Young
Talent in Action
#crediamoneltalentodelfuturo

ManpowerGroup™

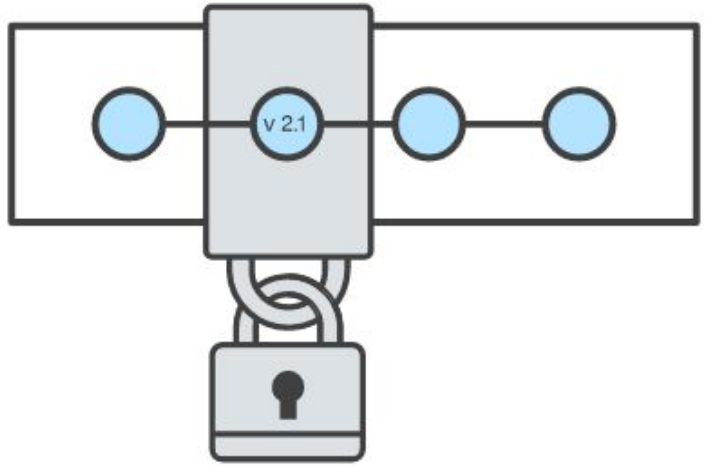
Git e Github

Programmazione Java - Modulo 9

Nicola Atzei

Young
Talent in Action
#crediamoneltalentodelfuturo

Version control systems



Young Talent in Action
#crediamoneltalentodelfuturo

Cosa sono

I *version control systems* (lett. sistemi di controllo di versione) sono applicativi software per lo sviluppo di programmi in maniera collaborativa.

Tengono traccia di **ogni modifica fatta nel codice**, registrando:

- l'autore della modifica
- data e ora
- quali file ha modificato
- quali righe ha modificato
- ...

Young
Talent in Action
#crediamoneltalentodelfuturo

Benefici

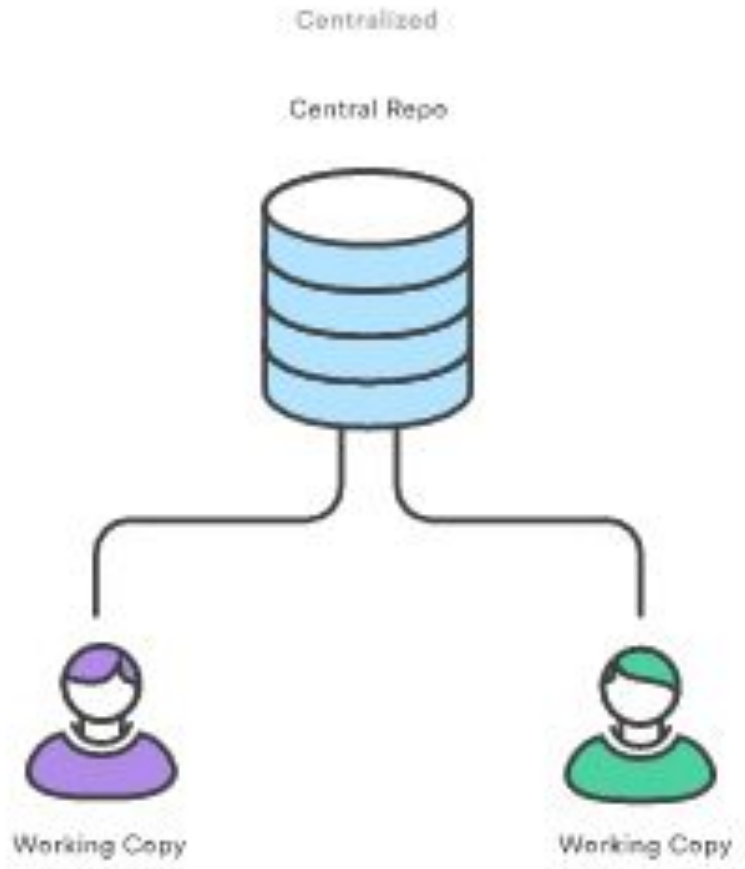
- ripristinare una versione precedente
- proteggere il codice da "catastrofi naturali"
- sviluppo simultaneo dello stesso programma
  - Topolino sviluppa la funzionalità X
  - Pippo sviluppa la funzionalità Y
  - Bugs Bunny scrive le classi di test per X e Y
- code review
  - Paperino controlla l'implementazione di X prima di integrarla nel programma principale

