

通过CMD或PowerShell使用Git进行家里和公司代码同步的详细步骤

如果不用用 VS Code，直接通过 **CMD 或 PowerShell** 操作 Git 同步代码，流程与命令行操作一致，核心是通过 Git 命令完成拉取、提交、推送。以下是详细步骤：

一、回到家后，将远程代码同步到家里的电脑

场景 1：家里电脑第一次操作这个项目（需要先克隆仓库）

1. 打开 PowerShell 或 CMD，进入你想存放项目的目录（例如 D:\Projects）：

```
cd D:\Projects # 切换到目标目录
```

2. 克隆远程仓库到本地（复制远程仓库地址，如 GitHub/Gitee 的 HTTPS 地址）：

```
git clone https://github.com/你的用户名/仓库名.git # 替换为你的远程仓库地址
```

3. 进入克隆后的项目目录：

```
cd 仓库名 # 仓库名与远程仓库名称一致
```

场景 2：家里电脑已有项目（直接拉取最新代码）

1. 打开 PowerShell 或 CMD，进入项目目录：

```
cd E:\Flutter_Projects\video_demo # 替换为你的项目路径
```

2. 确保当前分支是主分支（如 main）：

```
git branch # 查看当前分支，前面带 * 的是当前分支
```

若不在 main 分支，切换到 main：

```
git checkout main
```

3. 拉取远程最新代码（同步公司提交的修改）：

```
git pull origin main
```

- 首次拉取可能需要输入远程仓库的账号密码（或 GitHub 令牌），输入后即可拉取。

二、在家修改代码后，推送到远程仓库

1. 进入项目目录（同上）：

```
cd E:\Flutter_Projects\video_demo
```

2. 查看修改的文件（可选，确认修改内容）：

```
git status # 红色文件为未暂存的修改
```

3. 暂存所有修改（将修改加入提交队列）：

```
git add . # . 表示暂存当前目录所有修改
```

（如果只想暂存单个文件，替换为 `git add 文件名`）

4. 提交修改到本地仓库（填写提交信息，描述修改内容）：

```
git commit -m "在家修改：添加视频播放逻辑" # 替换为你的提交信息
```

5. 推送本地修改到远程仓库（供公司电脑同步）：

```
git push origin main
```

三、回到公司后，同步家里推送的代码

流程与「家里同步公司代码」完全一致：

1. 打开公司电脑的 CMD 或 PowerShell，进入项目目录：

```
cd 公司电脑的项目路径 # 例如 D:\Work\video_demo
```

2. 确保当前分支是 `main`:

```
git checkout main
```

3. 拉取远程最新代码（包含家里的修改）：

```
git pull origin main
```

核心命令总结

操作场景	命令	说明
首次获取项目	<code>git clone 远程地址</code>	克隆远程仓库到本地
切换到主分支	<code>git checkout main</code>	确保在主分支操作
拉取远程最新代码	<code>git pull origin main</code>	同步远程修改到本地
暂存修改	<code>git add .</code> 或 <code>git add 文件名</code>	准备提交的文件
提交到本地仓库	<code>git commit -m "提交信息"</code>	保存本地修改
推送本地修改到远程	<code>git push origin main</code>	同步本地修改到远程

按以上命令操作，无需 VS Code 也能轻松实现家里和公司的代码同步，关键是记住「拉取前先确认分支，修改后及时提交推送」。

（注：文档部分内容可能由 AI 生成）