

# git

Una gestione più elegante del codice



# Il progetto tipico

La problematica più comune legata alla gestione  
dei sorgenti

# I progetti contengono diversi file



.idea



assets



libs



res



src



tests



.gitignore



Android.mk



AndroidManifest.xml



CHANGELOG.md



CleanSpec.mk



MODULE\_LICENSE\_APACHE2



NOTICE



proguard.flags



README.md

# Archiviazione del progetto con tar



# Difficile gestire diverse versioni





# Soluzione:

Utilizziamo un software per il Controllo versione



# Cosa è il Controllo versione

- Tenere **traccia** delle **modifiche** effettuate su un insieme di file.
- Gestisce **versioni** multiple
- Permette agli sviluppatori di **collaborare** alle medesime risorse.



Version 1.0



Version 1.1



Version 2.0



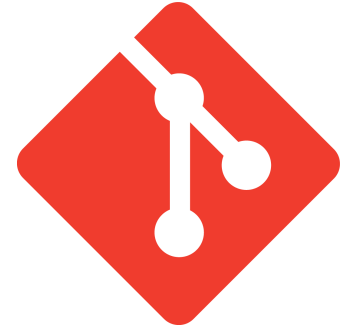
Version 2.5



Version 3.0

# Quali sono le soluzioni disponibili?

---



**Concurrent  
Versions System**

**Subversion**

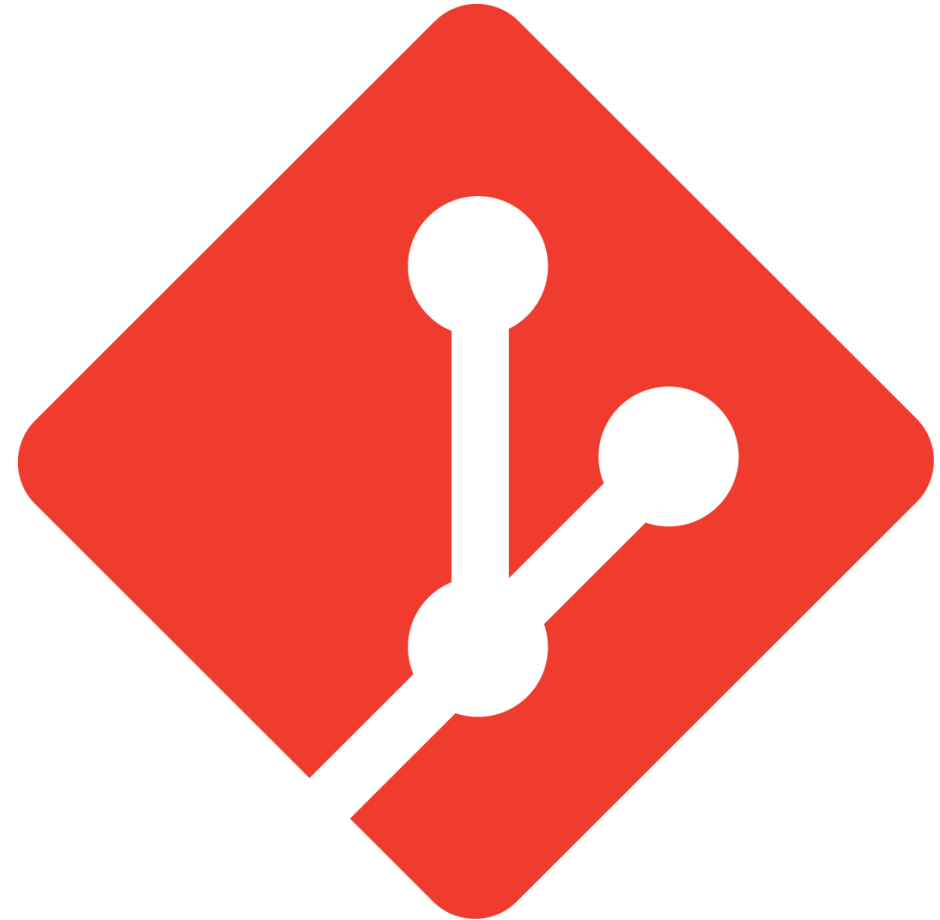
**Mercurial**

**Git**



# Cos'è Git

- Creato nel 2005 da **Linus Torvalds**
- Creato per la **scarsa** capacità di progetti datati nel gestire **grossi** progetti come quello del **Kernel**
- Non è necessario un **server** centrale



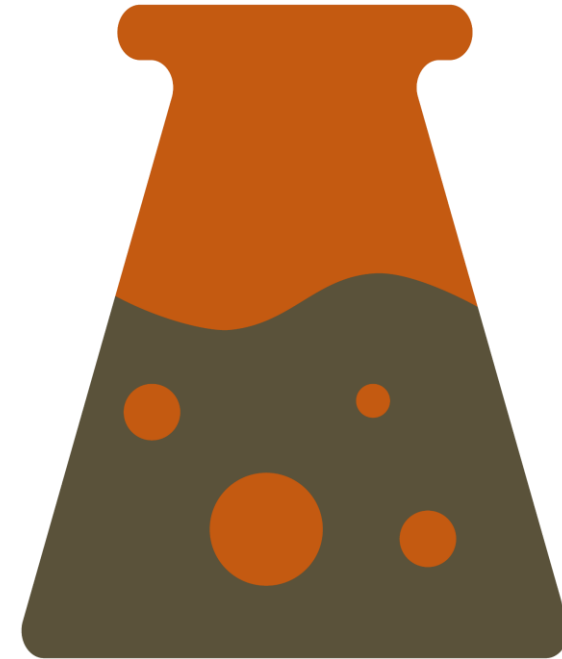
# Perché Git

- Sviluppo distribuito
  - Lavoro offline
- Modello di sviluppo ramificato
- Vasta community
  - Ottimo supporto
- Integrabile in altri strumenti
- Ciclo sviluppo più rapido



# Da chi è usato

- Kernel Linux
- Distribuzioni Linux
- Progetti open source
- Progetti closed source
- Documentazioni
- Windows(!)





