

# **git** POUR TOU ·TE ·S

Daphné Giorgi

CNRS – LPSM

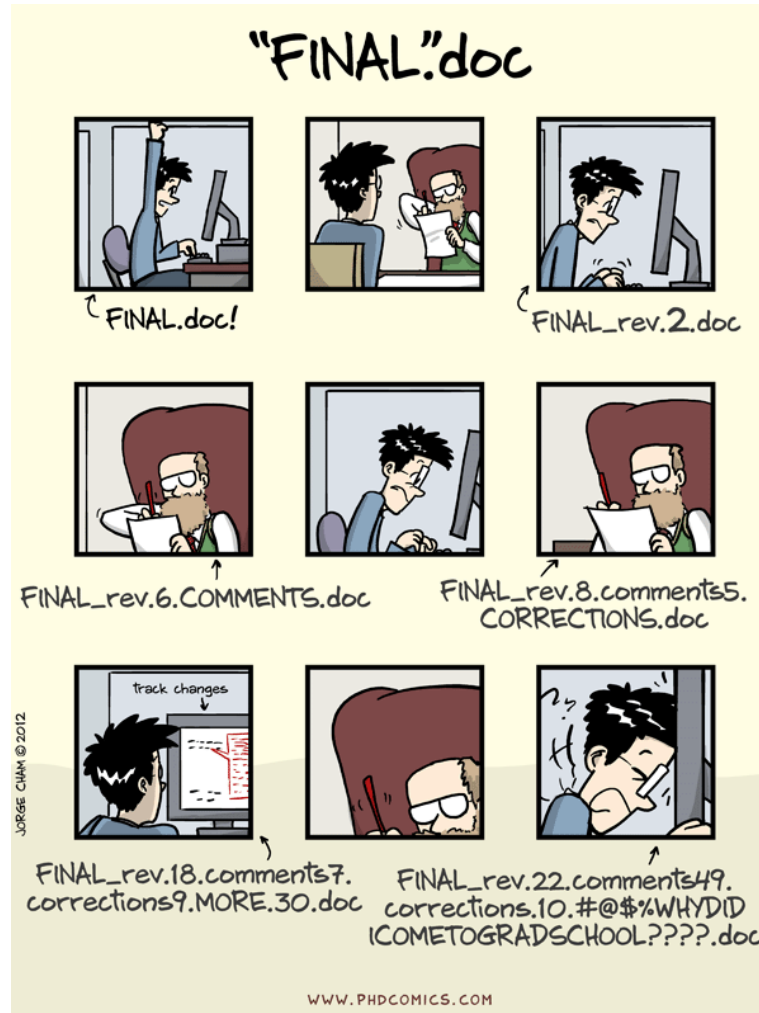
infomath – 06/01/2022

Qui Quoi Quand Où ~~Comment~~



## Si ceci vous est familier :

# Qui





Où bien vous avez trouvé la meilleure convention de nommage de fichiers qui soit :

Qui

```
19-11-07-doc.tex  
19-11-07-doc_initialEdits.tex  
19-11-20-doc.tex  
20-01-06-doc.tex  
20-03-24-doc_PNASsubmitted.tex  
20-03-24-doc_PLOSsubmitted.tex  
20-04-16-doc_PLOSrevision.tex  
20-05-08-doc_PLOSpublished.tex
```

Si vous avez entendu parler de (ou si vous avez vécu) au moins une des situations suivantes :

*Je n'avais pas de sauvegarde et mon ordinateur a planté/a été volé*

*Je travaillais avec un collègue et il a écrasé ma dernière version*

*J'ai deux dernières versions et je ne sais pas comment les fusionner correctement*



Qui

Quoi

**Système de contrôle de versions**

**distribué**

**libre**

**conçu pour être rapide et efficace**



Qui

Quoi

## Système de contrôle de version

Un système de contrôle de version permet de suivre l'historique des modifications sur des projets *partagés*.

