



Saverio Proto
(ZioPRoTo)

8 dicembre 2012
Fusolab 2.0

- **uname -a**
 - Io sono zioproto
 - Ho imparato git per colpa di Alessandro Gnagni a cui ho fatto da mentor per il Google Summer of Code 2012. Adesso mi tocca ringraziarlo ☺
- **Di giorno lavoro al GARR NOC**
 - La notte faccio il Ninux NOC
 - Sono nocchiere 24h ☺
- **Dimenticate il passato (svn, cvs)**
- **Basta con Facebook, oggi c'è github**
- **<http://github.com/zioproto>**



- Se avete in testa un client ed un server dimenticateveli
- Git init
 - La cartella .git contiene tutto il repository
 - Tutto in locale
- Meglio capire l'object store di git come è fatto
- Object type
 - Blobs: binary large object
 - Trees: represents one lever of directory information
 - Commits: Metadata pointing to a tree
 - Tags: human name connected to a object (usually a commit)

- each object in the object store has a unique name produced by applying SHA1 to the contents of the object
 - SHA1 = 160bit digest

```
git init
```

```
ls .git/
```

```
ls .git/objects/
```

```
touch file
```

```
git add file
```

```
git write-tree
```

```
ls .git/objects/
```

```
git cat-file -p df2b8
```



■ The git Index

- The index is a temporary and dynamic binary file that describes the directory structure of the entire repository. you execute Git commands to stage changes in the index. Changes usually add, delete, or edit some file or set of files. The index records and retains those changes, keeping them safe until you are ready to commit them

■ L'index bisogna essere coscienti che esiste per usare:

- Git add
- Git rm
- Git mv

■ I files possono essere:

– Tracked

- Any file staged in the index. Use git add to track files

– Ignored

- File that git ignores because specified in the .gitignore file

– Untracked

- Not tracked, not ignored

- **Commit introduces a change in the repository**
 - Un commit è freezare uno stato dell'index e mettere metadati
 - Commit by user
 - Commit by git itself (merge)
- **Access a commit with explicit or relative reference**
 - Git checkout ORIG_HEAD
 - Git checkout aefc6f
 - Git checkout HEAD~1
- **Refs and symrefs**
 - They are names that point to git objects
 - HEAD ORIG_HEAD FETCH_HEAD

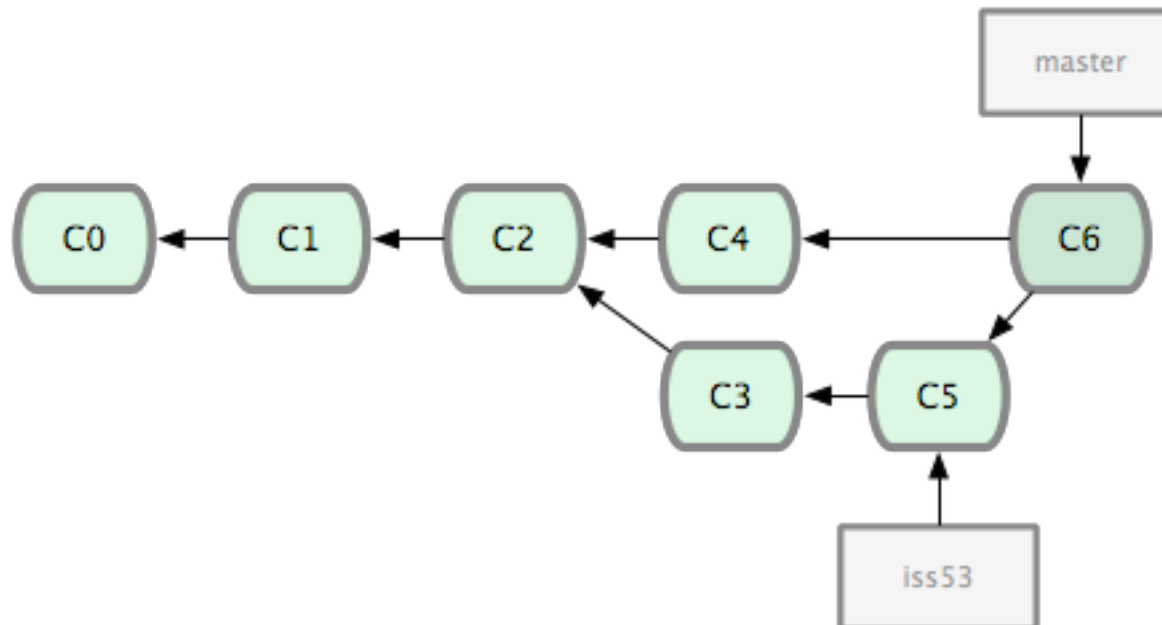
- **Un commit conosce i suoi genitori !**
 - E non conosce i suoi figli !



- **Git branch**
 - Nessuno ve lo spiega così ma è solo un ref che punta a un commit
- **Git checkout**
 - Carica un branch nella working directory
 - La working directory lavora su un branch alla volta
- **Quando committate su un branch il ref del branch si sposta al nuovo commit**
- **Cancellare le branch**
 - Cancellate solo il ref
 - Ogni tanto c'è un garbage collector che fa pulizia

- La cosa difficile sono le cose implicite
- Git diff
 - Tra working directory e index
- Git diff <commit>
 - Tra working directory e commit
 - Se mettete -cached allora tra index e commit
- Git diff <commit1> <commit2>
 - Tra i due commit
- Git diff <branch1>..<branch2>
 - Ve lo spiego a voce che si potrebbe scrivere un capitolo su quei due puntini
 - Infatti lo hanno scritto: “git diff and Commit Ranges”

- Current branch is always the target branch
- You can merge with conflicts
- A merge produces a new commit with maybe two parents



- Git cherry-pick and git revert and easy and usefull tools !



■ Altering commits

- Fallo solo su cose locali che sono solo sul tuo computer



■ Git rebase

- Cambia base ad una serie di patches

■ Git rebase interactive

- Git rebase –i
- Mi permette di fare squash dei commit inutili
- Mi permette di essere un vero git ninja !



- **Git reset –soft <commit>**
 - –soft changes the HEAD ref to point to the given commit
- **Git reset –mixed <commit>**
 - This is the default behaviour
 - –mixed changes HEAD to point to the given commit. Your index contents are also modified to align with the tree structure named by commit, but your working directory contents are left unchanged.
- **Git reset –hard <commit>**
 - This variant changes the HEAD ref to point to the given commit. The contents of your index are also modified to agree with the tree structure named by the named commit. Furthermore, your working directory contents are changed to reflect the state of the tree represented by the given commit.

- Adesso siete pronti per fare casino con gli altri

- Git remote
 - Gestisce i collegamenti ai repository remoti
 - Di fatto voi avete tutta la copia del remote in pancia come fosse una branch

- Git clone
 - Per iniziare

- Git push

- Git pull (meglio fare prima Git fetch)

- Git blame

- Git bisect

```
saverio@nockid:~$ cat .gitconfig
```

```
[user]
```

```
name = Saverio Proto
```

```
email = zioproto@gmail.com
```

```
[core]
```

```
editor = vim
```

```
[color]
```

```
diff = auto
```

```
branch = auto
```

```
status = auto
```

```
[alias]
```

```
superlog = log -graph -all -pretty=oneline -decorate
```

Fate un offerta



Se vi è piaciuto il corso lasciate qualche soldo così ne possiamo organizzare altri

Questions ?