# Glossario

Impariamo la terminologia di Git



# Repository

- **L'archivio** dove i file sono memorizzati
- Repository **Locale**
  - Unico
  - Rappresenta la **copia** sul PC



# Repository

- Repository **Remoto**
  - Servizi di **hosting**
    - Es. GitHub
  - Possono essere **uno o più**
  - Il principale si chiama **origin**



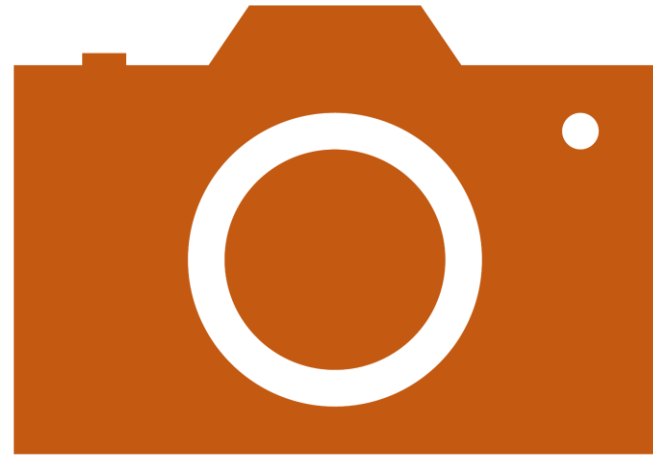
# Commit

- Un **insieme** di **modifiche** viene chiamato **commit**
- Ogni commit ha un suo **identificativo**
- Viene effettuato sui file in **staging**



# Commit come istantanea

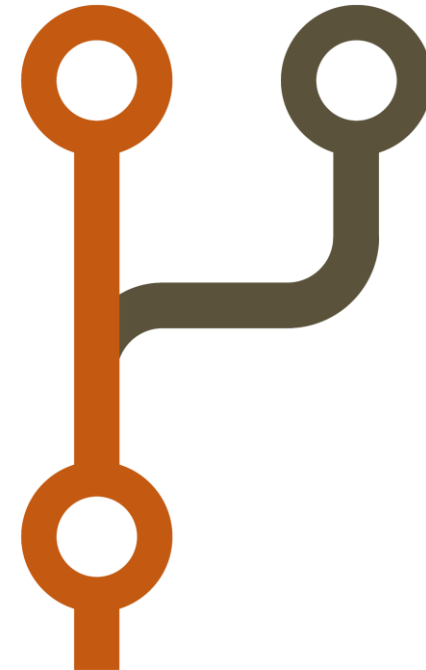
- «**Foto**» del progetto
- Stato **corrente** salvato all'interno dello **storico**





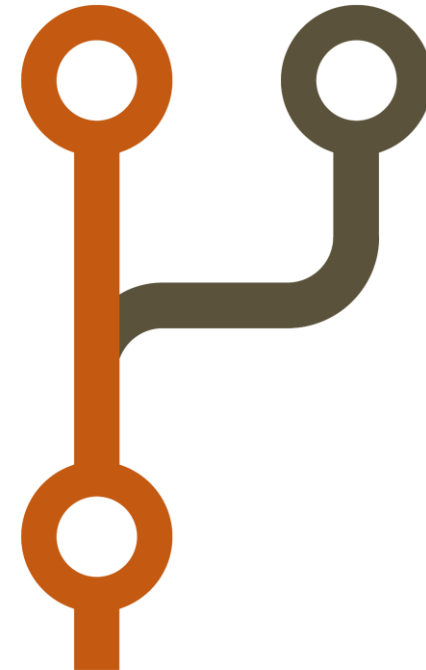
# Branch o rami

- Più **rami** per ogni repository
- Vengono **creati** mediante una **fork** (git branch)
- Ramo **principale**
  - Chiamato **master**
  - Versione **funzionante** del programma



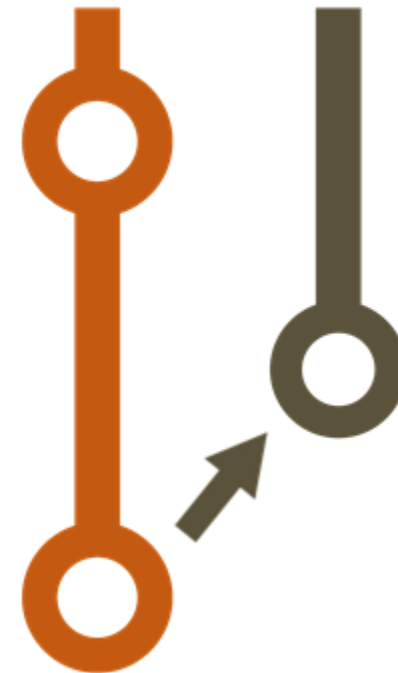
# Branch o rami

- Ramo di **sviluppo**
  - **Feature** branch
  - Aggiunta di codice in fase di «**Work in progress**»
  - Diversi **stadi** dello **stesso** progetto
- Ramo di Versione
  - «**Snapshot**» di una versione



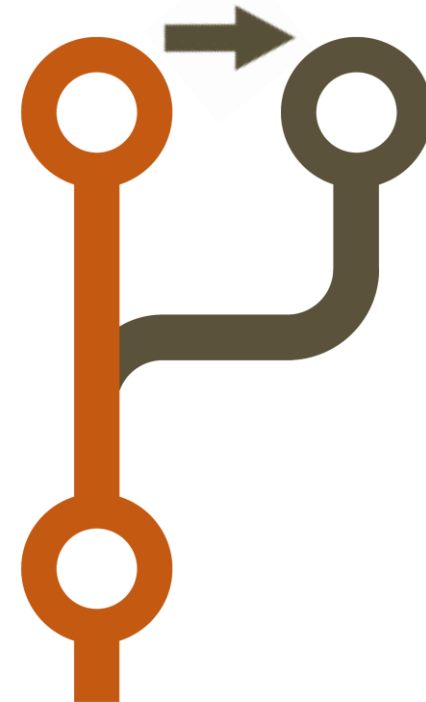
# Branch

- «Git branch nomenclatura»
- Creazione di un nuovo ramo
- Posizione in base al commit attuale
- Posso avere più rami allo stesso punto



