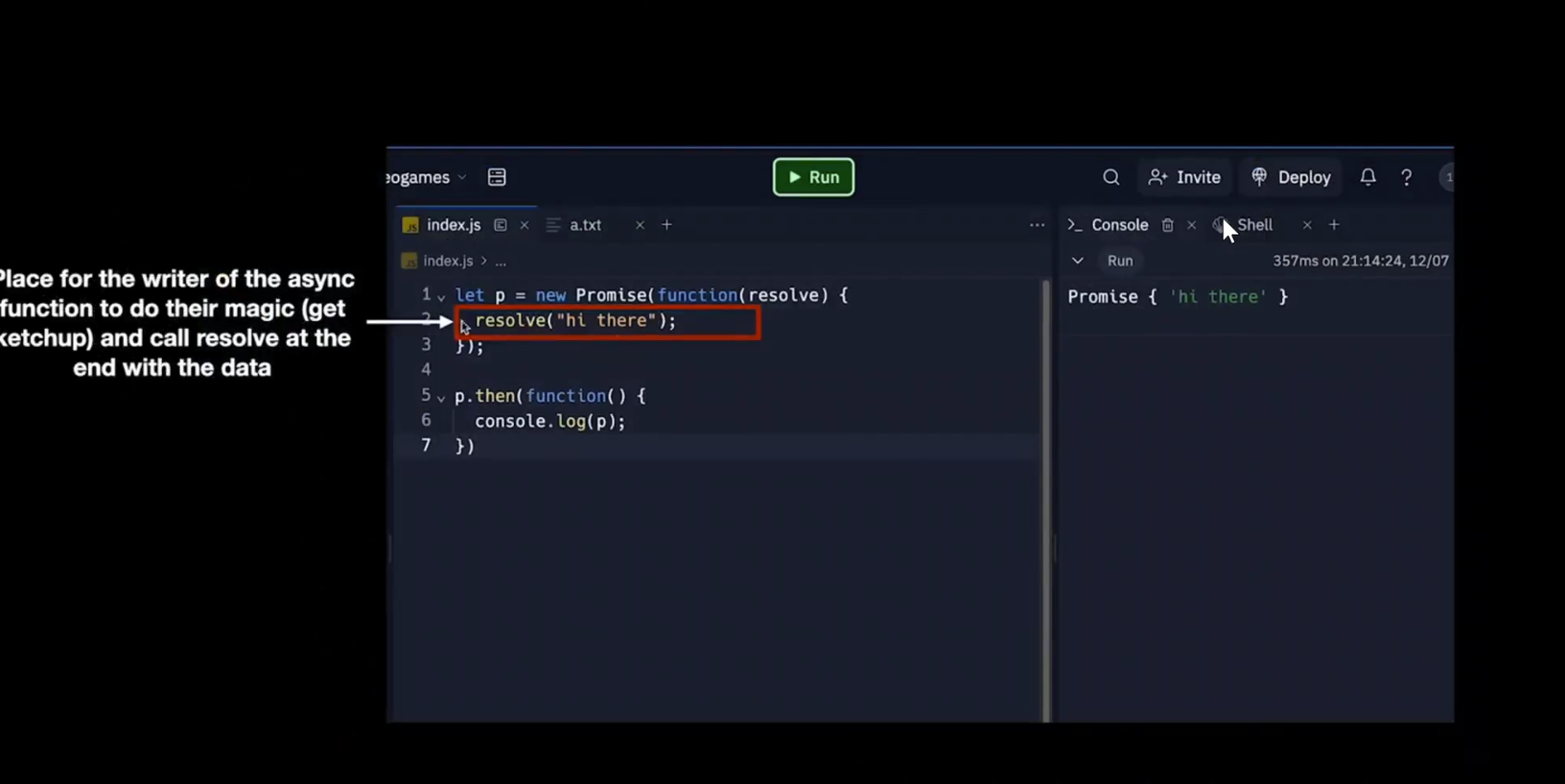
}

