# **VS Code 與 .NET Core 開發環境設定**

## **安裝 .NET SDK 與 VS Code**

### **Windows/macOS/Linux 安裝步驟**

1. 下載並安裝最新版本的 .NET SDK 8 <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>。
2. 安裝VS Code編輯器 <https://code.visualstudio.com/download>。
3. 在 VS Code 安裝「C#」擴充套件（由 Microsoft 提供），支援語法高亮、IntelliSense、自動補全與偵錯。

3.1 安裝C# Dev Kit Extension



* Install C# Dev Kit (The [C# extension](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp)[t](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.vscodeintellicode-csharp), and the [.NET Runtime Install Tool](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.vscode-dotnet-runtime) will automatically be installed)

3.2安裝IntelliCode for C# Dev Kit

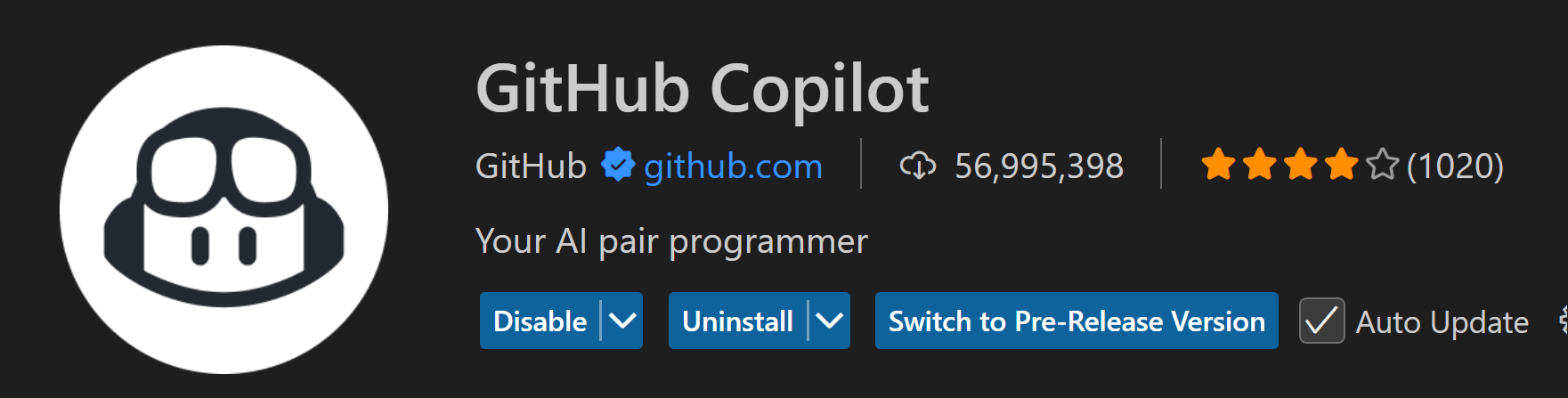


3.3 勾選Auto Save

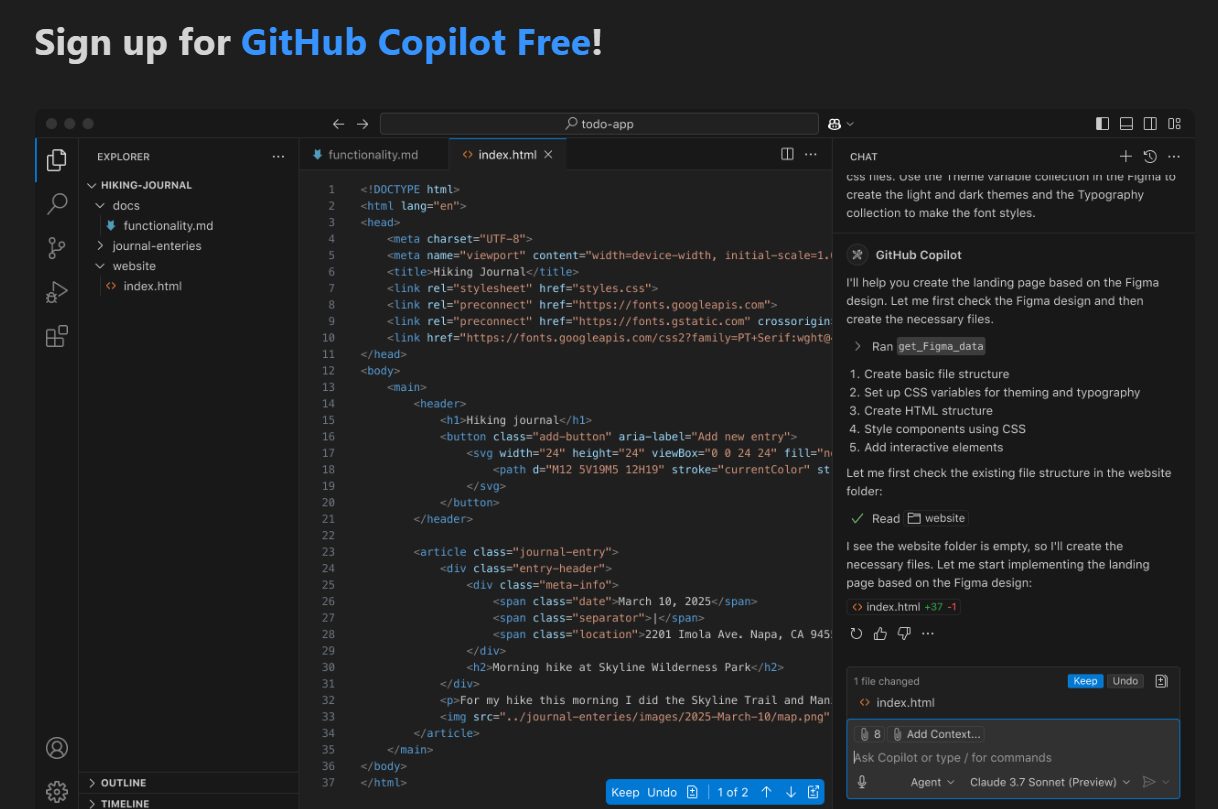


# **在 VS Code 設定 GitHub Copilot**

## **安裝與啟用步驟**

1. 安裝「GitHub Copilot」擴充套件及，啟用 AI 程式碼建議功能。
2. 登入 GitHub 帳號，授權 Copilot 使用。

Sign up for [GitHub Copilot Free](https://github.com/settings/copilot?utm_source=vscode-completions-readme&utm_medium=first&utm_campaign=2025mar-em-MSFT-signup)!



