

Suivez cette présentation sur votre ordinateur :

<https://louvainlinux.org/atelier-git>

Préparez-vous à utiliser **git** : vous utiliserez le logiciel GitHub Desktop durant cette présentation.

Prenez un peu d'avance, installez-le :

- Sur les ordinateur Windows UCL : suivez les instructions à l'adresse
<https://louvainlinux.org/git-windows-ucl>.
- Ou installez GitHub Desktop sur votre ordinateur :
 - Ubuntu : TODO : Tester RC et lien
 - Windows ou OS X : <https://desktop.github.com>



Présentation Git

Un outil de collaboration puissant

Gaëtan Cassiers Alexandre Fiset Pierre Ortegat

1^{er} Mars 2018

KAP Louvain-li-Nux

- Cette présentation est sous license libre CC-BY 4.0.
- En ligne (slides en pdf et sources \LaTeX , exercices...) :
<https://github.com/louvainlinux/atelier-git>

Table des matières

1. Introduction
2. Principes de Git
3. Utilisation : en pratique
4. Installation et configuration
5. Exercices
6. Fonctionnalités plus avancées
7. Informations et ressources

Introduction

Comment gérez-vous actuellement un projet ?

- L'envoyer à travers un message sur Facebook, ... (Très mauvaise idée)
- L'envoyer par mail (Un peu moins)
- Utiliser une Dropbox, Google Drive, ... (Déjà mieux mais toujours risqué ou manque de fonctionnalités)

Solution : Utiliser un **système de gestion de version décentralisé** (Distributed Version Control System (DVCS) pour les anglophiles).

Un DVCS ?

- **Version** Enregistre des « instantanés » du projet.
- **Gestion** Revenir en arrière, voir des différences, fusionner des modifications.
- **Décentralisé** Chacun
 - a sa copie (avec son historique) sur son PC,
 - peut mettre sa copie (et son historique) en ligne,
 - peut récupérer sur son PC les copies et historiques disponibles en ligne,
 - peut fusionner différentes copies (semi-)automatiquement.
- **Projet** n'importe quel répertoire (« dossier »). Donc n'importe quoi : Bureautique, \LaTeX , code, images, musique...

Et Git dans tout ça ?

Git a été créé en 2005 par Linus Torvalds (auteur de **Linux**); le plus connu et utilisé.

À l'origine, interface en ligne de commande.

Aujourd'hui : aussi des interfaces graphiques, dont GitHub Desktop.

Mais on m'avait parlé de GitHub!

Souvenez-vous...

- **Décentralisé** Chacun
 - peut mettre sa copie (et son historique) en ligne,
 - ...

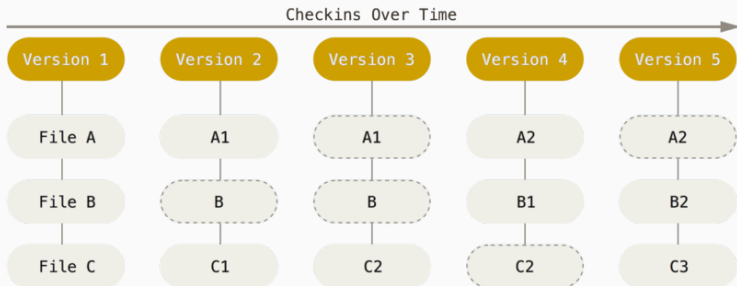
Il y a plein d'"endroits" en ligne où on peut envoyer son travail, GitHub est le plus connu.

En plus de ça, GitHub a des fonctionnalités pour interagir avec des collaborateurs.

Principes de Git

- espace de travail : les fichiers, répertoires... dans lesquels on travaille. Ils n'ont rien de spécial par rapport à d'autres sur l'ordinateur.
- Dépôt : espace de travail + historique, sur un ordinateur.
- Commit : "version", est le successeur d'une autre commit.
- Historique : la "chaîne" de tous les commits, du plus anciens.
- Dépôt distant : un dépôt qui se trouve chez GitHub.

Concept : le commit



Les illustrations non-sourcées viennent de <https://git-scm.com/book>.

