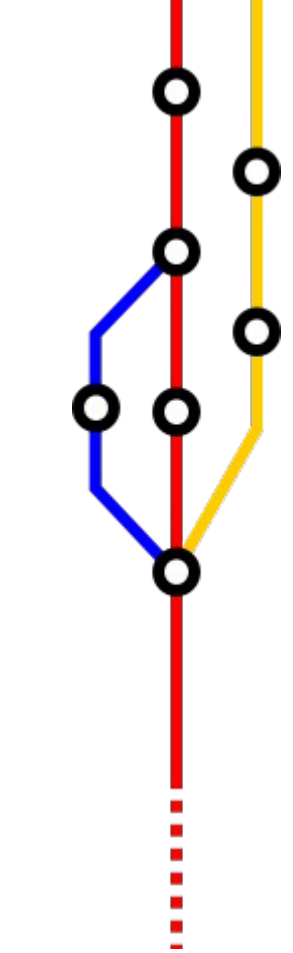
Club Informatique

Session git CIE

Sommaire



Introduction

Version control

Le problème :


Projet_vrai


Projet_vrai_Leo

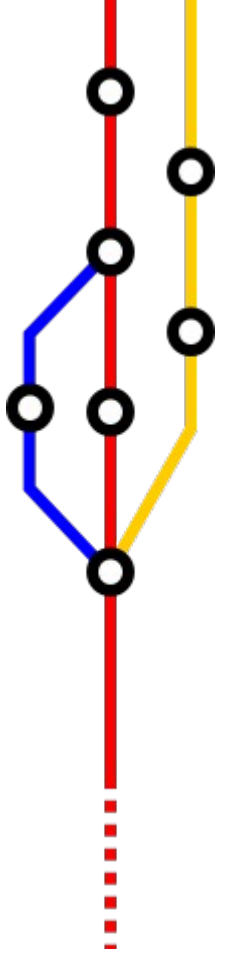

Projet_vrai_Remi


Projet_vrai_v2


Projet_vrai-Director_cut

Introduction

La solution git



