



Introduzione a Git

2018.02.27

Francesco Nigro

Di cosa parliamo oggi



- Le basi del versionamento
- Introduzione a Git
- Il mio primo repository
- working dir, staging, repo
- creare e gestire un branch
- navigare la storia
- diffs

Di cosa **NON** parliamo oggi



- Deploy
- Architetture
- Collaborazione
- Rebase

Materiale utile



- Version Control Intro
https://en.wikipedia.org/wiki/Version_control
<http://git-scm.com/video/what-is-version-control>
- Sito web <http://git-scm.com>
- Manuale <http://git-scm.com/doc>
- Salviamoci da noi stessi:
<http://git-scm.com/book/en/v2/Git-Basics-Undoing-Things>
- Git Cheat Sheet
training.github.com/kit/downloads/github-git-cheat-sheet.pdf
- Markdown Syntax Manual
<http://daringfireball.net/projects/markdown/syntax>

Le basi del versionamento



report.txt



report - very old.txt



report - real final.
txt



report - old.txt



report - new.txt



report - new new.
txt



report - final.txt



report - final v2.txt

File versioning: the bad old days

Le basi del versionamento



Report.txt



Le basi del versionamento



Report.txt

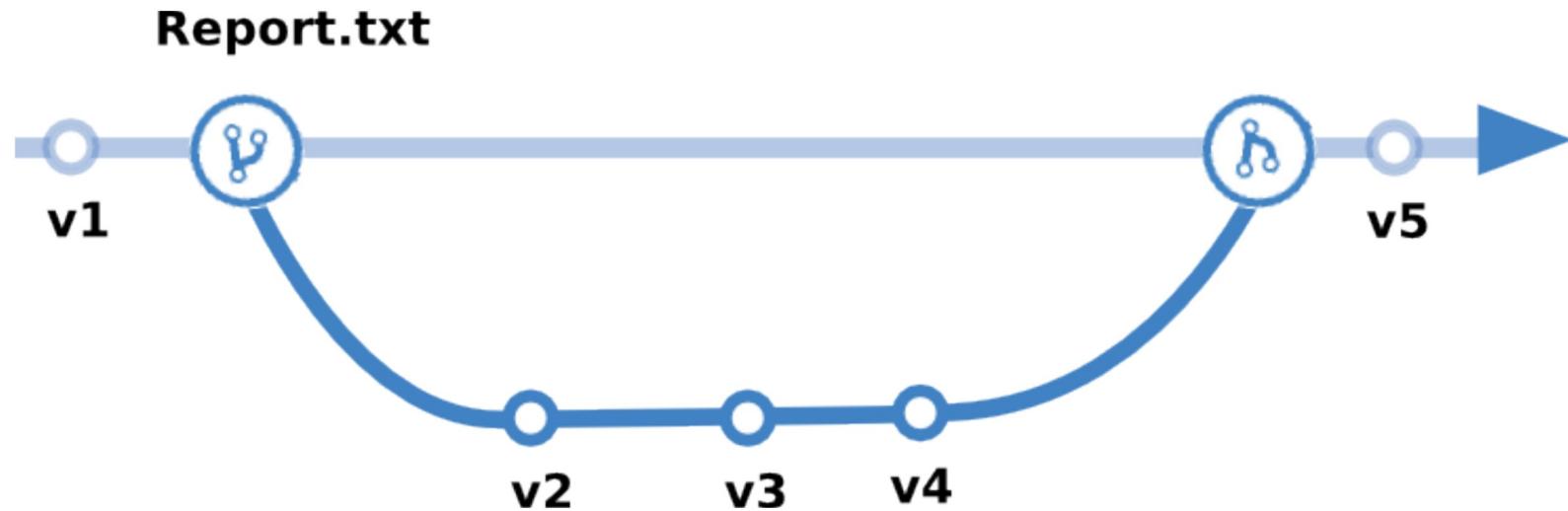


Le basi del versionamento



Report.txt





Un branch è una copia autonoma del branch principale (fino al prossimo merge)



