

Comandi Git

Comandi Git per l'uso quotidiano e le emergenze – cercabili e copiabili con un clic. Con avvisi sui comandi che scartano il lavoro in modo definitivo.

daniele.michelin // cheat sheet

Uso quotidiano: stato e commit

- Stato breve con branch e info di tracking
- git status -sb
- Mostra le modifiche non in staging
- git diff
- Mostra le modifiche in staging
- git diff --staged
- Aggiunge le modifiche a blocchi
- git add -p
- Crea un commit con messaggio
- git commit -m "Fix login redirect"
- ! Aggiunge le modifiche in staging all'ultimo commit
- git commit --amend --no-edit
- Crea un nuovo branch e ci passa
- git switch -c feature/login

Log e ricerca

- Ultimi 20 commit come grafo compatto
- git log --oneline --graph --decorate -20
- Commit di un autore nelle ultime 2 settimane
- git log --since='2 weeks ago' --author='Daniele'
- Trova i commit che hanno aggiunto o rimosso un testo
- git log -S 'password' --oneline
- Chi ha modificato per ultimo le righe 10-30?
- git blame -L 10,30 file.py

Stash

- Mette da parte le modifiche con una descrizione
- git stash push -m "wip login"
- Mette da parte anche i file non tracciati
- git stash push -u
- Applica uno stash e lo conserva
- git stash apply stash@{0}
- Applica lo stash più recente e lo rimuove
- git stash pop
- ! Scarta uno stash
- git stash drop stash@{0}

Rebase e pulizia della cronologia

- Riapplica il branch corrente sopra main
- git rebase main
- ! Riordina, unisce o riscrive in modo interattivo gli ultimi 4 commit
- git rebase -i HEAD~4
- Annulla il rebase, torna allo stato precedente
- git rebase --abort
- Unisce un branch con un merge commit
- git merge --no-ff feature/login

Annullare le modifiche

- ! Scarta le modifiche non in staging di un file
- git restore file.txt
- Toglie un file dallo staging, le modifiche restano
- git restore --staged file.txt
- Annulla un commit con un commit inverso (sicuro sui branch condivisi)
- git revert abc1234
- Annulla l'ultimo commit, le modifiche restano in staging
- git reset --soft HEAD~1
- ! Scarta l'ultimo commit e le sue modifiche
- git reset --hard HEAD~1
- ! Elimina definitivamente file e cartelle non tracciati
- git clean -fd

Reflog e recupero

- Mostra la cronologia di tutti gli spostamenti di HEAD
- git reflog
- ! Torna a uno stato precedente del reflog
- git reset --hard HEAD@{2}
- Recupera un commit perso tramite hash come branch
- git switch -c rescue abc1234
- Trova gli oggetti orfani
- git fsck --lost-found
- ! Annulla l'ultimo reset, merge o rebase
- git reset --hard ORIG_HEAD

Remote e bisect

- Scarica tutti i remote, elimina i branch rimossi
- git fetch --all --prune
- Scarica e riapplica i commit locali sopra
- git pull --rebase
- ! Force push che non sovrascrive i nuovi commit altrui
- git push --force-with-lease
- Avvia la ricerca binaria del commit che rompe
- git bisect start && git bisect bad && git bisect good v1.2
- Esegue bisect in automatico con uno script di test
- git bisect run ./test.sh
- Termina bisect e torna al branch di partenza
- git bisect reset