Réputé et opensource

Initiallement par Linus Torvald

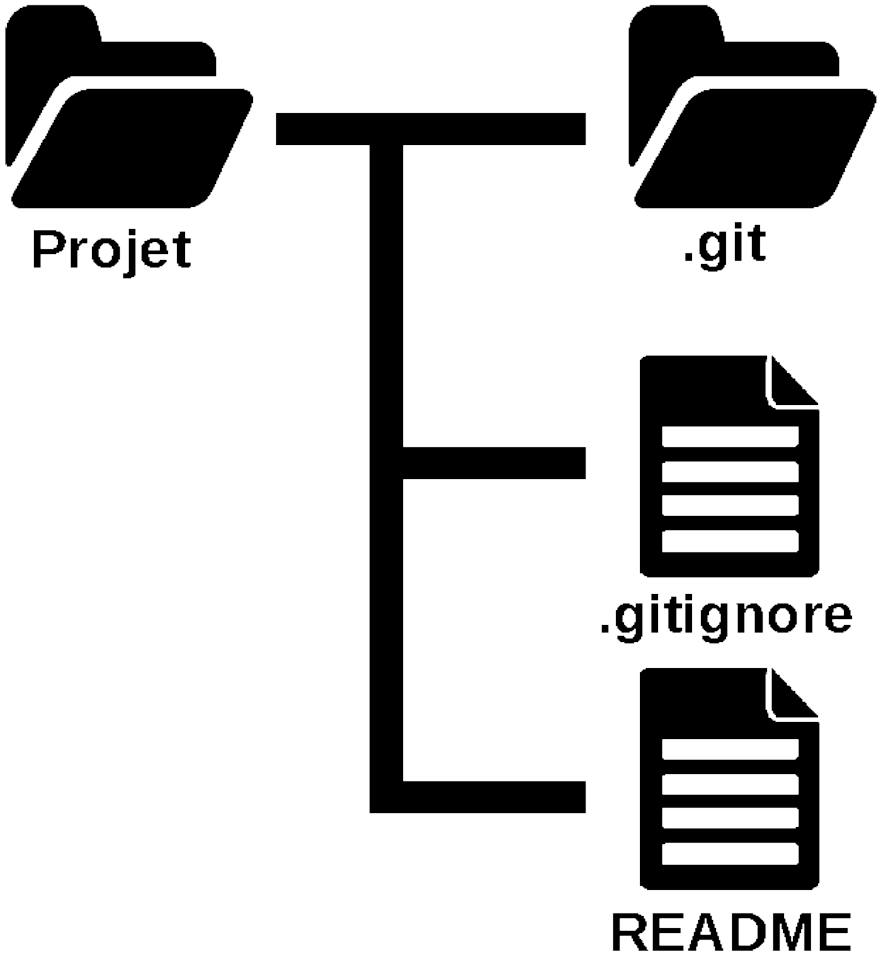
Multiples intégrations et GUI

Modèle distribué

Coucou les 4A

Les mécaniques fondamentales

le repo – le dossier de travail



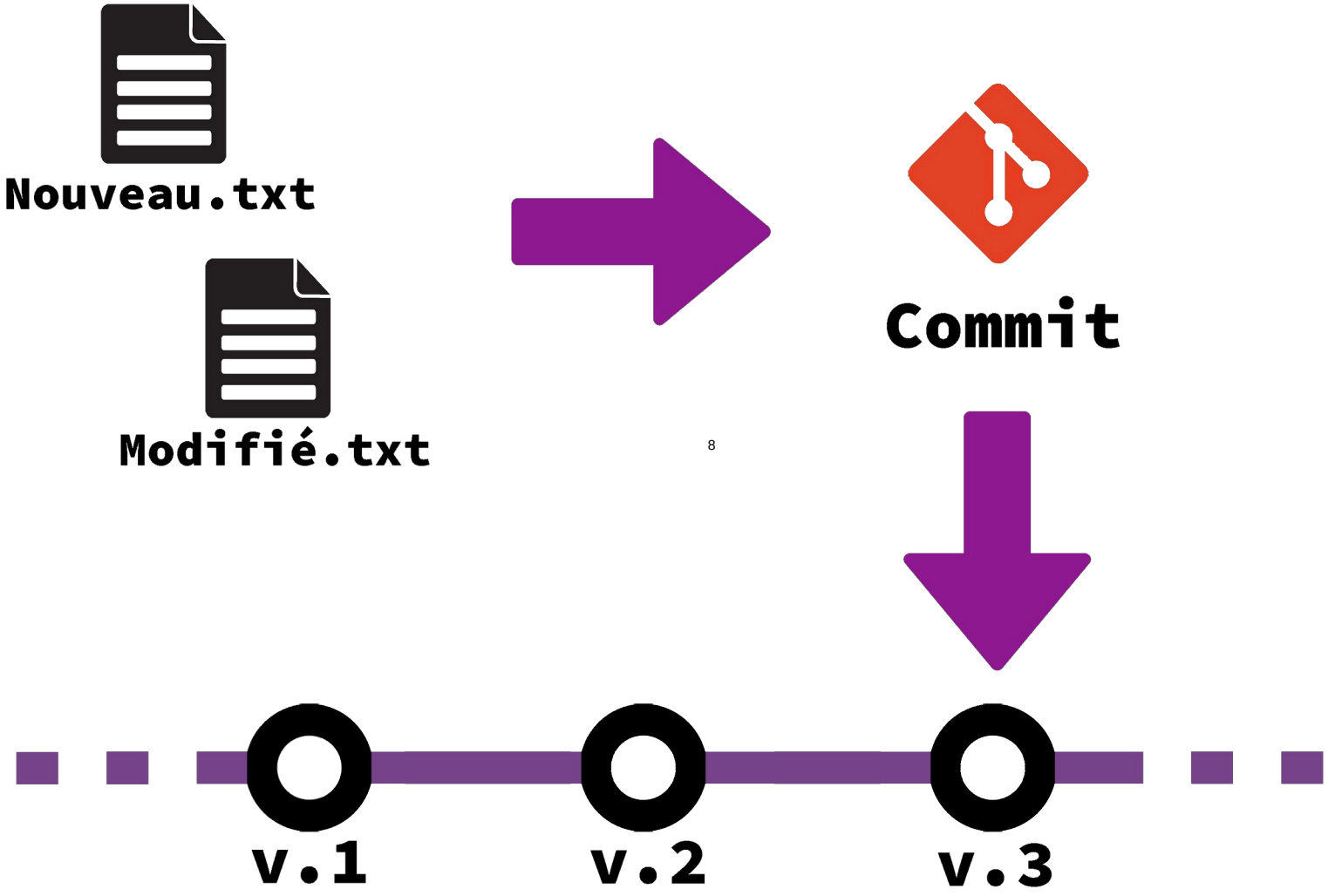
- Dossier magique
- Créé par git, géré par git

