

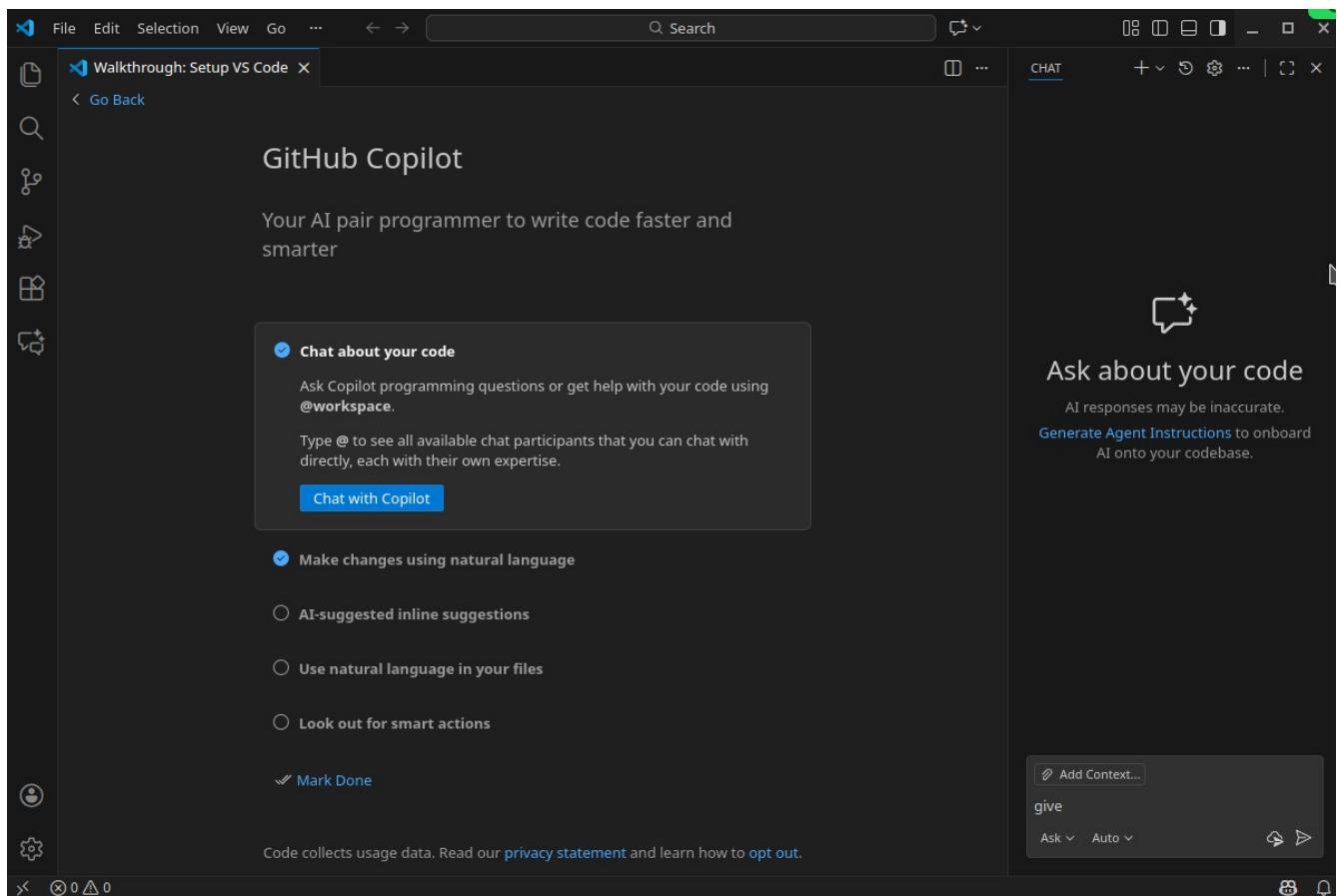
Lab 1 - Javascript Introduction

— Codepilot Setup in Visual Studio Code

Step 1. Download and Install Visual Studio Code

Visual Studio code is a lightweight code editor that we can use to write HTML and Javascript.

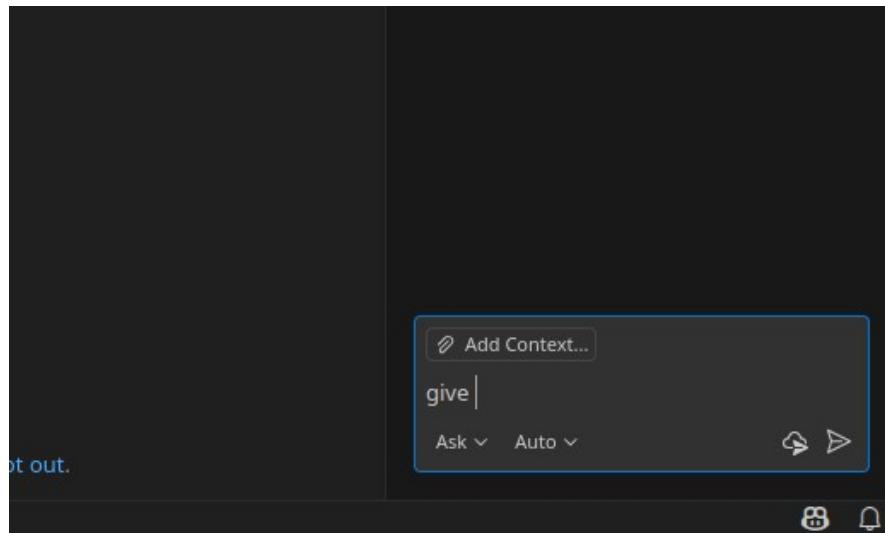
Once you open it, you will see the following tab window open. Click “Chat with Copilot”.



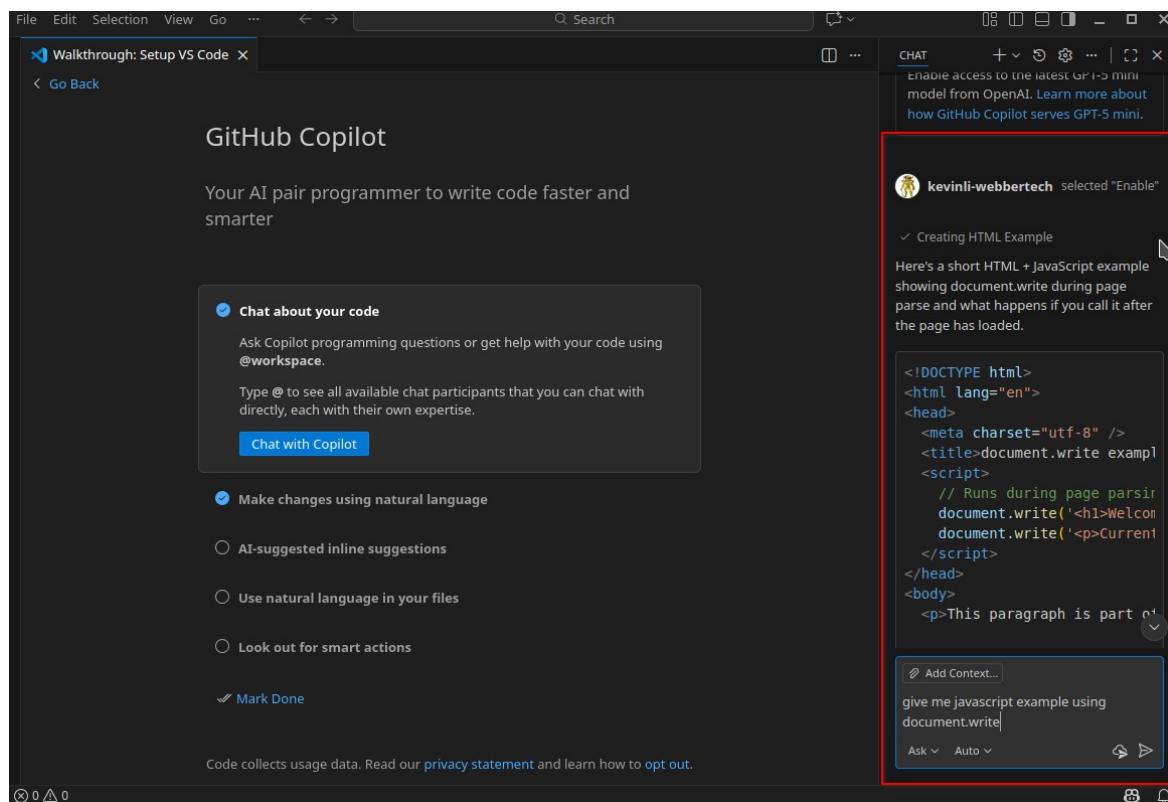
Step 2 Configure the ChatGPT or other AI Engine