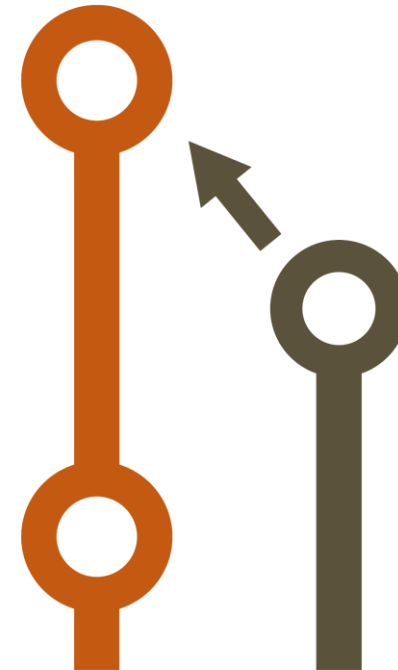
# Checkuout

- Passaggio da un ramo esistente a un altro esistente



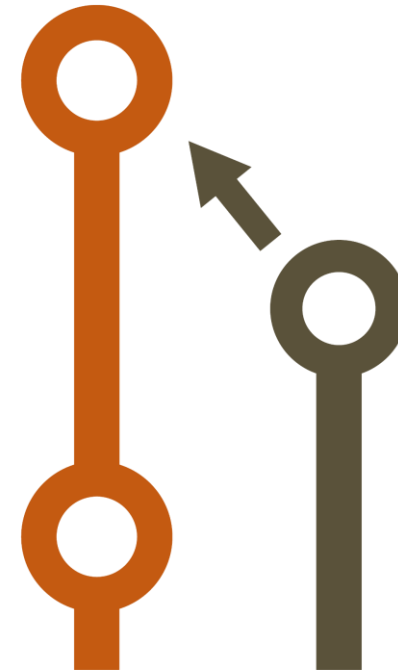
# Merge

- Usato spesso per **fondere** un ramo con un altro
- Creazione di un **commit** che contiene i commit dei rami da **unire**
- Git prova a fare un merge **automatico** ed incorporare le **modifiche**



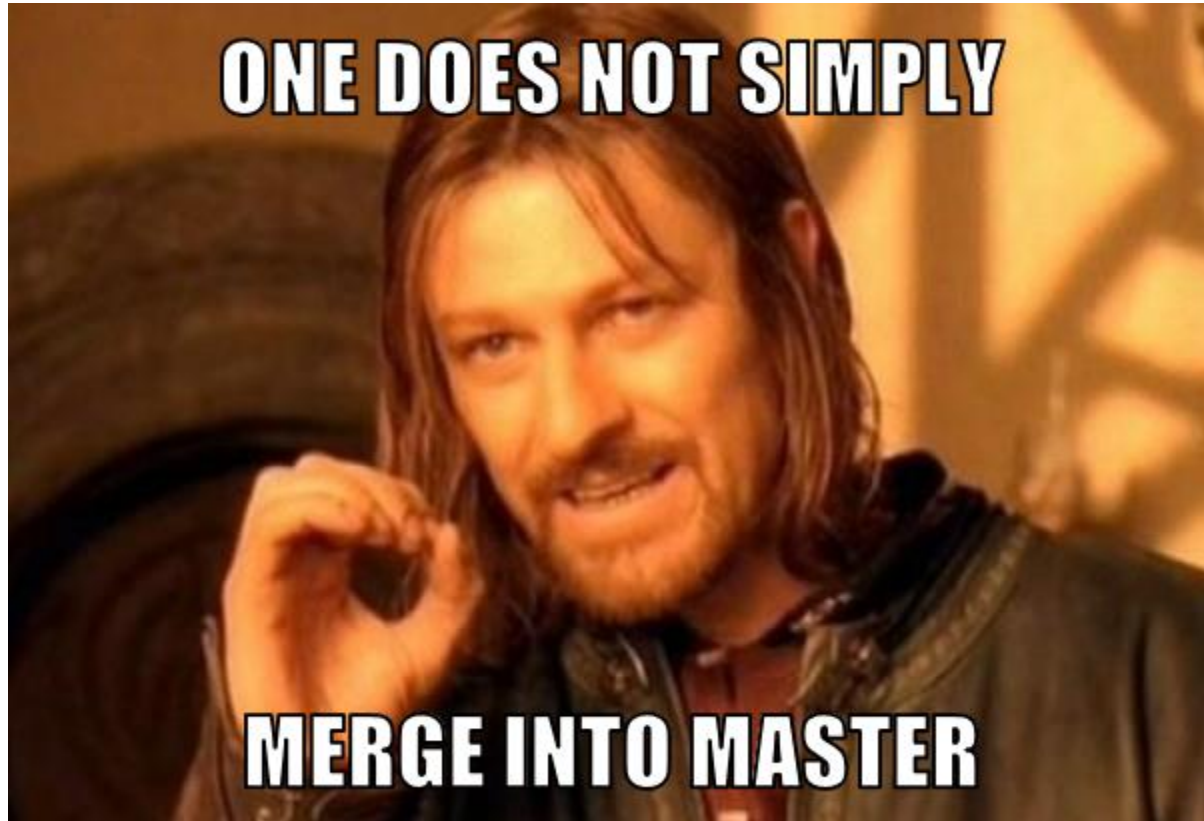
# Merge

- Fast-forward
  - Semplice «**avanzamento**» del ramo
- Octopus merge
  - Merge da **diversi** branch in un passo