Git: lo stupido sistema di versionamento

- Distribuito
- Off line
- Veloce
- Free and Open Source
- Diffuso

Introduzione a Git



Installare git su *Ubuntu

```
# add-apt-repository ppa:git-core/ppa  
  
# apt update  
  
# apt install git  
  
# git --version  
git version 2.16.1  
  
git config --global core.editor "vim"
```

Introduzione a Git



```
~$ git help
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]
```

Il mio primo repository

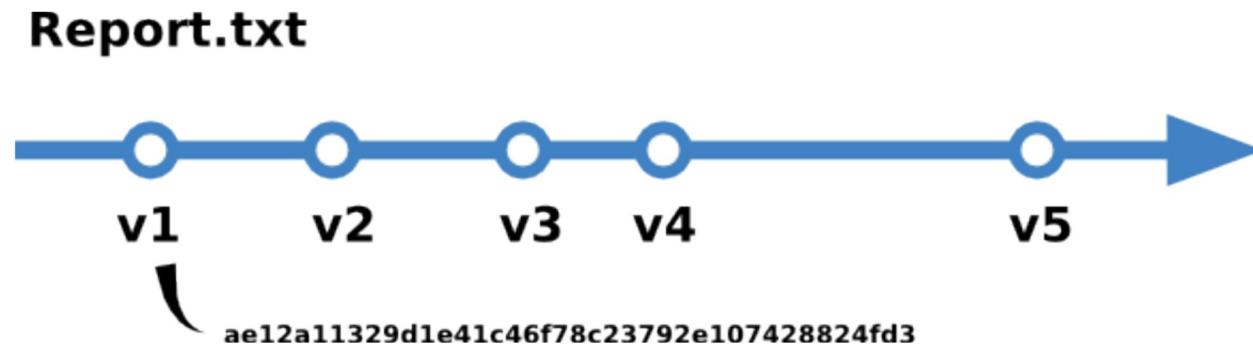


```
$ git status
fatal: Not a git repository (or any parent up to mount point /home)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).

$ git init
Inizializzato un repository Git in /home/fry/test/.git/

$ ls -a
.  ..  .git
```