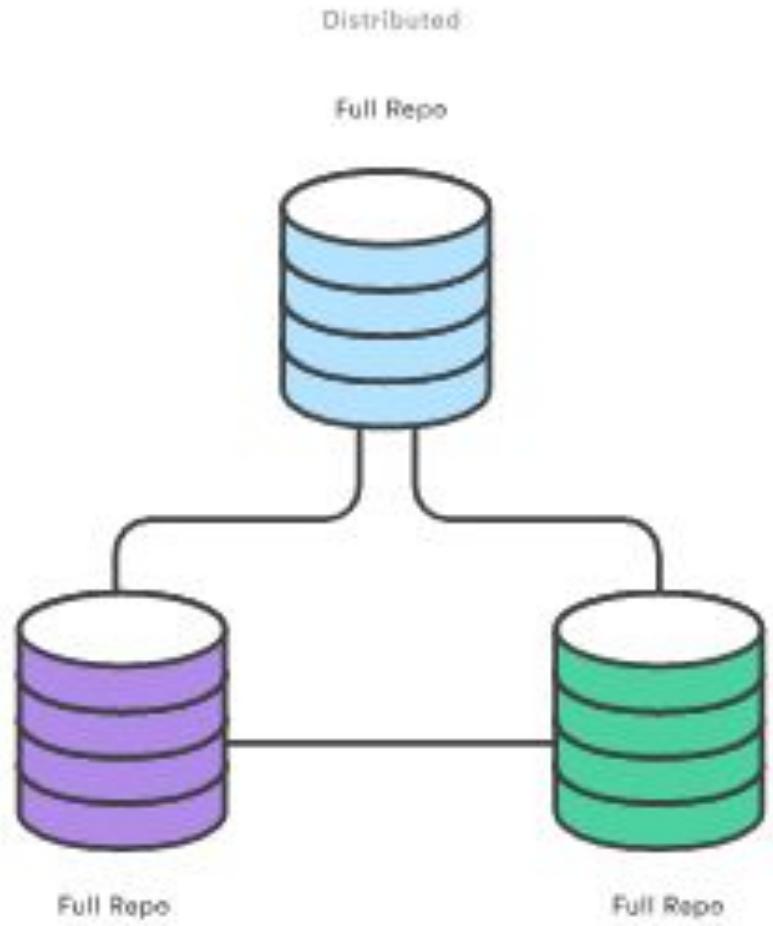
Young
Talent in Action
#crediamoneltalentodelfuturo

2005, Linus Torvalds

Young
Talent in Action
#crediamoneltalentodelfuturo

Distributed development





Young
Talent in Action
#crediamoneltalentodelfuturo

Installazione

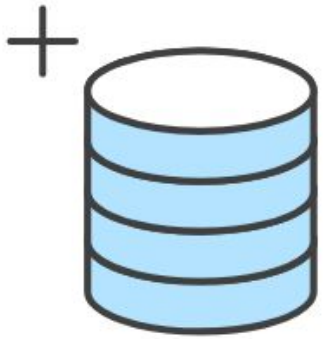
https://www.atlassian.com/git/tutorials/install-git#windows

Configurazione

```
git config --global user.name "Nicola Atzei"
git config --global user.email "atzeinicola@gmail.com"
git config --global alias.graph "log --all --abbrev-commit --decorate --graph"
```

Young
Talent in Action
#crediamoneltalentodelfuturo

Il nostro primo repository



Young Talent in Action
#crediamoneltalentodelfuturo

Cosa è un repository?

Un repository è uno storage virtuale, ossia il contenitore dentro il quale metteremo tutti i nostri file

Concretamente sarà una **cartella** nel vostro computer

Young Talent in Action
#crediamoneltalentodelfuturo

Creazione di un nuovo repository

Creiamo una nuova cartella

Apriamo un terminale

Entriamo nella cartella:

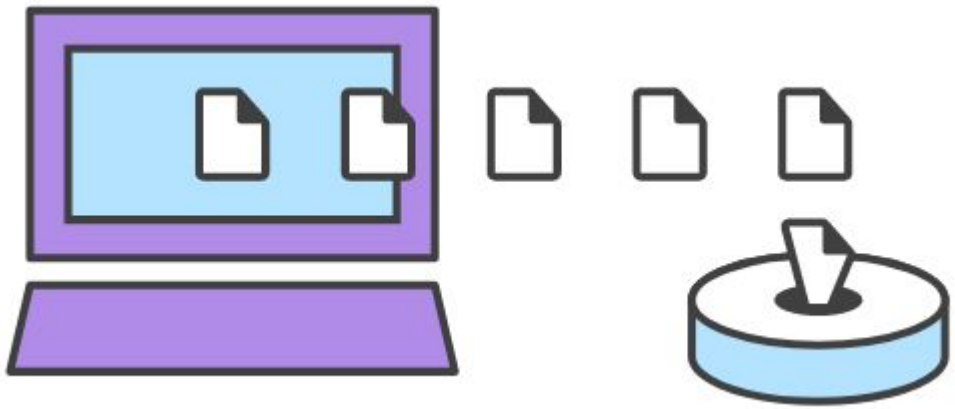
```
cd  /path/della/cartella/
dir /path/della/cartella/
```