- Document texte optionnel
- Liste de fichiers ignorés par git
- Regex

- Document optionnel
- Description du projet
- Lisez le

Les mécaniques fondamentales

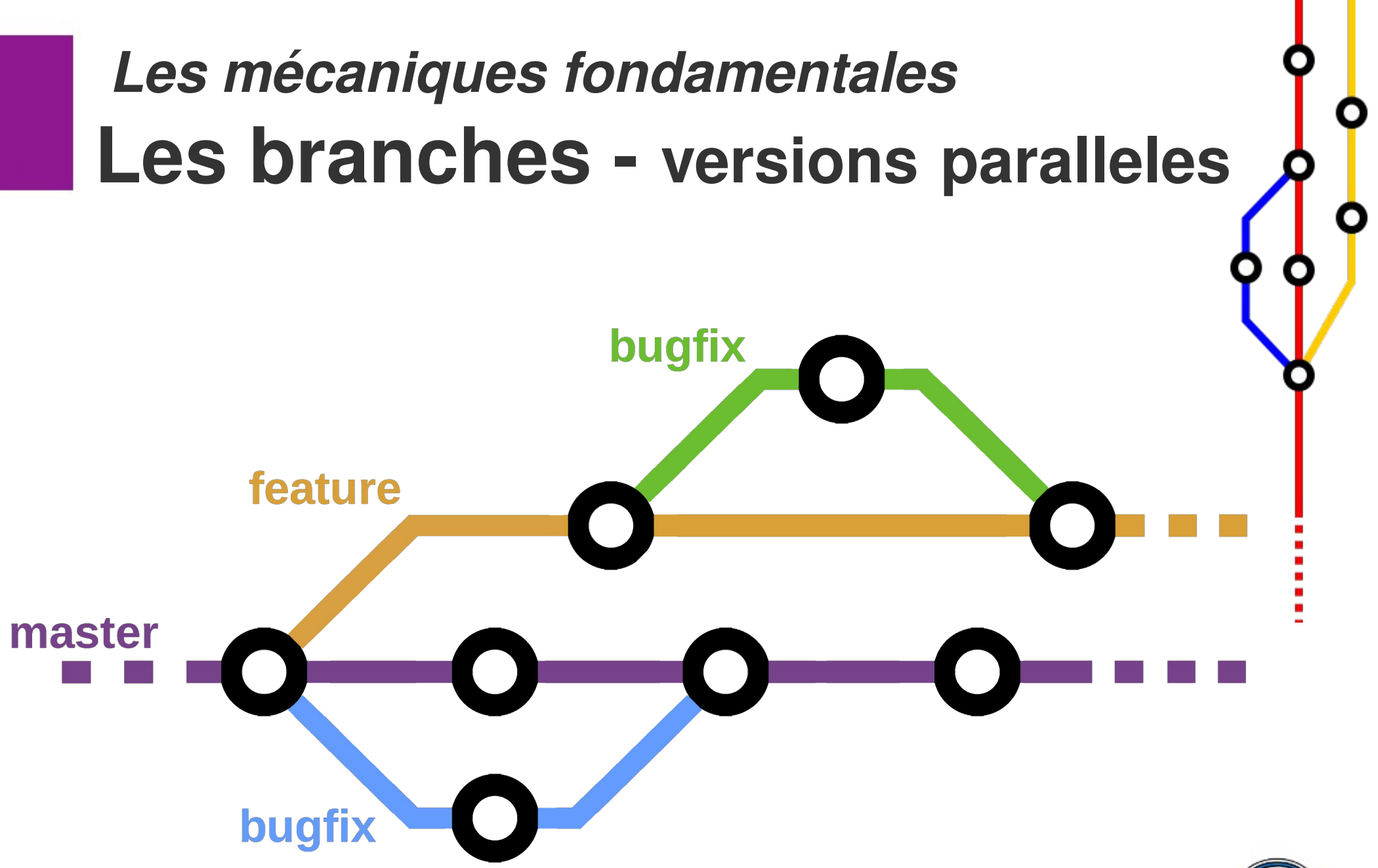
Les commits – la sauvegarde



Club Informatique

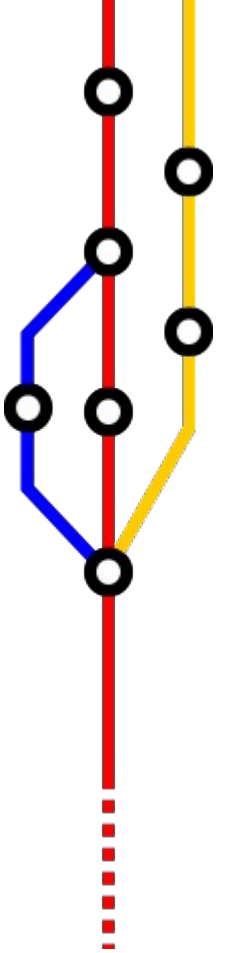
Les mécaniques fondamentales