Il mio primo repository



Un codice SHA-1 identifica il commit univocamente nell'universo

Il mio primo repository



```
$ echo "cippa lippa" > prova.txt
$ git hash-object prova.txt
6ce9cc1e90b2409c4db5bdeb1e2b968bd8ab2c9d

$ echo "cippa lippA" | git hash-object --stdin
735a7765b267776eb6bd909c363226d44acc0d00

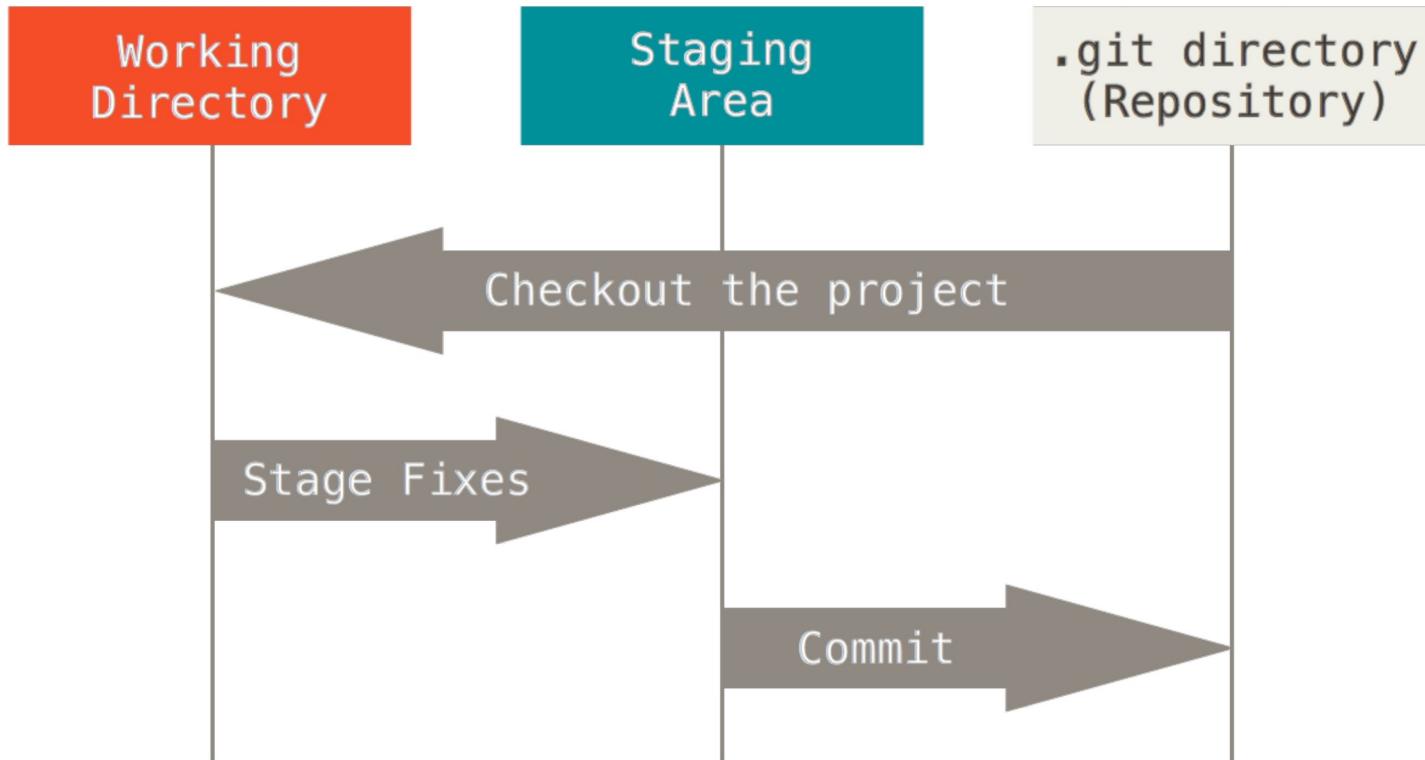
~/test/.git/objects/6c$ pigz -d < e9cc1e90b2409c4db5bdeb1e2b968bd8ab2c9d
blob 12cippa lippa

$ git cat-file 6ce9 -p
cippa lippa

$ git cat-file 6ce9 -t
blob
```