Les contributeurs peuvent consulter l'historique du projet pour retrouver :

- Quels changements ont été apportés ?
- Qui a effectué ces changements ?
- Quand ces modifications ont-elles été apportées ?
- Pourquoi des changements étaient-ils nécessaires ?



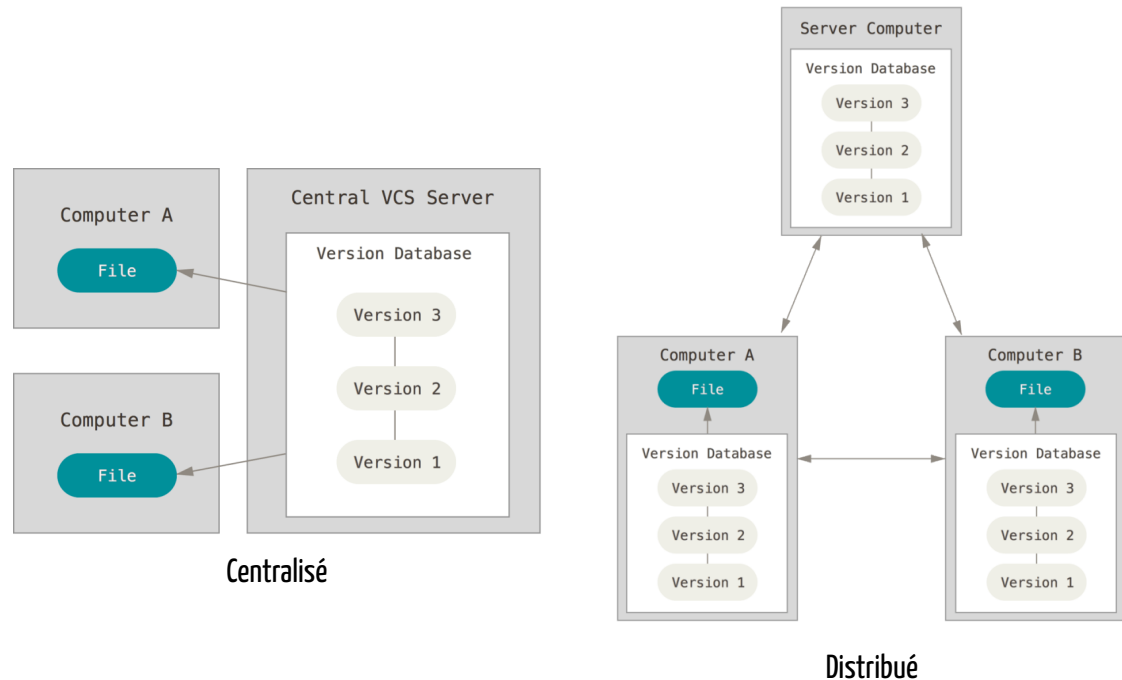
# Système de contrôle de version distribué

Qui

Quoi

Les systèmes de contrôle de version distribués n'ont pas besoin d'être constamment connectés à un dépôt central.

Au lieu de copies de dépôts à distance, on travaille avec des dépôts locaux.





# Attention à ne pas confondre Git avec

Qui

Les dépôts d'hébergement Git distants :

Quoi



Où

Les interfaces graphiques Git (GUI - Graphical User Interface) côté client :

- **SourceTree** (Mac, Windows)
- **GitHub Desktop** (Mac, Windows)
- **TortoiseGit** (Windows)
- **Git Extensions** (Linux, Mac, Windows)
- la ligne de commande :)

Plugins Git pour tout et n'importe quoi





## Installation Locale GitLab

Qui

Certains laboratoires disposent de leur propre serveur GitLab (**LPSM**).