Actions

- Créer un dépôt sur GitHub.
- Cloner (faire une copie d') un dépôt de GitHub sur son PC.
- Modifier/créer des fichiers (pas avec Git!).
- Ajouter un fichier modifié : il sera pris en compte dans le prochain commit.
- Faire un commit : créer une nouvelle version, qui contient les fichiers ajoutés. On y ajoute un commentaire (qui décrit les changements).
- Consulter un historique.
- Push : envoyer ses nouveaux commits sur GitHub.
- Pull : récupérer des changements de GitHub (qui ont été envoyés par quelqu'un d'autre).
- Merge : quand on Pull et qu'on a aussi des nouveaux commits sur son PC. Git essaye de fusionner automatiquement; s'il ne sais pas le faire, il demande.

Questions?

Utilisation : en pratique

TODO : screenshots

TODO : utile / nécessaire pour les exercices ?

TODO : screenshots

TODO : screenshots

TODO : screenshots

TODO : screenshots

Remarque : fichier texte vs binaire

- Fichiers texte : programme, \LaTeX ...
- Fichiers binaires : le reste : Word, Writer, images, sons, PDF...

TODO : screenshots

- Créer un `commit` sur base des fichiers ajoutés.
- Message de `commit` : décrit les changements effectués.

TODO : screenshots

TODO : screenshots

Récupérer un fichier d'un commit passé

TODO : screenshots

Astuce : ignorer des fichiers

Des fichiers qu'on ne veut jamais dans Git (résultats de compilation, fichiers temporaires...) Cachez-les!

NB : Cela crée un fichier **.gitignore** : celui-là, on le versionne.

TODO : screenshots

Push : envoyer des commits sur GitHub

TODO : screenshots

Pull : récupérer des commits qui sont sur GitHub

TODO : screenshots

Merge non-automatique : quand il y a des conflits

TODO : screenshots

On peut trouver de l'aide : TODO : A garder à cet endroit?
Github help : <https://help.github.com/>

Installation et configuration

- Sur les ordinateur Windows UCL : suivez les instructions à l'adresse
<https://louvainlinux.org/git-windows-ucl>.
- Ou installez GitHub Desktop sur votre ordinateur :
 - Ubuntu : TODO : Tester RC et lien
 - Windows ou OS X : <https://desktop.github.com>

Git a besoin de deux informations de base sur vous pour pouvoir travailler efficacement :

- Nom et Prénom
- Email

TODO : screenshots

Linux Windows Mac TODO : screenshots

Exercices

Exercise 1

TODO

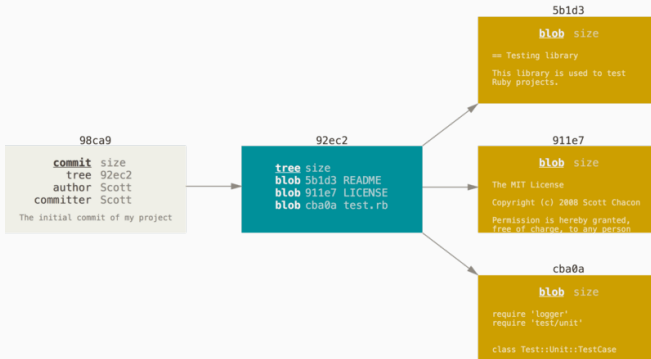
Exercise 2

TODO

Fonctionnalités plus avancées

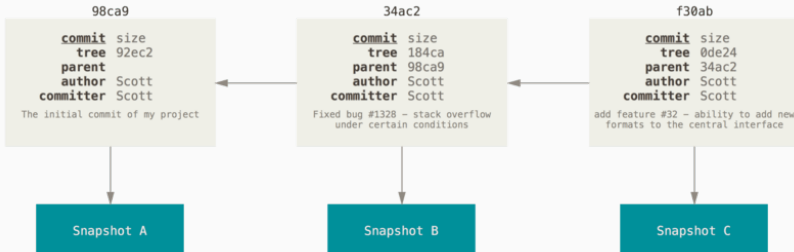
De derrière : les objets git

- Chaque commit a un identifiant :
12f87b95caff8cbeb5ce0717528d77e27db5669c.



