

GitLab 常用指令

git bracnch
看本機分支 + upstream
git branch -vv
看遠端有哪些分支
git branch -r

✓ 情況一：遠端已經有 `dev`，本機也有 `dev`（最常見）

先抓遠端最新分支資訊

git fetch origin

切到本機 dev

git checkout dev

讓本機 dev 追蹤遠端 dev（第一次一定要做）

git branch --set-upstream-to=origin/dev dev

合併遠端 dev 到本機 dev

git pull origin dev

✓ 情況二：遠端有 dev，但本機 dev 是「自己亂建的」想直接用遠端覆蓋

本機 dev 上的修改會被丟掉

git fetch origin

git checkout dev

git reset --hard origin/dev

✓ 情況三：遠端有分支，本機還沒有 dev（補充）

git fetch origin

git checkout -b dev origin/dev

git add .

git commit -m "20251022v2 Modified"

git push origin dev

（單獨只有一個main 就.. git push origin main）

`origin` 是 Git 裡的一個「遠端倉庫名稱（別名）」，不是關鍵字，也不是固定語法。

git clone <https://gitlab.com/xxx/project.git>

Git 會自動幫你做一件事：

遠端倉庫 URL → 取一個名字叫做 origin

`git pull origin dev` 等於 從 `origin` 的 `dev` 分支抓下來 + 合併到你目前的本機分支

- `git add .` → 加入目前目錄下所有修改、新增檔案，但不包含刪除。
- `git add -u` → 加入修改和刪除，但不包含新檔案。
- `git add -A` → 加入所有變更（新增、修改、刪除），最完整。

★由 Jerry.Zh 建立於 2025-12-30 15:06:54 CST

✍由 Jerry.Zh 更新於 2026-02-06 13:49:09 CST