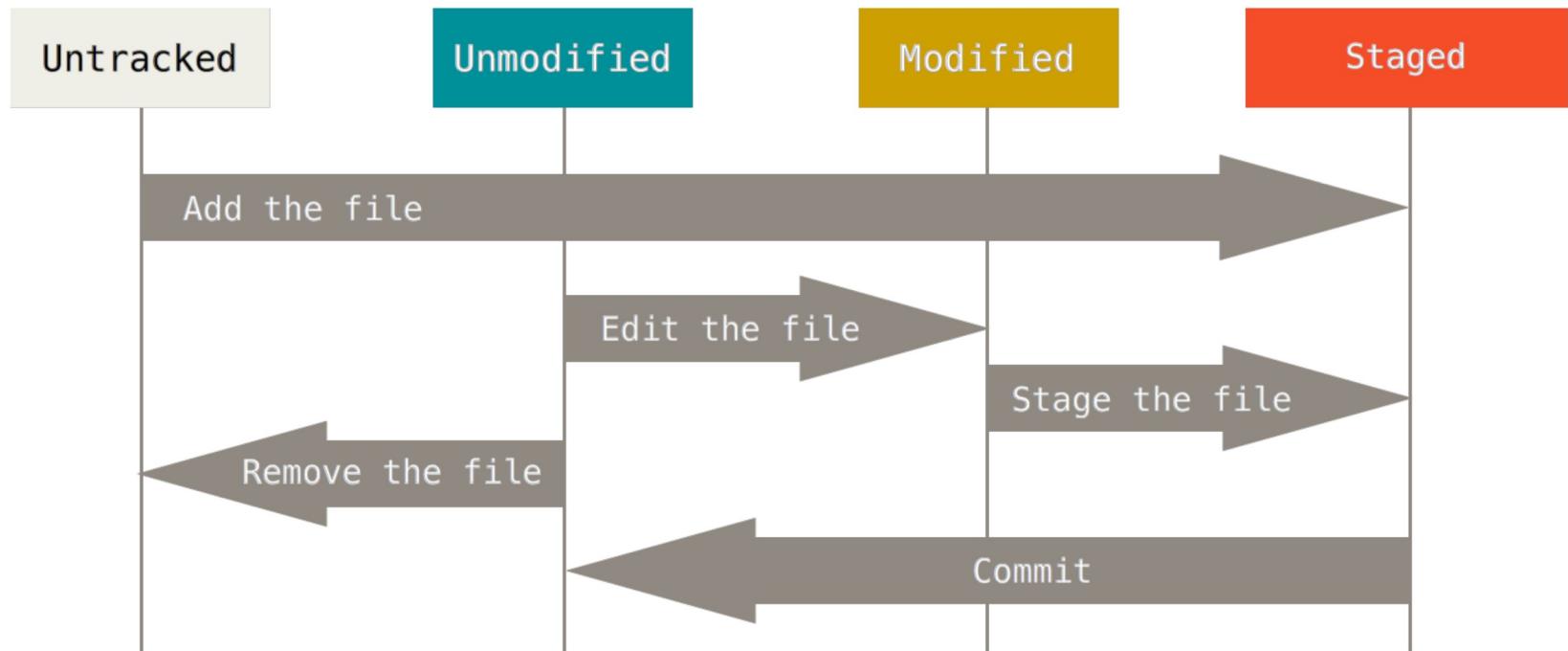
<https://git-scm.com/book/en/v2/Git-Internals-Git-Objects>

Working dir, stage, repo



- Working Directory: where your files are stored locally
- Staging Area: describes the files that will be inserted in the next commit
- Repository: contains the commits (complete history)

Working dir, stage, repo



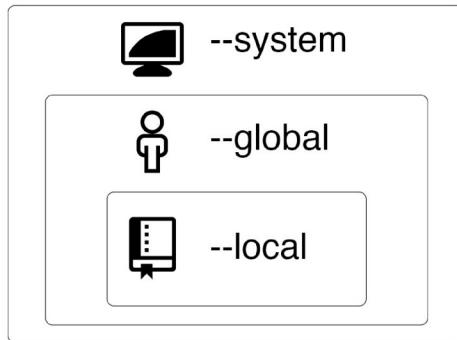
A file can be in one state at time

- Untracked: not tracked by git
- Unmodified: tracked by Git and no changes have been made
- Modified: file tracked with changes not committed
- Staged: file inserted in the staging area, ready to be committed