Eseguiamo i comandi

```
git init
git status
```

Young
Talent in Action
#crediamoneltalentodelfuturo

Il nostro primo commit

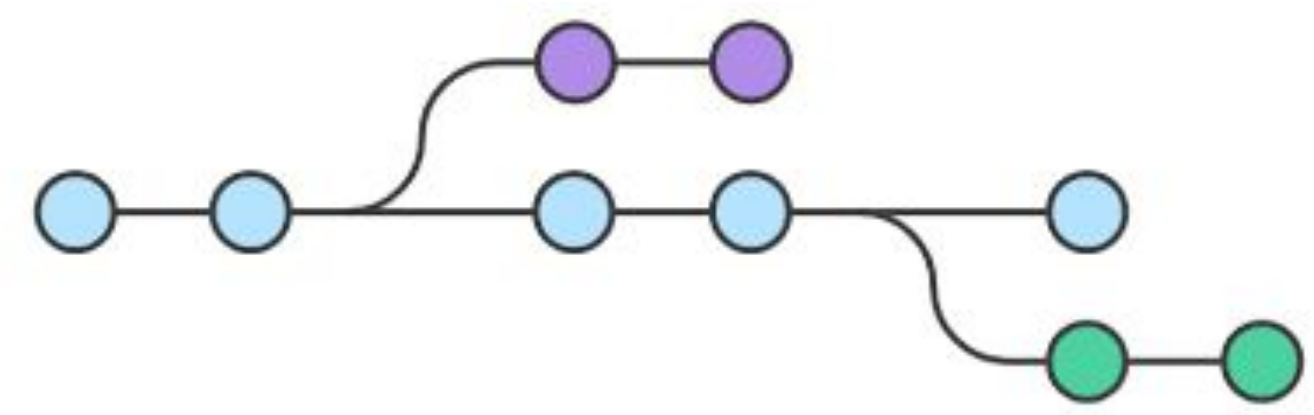


Young
Talent in Action
#crediamoneltalentodelfuturo

Commit

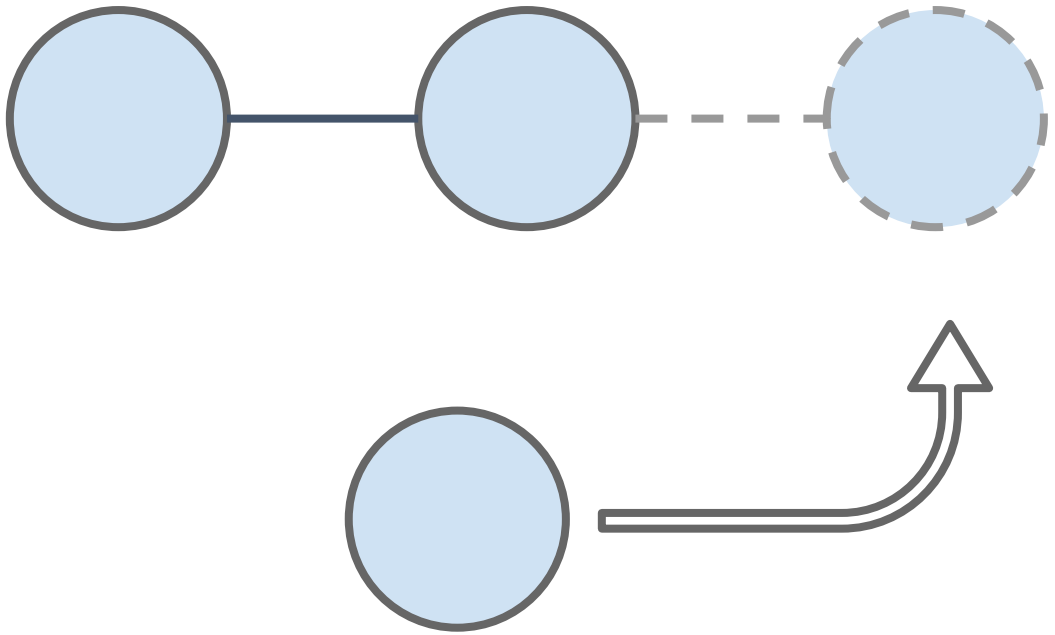
Un *commit* è un'istantanea (*snapshot*) del nostro repository

Quando facciamo delle modifiche, creiamo un nuovo commit per tenere traccia dei cambiamenti



Young
Talent in Action
#crediamoneltalentodelfuturo

Commit



Young Talent in Action
#crediamoneltalentodelfuturo

Esempio

- Creiamo un nuovo file `README.txt`
- Scriviamo dentro il file qualcosa a piacere. Ad esempio:
  `Leggere attentamente le seguenti istruzioni prima di procedere...`
- Salviamo
- Controlliamo lo stato del repository con
  `git status`

Il file appena creato comparirà come ***Untracked***

Young
Talent in Action
#crediamoneltalentodelfuturo

Ci siamo quasi

- Diciamo a GIT di tracciare il file con
  `git add README.txt`

- Controllate nuovamente lo stato del repository
  `git status`

Il file adesso è pronto per far parte del commit

Young
Talent in Action
#crediamoneltalentodelfuturo

L'ultimo sforzo

- A questo punto possiamo creare il nostro commit
  `git commit`

Si aprirà un editor di testo per inserire una descrizione del nostro commit. Ad esempio scriviamo:

`Creazione di README.txt`

Salviamo e chiudiamo l'editor. Fatto!