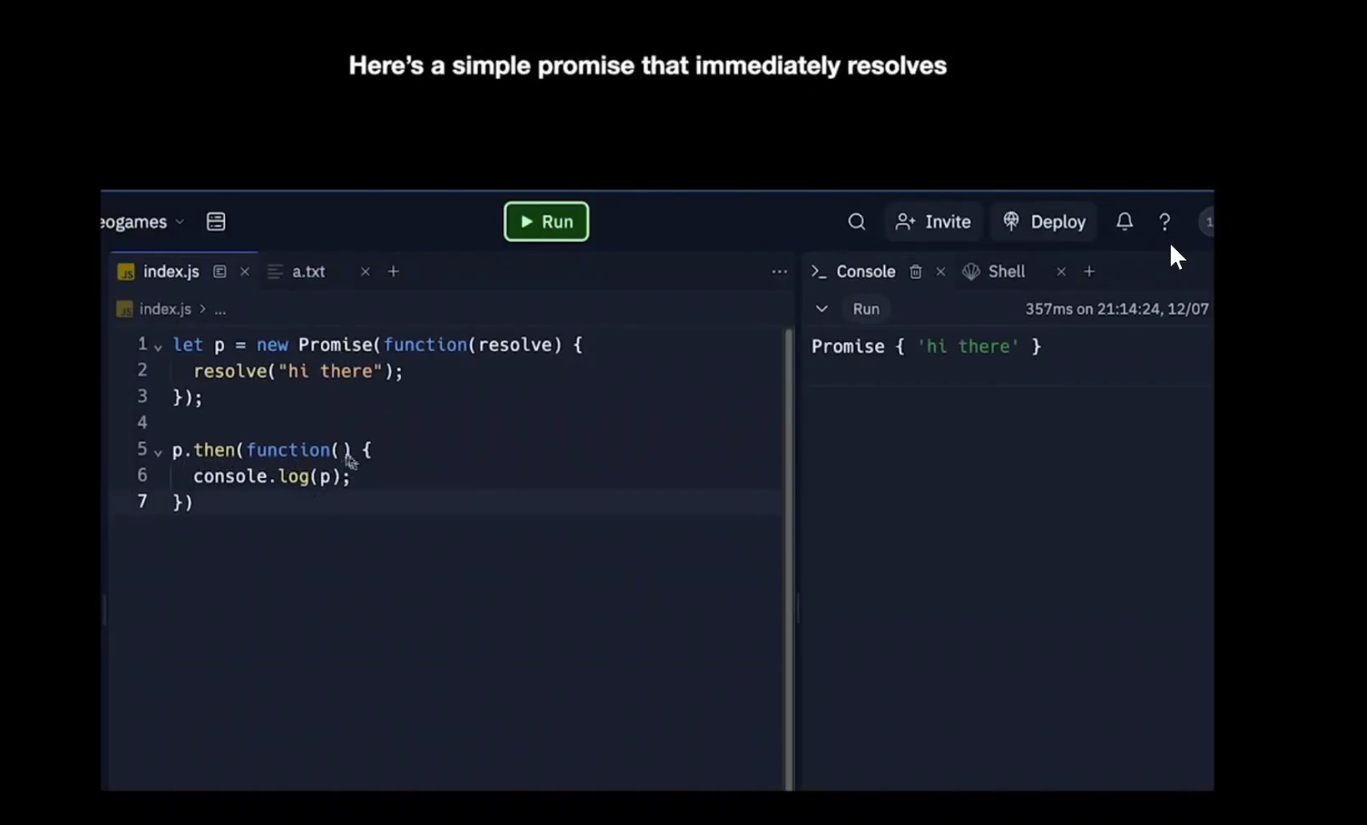
function onDone(data){