Working dir, stage, repo

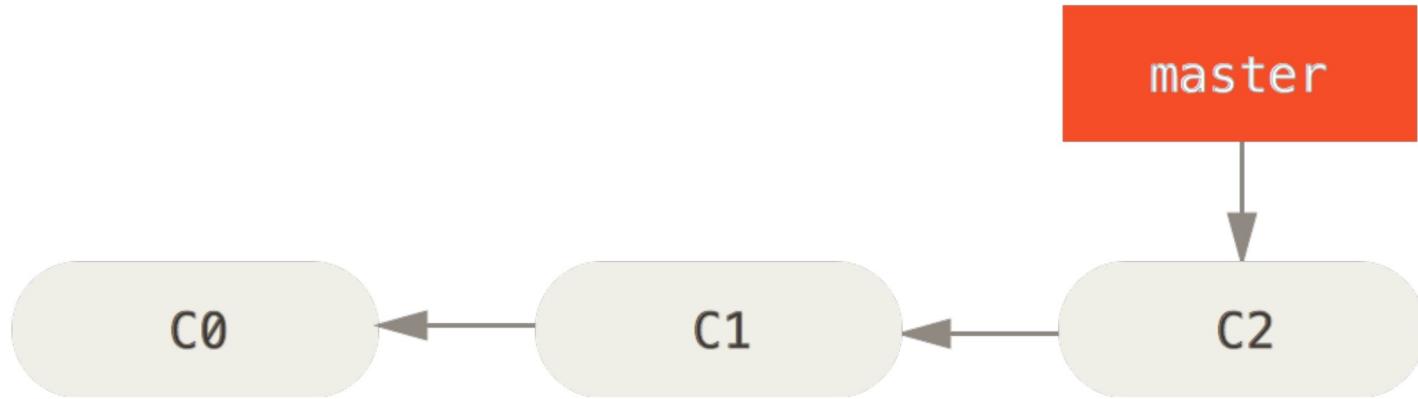


Config:



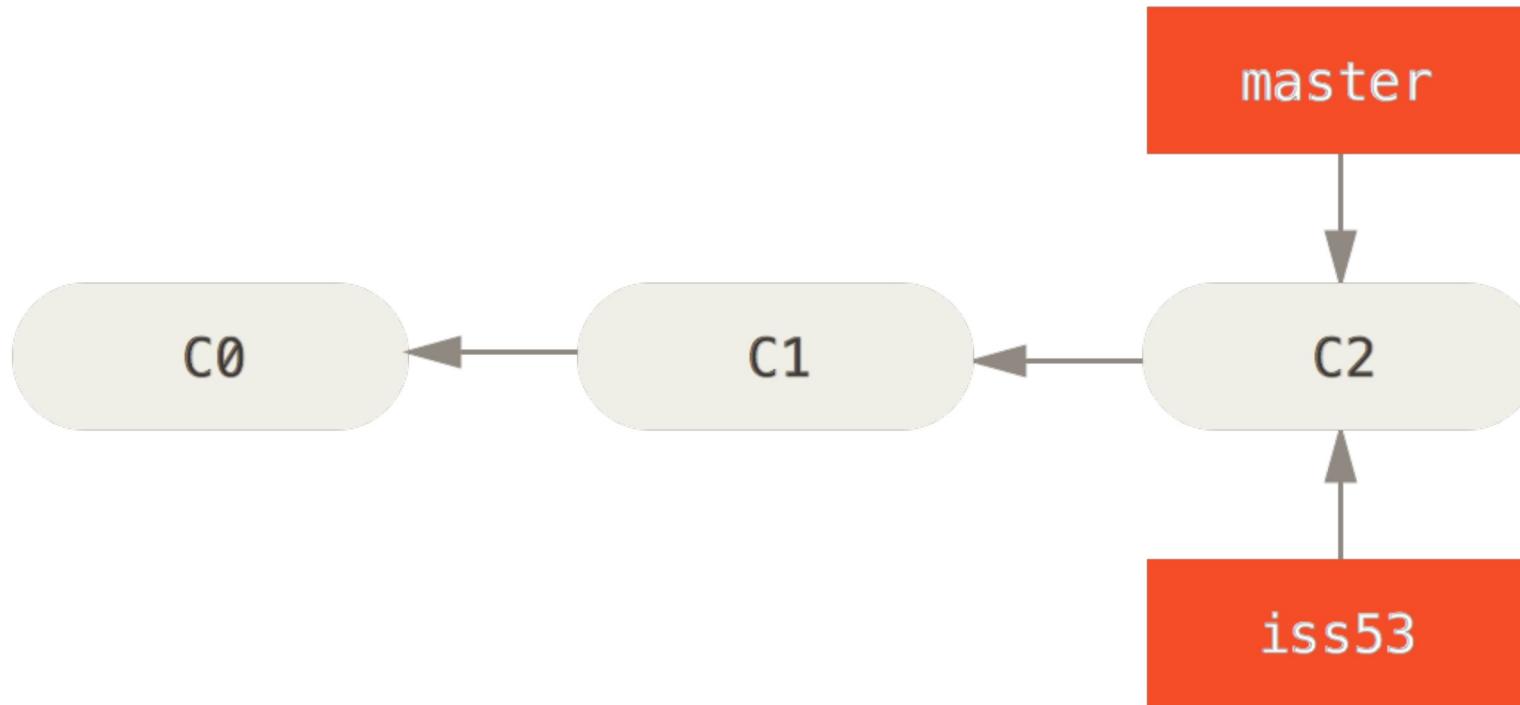
```
$ git config --list
user.email=f@test.com
user.name=Francesco Nigro
core.editor=vim
diff.tool=vimdiff
difftool.prompt=false
```

