1. La possibilità di gestire le varie versioni di uno stesso documento e che tiene traccia delle varie modifiche che vengono effettuate nel tempo, quindi si può lavorare in più persone contemporaneamente sullo stesso progetto.
2. Per creare una repo vado nella cartella attraverso i comandi da terminale cd nome della cartella e uso il comando “git init”.
3. Per origin si intende il remoto. es. se faccio una push -u origin nome significa che la invio a remoto
4. Git push Spinge tutti gli oggetti locali modificati nel repository remoto e fa avanzare i suoi rami.

Git pull Recupera i file dal repository remoto e li unisce a quello locale

1. Un branch viene utilizzato per l’implementazione di funzionalità. Cioè sviluppare in modo indipendente una funzionalità partendo da una radice.

Il branch master è la versione attualmente in produzione, conserva tutte le release.

Il branch develop è la base del nostro progetto e serve le future integrazioni.

1. Per creare un branch utilizzo il comando git branch <nomebranch>

Per entrare dentro il branch usiamo git checkout

Oppure possiamo creare il branch ed entrarci dentro con un solo passaggio attraverso git checkout -b <nomebranch>

1. git chechout develop

git merge sviluppo

1. La pull request è una richiesta, per far si che le modifiche che vogliamo aggiungere vengano accettate.
2. Git stash permette di “mettere da parte per dopo” le modifiche fatte al nostro codice, senza farne il commit, in modo che sia possibile, quindi, tornare all’ultimo “commit” locale e poter, eventualmente, cambiare branch in caso di necessità.

**git stash clear** Se hai più stashes salvati e non è più necessario utilizzarli, puoi rimuoverle con il comando clear.

**git stash list** guarda cosa hai attualmente nascosto

**git stash pop** con questo comando puoi tornare al punto in cui avevi interrotto.