# Git与Github使用笔记

1. Git简介

Git是一个分布式版本控制系统，由Linus Torvalds创建，用于有效、高速地处理从小到大的项目版本管理。

2. 安装Git

- 访问[Git官网](https://git-scm.com/)下载并安装Git。

3. 配置Git

- 设置用户名和邮箱，这将用于提交记录：

```

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

```

4. 创建仓库

- 在本地创建一个新的Git仓库：

```

mkdir my-project

cd my-project

git init

```

5. 添加文件

- 添加文件到仓库：

```

git add <file1> <file2> ...

```

6. 提交更改

- 提交更改到仓库：

```

git commit -m "Commit message"

```

7. 查看状态

- 查看仓库状态：

```

git status

```

8. 查看日志

- 查看提交日志：

```

git log

```

9. 分支管理

- 创建新分支：

```

git branch <branch-name>

```

- 切换分支：

```

git checkout <branch-name>

```

- 合并分支：

```

git merge <branch-name>

```

10. 远程仓库

- 添加远程仓库：

```

git remote add origin <repository-url>

```

- 推送到远程仓库：

```

git push -u origin <branch-name>

```

11. 克隆远程仓库

- 克隆远程仓库到本地：

```

git clone <repository-url>

```

12. 拉取更新

- 从远程仓库拉取最新更改：

```

git pull

```

GitHub使用笔记

1. 注册GitHub账号

- 访问[GitHub官网](https://github.com/)注册账号。

2. 创建仓库

- 在GitHub上创建一个新的仓库。

3. 连接本地仓库

- 使用`git remote add origin`命令将本地仓库与GitHub仓库连接。

4. 推送代码

- 将本地代码推送到GitHub仓库。

5. 拉取请求（Pull Request）

- 当你想要为别人的项目贡献代码时，可以创建一个Pull Request。

6. 问题跟踪（Issues）

- 使用Issues跟踪项目中的问题或功能请求。

7. 协作

- 通过GitHub的协作功能与他人共同开发项目。

8. GitHub Pages

- 使用GitHub Pages发布静态网站。

9. GitHub Actions

- 自动化你的软件工作流程。

10. 探索GitHub

- 使用GitHub探索其他项目，学习优秀的代码实践。