Young Talent in Action
#crediamoneltalentodelfuturo

Visualizzare i commit

Se controlliamo lo stato del repository, vedremo che non compare nessun file. Come visualizziamo i commit?

```
git log --all --abbrev-commit --decorate --graph
```

o semplicemente

```
git graph
```

(grazie all'alias che abbiamo creato prima)

Young Talent in Action
#crediamoneltalentodelfuturo

Saltare da un commit all'altro

```
git checkout <hash del commit>
```

Ad esempio:

```
git checkout e0fd8b8
```

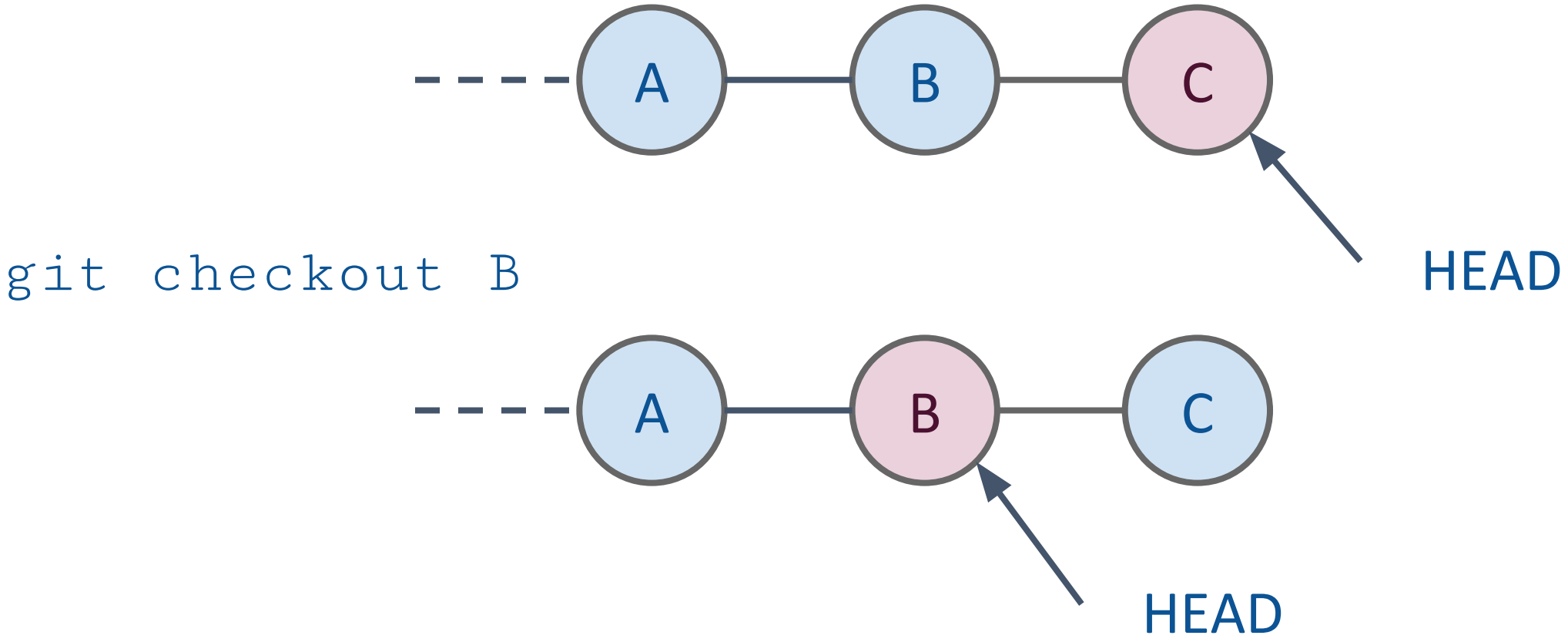
Young
Talent in Action
#crediamoneltalentodelfuturo

HEAD

Punta al commit che state visualizzando.
Quando saltate da un commit all'altro, HEAD si sposta di conseguenza.