**ONE DOES NOT SIMPLY**



**MERGE INTO MASTER**

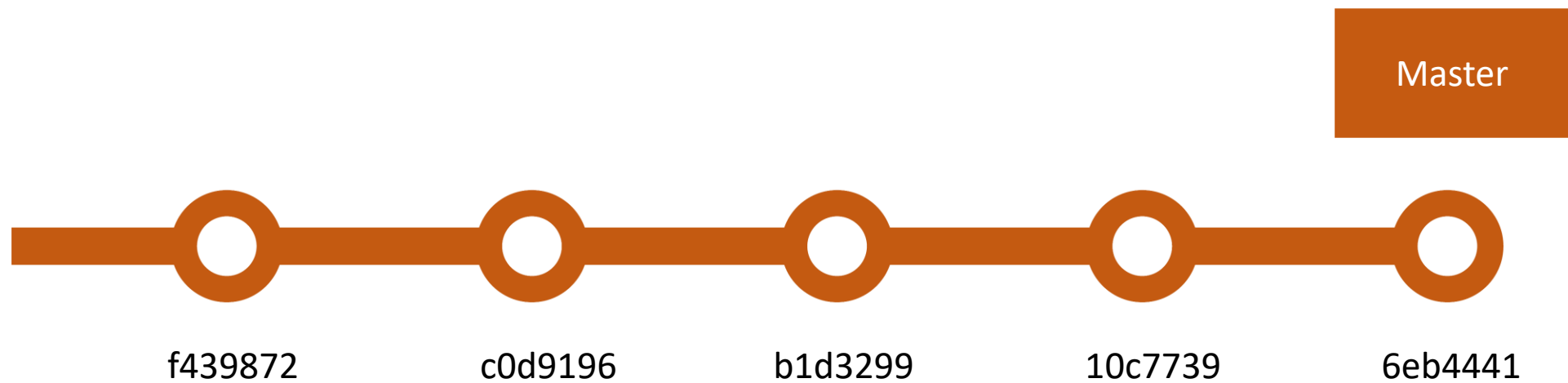
# Push e Pull

- **Sincronizzazione** con repository **remoto**
- Pull per fare una **fetch** ed incorporare mediante un **merge** le **modifiche** dal server remoto
- Push **invia** le **modifiche** al repository remoto

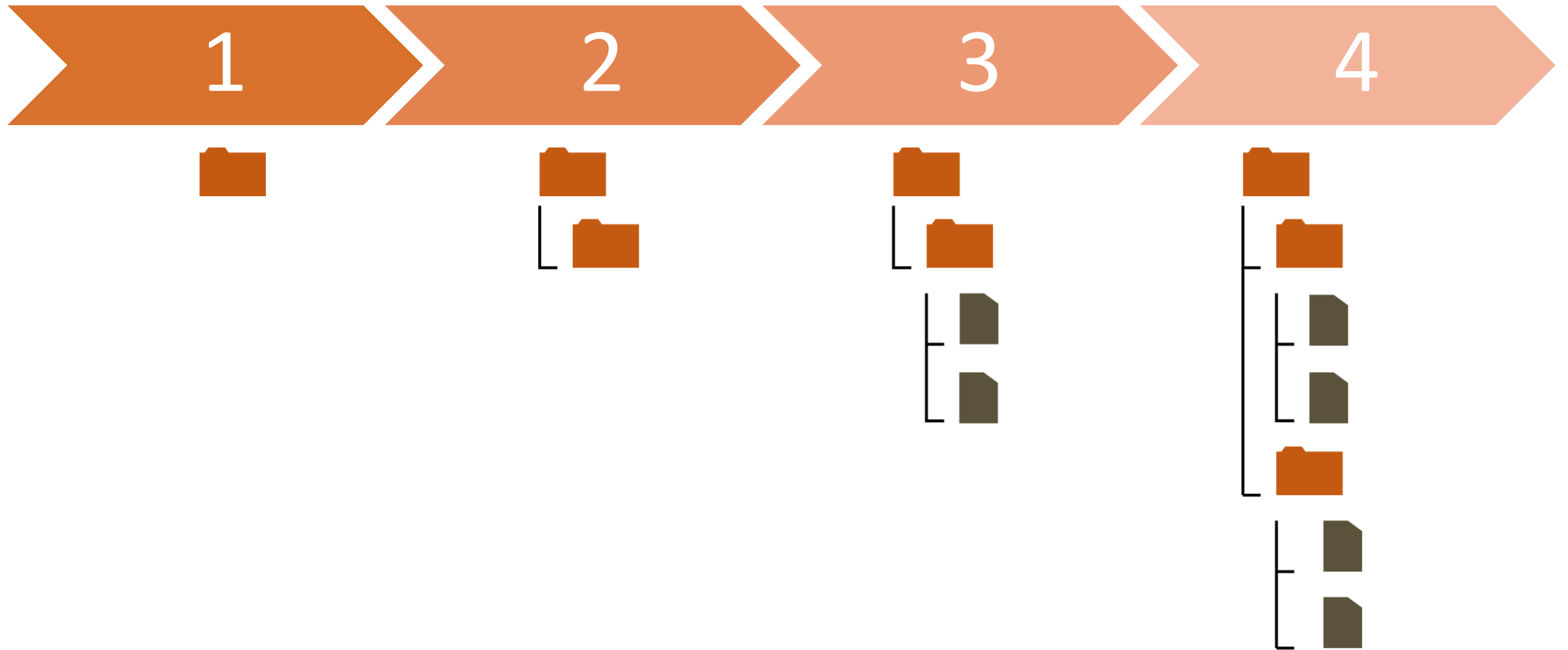




# Storico dei commit

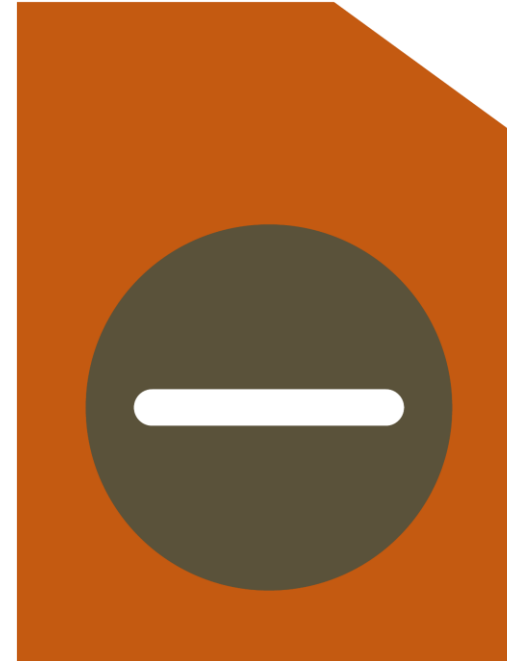


# Storico dei commit



# Ignoriamo i file non necessari

- Il file **.gitignore** contiene ciò che Git non deve **tracciare**
  - Definito dall'utente
  - File o cartelle
  - Cache, Binari, Configurazioni, ecc...
  - <https://gitignore.io>



