APPUNTI GIT

In git abbiamo tre stati: working copy -> stage -> repository

# Inizializzazione

* git init
* $ git config --global user.name "Il vostro nome"
* $ git config --global user.email [tuaemail@dominio.it](mailto:tuaemail@dominio.it)
* git clone <link github> or git pull ( se ero gia connesso al repository)

# Gittare un file

* git status
* git add . (oppure git <nome file>)
* $ git commit -m "Descrizione delle modifiche"
* $ git commit -am "Descrizione delle modifiche" ( significa add+commit, piu veloce)
* git log (mostra tutti i log passati)

# Modificare commit passati

* git diff --stage (mi fa vedere le modifiche effettuate hai file, anche nella fase di stage)
* git rm <nome file> (rimuove il file, DEVE ESSERE SUCCESSIVAMENTE COMMITATA)
* git rm <nome file> + git add <nuovo nome file> (rinominazione corretta)
* git reset HEAD <nome f> (utile per rimuovere il file dalla fase di stage)
* git checkout <prime cifre del ID del commit funzionante> -- <nome file da fare tornare alla versione voluta> (utile per commit errati)

Git hub (git online)

Repository name -> scrivo il nuovo del nuovo repository , ci da un link per collegare il git

git init

git add.

git commit -m <name commit>

git remote add <nikname repos> <link da github> (collegato il repository con la cartella

git push -u <nikname repos> master (carico file la prima volta seno basta git push)

## merge conflict

se due persone stanno modificando lo stesso file … il primo committa e fa il push invece il secondo committa solo poi dopo fa il pull ha un CONFLITTO nel suo file

se il secondo è il master puo fare le modifiche che vuole, git add. + git commit (senza nulla altro)+ esc + wq +git push

# branching

-git branch (controllo quanti branch ci sono e su quale sto lavorando)

-git branch feature1 (faccio una copia del master in feature1)

-git checkout feature1(ora lavoro su feature 1)(faccio modifiche)

-git add.

-git commit -m “name”

->ora facciamo la pull request<-

-git checkout master + git pull (per essere sicuri che il master non e stato modificato da altri)

-git checkout feature1

-git merge master (se mi esce il CONFLICT merge)(faccio modifiche)

-git add. + git commit + esc + wq + git push or git push –-set-upstream origin feature1

-> vado su github e faccio la pull request attraverso il pulsante(compare &pull request<-

->dopo di che qualchun altro magrari il master del github puo decidere di accettare la richiesta di merge sul master ed eliminare il branch temporaneo<-

git branch -d the\_local\_branch per eliminare il branch in locale