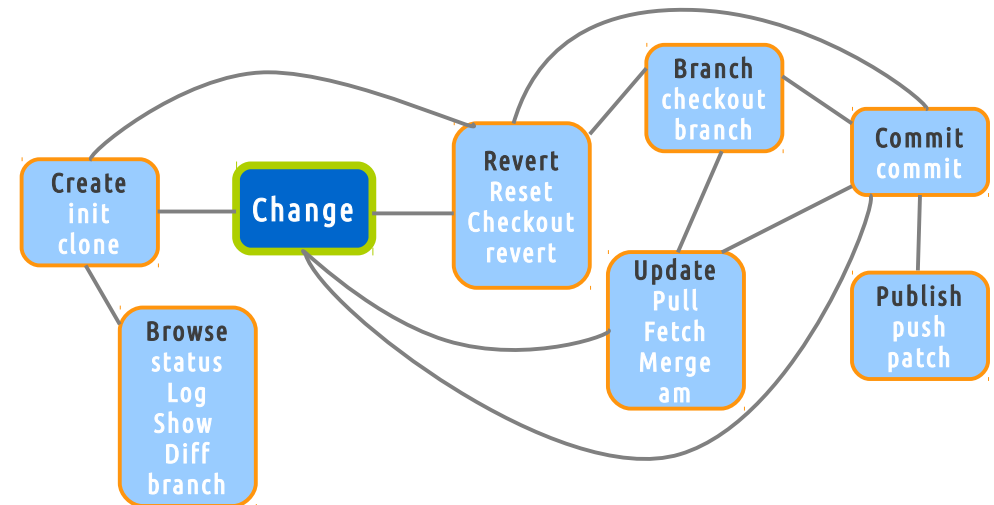


Git: Compendio

Da ricordare: `git command -help` per avere aiuto su un dato comando

Configurazione globale di Git: `$HOME/.gitconfig` (`git config --help`)

Creare	Tornare sui propri passi
Da dati esistenti <code>cd ~/projects/myproject</code> <code>git init</code> <code>git add</code>	Ritornare all'ultima commit <code>git reset --hard</code> (!! non si può annullare)
Da un repository esistente <code>git clone ~/exist/repo ~new/repo</code>	Annullare l'ultima commit <code>git revert HEAD</code> (crea una nuova commit)
Mostrare	Annullare una specifica commit
Files cambiati nella wd <code>git status</code>	<code>git revert \$id</code> (crea una nuova commit)
Modifiche rispetto all'ultima commit <code>git diff</code>	Correggere l'ultima commit <code>git commit --amend</code> (dopo aver corretto gli errori)
Cosa è cambiato tra due commit <code>git diff \$id1 \$id2</code>	Riprendere una certa versione di file <code>git checkout \$file</code>
Storia dei cambiamenti <code>git log</code>	Diramazioni (branches)
Cambiamenti per un file con diff <code>git log -p \$file \$dir/etc/tory/</code>	Spostarsi su un \$id branch <code>git checkout \$id</code>
Chi ha cambiato cosa e quando <code>git blame \$file</code>	Merge tra branch \$1 e \$2 <code>git checkout \$2</code> <code>git merge \$1</code> (destinatario \$2)
Una commit identificata \$id <code>git show \$id</code>	Creare un branch \$new partendo da HEAD <code>git branch \$new</code>
Uno specifico file da una specifica commit <code>git show \$id:\$file</code>	Creare il branch \$new partendo dal branch \$old, posizionandosi su \$new <code>git checkout -b \$new \$old</code>
Tutti i branch locali <code>git branch</code> (* indica il branch corrente)	Cancellare un \$branch branch <code>git branch -d \$branch</code>



Aggiornare	Publiccare
Carica gli ultimi aggiornamenti da origin <code>git fetch</code> (non esegue un merge)	Salva tutte le modifiche <code>git commit -a</code> (ma non i file aggiunti)
Carica gli ultimi aggiornamenti da origin <code>git pull</code> (ed esegue un merge)	Prepara una patch per altri programmatori <code>git format-patch origin</code>
Applicare una patch preparata da altri <code>git am apatch.patch</code> (in caso di conflitti, dopo averli risolti continuare con <code>git am --resolved</code>)	Salva le commit su server remoto (origin) <code>git push</code>
	Mette un tag/milestone <code>git tag V1.0</code>
Configurazione – cose utili	
Colori dell'interfaccia <code>git config --global color.ui auto</code>	Log più leggibili <code>git log --pretty=online</code>
Interfacce grafiche	
Gitk (integrato nell'installazione git standard) <code>\$prompt> gitk</code> https://delicious.com/matthew.mccullough/git+gui Elenco di client Git, per tutti i gusti	