



Git 101

Linux Day

2018, LUGMan



Paolo Dina, <https://linuxday.2018.php3.it/>



Di cosa parliamo oggi

- 1 Come si è arrivati a Git
- 2 Cosa può fare Git è perché ti è utile
- 3 Git comandi base
- 4 Lavorare con i branch
- 5 Lavorare con repository remoti
- 6 GitHub, GitLab, Bitbucket
- 7 Altri comandi utili
- 8 Conclusioni





Per la migliore esperienza di sviluppo, usa Git!

Non è l'unico ma imparerai ad amarlo





Git 101

Comandi base





1.

Ci serve un progetto

```
$ pwd
/tmp/myproject

$ touch {alice,bob}.txt
$ tree
.
├── alice.txt
└── bob.txt
```

Nulla di speciale per ora





2.

Ci serve un repo

```
$ git init
$ tree -a -L 2
.
├── alice.txt
├── bob.txt
└── .git <-- E' LUI!!!
    ├── branches
    ├── config
    ├── description
    ├── HEAD
    ├── hooks
    ├── info
    ├── objects
    └── refs
```





La directory .git

- 1 gestita da Git
- 2 non si dovrebbe toccare direttamente!
- 3 contiene tutte le informazioni sull'evoluzione dei nostri file



Split-screen heading and list body template.



Let's cook...



Come vedi cominciare con Git è stato
molto semplice.

Vediamo quali possibilità ci offre.





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git status

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git add

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git commit

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git rm

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git log

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

.gitignore

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git diff

todo





- 1 git status
- 2 git add
- 3 git commit
- 4 git rm
- 5 git log
- 6 file .gitignore
- 7 git diff
- 8 git clean

git clean

todo





Git 101

Lavorare con i branch





Branch

1.

E' un ramo di sviluppo indipendente.

- 1 bugfix
- 2 sviluppo di nuove funzionalità
- 3 ambiente isolato dove fare esperimenti





2.

Branch

Inoltre

- 1 in ogni repository Git esiste un branch di default chiamato `master`
- 2 le modifiche effettuate su un branch sono confinate a quel branch





PRO TIPS

- 1** crea tutti i branch che ti servono
- 2** Git ha una gestione intelligente e non rallenterà per questo
- 3** e neanche occuperà molto spazio su disco



Split-screen heading and list body template.



- 1 git branch
- 2 git checkout
- 3 git merge

git branch

todo





- 1 git branch
- 2 git checkout
- 3 git merge

git checkout

todo





- 1 git branch
- 2 git checkout
- 3 git merge

git merge

todo





Git 101

Repository remoti





Sui repo remoti...

1.

- 1** in Git ogni sviluppatore ha una propria copia locale del repository che contiene tutta la storia e i branch
- 2** lo sviluppatore lavora sulla propria copia ma può decidere di "inviare" le modifiche a un repository remoto, a GitHub o qualsiasi altro provider VCS (o anche in locale)





- 1 git remote
- 2 git pull
- 3 git push
- 4 git clone

git remote

todo





- 1 git remote
- 2 git pull
- 3 git push
- 4 git clone

git pull

todo





- 1 git remote
- 2 git pull
- 3 git push
- 4 git clone

git push

todo





- 1 git remote
- 2 git pull
- 3 git push
- 4 git clone

git clone

todo





Git 101

GitHub & Co.





GitHub, GitLab, whoat?

1.

Piattaforme collaborative per l'hosting di codice sorgente di progetti opensource o privati.

Offrono le funzionalità tipiche di altri social come Facebook e Twitter.





2.

- 1 soluzioni cloud (backup e 24X7 uptime, di solito)
- 2 review del codice collaborativa
- 3 gestione di team e organizzazioni
- 4 possibilità di fork / pull request
- 5 pages
- 6 molto altro





Qualche link

- 1 [Doc. GitHub](#)
- 2 [Doc. GitLab](#)
- 3 [Doc. Bitbucket](#)



Read The FINE Manual :-]



Demo time

*Vediamo **in pratica**
come funziona*





Git 101

Altri comandi





Altri comandi

1.

Git mette a disposizione moltissimi altri comandi per ogni evenienza e necessità.

E' impensabile elencarli tutti ora.





2.

Ne vedremo
qualcuno





Non indispensabili ma utili

- 1 git alias
- 2 git grep
- 3 git blame
- 4 git bisect
- 5 gitstats(*)



(*) non un comando Git nativo



- 1 git alias
- 2 git grep
- 3 git blame
- 4 git bisect

git alias

todo





- 1 git alias
- 2 git grep
- 3 git blame
- 4 git bisect

git grep

todo





- 1 git alias
- 2 git grep
- 3 git blame
- 4 git bisect

git blame

todo





- 1 git alias
- 2 git grep
- 3 git blame
- 4 git bisect

git bisect

todo





Git 101

Conclusioni





QUESTIONS



Domande del pubblico



QUESTIONS



Domande del pubblico





Questions?



Domande del pubblico