Young
Talent in Action
#crediamoneltalentodelfuturo

Checkout



Young Talent in Action
#crediamoneltalentodelfuturo

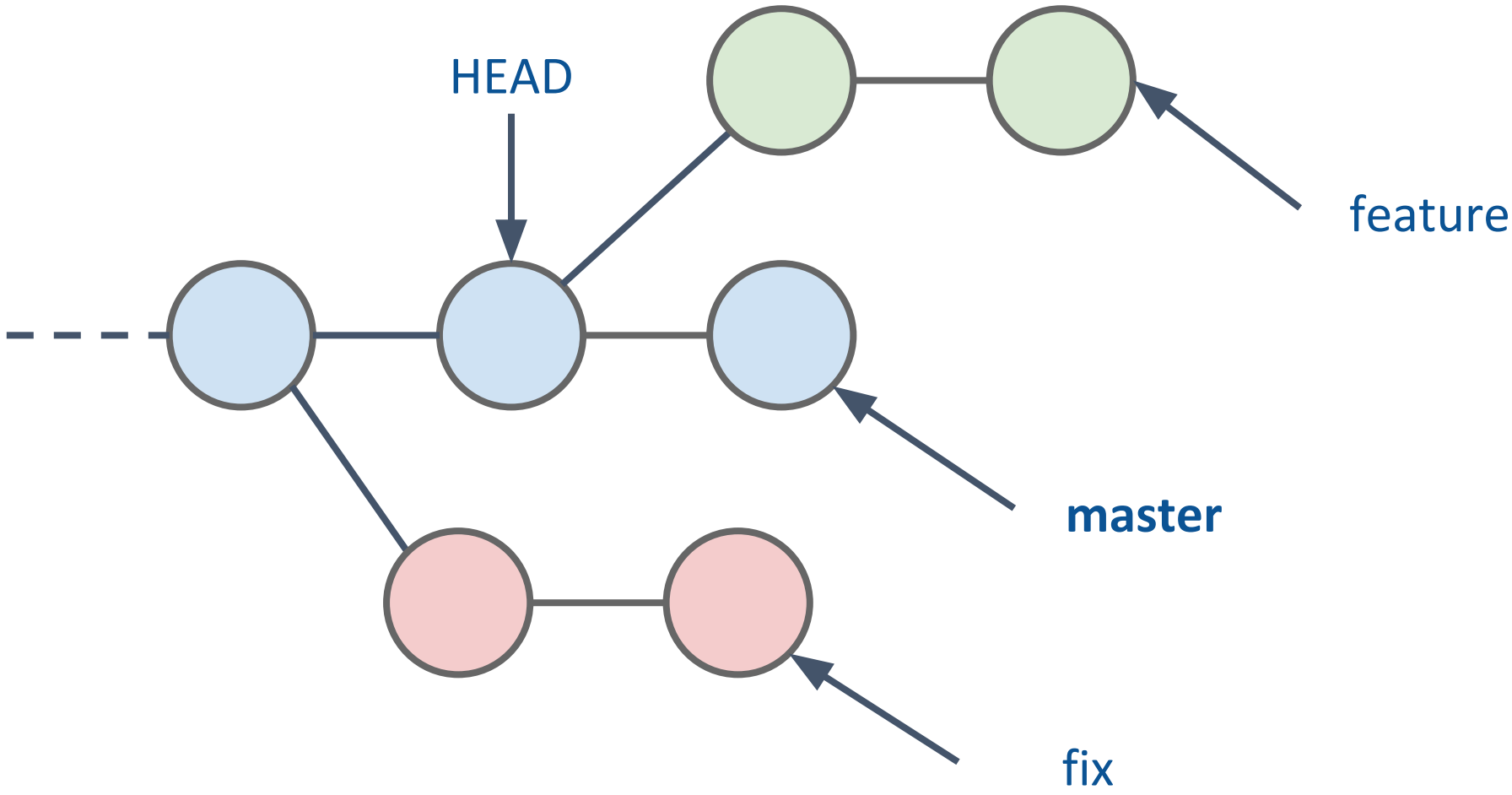
Altre funzionalità

- riordinare i commit*
- rimuovere commit*
- modificare commit*
- taggare un commit
- ...

* Queste operazioni sono sconsigliate quando si lavora in gruppo sullo stesso repository

Young
Talent in Action
#crediamoneltalentodelfuturo

Branches



Young
Talent in Action
#crediamoneltalentodelfuturo

Repository remoto

Young
Talent in Action
#crediamoneltalentodelfuturo

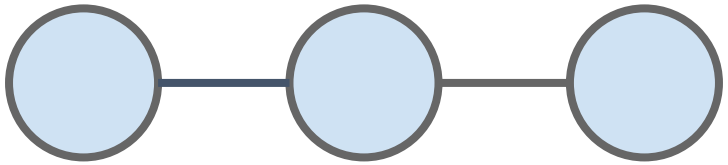
Repository remoto

Un repository può essere **remoto**, ossia memorizzato su un server.

Un repository remoto può essere **clonato** localmente. La copia locale può essere modificata liberamente aggiungendo nuovi commit e infine **sincronizzata** con la copia remota, che riceverà i nuovi commit.