Creare e gestire un branch



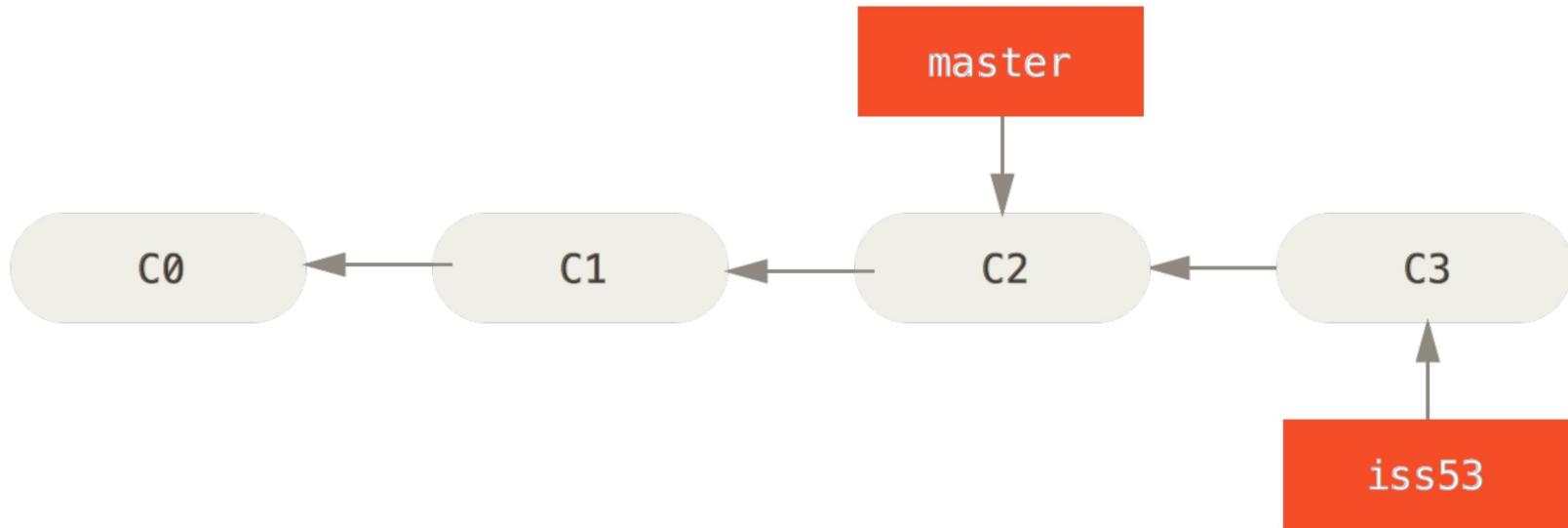
Master branch points to the last commit

Creare e gestire un branch



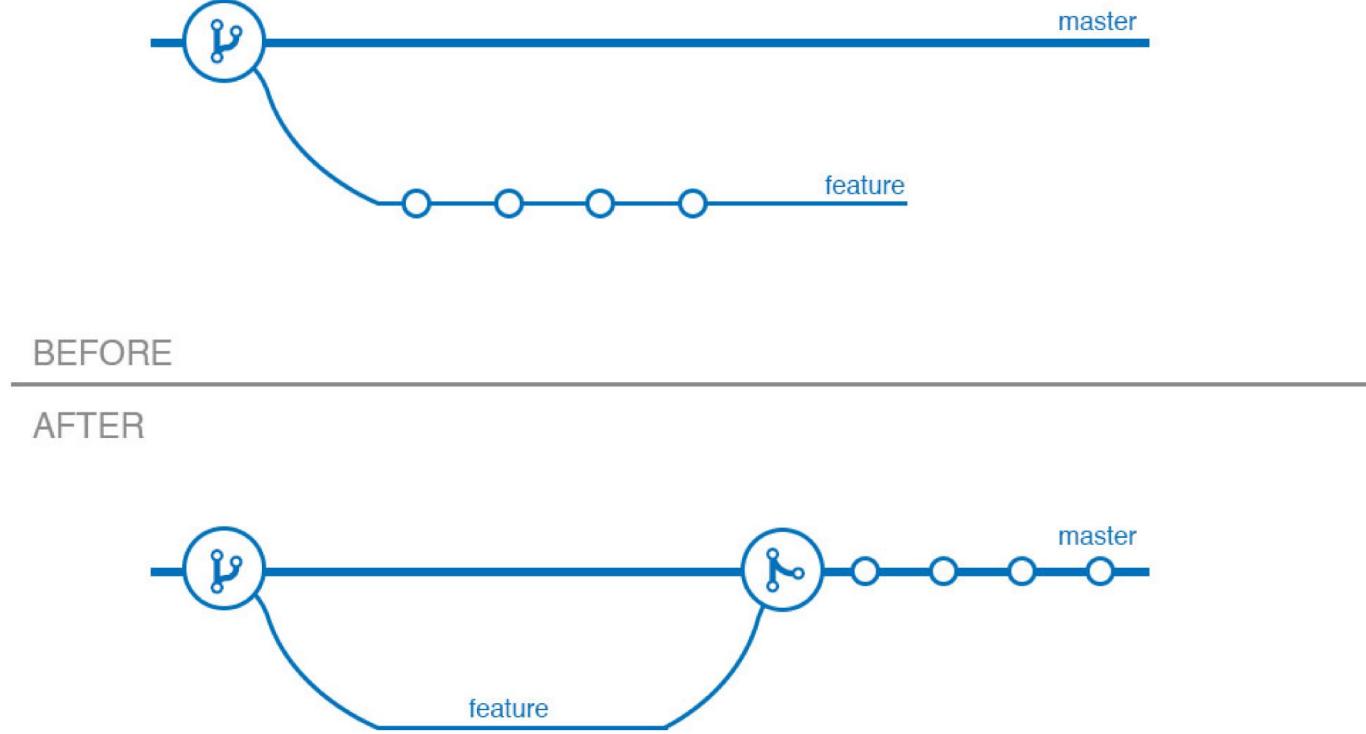
When a new branch is created it points to the same commit,
every changes made to one branch is reflected in the other until commit

Creare e gestire un branch



After commit the two branches differs

Creare e gestire un branch



Diffs



```
$ git log
commit b8778f09e6b88eae0b9a5bed45661c5375b131a5 (HEAD -> master)
[...]
commit 98226695949af836583134c2897dd6fa17cca74d
[...]
commit 4aa391a7d5356854d0e6b0aca33fd568ed3c7e03
[...]
```

Diff tra commit

```
$ git diff b8778 4aa39
```

Diff di file

```
$ git diff 4aa391a b8778f0 test.txt
```

Diff tra branch

```
$ git branch
  branch2
* master
$ git diff master branch2
```



Bye bye

Materiale utile



- Version Control Intro
https://en.wikipedia.org/wiki/Version_control
<http://git-scm.com/video/what-is-version-control>
- Sito web <http://git-scm.com>
- Manuale <http://git-scm.com/doc>
- Salviamoci da noi stessi:
<http://git-scm.com/book/en/v2/Git-Basics-Undoing-Things>
- Git Cheat Sheet
<https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>
- Command Help: `git help <command>` / `git <command> --help`
- Markdown Syntax Manual
<http://daringfireball.net/projects/markdown/syntax>