Quoi



Où

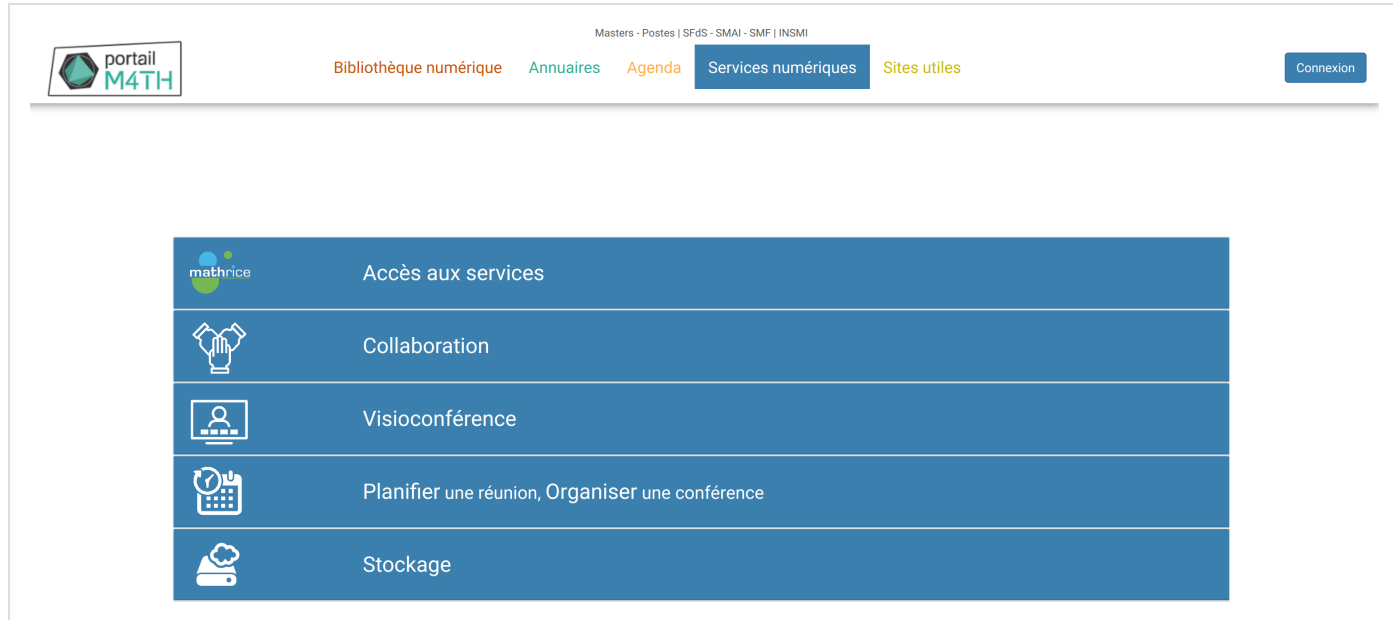
Réservé à la communauté mathématique et à ses collègues invités. Le nombre maximum de projets est fixé à 100.

Les services supplémentaires activés sont :

- **Mattermost** : *logiciel de communication en équipe*
- **Pages** : *publication de sites web*
- **Container Registry** : *images Docker*
- **CI** (Intégration Continue) : *test et validation de code*
- **LFS** (Large Files Storage) : *taille recommandée du dépôt git inférieure à 1G*

De nombreux autres outils utiles sur





## Accès aux services

- Comptes invités

## Collaboration

- PLMbox : 50G
- PLMlatex
- PLMlab
- FileSender : 100G

## Visioconférence

- Rendez-Vous : dans un navigateur
- **BBB**

## Organiser une conférence

- Sciencesconf.org



# Premier commit git : 7 avril 2005, Linus Torvalds

Qui

'Initial version of "git" the information manager from hell'

Quoi

Où

Quand

---

*I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'.*

*Wikipedia*

*Linus Torvalds deserves a Nobel Prize for git, the Turing Award for Linux and swift kick in the ass for almost everything else he's done*

*Twitter, 7 July 2018*

*[...] Maybe I can get an email filter in place so at when I send email with curse-words, they just won't go out. [...]*

*Linux 4.19-rc4 released, an apology, and a maintainership note, 16 September 2018*