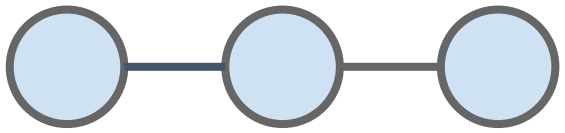
Remote repository

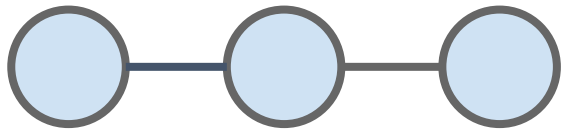
Repository remoto



clone

clone

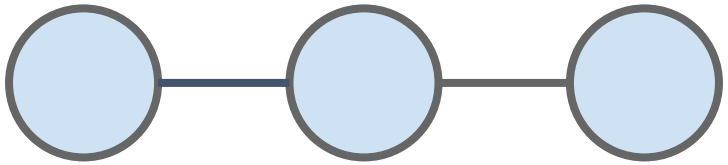


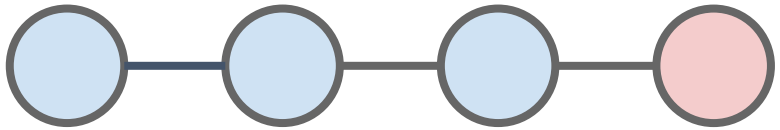


Young
Talent in Action
#crediamoneltalentodelfuturo

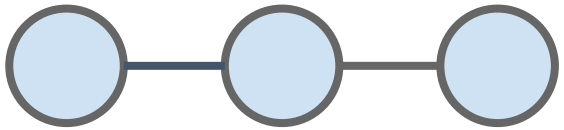
Remote repository

Repository remoto





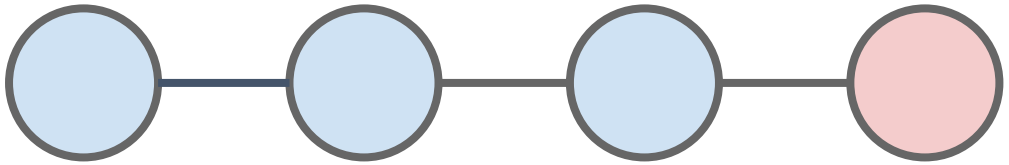
commit



Young Talent in Action
#crediamoneltalentodelfuturo

Remote repository

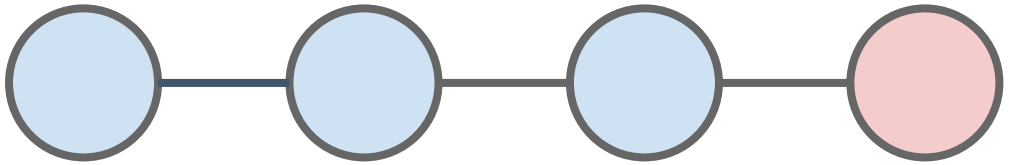
Repository remoto



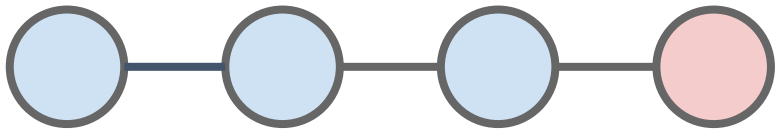
Young
Talent in Action
#crediamoneltalentodelfuturo

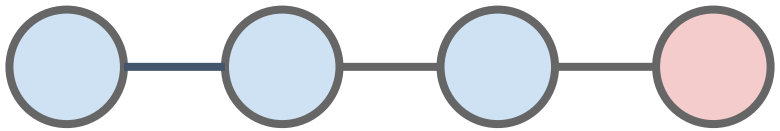
Remote repository

Repository remoto



fetch





Young
Talent in Action
#crediamoneltalentodelfuturo

Operazioni di base

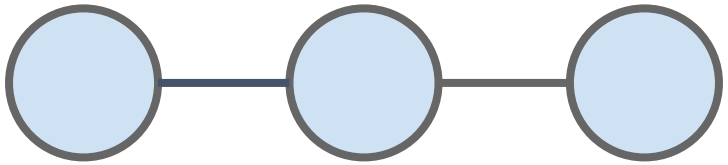
PUSH: invia la propria versione del repository al server remoto

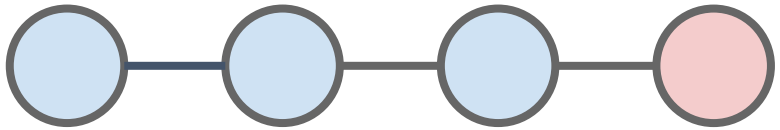
FETCH: ricevi i cambiamenti dal repository remoto

