

 **git** POUR TOU ·TE ·S

Daphné Giorgi

CNRS – LPSM

infomath – 09/11/2023

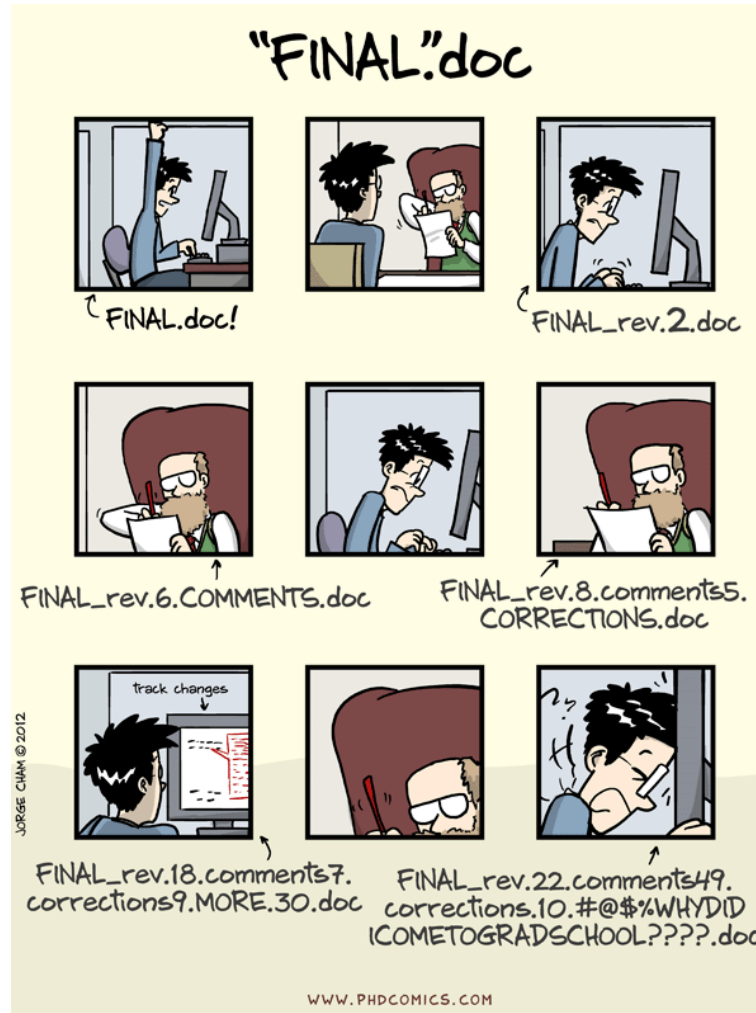
Qui Quoi Quand Où ~~Comment~~





Si ceci vous est familier :

Qui





Ou 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

Systeme de controle de versions

distribue

libre

concu pour etre rapide et efficace



Qui

Quoi

Systeme de controle 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 ?



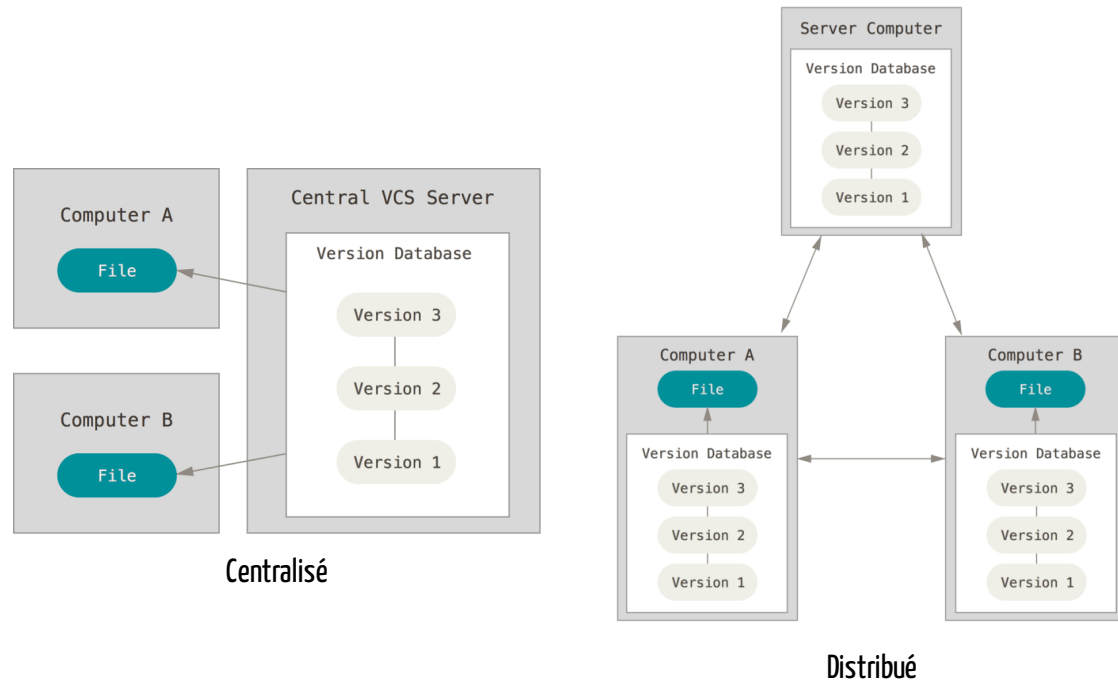
Systeme de controle de version distribue