Les branches - versions paralleles

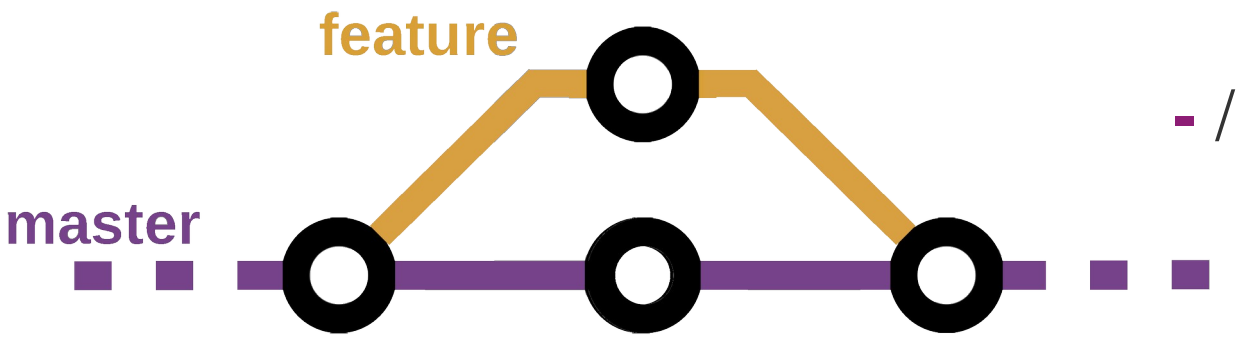


Les mécaniques fondamentales

Merge et rebase – les noeuds

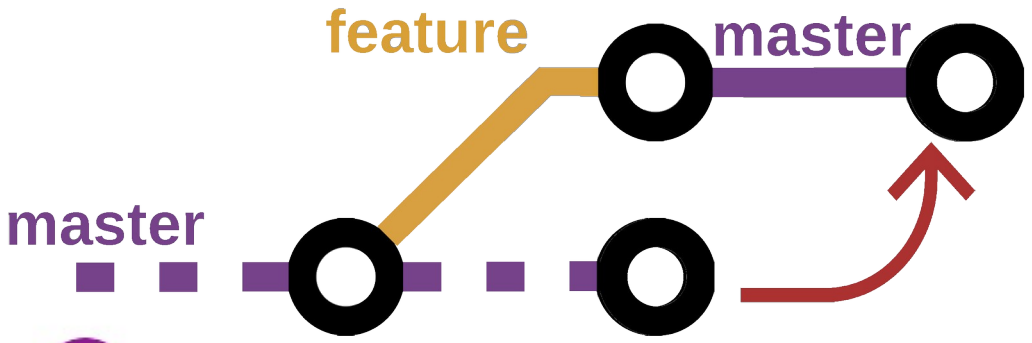


Merge



- /!\ sac de noeuds

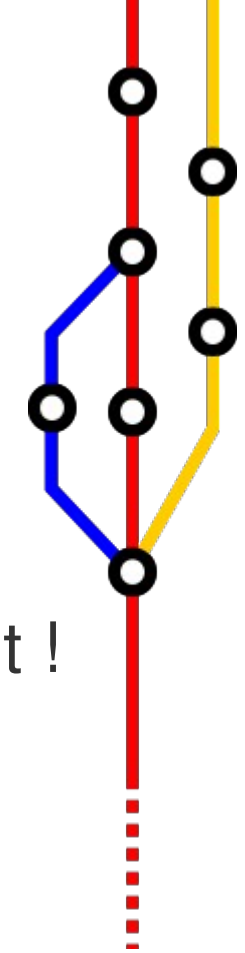
Rebase



- Ré-écriture de l'historique

Les mécaniques fondamentales

Les conflits – le problème