Then next you will need to configure the chatGPT or some AI engine with an authentication method, such as gmail...etc.

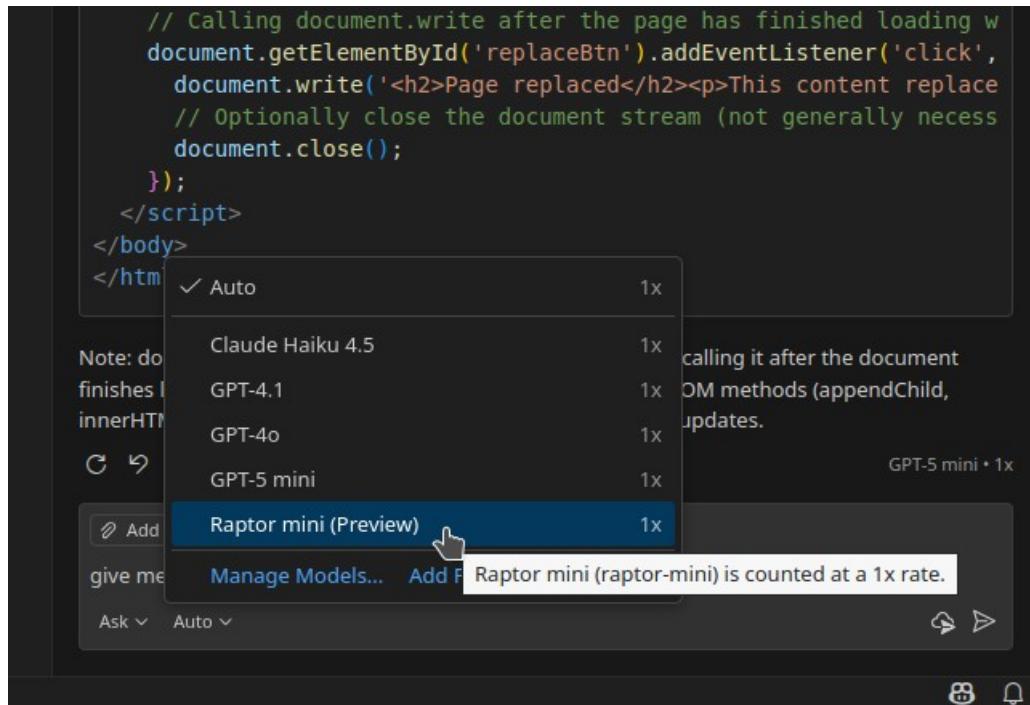
Once that is done, you will see a chat box like the following,



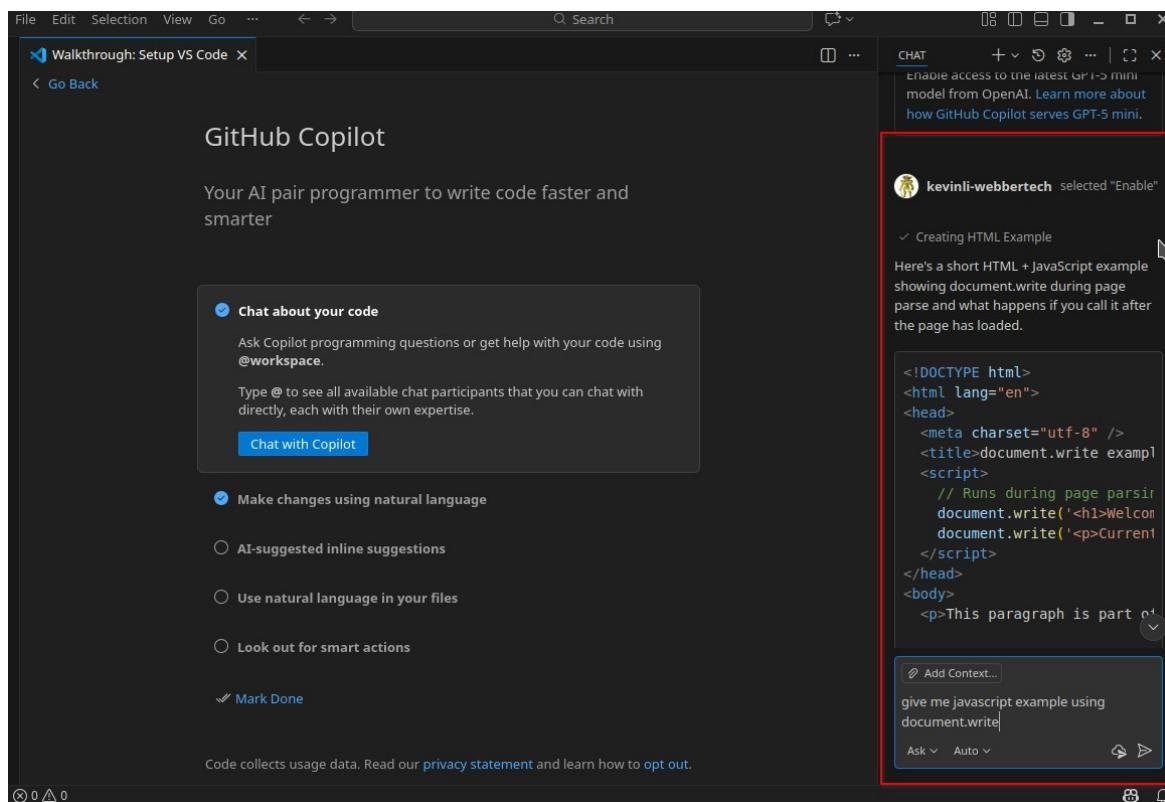
For our example, we could ask a question, and it will look like the following,



Click the auto dropdown menu in the following image, and you will see the menu shows you a few options,