Qui

Quoi

Les systemes de controle de version distribues n'ont pas besoin d'etre constamment connectes a un depot central.

Au lieu de copies de depots a distance, on travaille avec des depots 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 :

- **GitKraken** (*Linux, Mac, Windows*)
- **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



Installations locales GitLab

Qui

Laboratoires

Quoi

Certains laboratoires disposent de leur propre serveur GitLab, n'hésitez pas à vous renseigner auprès de vos services informatiques !

Où

CNRS : PLMlab

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*

Services numériques

- PLM
- portail M4th

Accès aux services

- Comptes invités

Collaboration

- PLMbox : 50G
- PLMlatex : basé sur Overleaf
- PLMlab : basé sur GitLab
- FileSender : 100G

Visioconférence

- Rendez-Vous : dans un navigateur
- Web conf : basé sur 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



Credits : Kimmo Mantyla/REX/Shutterstock

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

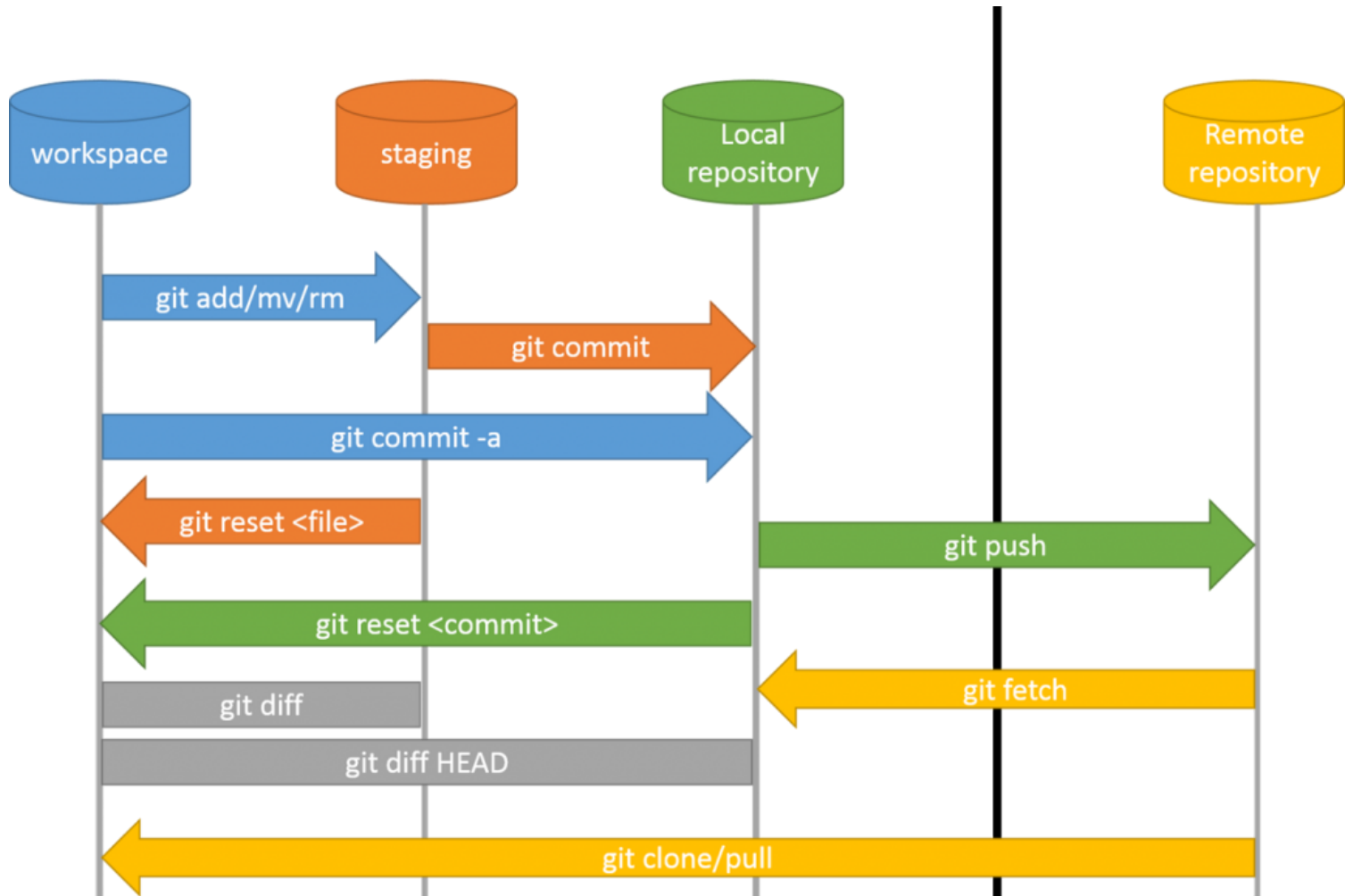
Bases de Git

Première année d'utilisation



- Pas besoin de comprendre la **théorie des graphes**
- Quelques commandes de base
 - Création d'un projet : `init`
 - Ajout dans l'historique : `add`, `commit`
 - Echange avec un remote : `push`, `fetch`, `pull`, `merge`
 - Organisation du travail : `branch`, `checkout`, `merge`
- Sauvegardez ailleurs et téléchargez à nouveaux

git --everything-is-local

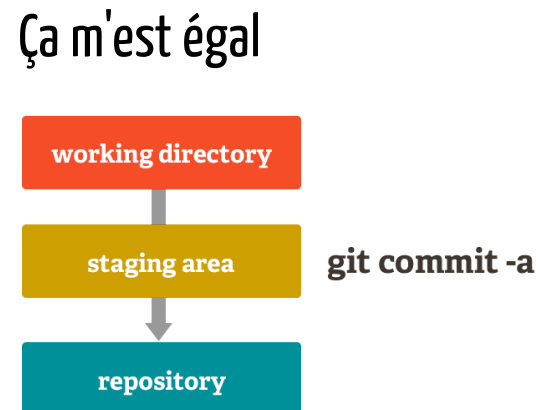
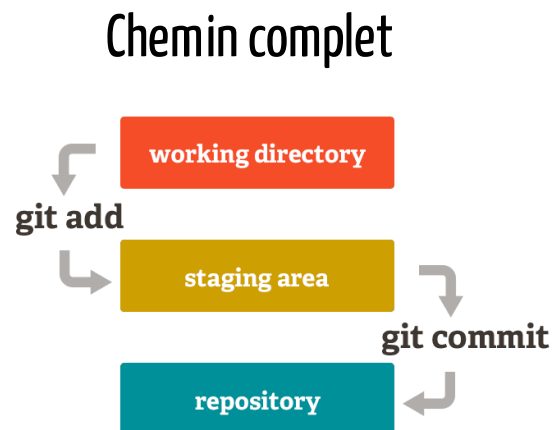


Index -- Staging area

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
- Faufilet 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