Une version est une liste de modifications

Modifications contradictoires suite a une fusion = conflit !

Des solutions automatisés

- Toujours favoriser une branche
- Algorithmes (3-way merge)

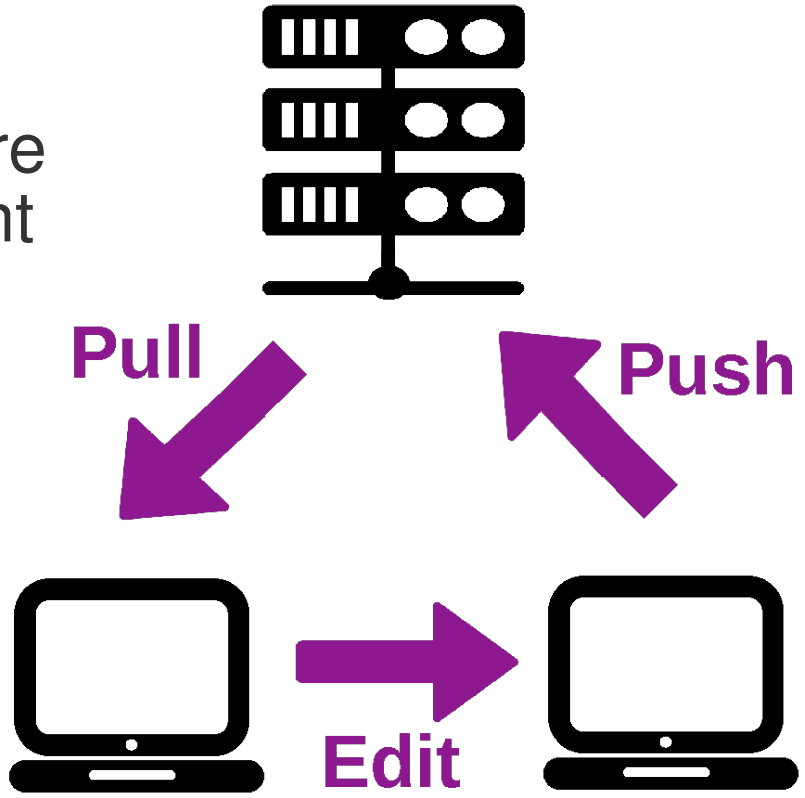
A la main (meld, kdiff3 , … , éditeur de texte)

Le modèle distribué

Chacun son repo !