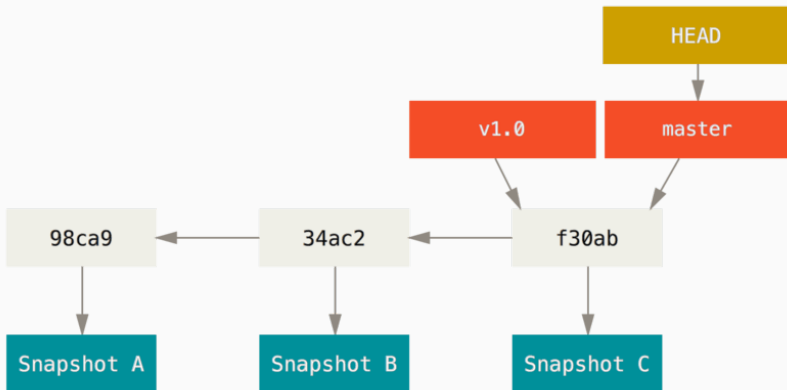
De derrière : les parents

- Chaque commit a un parent.



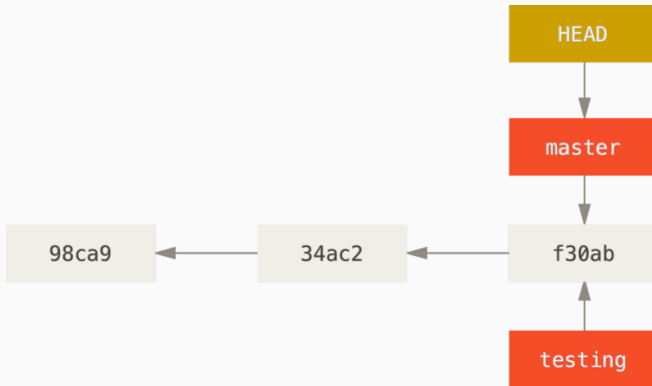
De derrière : les étiquettes

- On peut mettre des étiquettes sur des commits.
- **HEAD** est la position actuelle.



Créer une branche

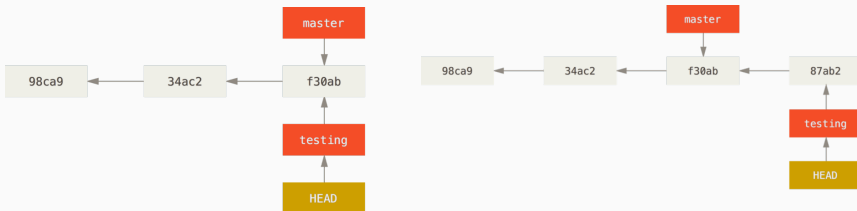
- Une branche est une nouvelle étiquette.
- La branche par défaut est **master**.



TODO : screenshots

Changer de branche

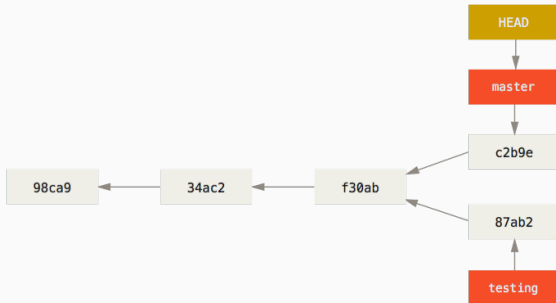
La branche courante est celle qui suit les nouveaux commits.



TODO : screenshots

Branches divergentes

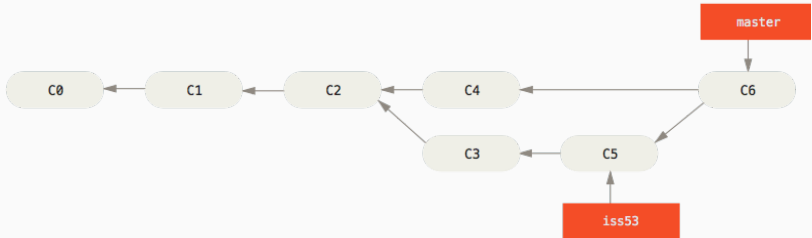
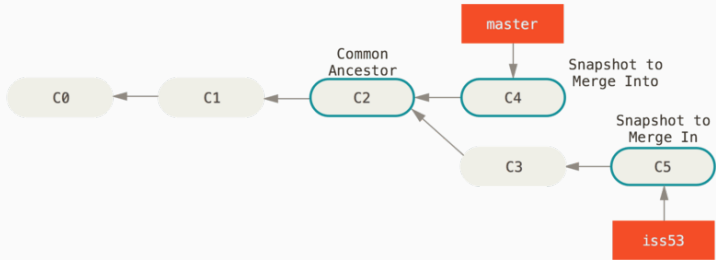
- Utilité : travailler sur des modifications indépendantes.



