Write HTML in VSCode & push it to GitHub

### **1. Install Visual Studio Code**

* If you haven't installed VSCode, download and install it from [Visual Studio Code's official website](https://code.visualstudio.com/).

### **2. Set Up VSCode for HTML Development**

* **Install Extensions (Optional but Helpful)**:
  + Basic ones are included. See https://code.visualstudio.com/docs/languages/html
  + Recommended ones
    - "Live Server" extension to preview your HTML files in a browser in real-time.
    - HTML CSS Support: Enhances your workflow by integrating CSS support directly into your HTML files. This is especially helpful in projects where you are constantly switching between HTML and CSS.
    - Auto Close Tag and Auto Rename Tag: Both are essential for writing cleaner and more error-free HTML code.
    - Prettier - Code Formatter: If you don’t already use a code formatter, Prettier is an excellent choice for keeping your HTML (and other code) consistently formatted.
    - HTML Boilerplate: This can be a nice convenience if you frequently start new HTML documents from scratch.

### **3. Create a New HTML File**

* Open VSCode.
* Create a new file by clicking on File > New File, or press Ctrl + N.
* Save the file as index.html by clicking on File > Save As, or press Ctrl + S, and then give it a .html extension.

### **4. Write Basic HTML Code**

* Start by writing some basic HTML code in your index.html file. Here’s a simple example:

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My First Webpage</title>

</head>

<body>

<h1>Hello, World!</h1>

<p>This is my first HTML page.</p>

</body>

</html>

### **5. Preview Your HTML in a Browser (Optional)**

* If you installed the "Live Server" extension, you can right-click your index.html file in VSCode and select "Open with Live Server". This will launch your HTML file in a web browser and update automatically as you make changes.

### **6. Initialize a Git Repository**

* Open the terminal in VSCode by going to View > Terminal, or press Ctrl + ` .
* Navigate to the directory where your HTML file is located.

Initialize a Git repository with the following command:  
git init

* This creates a new Git repository in your project folder.

### **7. Stage and Commit Your Files**

Add your HTML file to the staging area:  
git add index.html

Commit your changes:  
git commit -m "Initial commit with index.html"

### **8. Create a New Repository on GitHub**

* Go to [GitHub.com](https://github.com/) and log in.
* Click on the "+" icon at the top right of the page and select "New repository".
* Name your repository, add an optional description, and choose whether it should be public or private.
* Do **not** initialize the repository with a README, .gitignore, or license if you're pushing an existing project from your computer.
* Click "Create repository".

### **9. Push Your Code to GitHub**

#### Link your local repository to the GitHub repository

git remote add origin https://github.com/xiaoshuqian/skills.git

#### Set up a personal access token (classic)

To create a personal access token (PAT) for GitHub, you can do the following:

* Go to your profile photo in the upper-right corner of any GitHub page
  + Select Settings
* Click Developer settings in the left sidebar
* Under Personal access tokens, click Tokens (classic)
* Select Generate new token
* In the Note field, give your token a name
* To set an expiration date, select Expiration and choose a default option or enter a date
* Select the scopes you want to grant to the token (public\_repo)
* Click Generate token
* GitHub will only show the token once, so it's important to copy it and store it in a safe place.
* Paste the token to 1password

A PAT is a way for developers to securely interact with GitHub's platform using the command line or the GitHub API. Fine-grained PATs offer more security advantages than classic PATs because they can only access specific repositories and resources owned by a single user or organization.

#### Push your code to GitHub

git push https://xiaoshuqian@github.com/xiaoshuqian/skills.git main  
git push -u origin main # -u: = --set-upstream, sets the main branch in this local machine to the main branch on the remote, i.e. origin, such that in the future, you can use git push/pull w/o specifying the branch name

### **10. Access Your HTML Page on GitHub**

* After pushing your code to GitHub, you can view your repository by going to https://github.com/yourusername/your-repository-name.
* You’ll see the index.html file listed there. If you want to host this page using GitHub Pages, follow these additional steps:

### **11. Set Up GitHub Pages (Optional)**

* Go to the "Settings" tab of your repository on GitHub.
* Scroll down to the "GitHub Pages" section.
* Under "Source", select the branch you want to use (usually master or main).
* GitHub will provide you with a URL where your HTML page will be hosted, typically https://yourusername.github.io/your-repository-name/.

Now you can share the URL with others to view your webpage online!