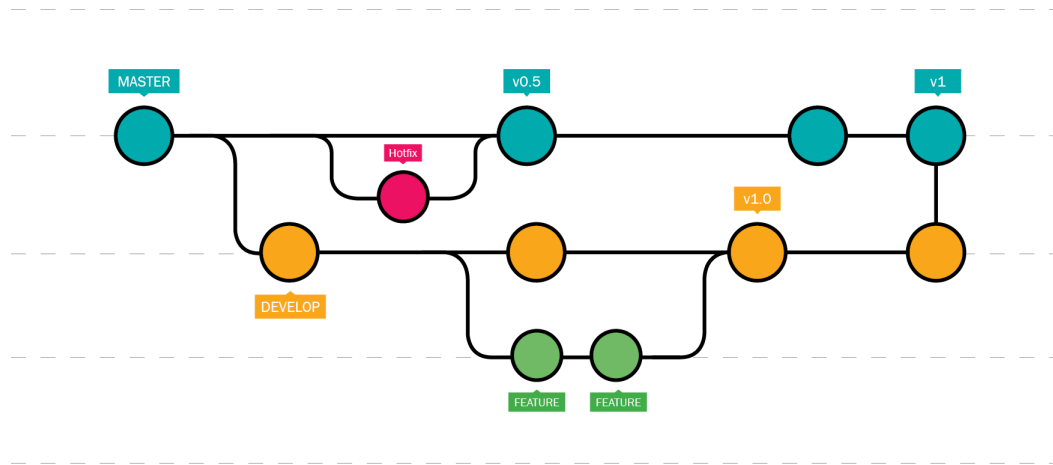
```
$ git add [filename]  
$ git reset [filename]
```

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

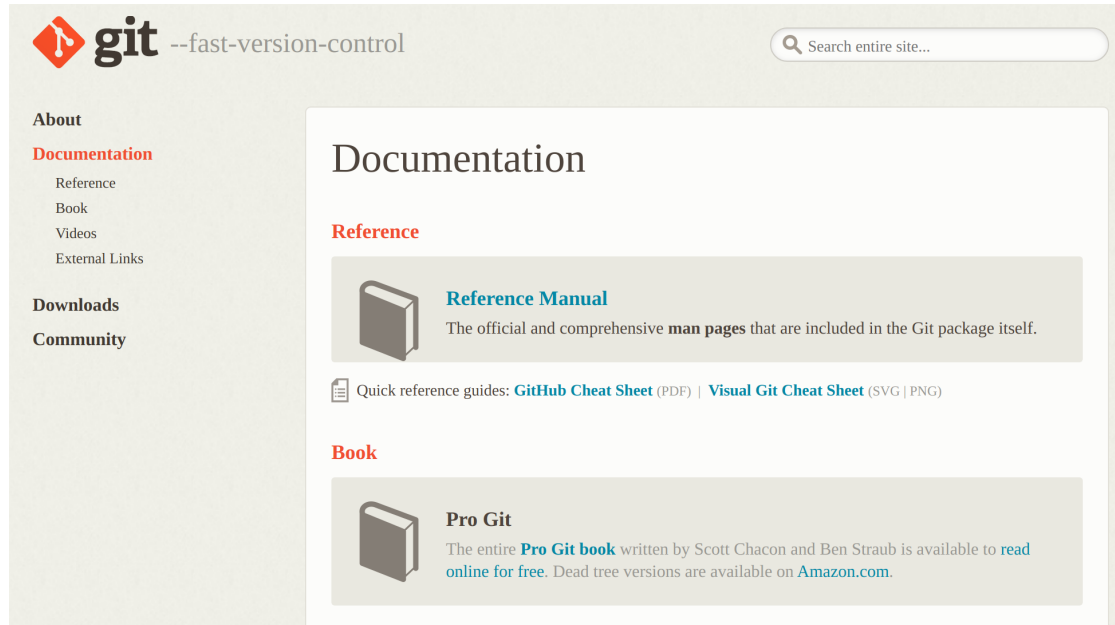
```
$ git pull
```

Envoyer sur le répertoire distant

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

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

Documents de référence



The screenshot shows the Git website's documentation page. At the top left is the Git logo with the tagline "--fast-version-control". A search bar is located at the top right. The left sidebar contains navigation links: "About", "Documentation" (highlighted), "Reference", "Book", "Videos", "External Links", "Downloads", and "Community". The main content area is titled "Documentation" and features two sections: "Reference" and "Book".

Reference

- Reference Manual**
The official and comprehensive **man pages** that are included in the Git package itself.

Quick reference guides: [GitHub Cheat Sheet](#) (PDF) | [Visual Git Cheat Sheet](#) (SVG | PNG)

Book

- Pro Git**
The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Doc officielle

- Manuel de référence
- Livre
- Vidéos
- Liens externes

Cheat sheets

- Interactive ndpsoftware
- Github

Merci pour votre attention !