# Il flusso di lavoro con Git





# Demo #1

Creiamo una repository locale, aggiungiamo file  
e creiamo un commit





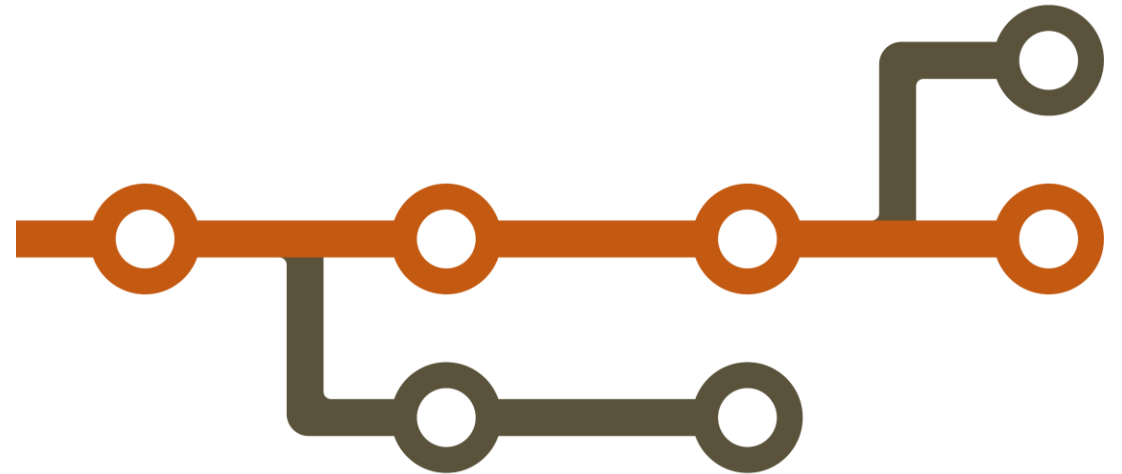
# Utilizzo dei rami

Lavoriamo con diversi rami separando le  
tipologie di modifiche

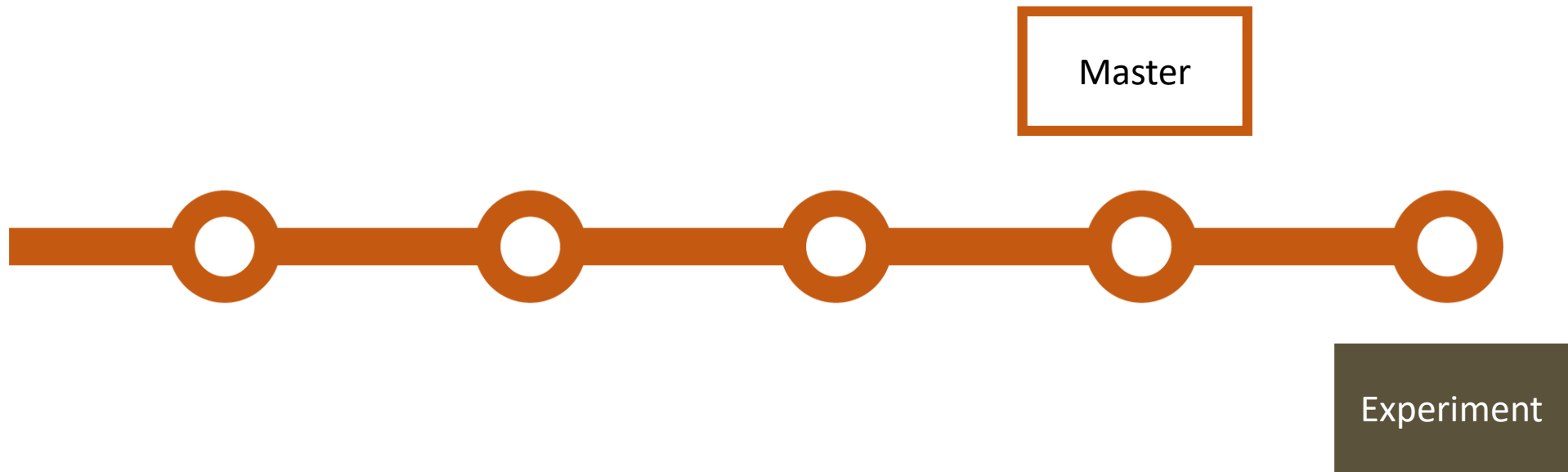


# Modello ramificato di lavoro

- Il **ramo** principale si chiama **master**
- HEAD è il **puntatore** al ramo corrente