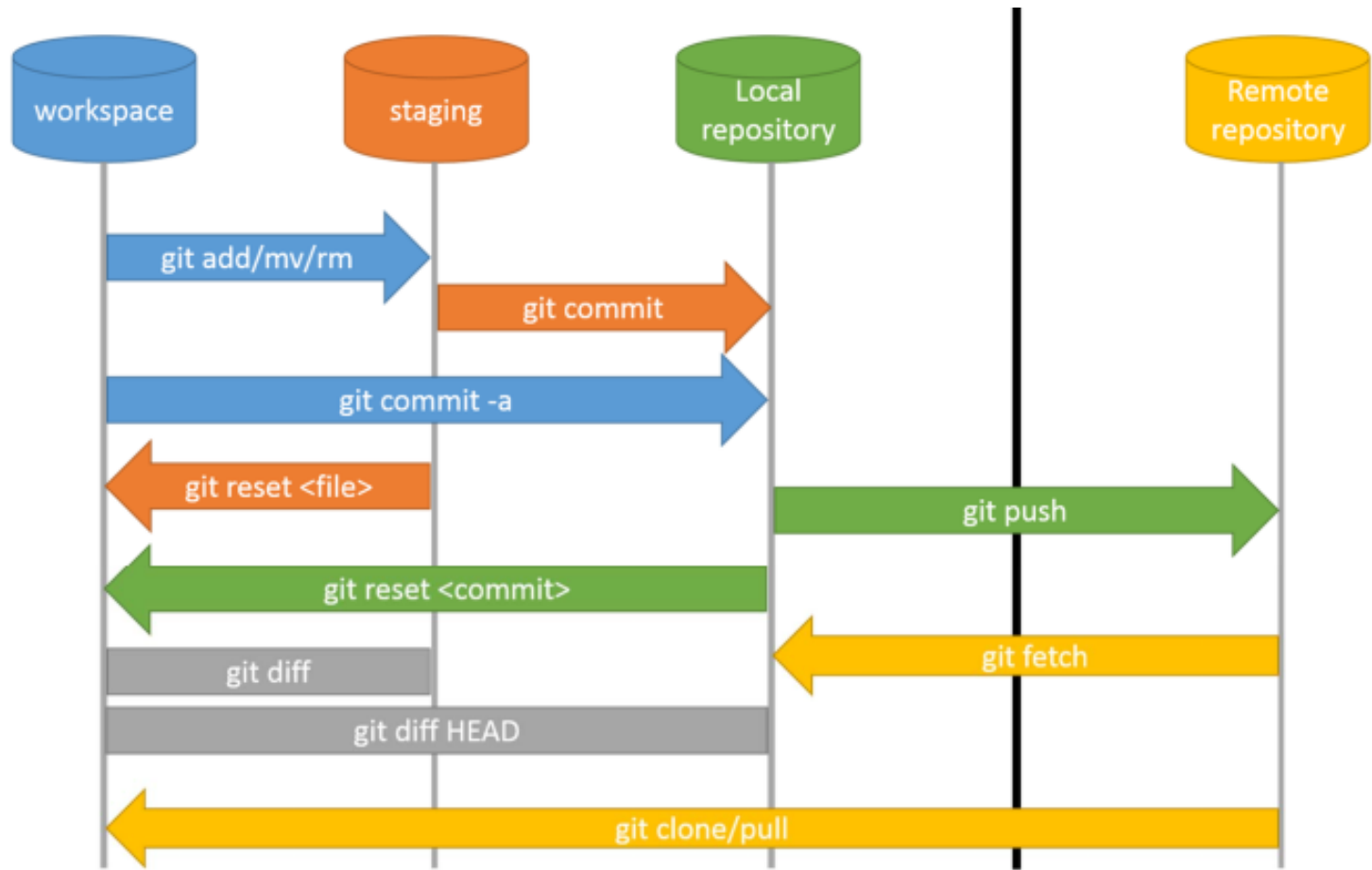
# Bases de Git

# Première année d'utilisation



- Pas besoin de comprendre la **théorie des graphes**
- Quelques commandes de base
  - init
  - add
  - commit
  - push
  - pull
  - branch
  - checkout
  - merge
- Sauvegardez ailleurs et téléchargez à nouveaux

