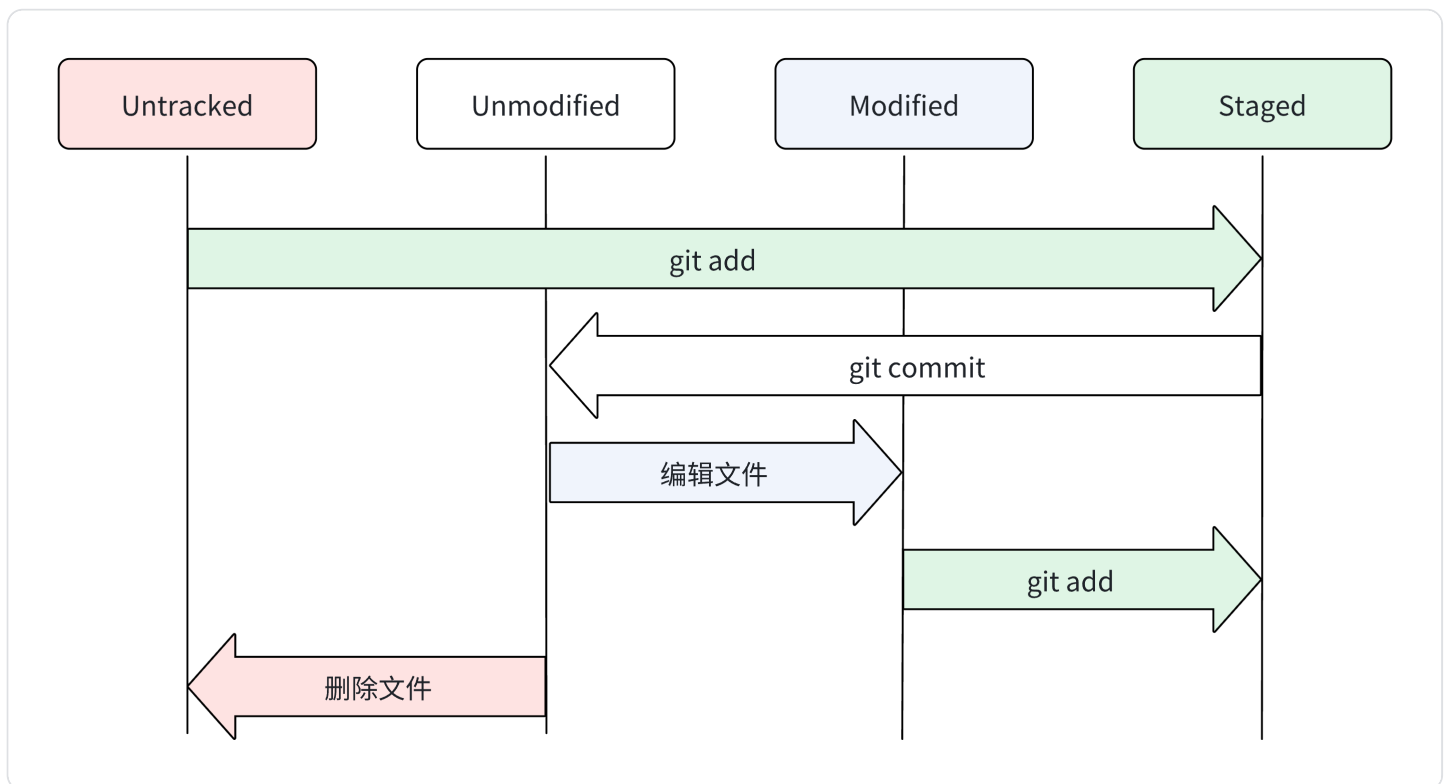


Git 基本使用



- 初学者入门: [可视化教程](#)
- Git命令详情: [官网](#)

1. 文件状态



2. 常用命令

2.1 提交

提交:

```
1 # 将指定的Untracked或Modified状态的文件，添加至暂存区
2 git add <file_name>
3
4 # 将所有的Untracked或Modified状态的文件，添加至暂存区
5 git add .
```

```
6
7 # 将暂存区的修改提交至本地仓库
8 git commit -m"<commit_message>"
```