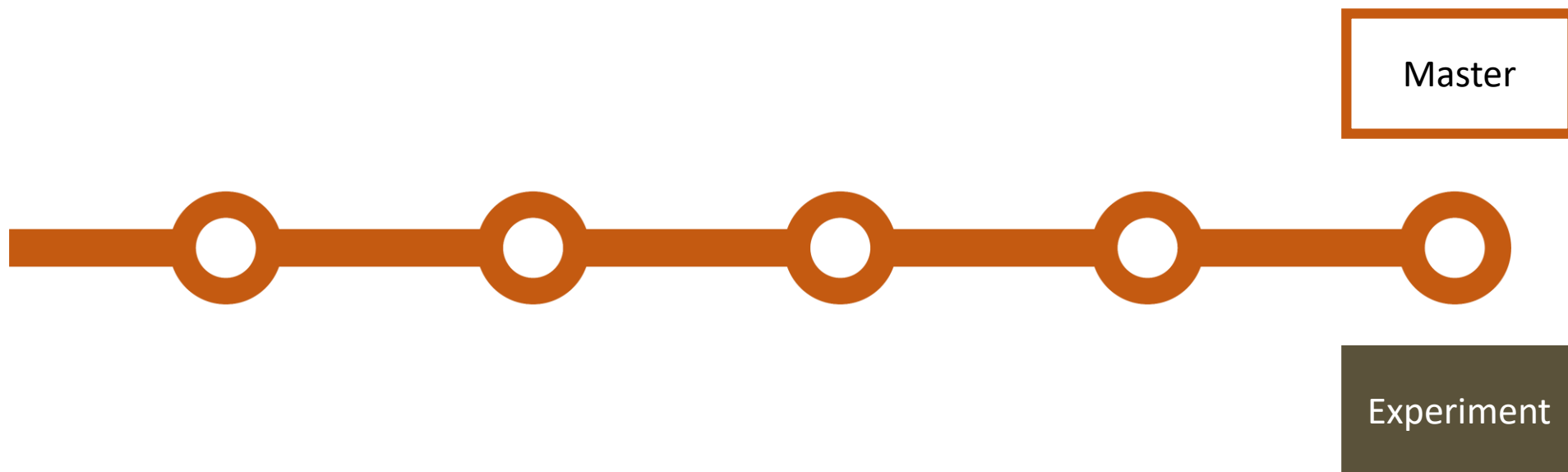


# Lavorare sui rami

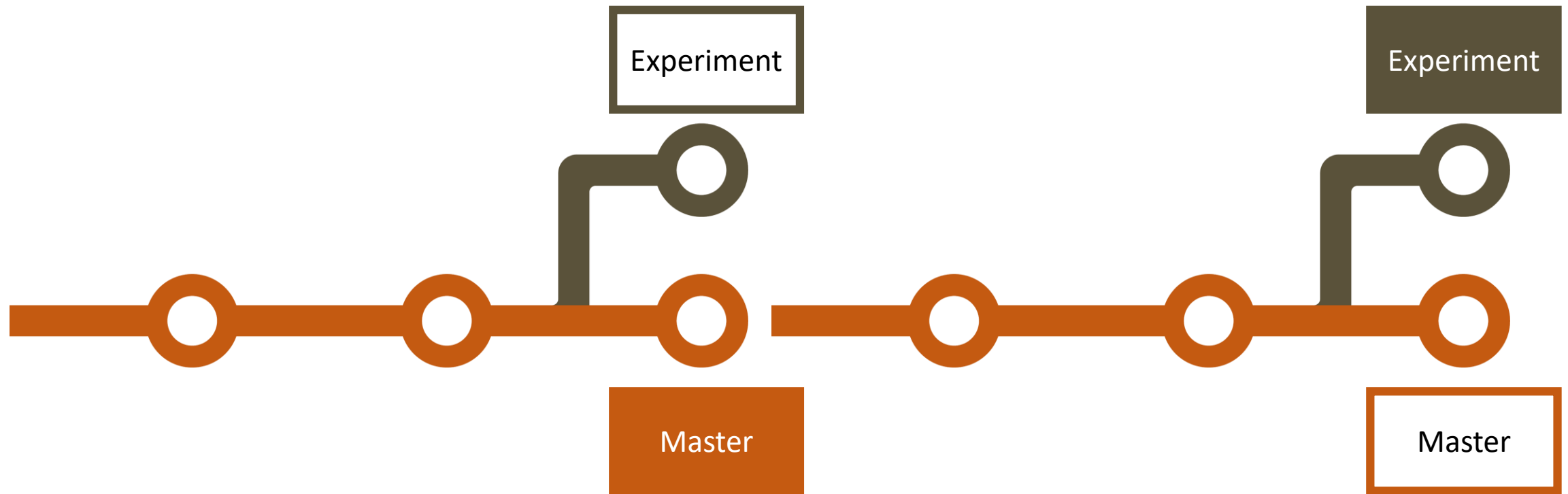




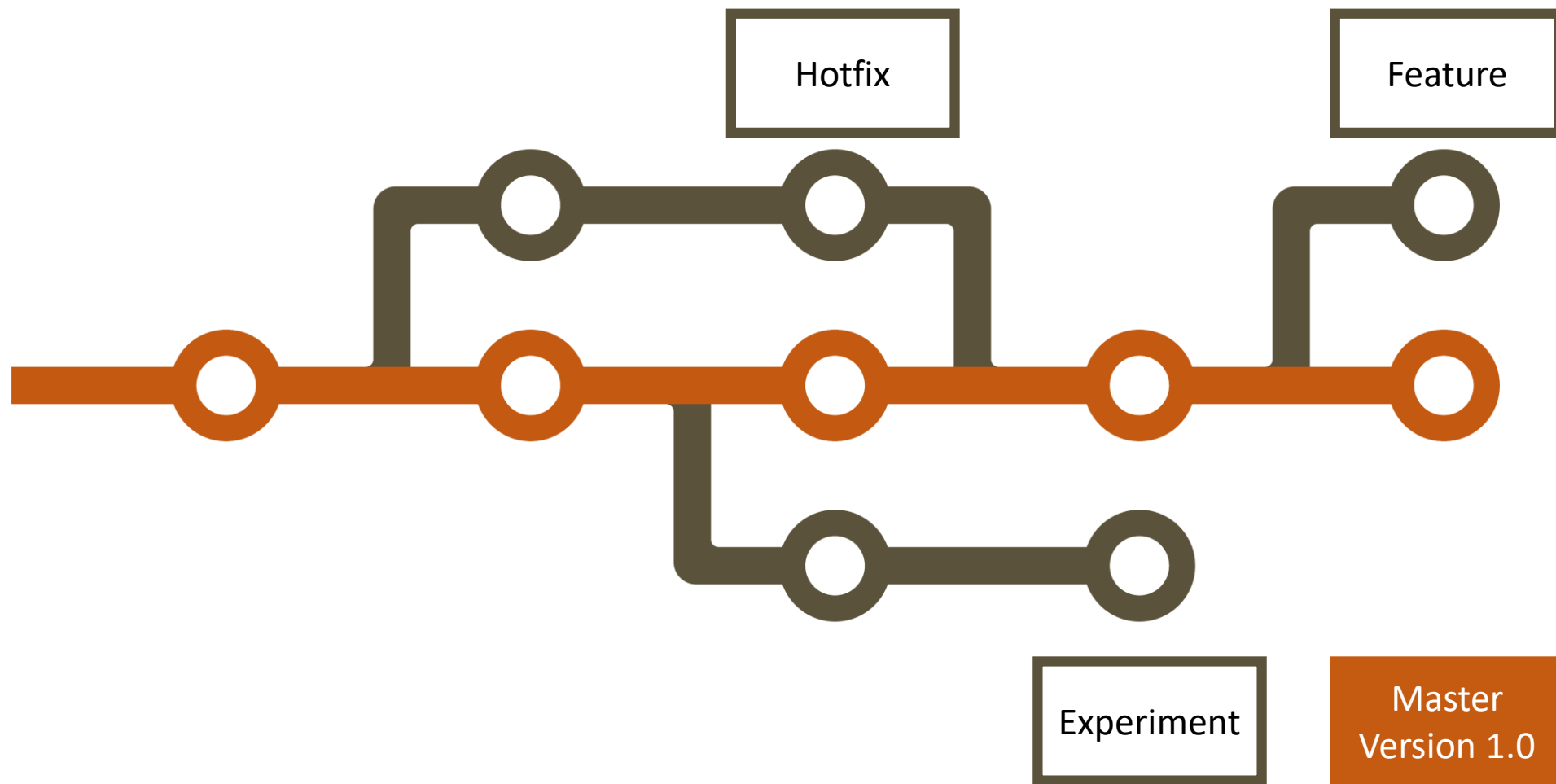
# Rami aggiornati allo stesso commit



# Switch tra rami con diversi commit



# Flusso di lavoro git





# Demo #2

Effettuiamo operazioni sui rami

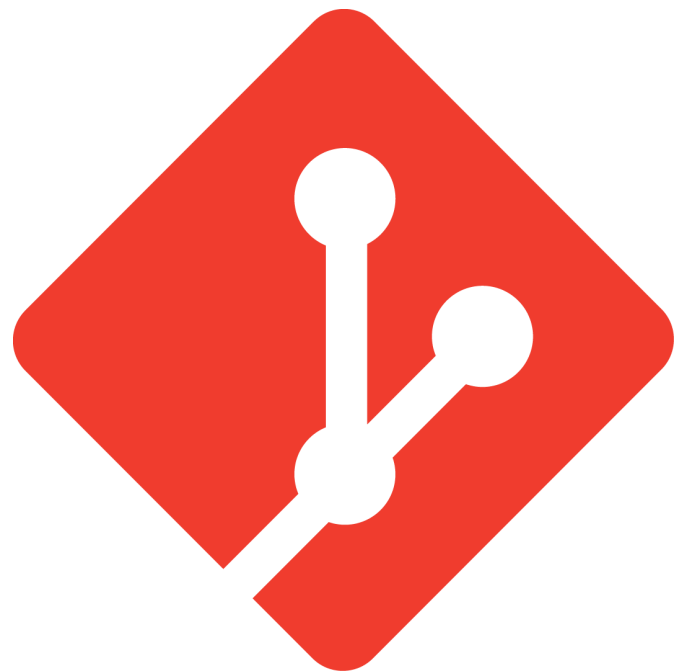




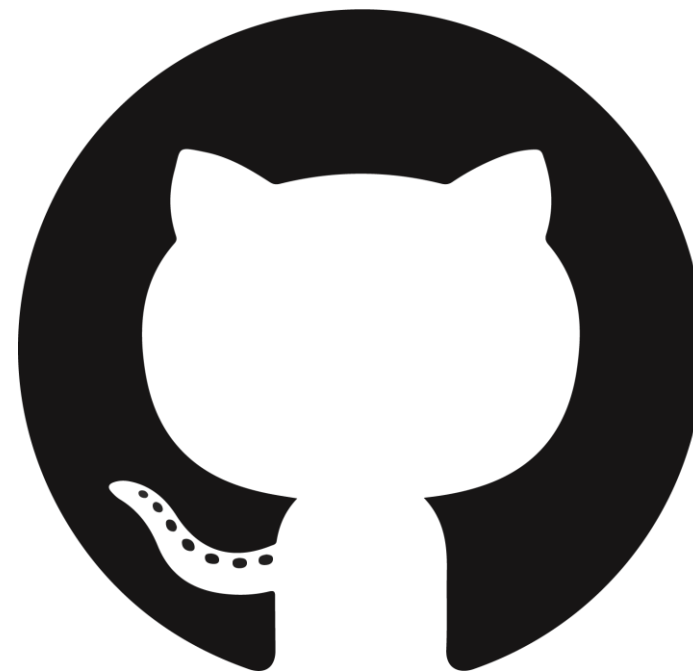
# Hosting Git

Collaborazione e backup





