

# 新手使用github（vscode）

## 🌟 从零开始：Git + VSCode + GitHub 最完整的新手友好教程

包括：

- ✓ 第一次创建并上传项目（最完整流程）
  - ✓ 之后如何更新已有项目（每次更新流程）
  - ✓ 每一步点哪里 + 为什么（背后原理）
- 

### 整体分两部分：

Part A — 第一次把一个新项目上传到 GitHub（init → commit → push）

Part B — 后续更新项目（commit → push）

---

## Part A：第一次上传项目到 GitHub（新手全流程）

你从零开始，需要做：

1. GitHub 创建远程仓库
  2. VSCode 初始化本地仓库
  3. 连接 GitHub（设置 remote）
  4. Commit（保存版本）
  5. Push（上传）
-



## 步骤 1：在 GitHub 上创建一个仓库（repo）

1. 打开 GitHub → 登录
2. 右上角 + → **New repository**
3. 填写：
  - Repository name: `hpc_ode_sde_project`
  - Public / Private 都可以
  - README 不要勾选（为了避免冲突）
4. 点击 **Create Repository**

你现在得到了一个空的 GitHub 仓库。

GitHub 会显示一段像这样的话：

```
git remote add origin https://github.com/.../hpc_ode_sde_project.git
git branch -M main
git push -u origin main
```

等会儿我们要用。



## 步骤 2：打开 VSCode → 打开你的项目文件夹

点：

```
File → Open Folder
```

选择你的 HPC 项目根目录。



## 步骤 3：初始化 Git 仓库（如果没有）

方法：

1. 左侧点 Source Control 图标（树枝状）
2. 如果没有 Git 仓库，会有按钮：

Initialize Repository

点它。

📌 VSCode 自动执行：

```
git init
```

你的项目就变成一个本地 Git 仓库了。



## （原理解释）

`git init` = 创建一个隐藏的 `.git/` 文件夹

它是 **本地版本库**，用来记录所有未来的 commit。



## 步骤 4：把所有文件加入版本控制

在左侧 Source Control 面板中：

- 所有文件都是 **U (Untracked)**

在顶部 message 输入框写：

```
Initial commit - upload HPC Numerical Engine project
```

然后点击 **Commit (✓)**

VSCode 会问：

| Stage all changes?

点击 **Yes**

VSCode 自动执行：

```
git add .  
git commit -m "Initial commit - upload HPC Numerical Engine project"
```

📌 意味着：

所有文件已经被 Git 记录下来。

## 步骤 5：连接你的 GitHub 仓库（添加 remote）

打开 VSCode 内置终端（Terminal → New Terminal）：

粘贴 GitHub 给你的这条命令：

```
git remote add origin https://github.com/你的账户/hpc_ode_sde_project.git  
git branch -M main
```

## 步骤 6：第一次 push

继续在终端输入：

```
git push -u origin main
```

第一次 push 会要求你：

- 登录 GitHub
- 输入 GitHub Token（不是密码）

如果你需要，我可以帮你生成 token。

如果 push 成功，你会看到：

```
Enumerating objects...  
Total ...  
To https://github.com/....
```

刷新 GitHub

🎉 你的项目已经成功展示在 GitHub 上！

---

## 🎉 Part B：后续更新项目（非常简单）

你以后每次修改代码，比如新加 solver、加单元测试、加 GPU 版本，只需要三步：

---

### 🟠 步骤 1：修改文件 → VSCode 左侧出现“M”

VSCode Git 会标记你所有修改过的文件：

- 橘色 M = Modified
- 绿色 A = Added
- 灰色 U = 新文件

代表 Git 检测到了你的更改。

---

### 🟢 步骤 2：写 Commit message → 点击 Commit

在顶部 Message 写：

```
Add Milstein integrator + Adaptive ODE solver
```

点击右上角 ✓ Commit

VSCode 自动执行：

```
git add .  
git commit -m "Add Milstein integrator + Adaptive ODE solver"
```

---

## 步骤 3：点击 Sync Changes（推送到 GitHub）

Source Control 界面的：

 **Sync Changes**

VSCode 自动执行：

```
git pull  
git push
```

如果没有冲突 → GitHub 立即更新。

## 更新项目只需要这三步：

```
写 message  
→ commit  
→ push (sync)
```

最快只需要 3 秒钟。

## 关键理解总结（让你以后不会迷路）

操作	VSCode 按钮	Git 命令	作用
保存版本	Commit	git commit	保存到本地仓库
上传版本	Sync / Push	git push	上传到 GitHub
标记文件	自动	git add	把文件加入提交

核心原则：

★ 没有 commit，就不会有 push

★ 没有 message，就不会有 commit

★ Sync Changes = push + pull

你之前 GitHub 上空，是因为你没有 commit，而不是 push 错了。

---



## 你现在已经掌握了 Git 最完整的工作流！

你已经会：

- 初始化 git
  - 连接 GitHub
  - commit（保存）
  - push（上传）
  - 更新项目
  - 使用 VSCode 图形界面管理 git
-