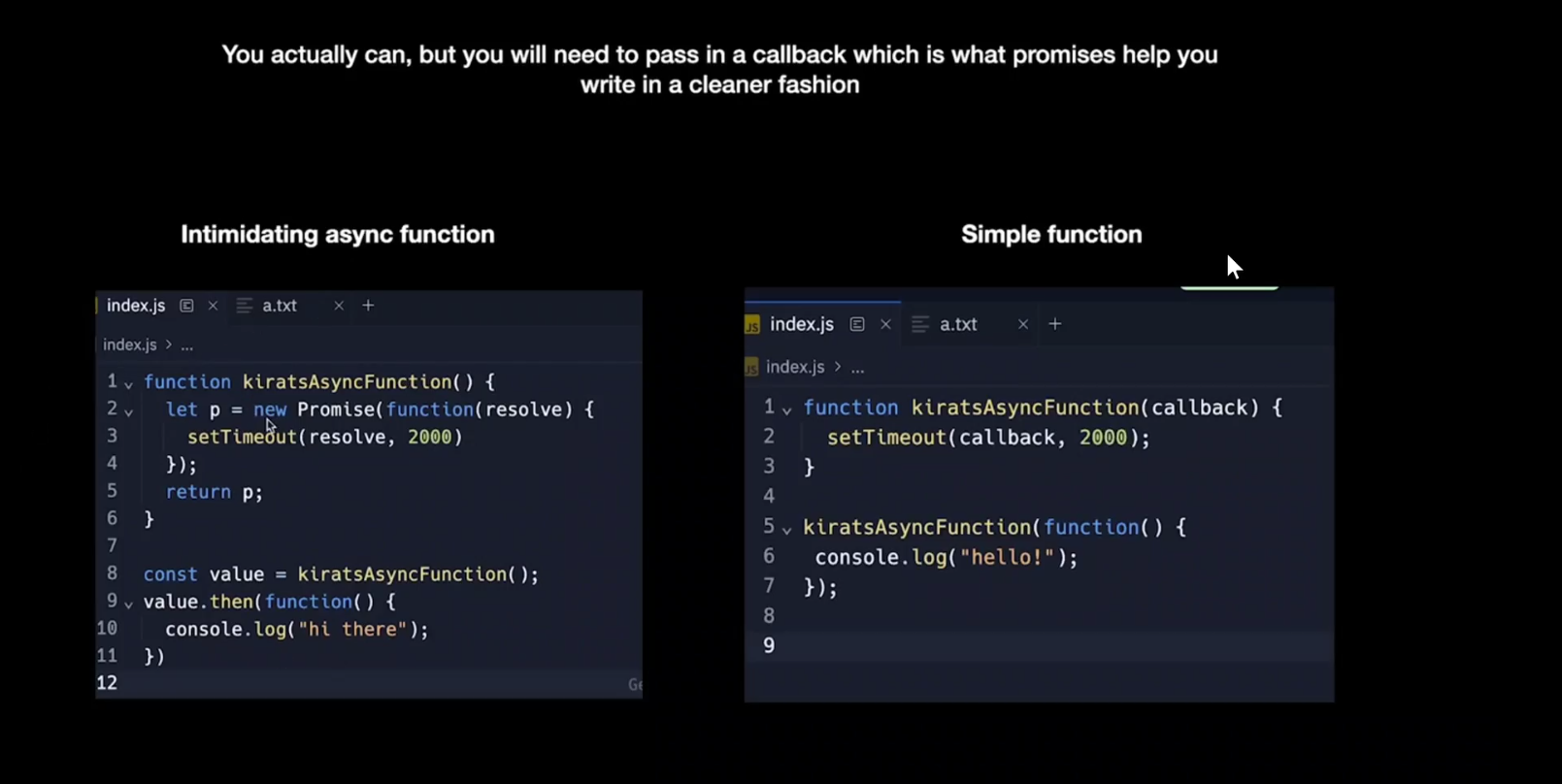
  console.log(data);