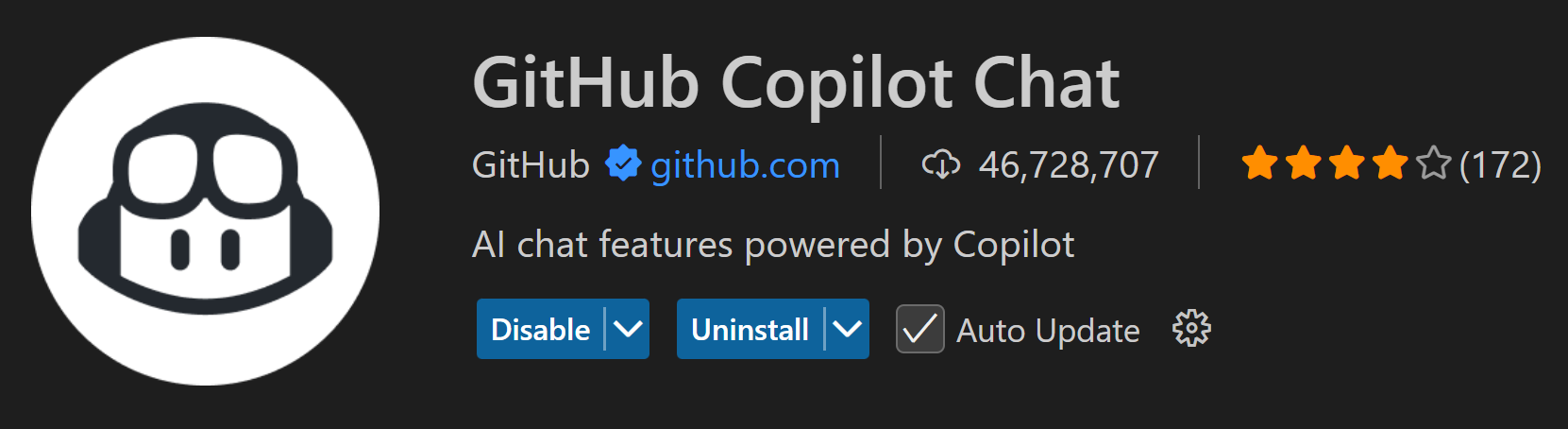
當你在 Visual Studio Code 安裝 Copilot 時，會獲得兩個擴充套件：

### **🧠 GitHub Copilot（此擴充套件）**

* 提供即時的程式碼建議，當你在編寫程式碼時，它會根據上下文自動補全或提出建議。
* 支援多種程式語言與框架，幫助你更快速地撰寫程式碼。

### **💬 GitHub Copilot Chat**

* 是 GitHub Copilot 的對話式 AI 夥伴。
* 你可以在 VS Code 的側邊欄中與它對話，詢問程式碼問題、除錯建議、最佳實踐等。
* 它能理解你的程式碼上下文，提供更深入的解釋與協助。



1. 開啟 C# 專案，開始撰寫程式碼時，Copilot 會即時產生建議，例如輸入 public class 或 public IActionResult，Copilot 會自動補全類別或 API 方法結構。

6.1 以系統管理員身份啟動Powershell



6.2 在根目錄建立程式資料夾 C:/Docker並設定ExecutionPolicy

PS C:\Windows\system32> cd c:/

PS C:\> mkdir docker

PS C:\> cd docker

PS C:\docker> Get-ExecutionPolicy

AllSigned

PS C:\docker> Set-ExecutionPolicy RemoteSigned

執行原則變更

執行原則有助於防範您不信任的指令碼。如果變更執行原則，可能會使您接觸到 about\_Execution\_Policies 說明主題 (網址為

https:/go.microsoft.com/fwlink/?LinkID=135170) 中所述的安全性風險。您要變更執行原則嗎?

[Y] 是(Y) [A] 全部皆是(A) [N] 否(N) [L] 全部皆否(L) [S] 暫停(S) [?] 說明 (預設值為 "N"): Y

PS C:\docker>

Get-ExecutionPolicyRemoteSigned

PS C:\docker>



6.3 VS CODE開啟資料夾C:\docker