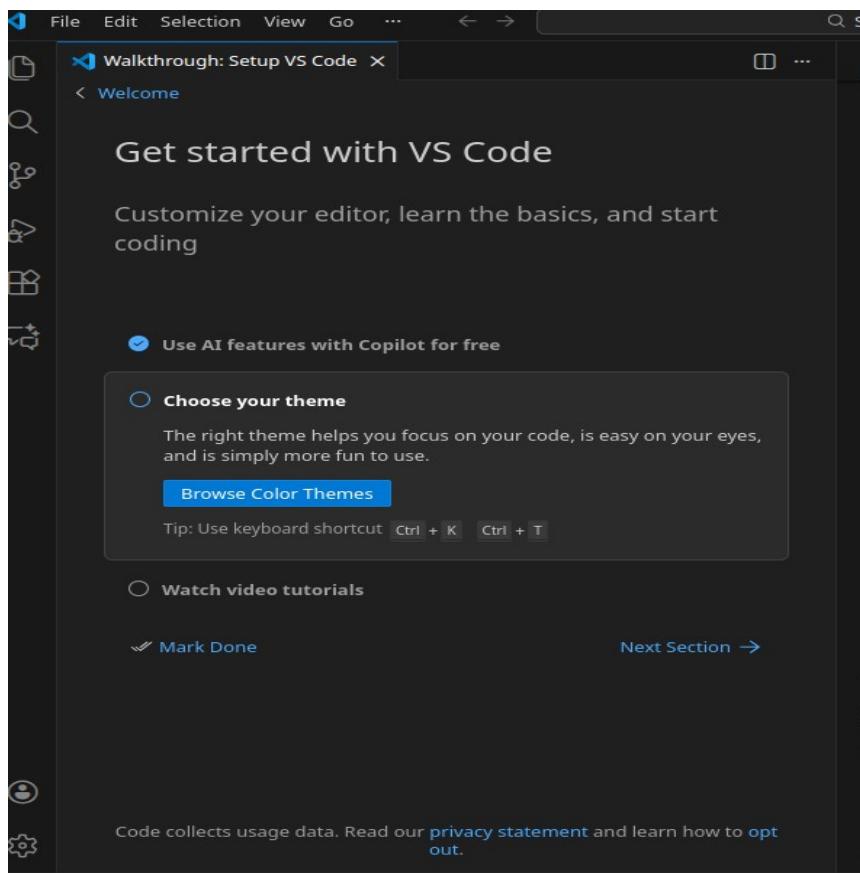


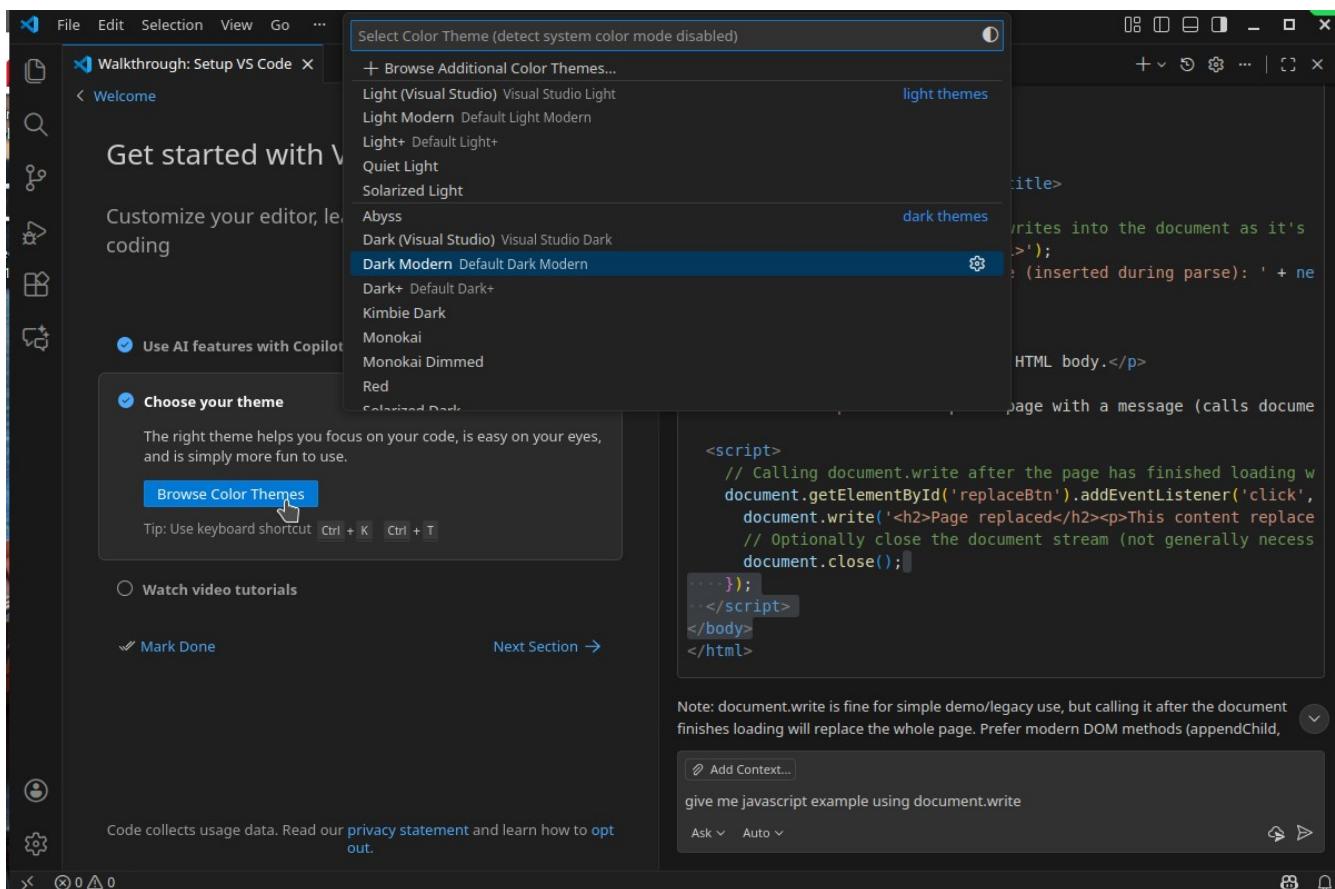
Check some other boxes in the following image, and click on “Mark Done”,



