

# 1. Git 是什么

---

- 分布式版本控制系统，不同于 SVN/CVS，不依赖中心服务器，每个开发者本地即可完整管理版本记录。
  - 由 Linus Torvalds 在 2005 年发起，目的是管理 Linux 内核开发
- 

## 2. 基本概念

---

- **仓库 (Repository)**：包含 `.git` 目录的文件夹，是 Git 管理的项目根目录。
  - **提交 (Commit)**：保存快照，一次更改对应一个提交，由 SHA-1 哈希唯一标识
  - **分支 (Branch)**：提交历史的可变引用，创建与切换分支用于隔离不同开发线路
  - **标签 (Tag)**：为某次提交贴标签，代表版本标志点
  - **暂存区 (Staging)**：临时缓存变更，用于组织在 commit 前的改动
  - **冲突与合并**：多分支协作中常见，需解决冲突并完成合并操作
- 

## 3. 安装 & 初始化

---

```
git init
```

- 初次配置：设置用户名与邮箱（例如 `git config --global user.name "Name"`）；
  - 可查看帮助：`git help <command>` 或 `git <command> --help`。
- 

## 4. 仓库基础操作

---

```
git init          # 初始化仓库
git clone <url>   # 克隆远程仓库

git add <file>    # 添加单个文件
git add . / -A    # 添加所有变更
git status        # 查看仓库状态
git commit -m "msg" # 提交暂存区内容到本地
git log          # 查看提交历史
```

---

## 5. 远程仓库管理

---

```
git remote add origin <url> # 添加远程仓库
git remote -v                # 查看已添加的远程库
git remote rename old new    # 重命名远程库（如 origin -> oschina）
git remote set-url origin <url> # 修改远程仓库连接地址
```

---

## 6. 拉取与推送

```
git pull origin master    # 拉取远程仓库最新提交并合并
git push origin master    # 推送本地 master 到远程
git push origin master -f # 强制推送（慎用）
```

- `pull` = `fetch` + `merge`，会合并远程变更；
- `push` 前若与远程有差异，建议先 `pull`。若确定覆盖可 `-f` 强制推送

---

## 7. 暂存与恢复

```
git stash          # 暂存当前更改
git stash list     # 查看 stash 列表
git stash pop      # 恢复最近一次 stash
```

---

## 8. 撤销与重置

```
git reset --hard          # 丢弃所有未提交更改
git reset <commit_id>     # 回退到指定提交，保留后续修改
git checkout -- <file>    # 恢复文件到上次 commit 状态
```

---

## 9. 多分支开发与合并

```
git branch new-feature    # 创建新分支
git checkout new-feature  # 切换分支
git merge new-feature      # 将 new-feature 合并回当前分支
git branch -d new-feature  # 删除已合并分支
```

多人协作推荐分支开发，完成后合并至主分支（如 master 或 main）

---

## 10. Git 与 Gitee/GitHub 同步

## GitHub → Gitee

```
git clone git@github.com:...repo.git
cd repo
git remote add gitee git@gitee.com:...repo.git
git push gitee master
```

也可在 Gitee 网页点击「从 GitHub 导入」

## 双向同步（适用于多分支仓库）

```
# 拉取所有远程分支
for b in `git branch -r | grep -v '->'; do
    git branch --track ${b##origin/} $b;
done
git pull origin master
git push gitee master
```

## VS Code 同步

- 克隆任何一端仓库；
- 在源码管理视图中将另一个远程添加为“GitHub”或“Gitee”；
- 使用 Pull & Push 即可选择源头/目标。