**Git**



**GitHub**

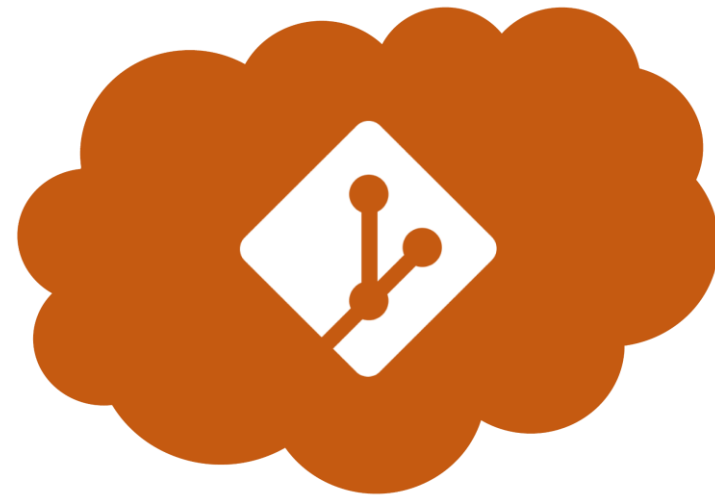
# Hosting Git

- Git **non** deve essere **costantemente** connesso a un **server**
- Può essere utilizzato come **backup** dei progetti
- Usato per molti progetti **open source** per la **condivisione** e la **collaborazione** tra **diversi** utenti



# Hosting Git

- Si possono usare **service provider** su internet
- Creare un **proprio** server git
  - Nas
  - Raspberry
  - VPS
  - You tell me...