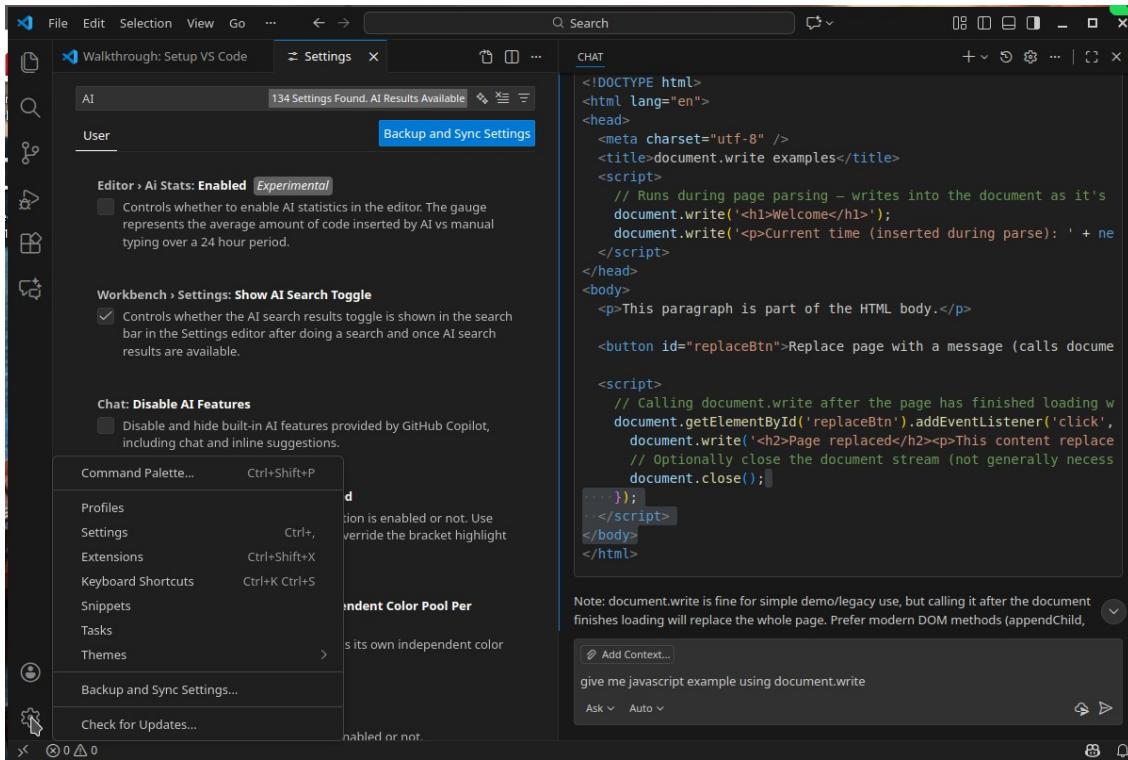
Walkthrough setup

Please configure the color theme if you like,

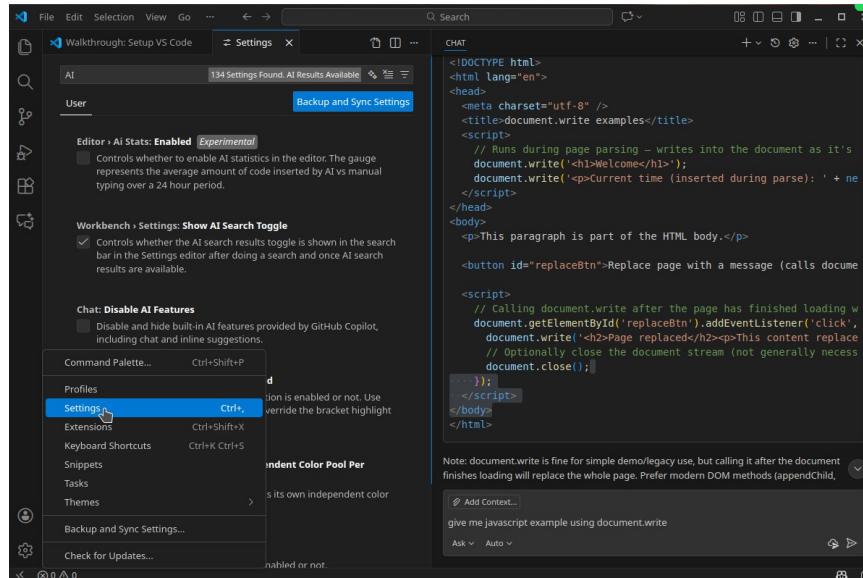




If you want to reconfigure the settings, such as the AI or the Theme, you can go to the “Setting”. It is a gear button in the bottom left.