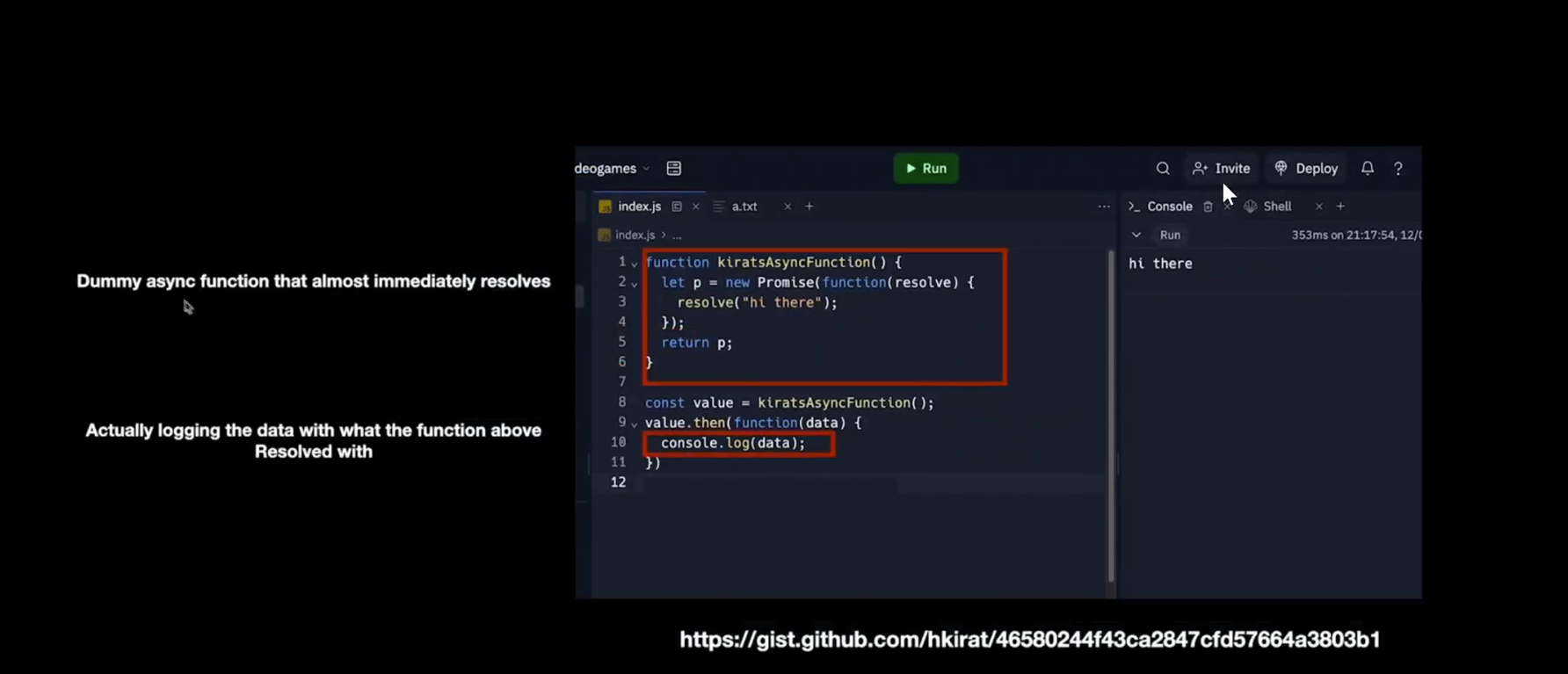
}

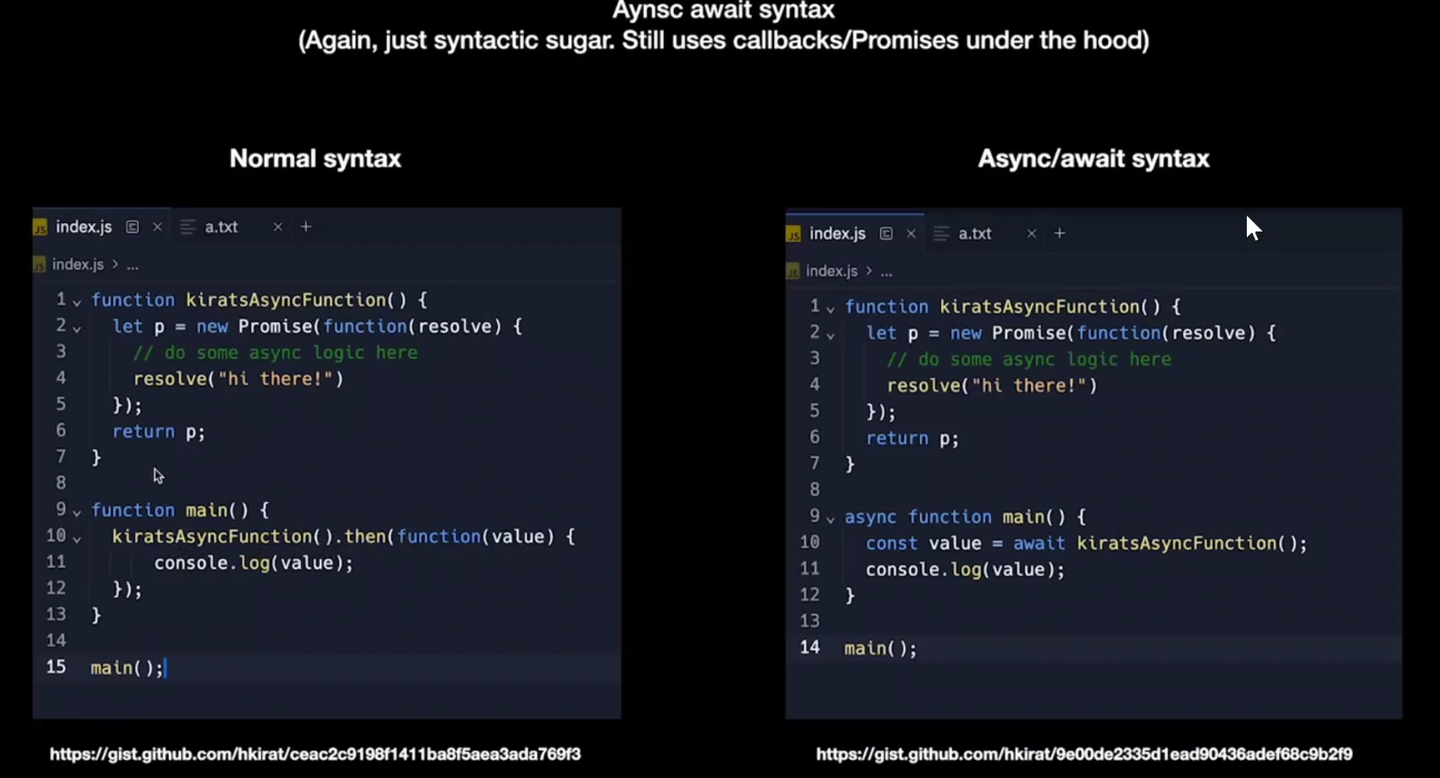
kiratReadFile(onDone);











==================================================================================async/await

Only async no await I used check below output both with async and await

function kiratAsyncFunction(){

  let p=new Promise(function(resolve){

    // do some async logic here

    setTimeout(function(){

      resolve("hi i am a superstar")

    },2000);

  });

  return p;

}

async function main(){

 let value= kiratAsyncFunction();

   console.log(value); // prints a promise object

}

main();

PS C:\Users\MARIMUTHU SANTHOSH\OneDrive\Documents\PRACTICE\Cohort2\class1> node main.js

Promise { <pending> }

function kiratAsyncFunction(){

  let p=new Promise(function(resolve){

    // do some async logic here

    setTimeout(function(){

      resolve("hi i am a superstar")

    },2000);

  });

  return p;

}

async function main(){

 let value= await kiratAsyncFunction();

   console.log(value); // prints a promise object

}

main();

PS C:\Users\MARIMUTHU SANTHOSH\OneDrive\Documents\PRACTICE\Cohort2\class1> node main.js

hi i am a superstar

Here it prints the output “hi I am a superstar”

