

- **Git 新手快速上手指南**
 - 一、安装 Git
 - 二、检查 Git 是否安装成功
 - 三、必要的配置（只需配置一次）
 - 四、克隆仓库
 - 五、切换到工作分支
 - 六、修改文件并提交
 - 1. 修改文件
 - 2. 查看修改
 - 3. 添加到暂存区
 - 4. 提交到本地仓库
 - 七、完整操作流程示例
 - 八、常见问题
 - 问题1：提交时报错 "Please tell me who you are"
 - 问题2：不知道当前在哪个分支
 - 问题3：想放弃修改，恢复到修改前
 - 问题4：已经 add 了，想取消
 - 九、常用命令速查
 - 总结

Git 新手快速上手指南

适合刚安装完 Git 的同学，帮助你快速配置并开始使用

一、安装 Git

如果你还没有安装 Git，请使用以下命令安装：

Ubuntu/Debian：

```
sudo apt update
sudo apt install git
```

Fedora/CentOS/RHEL：

```
sudo dnf install git
```

Arch Linux:

```
sudo pacman -S git
```

二、检查 Git 是否安装成功

打开终端（命令行），输入：

```
git --version
```

如果显示版本号（如 `git version 2.x.x`），说明安装成功。

三、必要的配置（只需配置一次）

在使用 Git 之前，必须告诉 Git 你是谁。这样每次提交代码时，Git 才知道是谁提交的。

```
# 配置你的名字（可以是中文或英文）
git config --global user.name "你的名字"

# 配置你的邮箱
git config --global user.email "你的邮箱@example.com"
```

示例：

```
git config --global user.name "张三"
git config --global user.email "zhangsan@qq.com"
```

 **重要提示：关于 GitHub 账号关联**

邮箱是关键：

- GitHub 通过**邮箱地址**来关联你的提交记录

- 如果你有 GitHub 账号，请使用 GitHub 账号中**已验证的邮箱**
- 如果还没有 GitHub 账号，以后注册时使用这个邮箱即可

用户名随意：

- `user.name` 只是显示用的，可以是任何名字
- GitHub 不用它来关联账号

不需要 SSH：

- 即使不配置 SSH 密钥，只要邮箱匹配，提交就能关联到你的 GitHub 账号
- SSH 只是用于身份验证（push/pull），和提交记录关联无关

验证配置：

```
# 查看配置是否成功
git config --global user.name
git config --global user.email
```

四、克隆仓库

现在可以克隆我的仓库到你的电脑了。

```
# 克隆仓库
git clone https://github.com/Furina318/practice.git

# 进入仓库目录
cd practice
```

说明：

- `git clone` 会把远程仓库的所有代码下载到本地
- 下载完成后会自动创建一个 `practice` 文件夹
- 使用 `cd practice` 进入这个文件夹

验证克隆成功：

```
# 查看当前目录的文件
ls
```

```
# 查看 Git 状态
git status
```

五、切换到工作分支

仓库有多个分支，我们需要切换到 `main` 分支进行开发。

```
# 查看所有分支
git branch -a

# 切换到 main 分支
git checkout main

# 验证当前分支
git branch
```

说明：

- `git branch -a` 显示所有分支（本地和远程）
- `git checkout main` 切换到 `main` 分支
- `git branch` 前面带 `*` 的就是当前分支

六、修改文件并提交

现在可以修改代码了！

1. 修改文件

用你喜欢的编辑器打开并修改文件，比如：

- 修改 `README.md`
- 修改其他代码文件
- 或者创建新文件

2. 查看修改

```
# 查看哪些文件被修改了
git status

# 查看具体修改了什么内容
git diff
```

3. 添加到暂存区

```
# 添加单个文件
git add 文件名

# 或者添加所有修改的文件
git add .
```

4. 提交到本地仓库

```
# 提交修改
git commit -m "描述你做了什么修改"
```

示例：

```
# 假设你修改了 README.md
git add README.md
git commit -m "更新了README文件"
```

七、完整操作流程示例

从头到尾完整演示一遍：

```
# 1. 配置 Git (只需要做一次)
git config --global user.name "张三"
git config --global user.email "zhangsan@qq.com"

# 2. 克隆仓库
git clone https://github.com/Furina318/practice.git
cd practice
```

```
# 3. 切换到 main 分支
git checkout main

# 4. 修改文件（用编辑器修改）
# 比如修改 README.md

# 5. 查看修改
git status
git diff

# 6. 添加修改
git add README.md

# 7. 提交
git commit -m "我的第一次提交"

# 8. 查看提交历史
git log
```

八、常见问题

问题1：提交时报错 "Please tell me who you are"

原因：没有配置用户名和邮箱

解决：

```
git config --global user.name "你的名字"
git config --global user.email "你的邮箱"
```

问题2：不知道当前在哪个分支

解决：

```
git branch
```

前面带 ***** 的就是当前分支

问题3：想放弃修改，恢复到修改前

解决：

```
# 放弃单个文件的修改
git restore 文件名

# 放弃所有修改
git restore .
```

问题4：已经 add 了，想取消

解决：

```
git restore --staged 文件名
```

九、常用命令速查

```
# 查看状态
git status

# 查看修改内容
git diff

# 查看提交历史
git log

# 查看当前分支
git branch

# 切换分支
git checkout 分支名
```

总结

恭喜你！现在你已经学会了：

✓ 配置 Git 用户信息 ✓ 克隆远程仓库 ✓ 切换分支 ✓ 修改文件 ✓ 提交到本地仓库

记住这个基本流程：

1. 配置 → 2. 克隆 → 3. 切换分支 → 4. 修改 → 5. add → 6. commit

注意：

- 本指南只涉及本地操作，不包括推送（push）到远程仓库
- 如果需要推送代码，需要额外的权限配置

有问题随时问我！