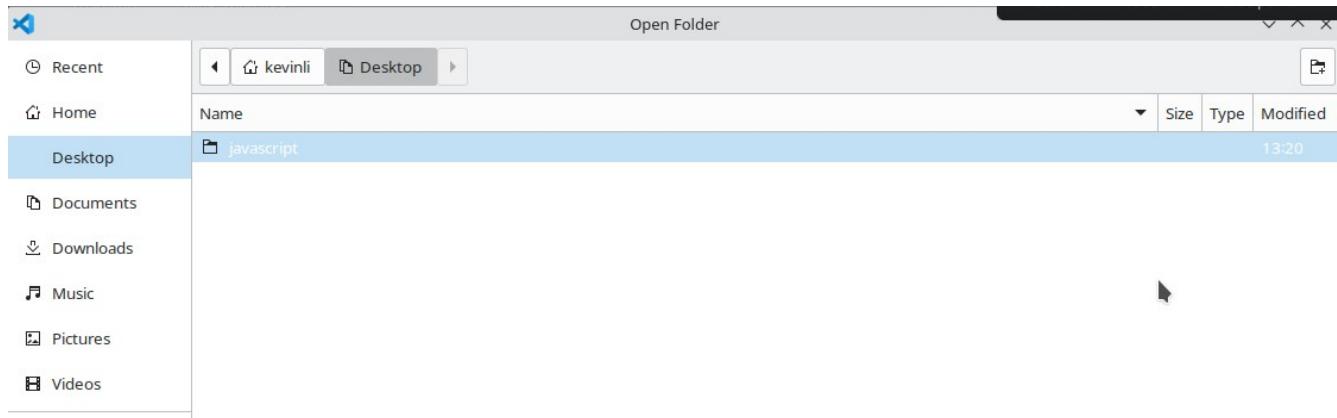


Click on the settings,

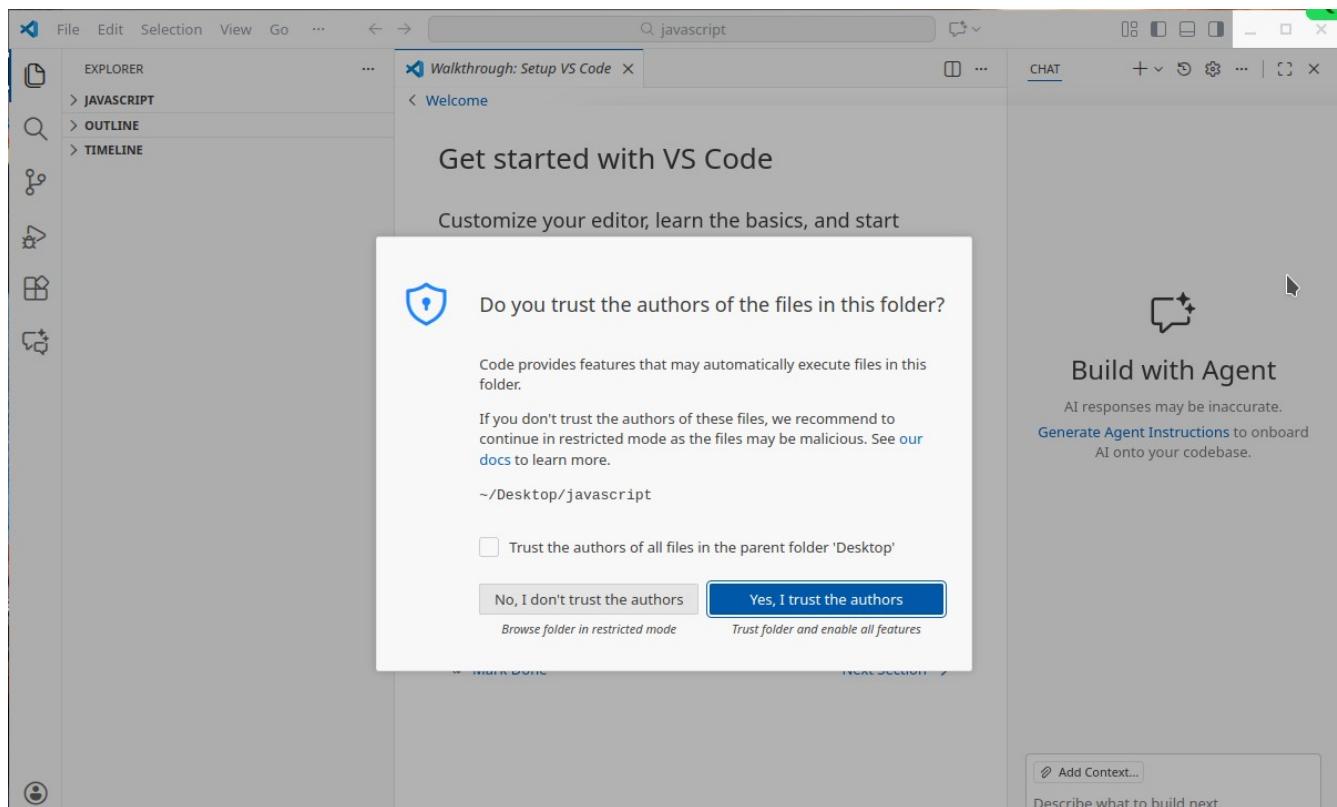


Step 3 Set up a directory for development

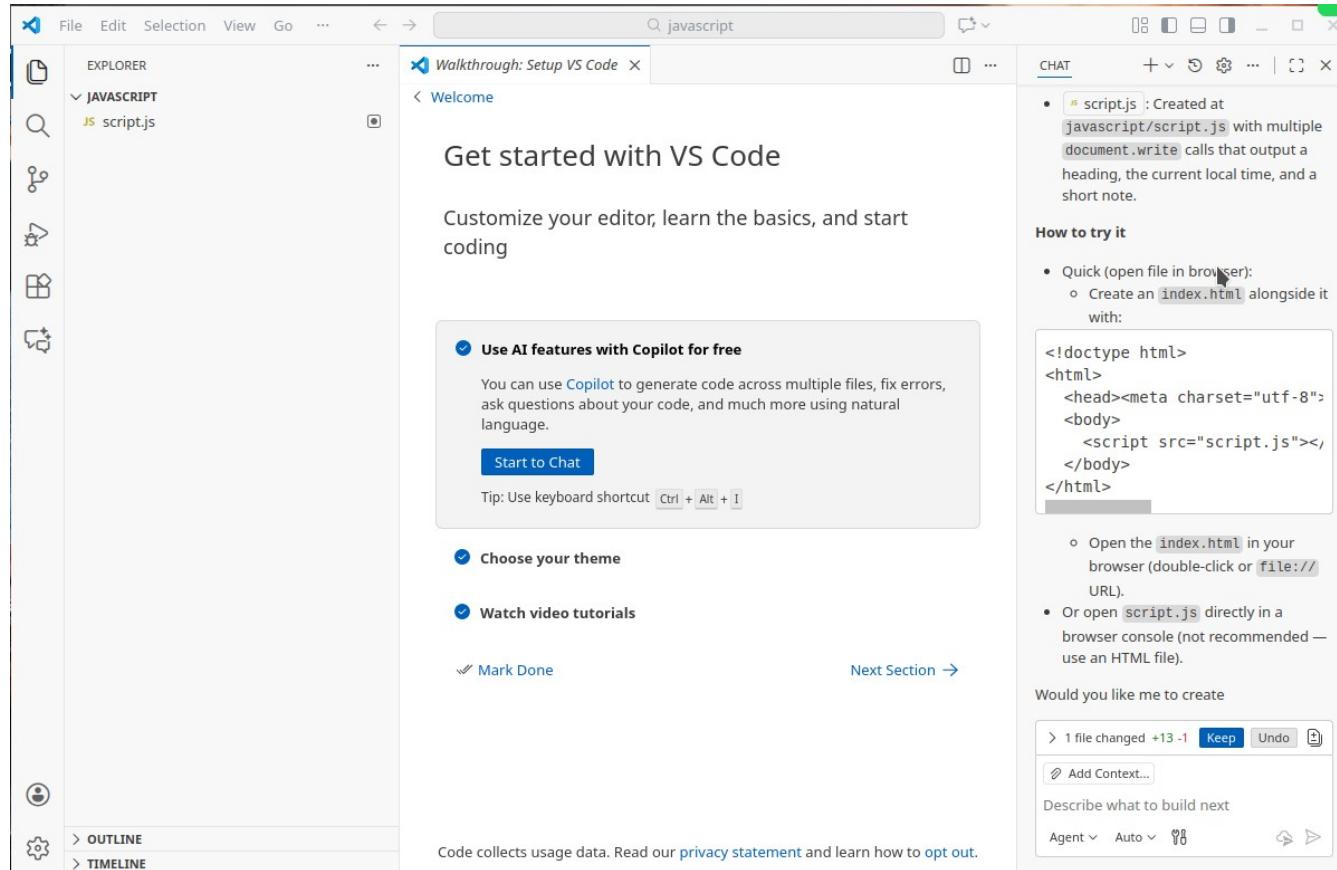
Next, create a directory in your home drive or desktop, in my case I call it “Javascript”, and go to “File” menu, and pick the Javascript directory you created, and you will see the following image.



Note that, I have changed my color scheme to light instead of dark.

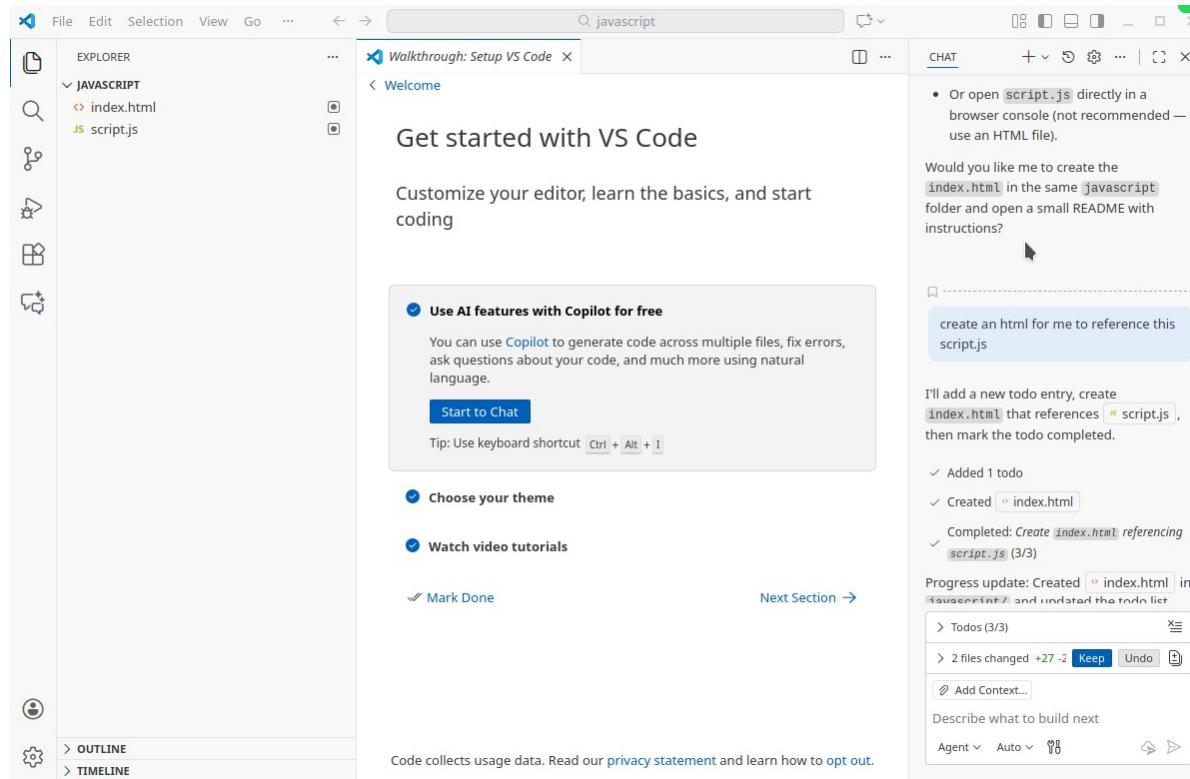


Next, in the chat box and ask questions to generate a Javascript file. We need not only to generate the

