# git中的文件状态

## 一、文件的四种核心状态

1. **未跟踪（Untracked）**

* 1. 含义：新创建的文件，Git 尚未开始跟踪

* 1. 特征：git status 中显示为红色

* 1. 如何处理：使用 git add <文件名> 将其加入暂存区

1. **已暂存（Staged）**

* 1. 含义：已通过 git add 命令标记，准备提交到本地仓库

* 1. 特征：git status 中显示为绿色

* 1. 如何处理：使用 git commit 提交到本地仓库

1. **已提交（Committed）**

* 1. 含义：文件的修改已被安全存储到本地仓库

* 1. 特征：git status 中显示 "nothing to commit"

* 1. 状态转换：提交后文件从暂存区转移到本地仓库

1. **已修改（Modified）**

* 1. 含义：已跟踪的文件被修改但尚未暂存

* 1. 特征：git status 中显示为红色的 "modified"

* 1. 如何处理：需要重新执行 git add 暂存修改

## 二、文件状态转换图

**plaintext**

未跟踪（Untracked）  
 ↓ git add  
已暂存（Staged）  
 ↓ git commit  
已提交（Committed）  
 ↓ 对文件进行修改  
已修改（Modified）  
 ↓ git add  
已暂存（Staged） ← 循环

## 三、查看文件状态的命令

1. **基本状态查看bash**

git status

显示工作区和暂存区的状态差异

1. **简洁状态查看bash**

git status -s # 或 git status --short

以紧凑格式显示：

* 1. ??：未跟踪文件

* 1. A ：新文件已暂存

* 1. M：已修改但未暂存

* 1. M ：已修改且已暂存

1. **查看具体修改内容bash**

# 工作区与暂存区的差异  
git diff <文件名>  
  
# 暂存区与本地仓库的差异  
git diff --cached <文件名> # 或 git diff --staged

## 四、状态操作常见场景

1. **放弃工作区修改bash**

git checkout -- <文件名>

将文件恢复到最近一次提交或暂存的状态

1. **取消暂存bash**

git reset HEAD <文件名>

将文件从暂存区退回到工作区（保持修改）

1. **删除文件bash**

# 从工作区和暂存区同时删除  
git rm <文件名>  
  
# 只从暂存区删除，保留工作区文件  
git rm --cached <文件名>

## 五、总结要点

* Git 工作流程本质上是文件在这四种状态间的转换

* 核心操作路径：修改文件 → 暂存（add）→ 提交（commit）

* 未跟踪文件需要先通过 git add 加入版本控制

* 任何时候都可以用 git status 检查文件当前状态