Young
Talent in Action
#crediamoneltalentodelfuturo

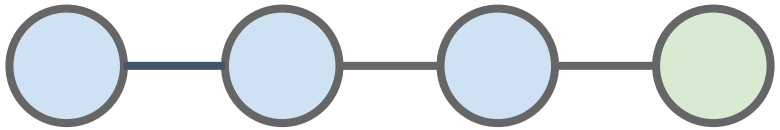
Remote repository

Repository remoto





commit

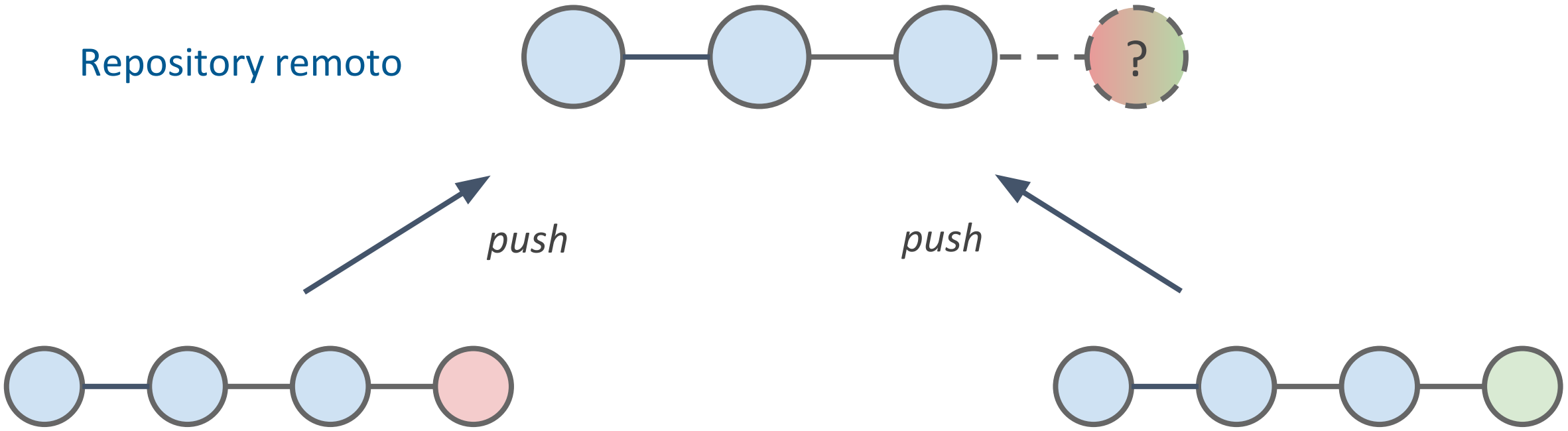


commit

Young Talent in Action
#crediamoneltalentodelfuturo

Remote repository

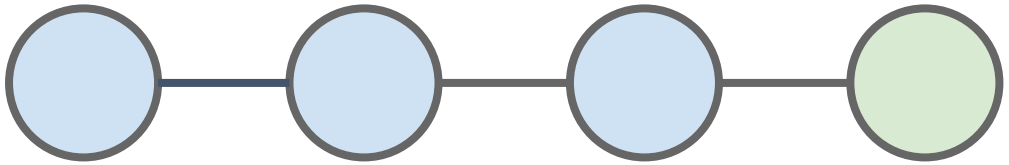
Repository remoto



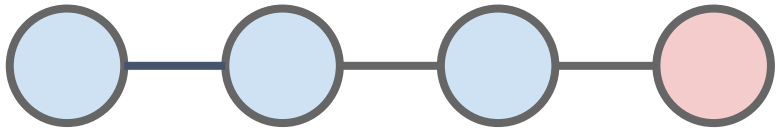
Young
Talent in Action
#crediamoneltalentodelfuturo

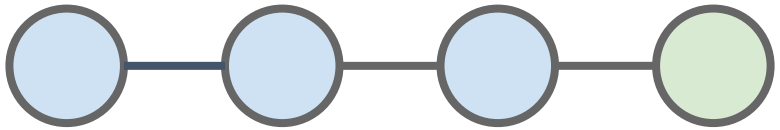
Remote repository

Repository remoto



solo un push vince, l'altro fallisce

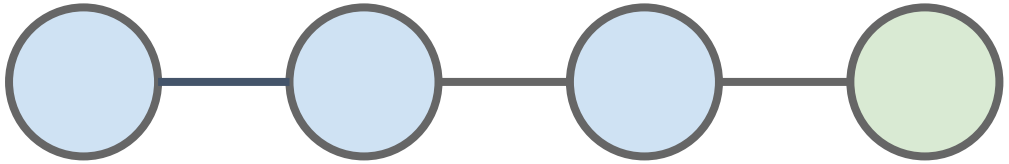


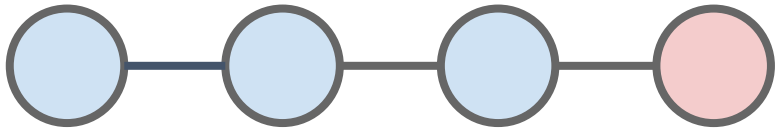


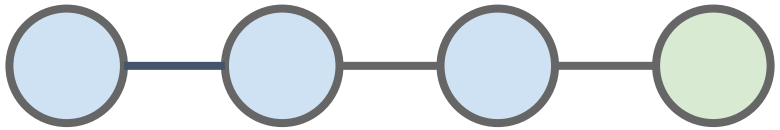
Young
Talent in Action
#crediamoneltalentodelfuturo

Remote repository

Repository remoto





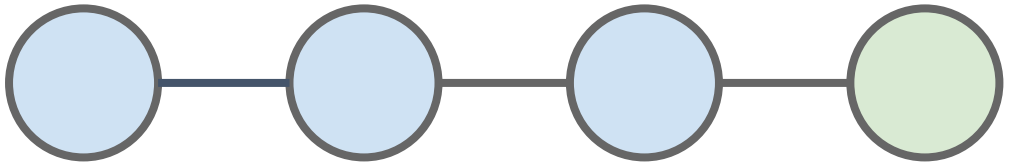


come faccio?

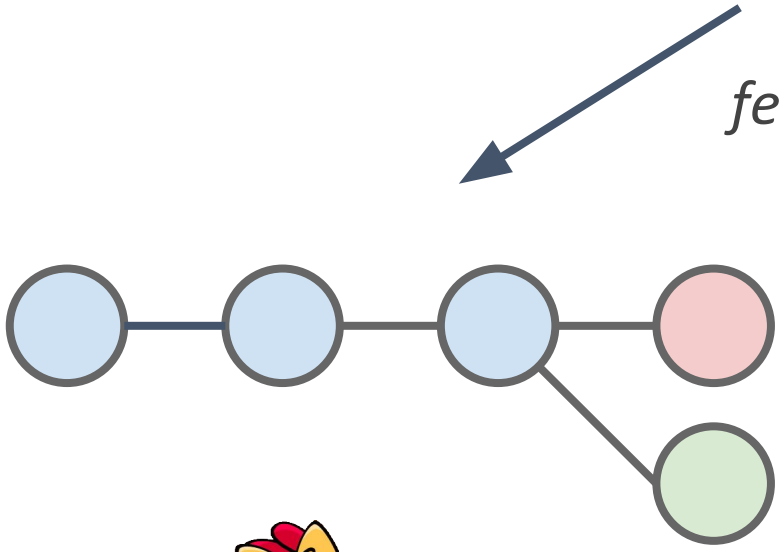
Young
Talent in Action
#crediamoneltalentodelfuturo

