



Comandi git avanzati



Git offre molte funzionalità avanzate che consentono di gestire in modo più efficace e flessibile lo sviluppo del software

1. Branching e Merging Avanzati

- Creazione di branch per lo sviluppo di nuove funzionalità o risoluzione di bug.
- Fusione di branch in modo flessibile, inclusi metodi come il rebase per mantenere una storia di commit più lineare.

2. Rebase

- Consente di riapplicare i commit di un branch su un altro, utile per mantenere una storia di commit più pulita.
- Può essere usato per integrare le modifiche da un branch all'altro in modo più flessibile rispetto alla fusione.

3. Stashing

- Temporaneamente nasconde le modifiche locali non commesse in modo da poter passare a un altro branch o eseguire altre operazioni senza committare le modifiche.

4. Cherry-pick

- Permette di selezionare singoli commit da un branch e applicarli su un altro branch.

5. Submodule

- Consente di includere altri repository Git all'interno di un repository principale.

6. Git Bisect

- Aiuta a individuare il commit specifico in cui è stato introdotto un bug attraverso una procedura di ricerca binaria.

7. Reflog

- Registra tutte le azioni che modificano la storia del repository, consentendo il ripristino di commit persi o branch eliminati accidentalmente.

8. File .gitignore

- Specifica i file e le directory che Git deve ignorare, utile per evitare di tracciare file non desiderati.

9. Hooks

- Consente di eseguire script personalizzati automaticamente in risposta a determinati eventi Git, come il pre-commit o il post-merge.

10. Bisect

- Aiuta nell'individuare il commit che ha introdotto un bug utilizzando una ricerca binaria tra i commit.

11. Note

- Consente di allegare annotazioni ai commit, utile per memorizzare informazioni aggiuntive o metadati.

12. Alias

- Consente di creare alias per comandi Git complessi o frequentemente utilizzati, semplificando la digitazione e la memorizzazione di comandi.