撤回:

```
1 # 撤回对指定文件的编辑 (未添加至暂存区)
2 git restore <file_name>
3
4 # 撤回 git add <file_name> 操作
5 git restore --staged <file_name>
6
7 # 当前分支回退到指定的某次提交节点
8 # soft: 两节点间的差异信息会保存到暂存区
9 # mixed (默认): 两节点间的差异会保存到工作区
10 # hard: 两节点间的差异会被丢弃
11 git reset --<soft|mixed|hard> <commit_id>
```

2.2 本地分支

2.2.1 Branch

```
1 # 列出本地所有分支, 当前所在分支以 "*" 标出
2 git branch
3
4 # 基于当前分支, 创建一个新分支
5 git branch <branch_name>
6
7 # 切换至指定分支
8 git switch <branch_name>
9
10 # 修改分支名称 (若不指定原分支名, 则默认是当前分支名)
11 git branch -m <原分支名> <新分支名>
12
13 # 强制修改分支名称 (若不指定原分支名, 则默认是当前分支名)
14 git branch -M <原分支名> <新分支名>
15
16 # 删除指定分支
17 git branch -d <branch_name>
18
19 # 强制删除指定分支
```

```
20 git branch -d <branch_name>
```

2.2.2 Checkout

```
1 # 基于当前分支，创建一个新分支，并切换到新分支
2 git checkout -b <branch_name>
3
4 # 切换至指定分支
5 git checkout <branch_name>
6
7 # 将 HEAD 指向指定的节点。
8 # 此时为 detachedHEAD 状态，可以随意提交，
9 # 但是该状态下的提交将被Git在不久的将来丢弃掉，所以需要尽快与一个分支绑定：
10 # 1.基于HEAD创建一个新分支: git branch <branch_name>
11 # 2.强制让某个分支指向HEAD: git branch -f <branch_name> HEAD
12 git checkout <commit_id>
13
14 # 手动将远程分支与本地分支关联
15 git checkout -u <remote_branch_name> <branch_name>
```

2.2.3 Merge

```
1 # 将指定的分支合并到当前所在的分支下，并自动进行新的提交
2 # 方便记忆: git merge branch_name into crrent_branch
3 git merge <branch_name>
```

2.2.4 Rebase

```
1 # 以指定分支为基点，将当前分支的提交复制到指定分支之上
2 # 方便记忆: git rebase crrent_branch onto branch_name
3 git rebase <branch_name>
```

2.2.5 Cherry-Pick

```
1 # 将指定的提交应用到当前分支
2 git cherry-pick <commit_id1> <commit_id2> <...>
```

2.3 远程分支

2.3.1 Remote

```
1 # 列出已经存在的远程仓库
2 git remote
3
4 # 添加远程仓库
5 git remote add <remote_name> <remote_url>
6
7 # 修改远程仓库的别名
8 git remote rename <remote_name> <new_remote_name>
9
10 # 删除指定名称的远程仓库
11 git remote remove <remote_name>
12
13 # 修改远程仓库的 URL 地址
14 git remote set-url <remote_name> <new_remote_url>
```

2.3.2 Fetch

```
1 # 将远程仓库所有分支全部下载到本地
2 git fetch <remote_name>
3
4 # 将远程仓库指定分支的下载到本地
5 git fetch <remote_name> <branch_name>
```

2.3.3 Pull

```
1 # 将指定远程仓库、指定分支下载到本地，并合并到本地相应的分支
2 git pull <remote_name> <branch_name>
```

2.3.4 Push

```
1 # 把本地仓库的分支推送到远程仓库
2 git push <remote_name> <branch_name>
3
```

```
4 # 把本地仓库的分支推送到远程仓库的指定分支
5 git push <remote_name> <branch_name>:<remote_branch_name>
6
7 # 删除指定的远程仓库的分支
8 git push <remote_name> :<branch_name>
9 git push <remote_name> --delete <branch_name>
```

2.3.5 Revert

```
1 # 生成一个新的提交来撤销某次提交
2 git revert <commit_id>
```