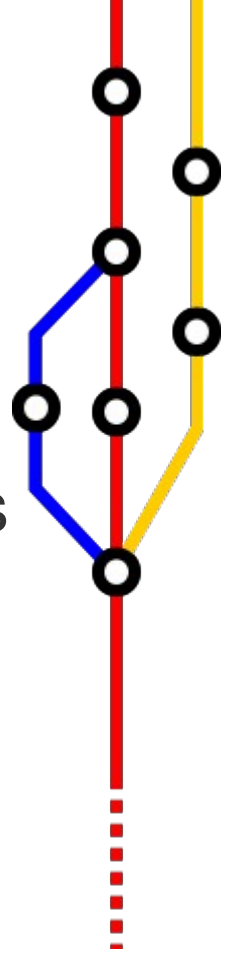
Un repo “distant” pour tout le monde

Synchronisation entre notre repo local et le repo distant



Git en équipe

Méthode de travail / Normes



On ne travaille pas directement sur les branches partagées

- Chacun ses branches
- Exploiter les merge requests (requête de fusion)

Le propriétaire de la branche cible valide les merge requests

- Les admins s'occupent des branches partagées (master)

Heberger le repo distant

Les hébergeurs publiques :

- Github (Microsoft)
- Framagit (Framasoft)

Auto-hébergement possible !

- Gitlab
- Serveur Git natif

Interfaces graphiques d'administration

Objectifs

Installer et configurer git

Créer / rejoindre un projet git

Découvrir / préparer le projet

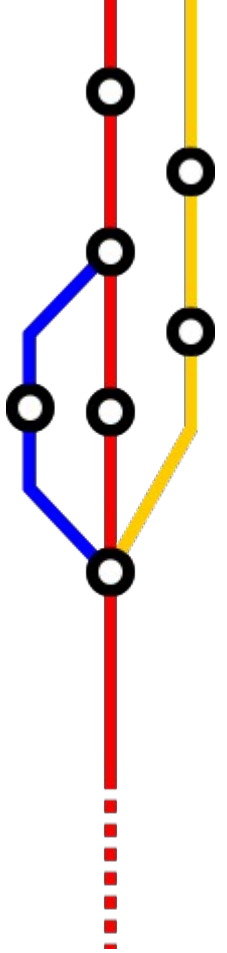
Editer le projet

Pousser nos modifications sur le repo distant

Fusionner nos modifications dans les branches partagées

/!\ Commandes avec l'utilitaire git "standard"
peuvent varier de l'utilisation avec votre GUI / IDE /!\

Installation et configuration



Téléchargement (package git pour linux)

http://git-scm.com/download

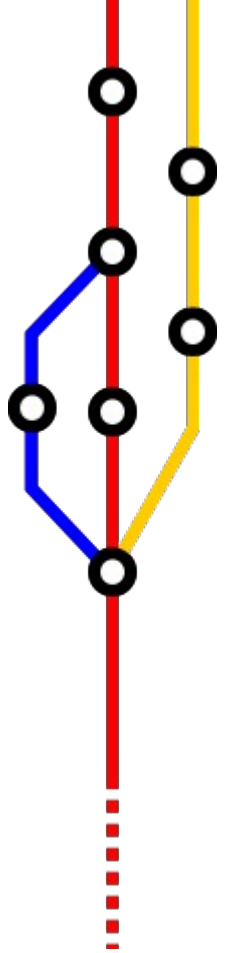
Création de son “identité”

$ git config (--global) user.name "<pseudo_git>"

$ git config (--global) user.email "<email_git>"

- /!\ Informations publiques dans chaque commit /!\

Créer / rejoindre un projet git



Créer un repo local dans le dossier courant

$ git init

- Création d'un dossier .git

Cloner un repo distant

$ git clone <url_repo_distant>

- Copie les fichiers et le .git (historique, branches ,...)