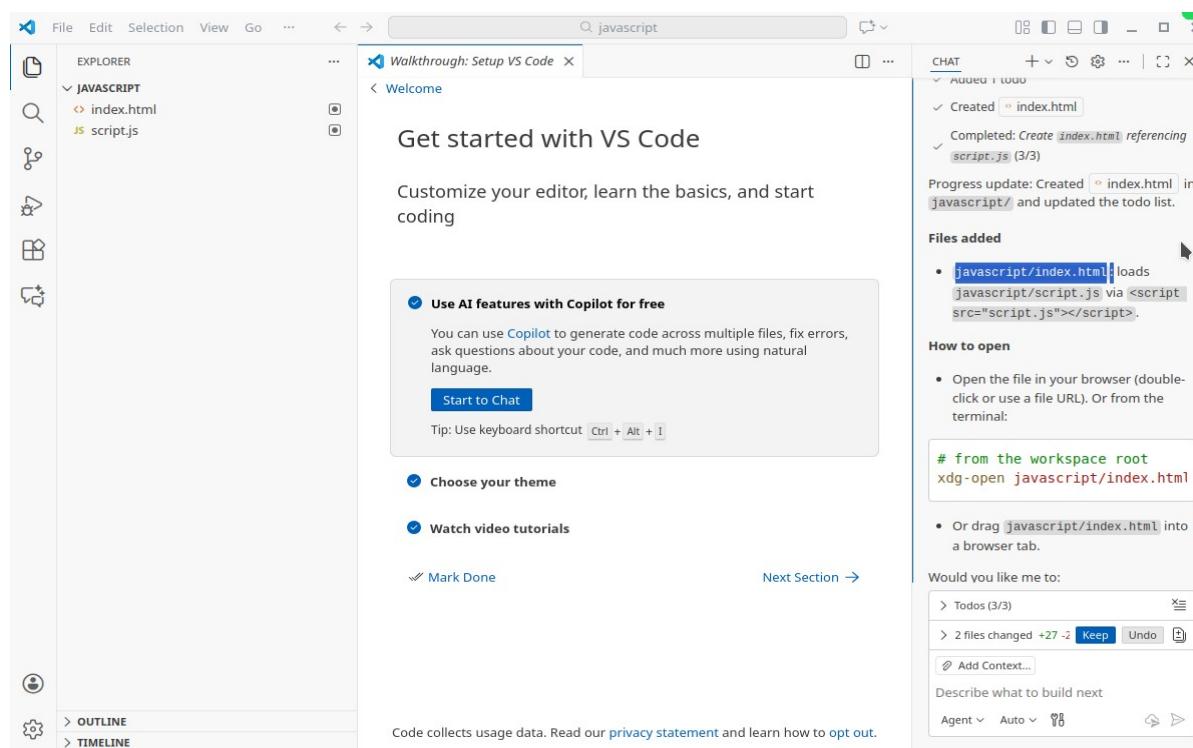


code but we also want to save it to my working directory, now in the above image, you can see script.js in the workspace in the left pane.

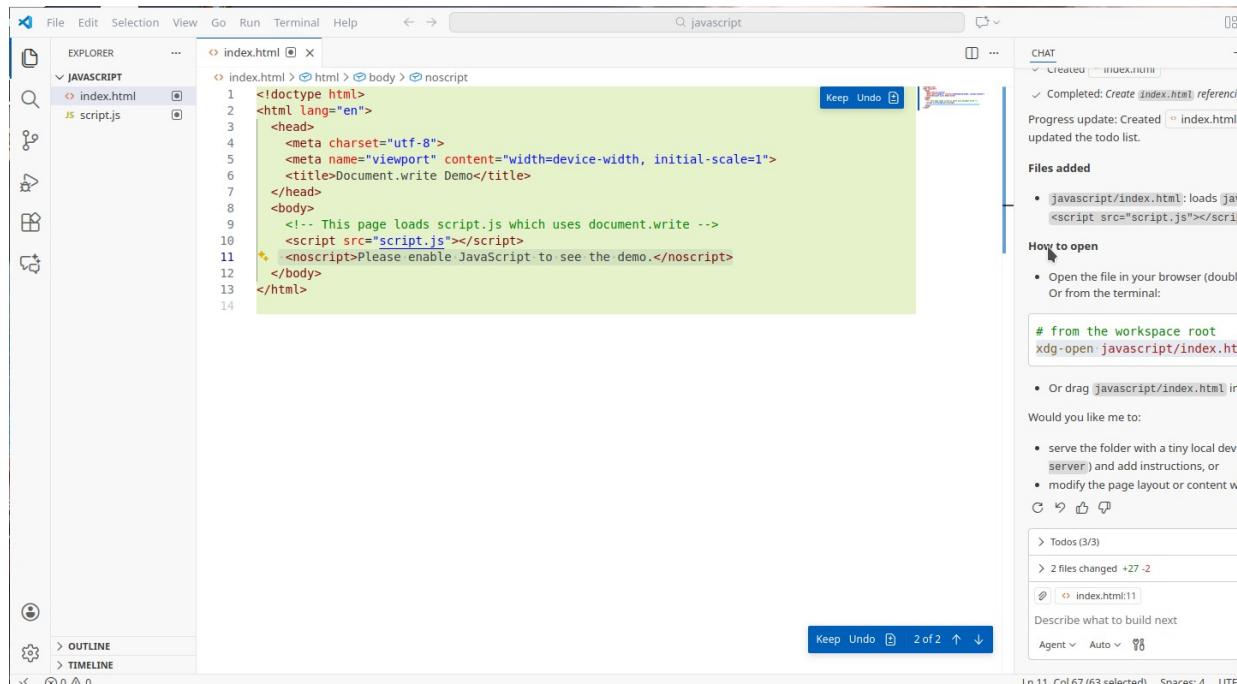
Now, I ask for another HTML to call/reference this script.js file. Now you can see two files on the left of the image blow,



On the right hand side where I highlighted, it also explains how to use it.



Now let us open the HTML file and take a look at how it writes the code. Now, this is a good way of learning and what is more, it saves your time, copy and paste from the web version. ChatGPT.

