

Git Command

這會列出所有設定過的 remote 名稱及其對應的 URL — 包含 fetch (抓取) 跟 push (推送) 的 URL。
git remote -v
顯示 remote 的 fetch/push URL、HEAD branch、remote branches、local-remote 分支對應、追蹤 (tracking) 狀態等
git remote show origin
可以直接打開 code .git/config
git branch 看分支

git restore . 還原 commit

git branch -d 分支名稱

把 GitLab 專案同步下載到本機(local)

git pull

強制推送指令 覆蓋遠端的 dev
git push --force origin dev

Git 常用指令總覽

Git 是目前最流行的分散式版本控制系統。以下為您整理了日常開發中，從專案初始化、基本操作到進階協作最常用的 Git 指令，並附帶簡要說明。

一、專案設定與初始化 (Setup & Init)

指令	說明
<code>git config --global user.name "Your Name"</code>	設定全域使用者名稱。
<code>git config --global user.email "email@example.com"</code>	設定全域使用者 Email。
<code>git init</code>	在當前目錄初始化一個新的 Git 倉庫。
<code>git clone <URL></code>	複製一個遠端倉庫到本地。

二、基礎工作流程 (Basic Workflow)

這是您日常開發中最常用到的指令。

指令	說明
<code>git status</code>	檢查工作目錄和暫存區的狀態（哪些檔案被修改、哪些未追蹤）。
<code>git add <file></code>	將指定的檔案從工作目錄添加到暫存區 (Staging Area)。
<code>git add .</code>	將所有變更過的檔案添加到暫存區。
<code>git commit -m "Commit Message"</code>	將暫存區的內容提交到本地倉庫，並加上描述訊息。
<code>git commit -am "Commit Message"</code>	跳過暫存區，直接將所有已追蹤的修改提交。
<code>git rm <file></code>	從工作目錄和暫存區中刪除檔案，並記錄刪除操作。

三、查看歷史記錄與差異 (History & Diffs)

指令	說明
<code>git log</code>	查看完整的提交歷史記錄。
<code>git log --oneline</code>	以單行簡潔格式顯示提交歷史。
<code>git diff</code>	比較工作目錄與暫存區之間的差異。
<code>git diff --cached</code>	比較暫存區與最近一次提交之間的差異。
<code>git diff HEAD</code>	比較工作目錄與最近一次提交之間的差異。

四、分支操作 (Branching)

分支是 Git 的核心功能，用於隔離開發工作。

指令	說明
<code>git branch</code>	列出所有本地分支，當前分支前有  標記。
<code>git branch <new-branch></code>	建立一個新的本地分支。
<code>git checkout <branch-name></code>	切換到指定分支。

指令	說明
<code>git checkout -b <new-branch></code>	建立並立即切換到新的本地分支（常用）。
<code>git branch -d <branch-name></code>	刪除指定的本地分支（該分支必須已合併）。
<code>git merge <branch-name></code>	將指定分支的內容 合併 到當前分支。

五、遠端協作 (Remote & Sync)

用於與 GitLab、GitHub 等遠端倉庫進行互動。

指令	說明
<code>git remote -v</code>	查看所有已配置的遠端倉庫名稱及 URL。
<code>git fetch <remote></code>	從遠端倉庫下載資料，但不自動合併到本地。
<code>git pull</code>	下載並自動合併遠端分支的內容到當前本地分支（相當於 <code>git fetch</code> + <code>git merge</code> ）。
<code>git push <remote> <branch></code>	將本地分支的提交推送到遠端倉庫。
<code>git push -u origin <branch></code>	首次推送時設定本地分支追蹤遠端分支。

↩ 六、還原與撤銷操作 (Undoing Changes)

Git 讓您可以安全地撤銷錯誤操作。

指令	說明
<code>git reset HEAD <file></code>	取消暫存區中指定檔案的變更（將檔案從暫存區移回工作目錄）。
<code>git checkout -- <file></code>	捨棄工作目錄中指定檔案的 未暫存 修改（回到最近一次提交或暫存的狀態）。
<code>git revert <commit-hash></code>	創建一個新的提交，來撤銷指定提交所做的更改（保留歷史記錄）。
<code>git reset --hard <commit-hash></code>	危險操作！ 永久丟棄當前分支後續的所有提交，將 HEAD 指向指定提交。
<code>git stash</code>	暫存當前工作目錄的變更，讓工作目錄回到乾淨狀態，方便切換分支。
<code>git stash pop</code>	重新應用最近一次暫存的變更。

★由 Admin 建立於 2025-11-26 12:59:03 CST
✎由 Jerry.Zh 更新於 2025-12-01 17:12:24 CST