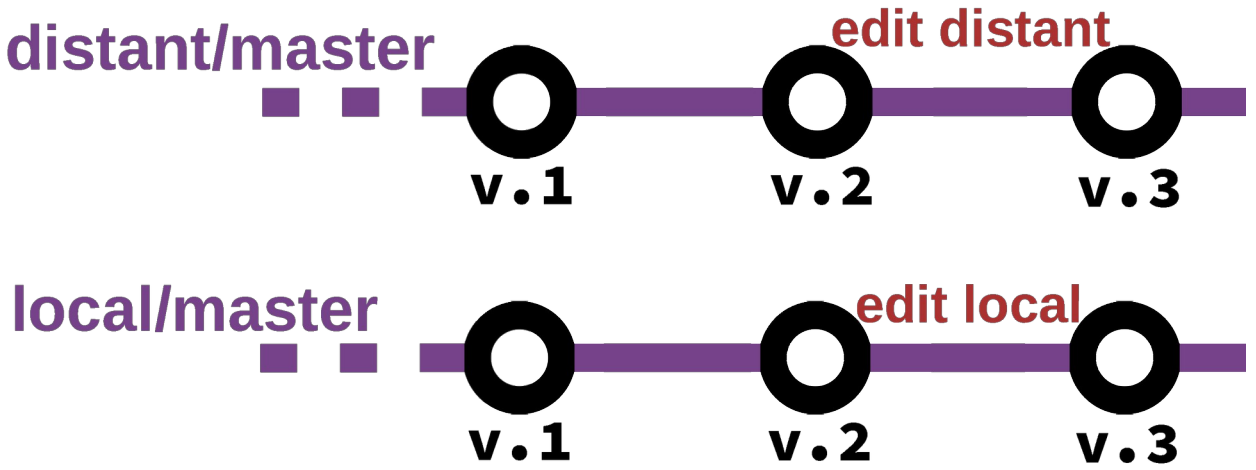
Synchroniser un projet déjà cloner

Synchroniser la branche avec le repo distant

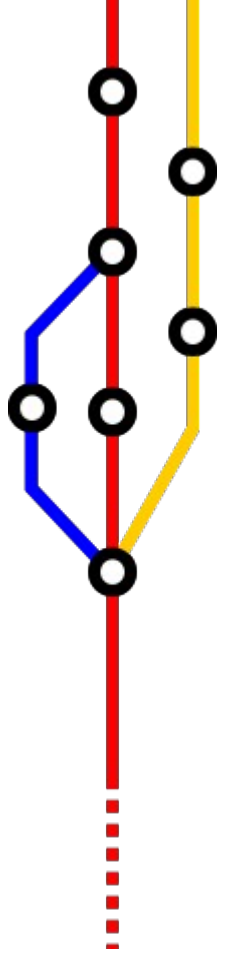
$ git pull

- Assurez vous de pull avant de travailler sinon ...

Potentiels conflits si la branche locale a été modifiée à partir d'une version antérieure de la branche



Découvir / préparer le projet



Le README !

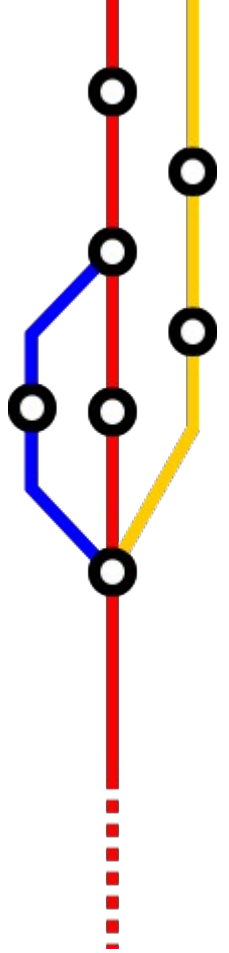
- Fichier texte / markdown
- Pour savoir ce qu'il se passe
- Description du projet, règles et normes de dev