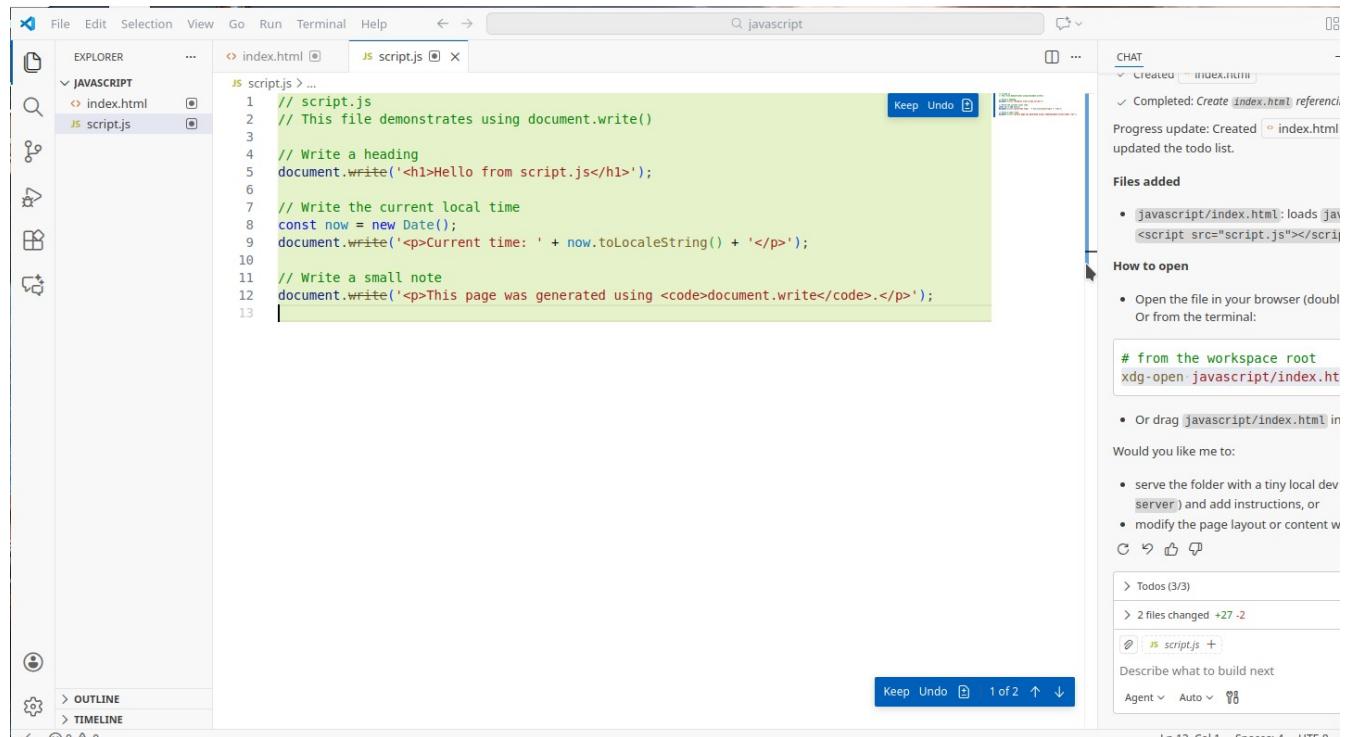


The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder named "JAVASCRIPT" containing "index.html" and "script.js".
- Editor:** The "index.html" file is open, displaying the following code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Document.write Demo</title>
  </head>
  <body>
    <!-- This page loads script.js which uses document.write -->
    <script src="script.js"></script>
    <noscript>Please enable JavaScript to see the demo.</noscript>
  </body>
</html>
```
- Output Panel:** Shows a "CHAT" section with a message about creating a reference to index.html. It also lists "Files added" and "How to open" instructions.
- Bottom Status Bar:** Shows file counts (11), terminal status (Col 67 / 62), and other UI elements.

Now the following is the Javascript file,



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder named "JAVASCRIPT" containing "index.html" and "script.js".
- Editor:** The "script.js" file is open, displaying the following code:

```
// script.js
// This file demonstrates using document.write()

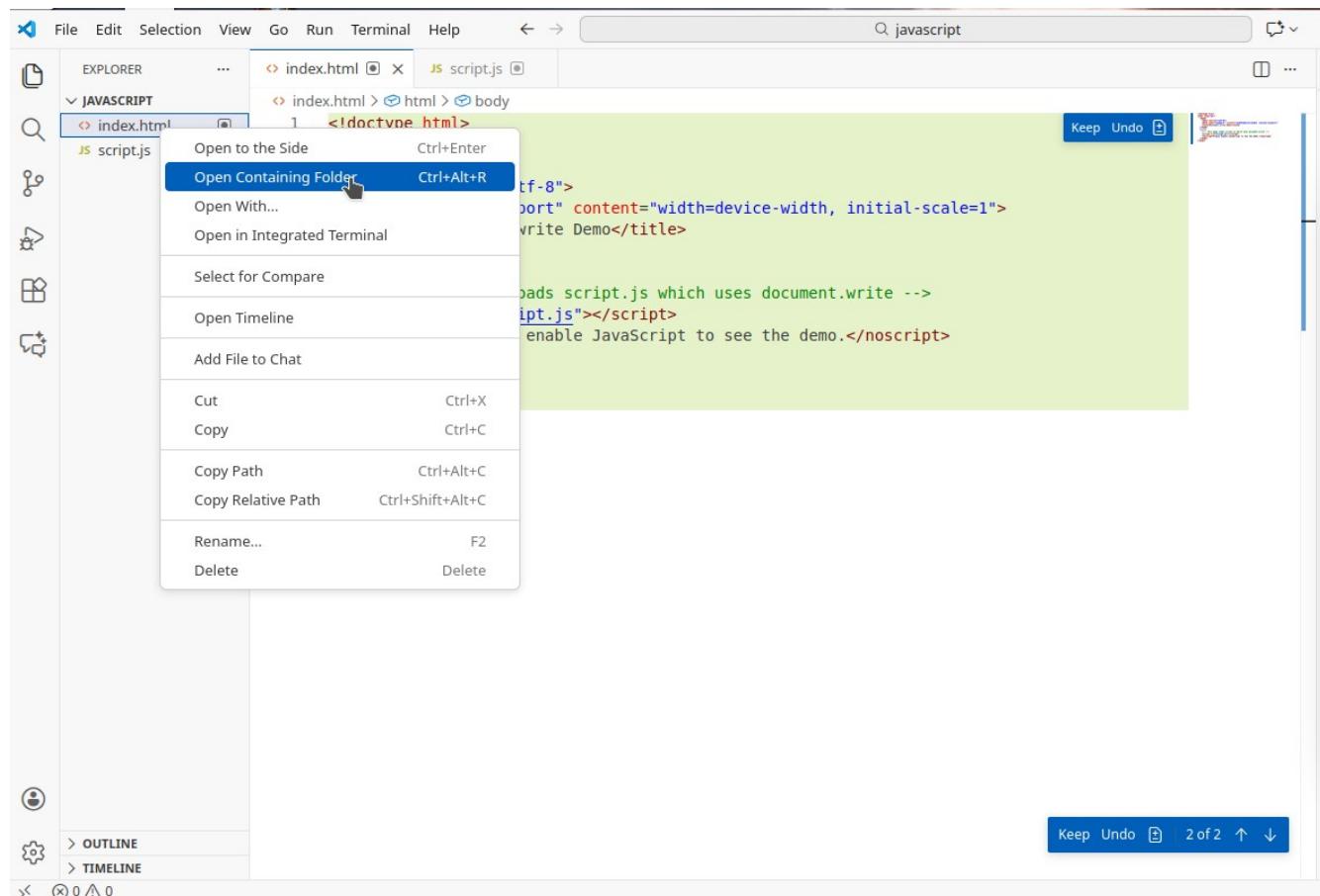
// Write a heading
document.write('<h1>Hello from script.js</h1>');

// Write the current local time
const now = new Date();
document.write('<p>Current time: ' + now.toLocaleString() + '</p>');