# In case of fire



1. git commit



2. git push



3. leave building

# Quale hosting scegliere

- Repository **privati**
- Numero di utenti nel **team**
- **Hosting** di un sito
- **Community** del sito



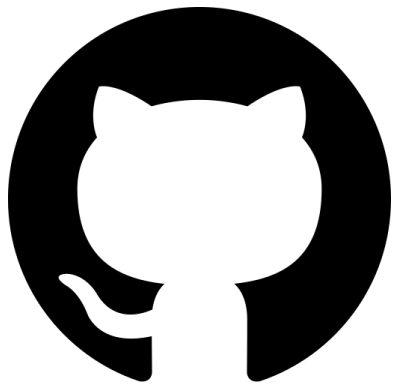
# Quale hosting scegliere

- **Prestazioni** dell'hosting
- Feature **aggiuntive**
  - Large File Storage (**LFS**)
  - Esecuzione di **test**
    - Travis-CI
    - Bitbucket Pipeline
  - Continuous Integration



# I più famosi hosting Git

---



## GitHub

**Gratis** per progetti  
**open source**  
Disponibile a pagamento  
On-premises



## Bitbucket

**Gratis** per progetti  
**open source** e  
team di 5 persone  
Disponibile a pagamento  
On-premises



## GitLab

**Gratis** per progetti  
**open source**  
Installabile gratuitamente  
On-premises  
Progetto **interamente** open source

---

# File speciali

- Utilizzo sintassi **Markdown**, facilita la **condivisione** su **GitHub**
  - [markdowntutorial.com](https://www.markdowntutorial.com)
  - README.MD
    - **Descrizione** del progetto
    - **Istruzioni** per l'utilizzo



# File speciali

- Utilizzo sintassi **Markdown**, facilita la **condivisione** su **GitHub**
  - LICENSE.MD
    - Istruzioni di distribuzione
    - Termini d'utilizzo sorgenti
    - Può essere una licenza open source
    - [choosealicense.com](https://choosealicense.com)



# Licenza open source

- Public Domain
- MIT
- BSD
- Apache
- LGPL
- GPL
- AGPL





# Demo #3

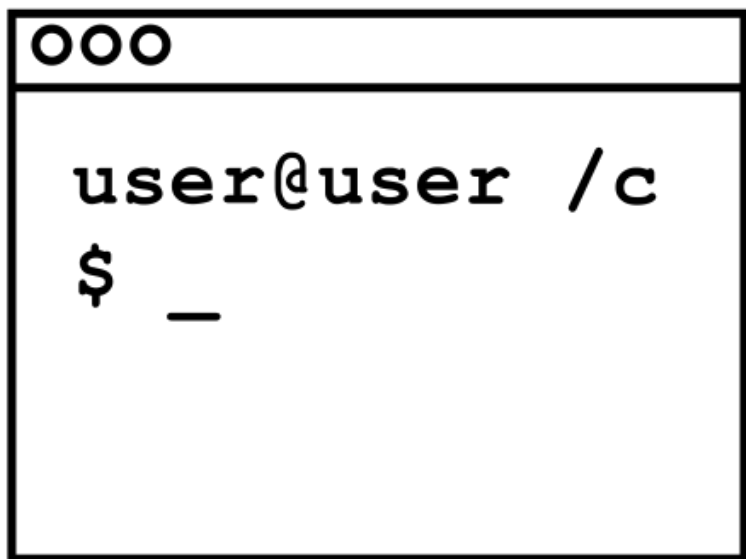
Usiamo GitHub per condividere il sorgente con la community





# Prossimi eventi

Introduzione a Bash  
Venerdì 27 Aprile



Accessibilità  
Mercoledì 2 Maggio




# Seguici

- [facebook/unixMiB](https://facebook.com/unixMiB)

- [@unixMiB](https://twitter.com/unixMiB)

- [unixmib.github.io](https://unixmib.github.io)





***«Non chiedetevi cosa può fare la community per voi,  
chiedete cosa potete fare voi per la community»***

**John Fitzgerald Kennedy**



**Grazie per l'attenzione**

# Fonti

- Atlassian: <https://www.atlassian.com/git/tutorials/using-branches>
- Wikipedia: [https://it.wikipedia.org/wiki/Git \(software\)](https://it.wikipedia.org/wiki/Git_(software))
- Git-Scm: <https://git-scm.com/book/it/v2/>
- Readthedocs: <http://get-git.readthedocs.io/>
- Partito Pirata:  
[http://www.piratpartiet.it/mediawiki/index.php?title=Mini Guida a GIT](http://www.piratpartiet.it/mediawiki/index.php?title=Mini_Guida_a_GIT)
- GitHowTo: <https://githowto.com/>

# Legal

- Git Logo by [Jason Long](#) is licensed under the [Creative Commons Attribution 3.0 Unported License](#).
- GITHUB®, the GITHUB® logo design, OCTOCAT® and the OCTOCAT® logo design are exclusive trademarks registered in the United States by GitHub, Inc.
- The GitLab logo and wordmark artwork are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
- The OSI logo is a trademark of Open Source Initiative.
- Bitbucket®, the Bitbucket® logo design are trademarks by Atlassian.
- unixMiB logo by Kowalski7cc is licensed under a Creative Commons Attribution-Share Alike 3.0 License.
- Tar gz icon is part of the Humanity icons project, and is licensed under the GNU General Public License v2.0
- Icons for events from Iconfinder by Elizabeth Arostegui and Softboxindia