(optionnel) Définir le serveur distant

$ git remote add <label_serv.> <@_serv.>

- Pour les nouveaux projets partagés

Editer le projet – les branches



Lister les branches

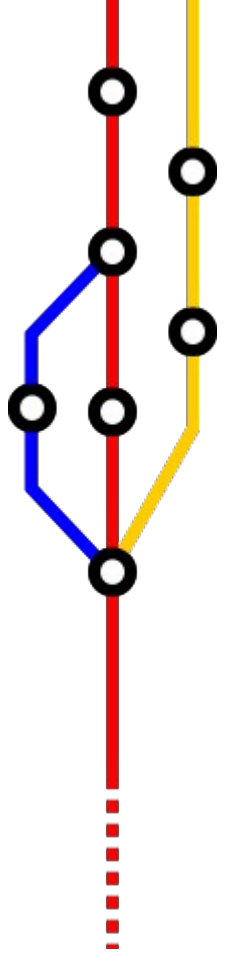
$ git branch

Aller sur une branche

$ git checkout [-b] <nom_de_branche>

- -b pour la créer

Editer le projet – le commit



On ajoute nos nouveaux fichiers au suivis de git

$ git add <fichier 1> <fichier 2>

- Expressions regex (wildcards * , range A-Z, ….)

(bonus) Vérifier les fichiers suivis

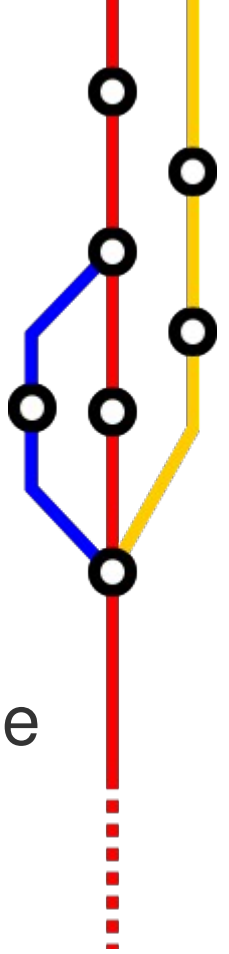
$ git status

Le commit (sauvegarde / nouvelle version git)

$ git commit [-m message de commit]

Une journée avec git

Pousser nos modifications sur le repo distant



Synchroniser la branche sur le repo distant

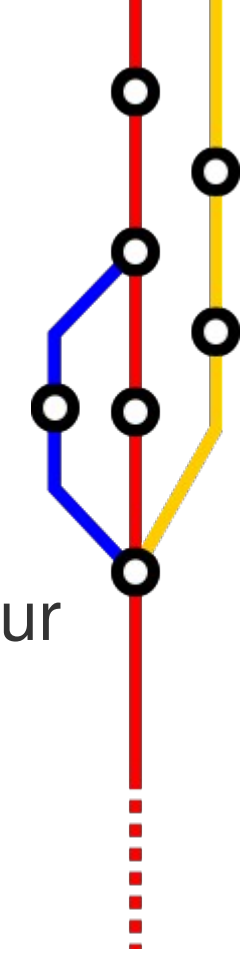
$ git push [-u] [branche]

- La branche actuelle par défaut
- -u lie une nouvelle branche à sa source en historique

Potentiels conflits si un autre push a été fait sur votre branche

- Force à toi

Une journée avec git
Fusionner avec les branches partagées



Faire une merge request sur le serveur distant

- Trouver le bouton “merge request” chez votre hébergeur

Valider la merge request

- Considérer les potentiels conflits
- Régler les conflits (souvent possible depuis le GUI)

Conclusion

Liens utiles

Documentation git

https://git-scm.com/docs/

Github cheatsheet

https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf

Guides en ligne

- Les copains de l'IUT Valence

  https://github.com/benito103e/revealjs-formation-git-iut

Conclusion

C’est fini