**Cos'è Git**

* **Sistema di Controllo di Versione**: Git permette di gestire e tracciare le modifiche al codice sorgente nel tempo, facilitando la collaborazione tra più sviluppatori.
* **Distribuito**: Ogni copia del repository Git è completa, con tutta la cronologia del progetto. Questo significa che ogni sviluppatore ha una copia completa del progetto e della sua storia.

**Cosa Serve Git**

* **Tracciamento delle Modifiche**: Git registra ogni modifica effettuata al codice sorgente, consentendo di tornare a versioni precedenti se necessario.
* **Collaborazione**: Facilita la collaborazione tra sviluppatori, permettendo a più persone di lavorare sullo stesso progetto simultaneamente.
* **Branching e Merging**: Supporta la creazione di rami (branches) separati per lo sviluppo di nuove funzionalità, che possono essere uniti (merged) al ramo principale (main) una volta completati.

**Cos'è il Main**

* **Ramo Principale**: main è il nome convenzionale del ramo principale di un repository Git. È il punto di partenza per la storia del progetto e il ramo in cui vengono generalmente integrate le modifiche stabili e definitive.

**Cos'è un Ramo (Branch)**

* **Ramo**: Un ramo in Git è una linea di sviluppo separata. I rami permettono di lavorare su diverse funzionalità o correzioni di bug in parallelo senza interferire con il ramo principale.
* **Uso dei Rami**: Permettono di isolare lo sviluppo di nuove funzionalità o correzioni di bug, facilitando il lavoro collaborativo e la gestione del codice.

**Cos'è un Merge**

* **Merge**: Il merge è il processo di integrazione delle modifiche da un ramo all'altro. Solitamente, le modifiche di un ramo secondario vengono integrate nel ramo principale.
* **Conflitti di Merge**: Quando due o più modifiche concorrenti vengono integrate, possono verificarsi conflitti. Un conflitto si verifica quando Git non può decidere automaticamente come integrare le modifiche, richiedendo l'intervento manuale per risolvere.

**Comandi Principali di Git**

**Configurazione di Git**

git config --global user.name "Il Tuo Nome"

git config --global user.email "tuoemail@example.com"

**Creazione e Clonazione di Repository (uno o l’altro)git**

git init # Inizializza un nuovo repository Git

git clone <url-repository> # Clona un repository esistente

**Stato e Cronologia**

git status # Mostra lo stato dei file nel repository

git log # Mostra la cronologia dei commit

**Aggiunta e Commit di Modifiche**

git add <file> # Aggiunge un file all'area di staging

git add . # Aggiunge tutti i file all'area di staging

git commit -m "Messaggio del commit" # Registra le modifiche nell'area di staging

**Creazione e Gestione dei Rami**

git branch # Elenca i rami nel repository

git branch <nome-ramo> # Crea un nuovo ramo

git checkout <nome-ramo> # Passa a un altro ramo

git checkout -b <nome-ramo> # Crea e passa a un nuovo ramo

**Merge e Risoluzione dei Conflitti**

git merge <nome-ramo> # Esegue il merge del ramo specificato nel ramo corrente

* **Risoluzione dei Conflitti**: Quando si verifica un conflitto, Git segnala i file in conflitto e richiede di risolvere manualmente le differenze prima di completare il merge.

**Push e Pull**

git push origin <nome-ramo> # Invia le modifiche locali al repository remoto

git pull origin <nome-ramo> # Recupera e integra le modifiche dal repository remoto

**Riepilogo**

* **Git**: Sistema di controllo di versione distribuito per tracciare le modifiche al codice sorgente.
* **Main**: Il ramo principale del repository.
* **Ramo (Branch)**: Una linea di sviluppo separata.
* **Push**: Invia le modifiche locali al repository remoto.
* **Pull**: Recupera e integra le modifiche dal repository remoto.
* **Merge**: Integra le modifiche da un ramo all'altro, potenzialmente causando conflitti che devono essere risolti manualmente.

Git è uno strumento fondamentale per lo sviluppo software moderno, che facilita il lavoro collaborativo e la gestione efficiente del codice.

Link utili

<https://www.html.it/guide/git-la-guida/>

<https://it.wikipedia.org/wiki/Git>

<https://gist.github.com/tesseslol/da62aabec74c4fed889ea39c95efc6cc>

<https://www.programmareinpython.it/blog/come-usare-git-e-github/>

<https://www.youtube.com/watch?v=wPAE9-DdMtI&t=1590s&pp=ygUDZ2l0>

<https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/chiavi-ssh-con-github/#:~:text=Per%20l'accesso%20in%20lettura,tramite%20nome%20utente%20e%20password>