*Confronto tra Staging Area (Index) e Repository*

Con get diff si vedono solo le differenze tra i file con cui si sta lavorando e quelli nella repository.

Per risalire invece alle differenze tra i file nella staging area e l’index, il comando da usare è :

git diff --staged

*Rimozione file dalla repository e la versione working copy*

Per fare ciò si esegue

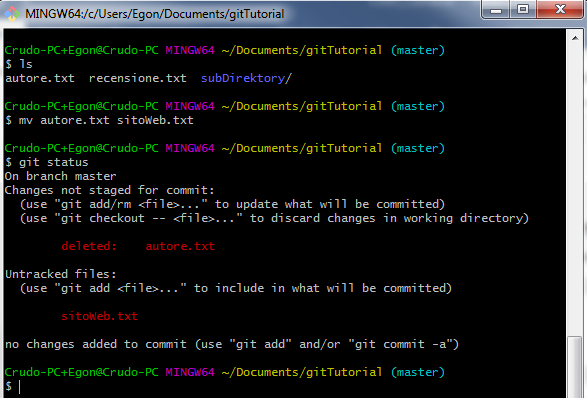
git rm <nome file>

La sintassi è analoga a quella vista nei comandi per la gestione dei file e navigazione nelle directory, con l’aggiunta della keyword git.

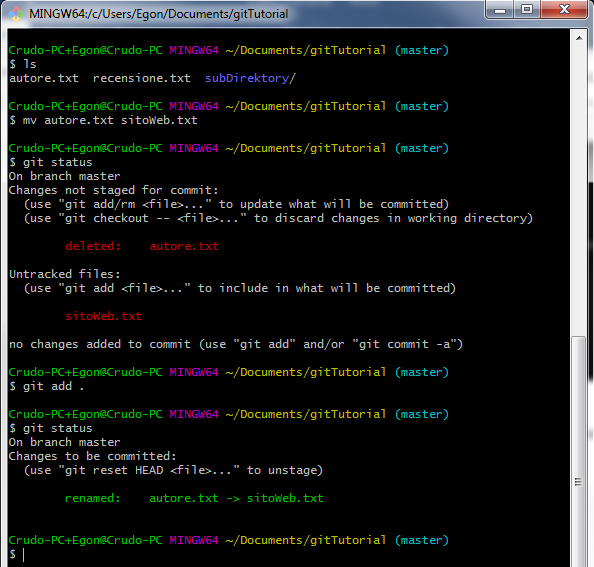
Ad ogni modo bisogna commitare l’evento dela cancellazione del file, di fatti se si fa git status il file rimosso sarà riportato in verde affiancato da “deleted”

*Rinominare e spostare files*

Quando si rinomina un file dall’explorer o mediante il comando “mv” git percepirà la suddetta azione come l’eliminazione del file con il vecchio nome, e la creazione del nuovo file con un nuovo nome



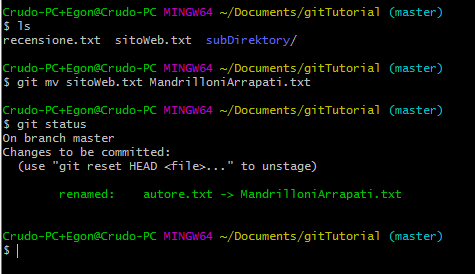
Adesso aggiungendo le modifiche all’index nello staging area, git è in grado di riconoscere il processo di rinominazione



Questo procedimento risulta essere un po' macchinoso, ma può essere evitato mediante il comando

git mv <vecchio\_nome\_file> <nuovo\_nome\_file>

grazie al quale si ha la rinominazione del file che viene immediatamente posto nello staging area



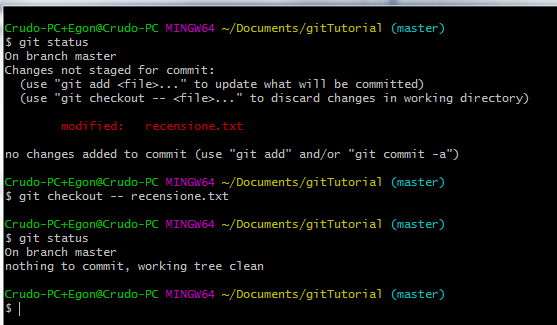
Lo stesso discorso vale sia per lo spostamento in altre directory, non per la copia!

Estrarre un file dalla repository e metterlo nella cartella di lavoro

Nel caso avessimo modificato erroneamente un file e non fossimo in grado di annullare le modifiche, si può estrarre il suddetto file dalla repository e sostituirlo al working file mediante il seguente comando

git checkout -- <nomefile>

come nel seguente esempio:



Togliere file dallo staging area

git reset HEAD <nome file>

il file verrà rimesso fuori dall staging area e verrà scritto di colore rosso proprio come un working file! Bitches!