

# git

xVss

- **Attenzione:** Ogni cosa che scrivo nel seguito va presa con le pinze, visto che ancora devo capire bene come funziona il tutto. Riporto comunque i comandi che ho testato e le cose che credo di aver capito, quindi non dovrebbero esserci problemi.

## 1. Comandi

git serve a mantenere una sorta di storia del progetto, con il log di tutti i cambiamenti, e la possibilità di tornare ad una versione precedente. Con il termine di repository non ci si riferisce necessariamente soltanto a quello che si trova in remoto. Anche una cartella locale può essere trasformata in un repository con i comandi appropriati.

### git clone [HTTPS]

Serve a copiare, nella cartella in cui si invoca il comando, l'intera cartella del repository di indirizzo [HTTPS]. Il link del repository si trova nella pagina GitHub del repository, a destra, copiando il link che compare cliccando sul tasto verde di 'clone'. Una volta copiato, si possono apportare modifiche in locale.

**Es:** git clone <https://github.com/xVss/hello-world.git>

### git status

Visualizza i file creati/modificati che non sono più sincronizzati con il repository nel cloud. I file in rosso sono quelli che non sono stati aggiunti ad un eventuale commit. I file in verde sono quelli segnati per essere riportati sul repository alla prossima commit.

### git add [FILE1] [FILE2] ...

Serve a segnare i file che si vuole richiedere vengano aggiunti al branch del repository in cui si sta lavorando Per aggiungere tutti i file della cartella: "git add ."

**Es:** git add file1.java file2.java

### git commit

Serve a fare la commit, a caricare in remoto i file selezionati con *git add*.

- -a : con questa opzione vengono messi nel commit tutti i file nuovi/modificati ma non quelli cancellati

- -m : scrivi il messaggio/commento del commit direttamente nel comando

**Es:** git commit -m "Modificato file sui DiGraf.java"

### **git log**

Restituisce il log dei cambiamenti apportati nella storia del repository

- --oneline : Restituisce una versione compatta del log

### **git pull**

Sincronizza i file in locale prendendoli da remoto

## 2. Esempi

### **Copiare, modificare e ricaricare un repository nel cloud**

- `git clone https://github.com/Alessandrostr95/ProgettiJava.git`
- `cd ProgettiJava/`
- `touch nuovo_file.txt`
- `git status`
- `git add nuovo_file.txt`
- `git commit -m "Creato nuovo_file.txt"`
- `git push origin master`
- `git pull`

## 3. Cose da scrivere/spiegare vedere

`git branch` e branching | `git remote` | origin | HEAD del repository | Creare, selezionare e lavorare su diversi branch |