# **git** --everything-is-local

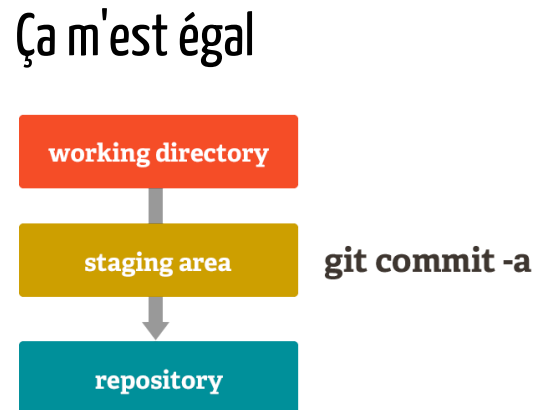
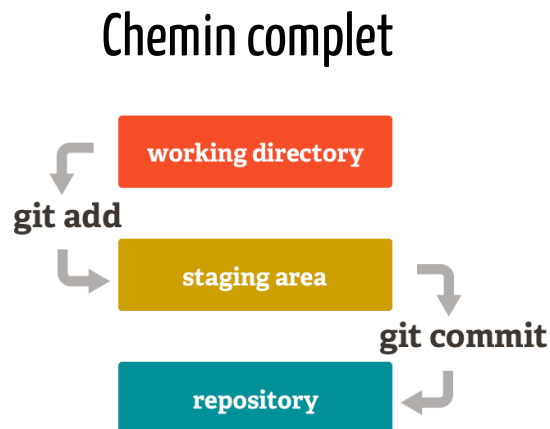


# Index -- Staging area

*La meilleure description de l'index (staging area) est un aperçu de votre prochaine soumission (commit).*

## Avantages:

- Séparer un grand changement en plusieurs soumissions
- Examiner les changements
- Une fusion (merge) est source de conflits
- Permet de garder des dossiers locaux sans les soumettre
- Faufler des petits changements



## Première configuration

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

## Création d'un répertoire Git

Depuis le dépôt local

```
$ cd path/to/directory  
$ git init [project-name]
```

Depuis un dépôt distant

```
$ git clone [url]
```



## Après avoir édité un fichier

Lister les nouveaux fichiers ou des modifications à soumettre

```
$ git status
```

Montrer les différences

```
$ git diff [optional filename/repository]  
$ git diff --staged [optional filename/repository]
```

Ajouter ou enlever un fichier de l'index

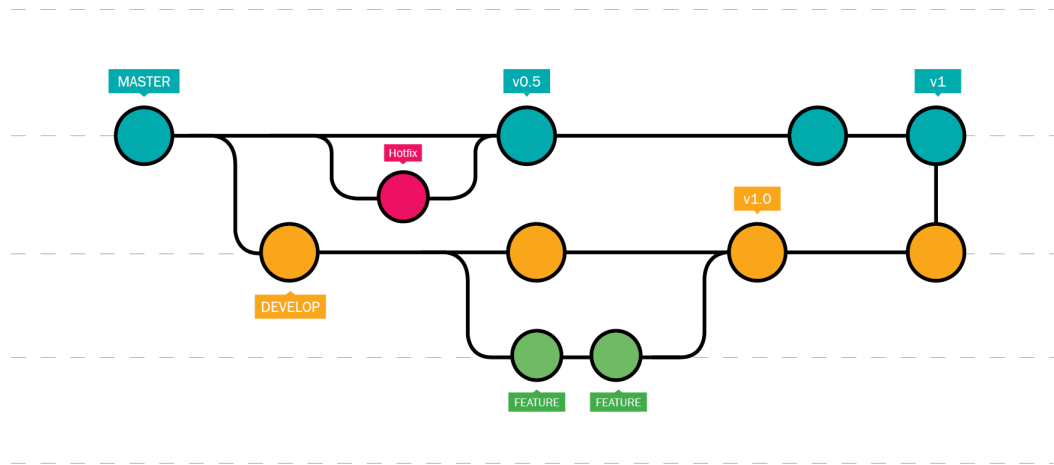
```
$ git add [filename]  
$ git reset [filename]
```