Résultat : embranchement dans l'historique

TODO : screenshots

Fusionner des modifications

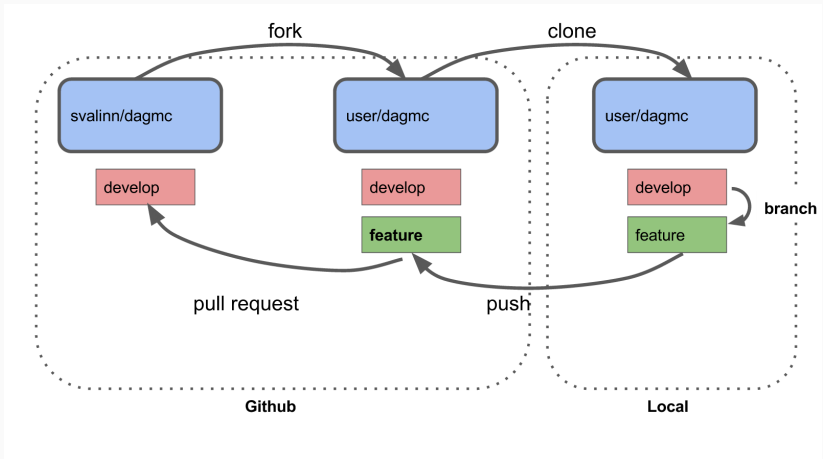


TODO : screenshots

Et parfois il faut résoudre des conflits...

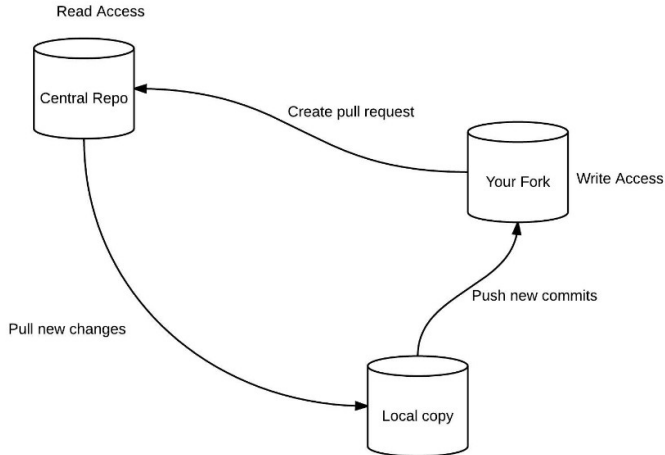
Fork – Pull Request

Une autre méthode de collaboration, très utilisée pour des larges projets et/ou projets où la contribution est ouverte à tous.



Fork – Pull Request : Méthode de travail

Ajouter un "upstream". TODO : ref vers une explication / voir comment ça marche ave github desktop



TODO : screenshots

TODO : screenshots

Informations et ressources



Pratiquement identiques (tous fonctionnent avec GitHub Desktop).

Dépôts privés gratuits (tout comme sur Gitlab & Bitbucket), et d'autres avantages :

<https://education.github.com/pack>.

Nécessite d'ajouter l'adresse `...@student.uclouvain.be` au compte GitHub.

Interface en ligne de commande

Utilisée par beaucoup de gens , très puissante si vous êtes à l'aise avec un terminal.

Installation :

- Ubuntu : `sudo apt-get install git`
- OS X : <https://sourceforge.net/projects/git-osx-installer/>
- Windows : <https://git-for-windows.github.io/>
(déjà installé à l'UCL)

Documentation :

- La référence : Git book : <https://git-scm.com/book> :
abordable, bien expliqué et très complet!
- `git help`, `git <command> help`
- TODO : autre bon tuto?

- <https://git-scm.com/docs/gitk> (Installé par défaut sur PC UCL)
- <https://www.gitkraken.com/>
- <https://desktop.github.com/>
- D'autres : <https://git-scm.com/downloads/guis>

Questions?