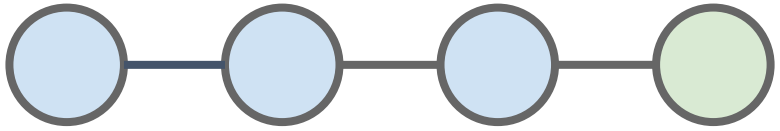
Remote repository

Repository remoto



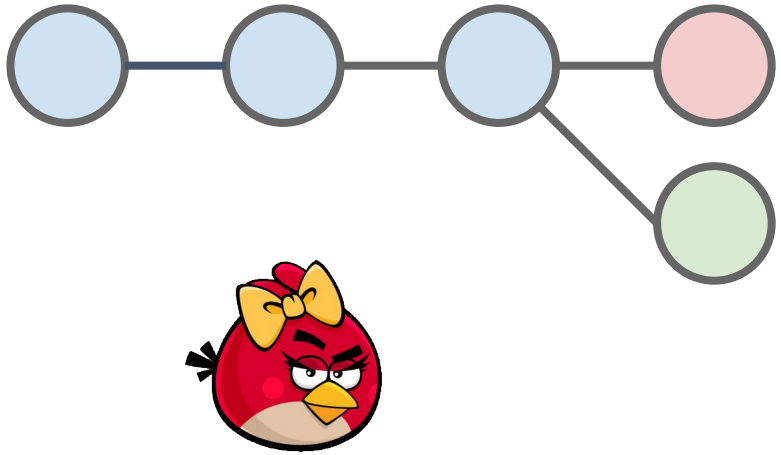
fetch





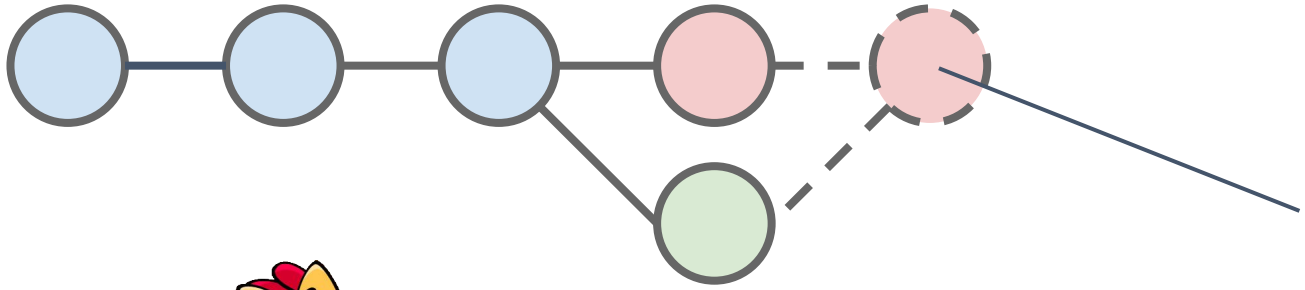
Young
Talent in Action
#crediamoneltalentodelfuturo

Remote repository



Devo portare le mie modifiche
davanti al commit verde

Merge

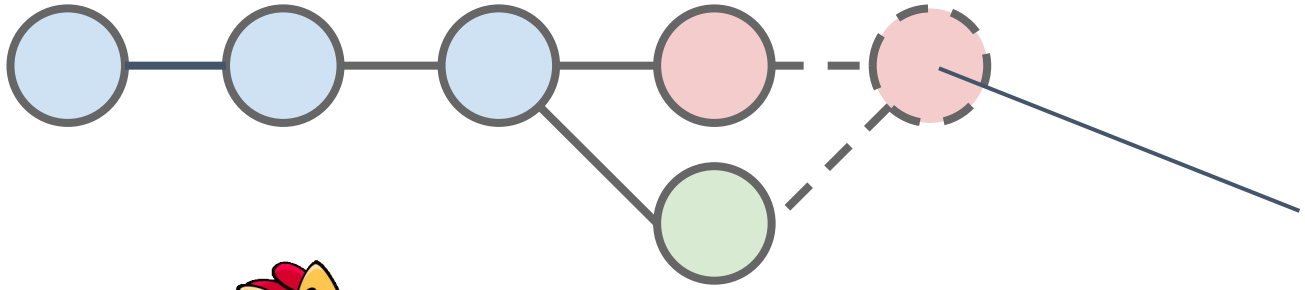


commit di merge
(sospeso)

Cosa succede se entrambi hanno modificato gli stessi file?

GIT riesce a mettere insieme le modifiche, fintanto che esse **riguardino righe diverse.**

Merge



commit di merge
(sospeso)

CONFLITTI!

GIT indicherà le righe che sono state modificate da entrambi, chiedendovi di risolvere il problema

Young Talent in Action
#crediamoneltalentodelfuturo

Riepilogo dei comandi

`git clone http://..`

clona un repository remoto, che viene chiamato **origin**

`git fetch origin`

scarica le modifiche dal repository remoto

`git push origin master`

invia a **origin** le modifiche relative al ramo **master**

Young Talent in Action
#crediamoneltalentodelfuturo

Riepilogo dei comandi

`git merge <commit>`

fonde le modifiche del commit specificato all'interno del ramo corrente (ad es. master)

`git merge --continue`

dopo che abbiamo risolto i conflitti, conclude il merge

`git merge --abort`

se il merge presenta dei conflitti, posso abortire l'operazione

Young
Talent in Action
#crediamoneltalentodelfuturo

GitHub

Young
Talent in Action
#crediamoneltalentodelfuturo

Reference

https://www.atlassian.com/git/tutorials