Enregistrer les changements dans l'historique des versions

```
$ git commit -m "[message]"
```

<https://plmbox.math.cnrs.fr/f/2e8ec31e92534139957e/>

## Flux de travail



### Créer et changer de branche

```
$ git branch [branch-name]  
$ git switch [branch-name]
```

### Fusionner une branche secondaire dans la branche courante et effacer la branche secondaire

```
$ git merge [branch]  
$ git branch -d [branch]
```

## Historique

Historique à partir du HEAD

```
$ git log
```

Synthèse

```
$ git log --oneline
```

Synthèse et graphe

```
$ git log --oneline --graph
```

<https://plmbox.math.cnrs.fr/f/047360fd9e9044d1936f/>

## Travailler avec un répertoire distant

Cloner un répertoire distant

```
$ git clone [url]
```

Télécharger l'historique distant et ensuite fusionner avec le répertoire local

```
$ git fetch [remote]  
$ git merge [branch]
```

Télécharger et fusionner

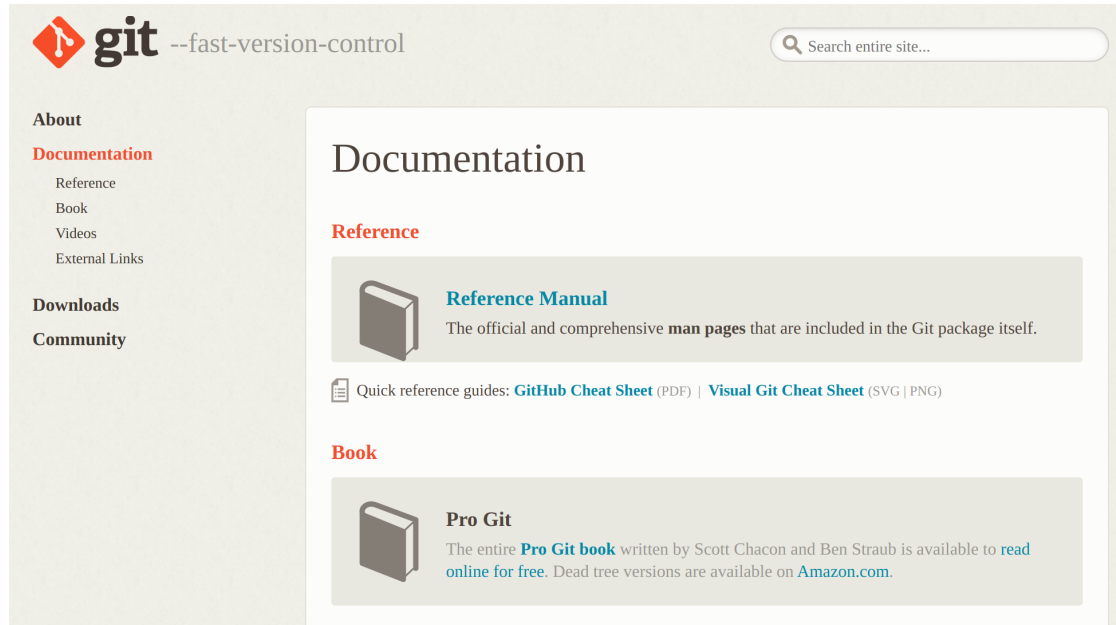
```
$ git pull
```

Envoyer sur le répertoire distant

```
$ git push [remote] [branch]
```

<https://plmbox.math.cnrs.fr/f/09f39678f3444f7d88f0/>

# Documents de référence



## Doc officielle

- Manuel de référence
- Livre
- Vidéos
- Liens externes

## Cheat sheets

- Interactive ndpsoftware
- Github

**Merci pour votre attention !**