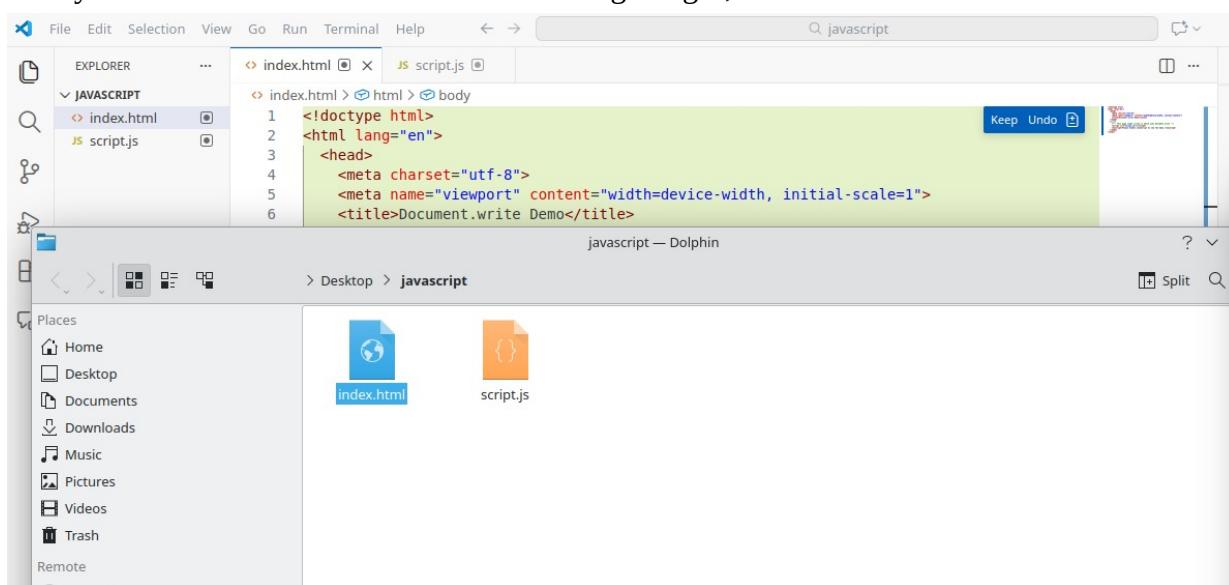
// Write a small note
document.write('<p>This page was generated using <code>document.write</code>.</p>');
```
- Output Panel:** Shows a "CHAT" section with a message about creating a reference to index.html. It also lists "Files added" and "How to open" instructions.
- Bottom Status Bar:** Shows file counts (12), terminal status (Col 1 / 1), and other UI elements.

Step 4 Test the Javascript file

From the left of the workspace in the image above, open it in its containing folder, and we will see two files here, shown in the following image.



Then you will see the files show in the following images,



Note that, you can only test the HTML file in the browser, and your Javascript file is being called and referenced in the html file.

So let us double click it or drag and drop it to a web browser, and you will see the html file open. In the following image, I put them side-by-side.

The screenshot shows the VS Code interface with two panes. On the left, the code editor displays `script.js` containing the following code:

```
// script.js
// This file demonstrates using document.write()

// Write a heading
document.write('<h1>Hello from script.js</h1>');

// Write the current local time
const now = new Date();
document.write('<p>Current time: ' + now.toLocaleString() + '</p>');

// Write a small note
document.write('<p>This page was generated using <code>document.write</code>.</p>');
```

On the right, a browser window shows the rendered output of the script:

Hello from script.js

Current time: 11/25/2025, 1:29:53 PM

This page was generated using `document.write`.

Now I ask another question to add another feature “add console.log example” to the javascript file.

The screenshot shows the VS Code interface with the AI Assistant extension open. The code editor contains the same `script.js` code as before. The AI panel on the right has a suggestion: "add console.log example into this javascript file".

Then the code was automatically generated on the left, shown in the following image,

The screenshot shows the VS Code interface with the code editor displaying the updated `script.js` file. The new code includes several `console.log` statements:

```
// script.js
// This file demonstrates using document.write()

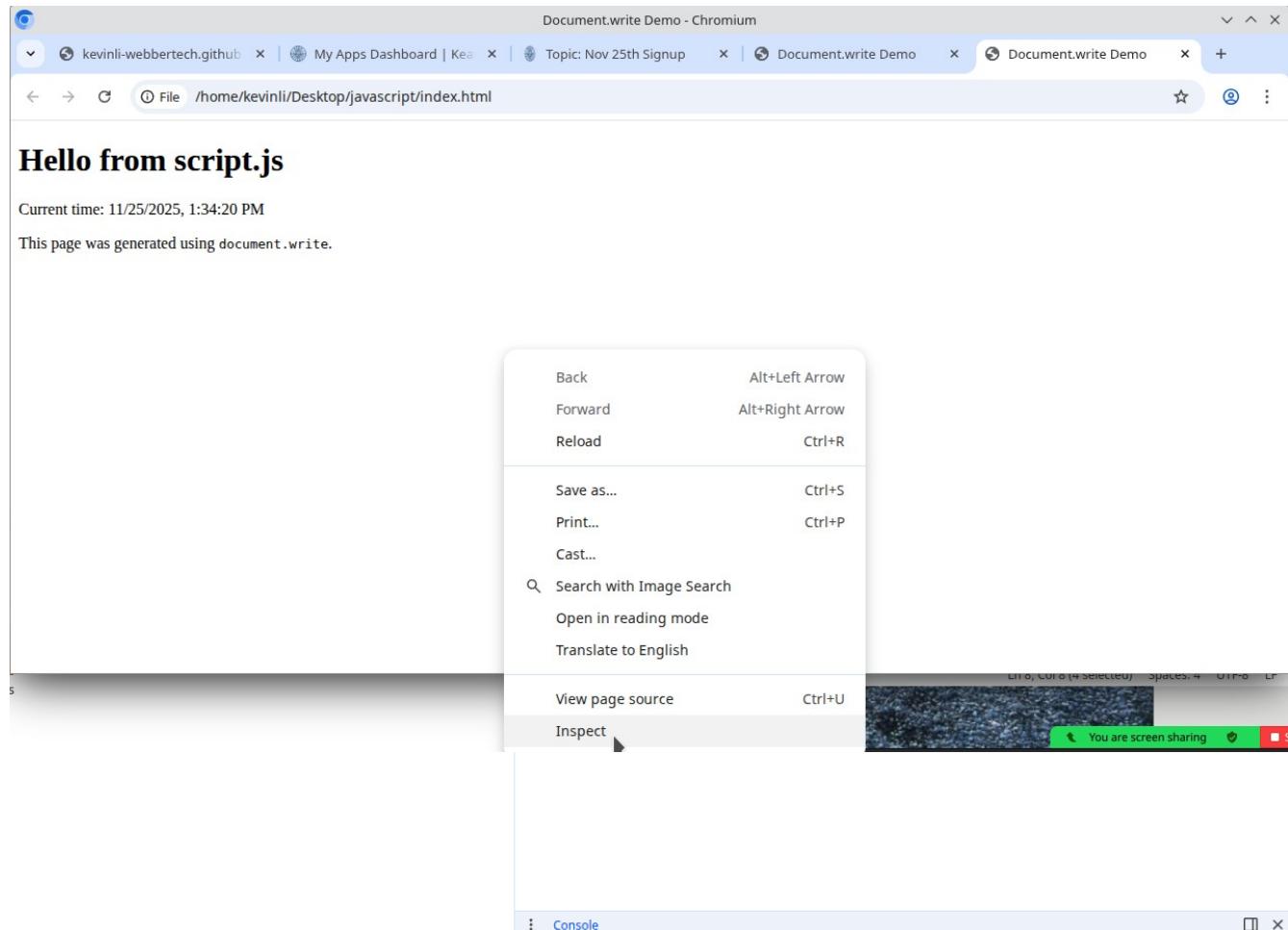
// Write a heading
document.write('<h1>Hello from script.js</h1>');

// Write the current local time
const now = new Date();
document.write('<p>Current time: ' + now.toLocaleString() + '</p>');

// Write a small note
document.write('<p>This page was generated using <code>document.write</code>.</p>');

// Console log examples
console.log('script.js loaded');
console.log('Current Date object:', now);
console.log('Current locale time:', now.toLocaleString());
console.log('Document title:', document.title);
console.log('Page URL:', location.href);
```

Open the web page in browser (all the browser supports the debugging), right click in the white area, and in the contextual menu, and click on the “Inspect” button,



Let us open the console tab on the right hand side of the following image,

