Git - Versionning

Bocar COLY

GIT

Qu'est-ce que GIT?

- Git est un système de contrôle de version de projet informatique
- Git est Open Source donc gratuit et mis à jour par la communauté
- Capable de gérer à la fois des petits et des très grand projets

Un peu d'histoire…

Historique

Créé en 2005 par **Linus Torvalds.**

git

Avantages

Performances

- Commiter de nouveaux changements
- Créer des branches
- Faire des merges
- Comparer les anciennes versions

Avantages

Sécurité

- L'intégrité du code source
- Sécurisé à l'aide d'un algorithme de hachage sécurisé appelé le SHA-1
- Protège le code et l'historique des changements contre toute modification accidentelle ou malveillante, tout en assurant une traçabilité complète de l'historique.

Avantages

Flexibilité

- Efficacité dans l'élaboration de projets de différente envergure
- Compatibilité avec de nombreux systèmes et protocoles existants
- Prend en charge le branching et le tagging en priorité
- Les opérations sur les branches et les tags sont également stockées dans l'historique des modifications

Contrôle de version

- Quand le fichier a été modifié
- Qu'est-ce qui a été modifié
- Pourquoi il a été modifié
- Qui a modifié le fichier

Exemple

Installation

http://git-scm.com

Lignes de commande

Git Bash que nous venons d'installer fonctionne avec les lignes de commande, avant de passer à la suite nous allons prendre rapidement en main l'invité de commandes de Git Bash.

Un peu de pratique…

Commande GIT

Voici les principales commandes GIT que nous allons utiliser, bien évidemment il en existe d'autres que vous allez rencontrer dans votre carrière de développeur mais pour l'instant nous allons nous concentrer sur l'essentiel.

- Clone: copie sur sa machine (en local) un repository qui est hébergé sur un site distant tel GitHub
- Add: Ajoute le(s) fichier(s) et modification(s) dans Git
- Commit: sauvegarde vos fichiers dans Git
- Push: met à jour les commit git sur un repository distant comme GitHub par exemple
- Pull: télécharge les changements d'un repository (dépôt) distant sur votre machine local (c'est l'inverse de push)

Commande GIT

GitHub Cheat Sheet

Versionner son travail

Versionner en local

git init	initialise le dépôt (se mettre sur le bon dossier), mieux à faire depuis Github.com
git add .	ajoute toutes les modifications (le . symbolise tout)
git commit -m "explication"	créer un nouveau commit. **git add** pousse les fichiers en zone d'index, **git commit** les sauvegarde réellement dans un nouveau commit

Gérer les commits

git log	liste des commits
git log -n2	affiche les 2 derniers commits
git show sha-1	voir commit spécifique (cliquer molette souris pour coller)
git checkout sha-1	remettre la version du sha-1
git checkout main	remettre version la plus récente

Versionner sur un dépôt distant

git clone lien-github.com	récupérer travail depuis dépôt distant
git push -u origin main	pousse les modifications vers serveur
git push -f origin main	pousse de force des modifications (à manipuler avec précaution)

Naviguer dans Git Bash

pwd	savoir dans quel dossier je suis
mkdir "dossier"	créer un dossier (Make Directory)
touch fichier.txt	créer fichier
ls	liste le dossier courant
ls -la	liste tout plus précisément que ls
cd dossier	aller dans le dossier (Change Directory)
cd ..	Remonter d'un dossier

Initialisation de Git

git config --global user.name	"Mon Nom"
git config --global user.email	mon@mail.com
git config --global --list	Affiche nom et mail

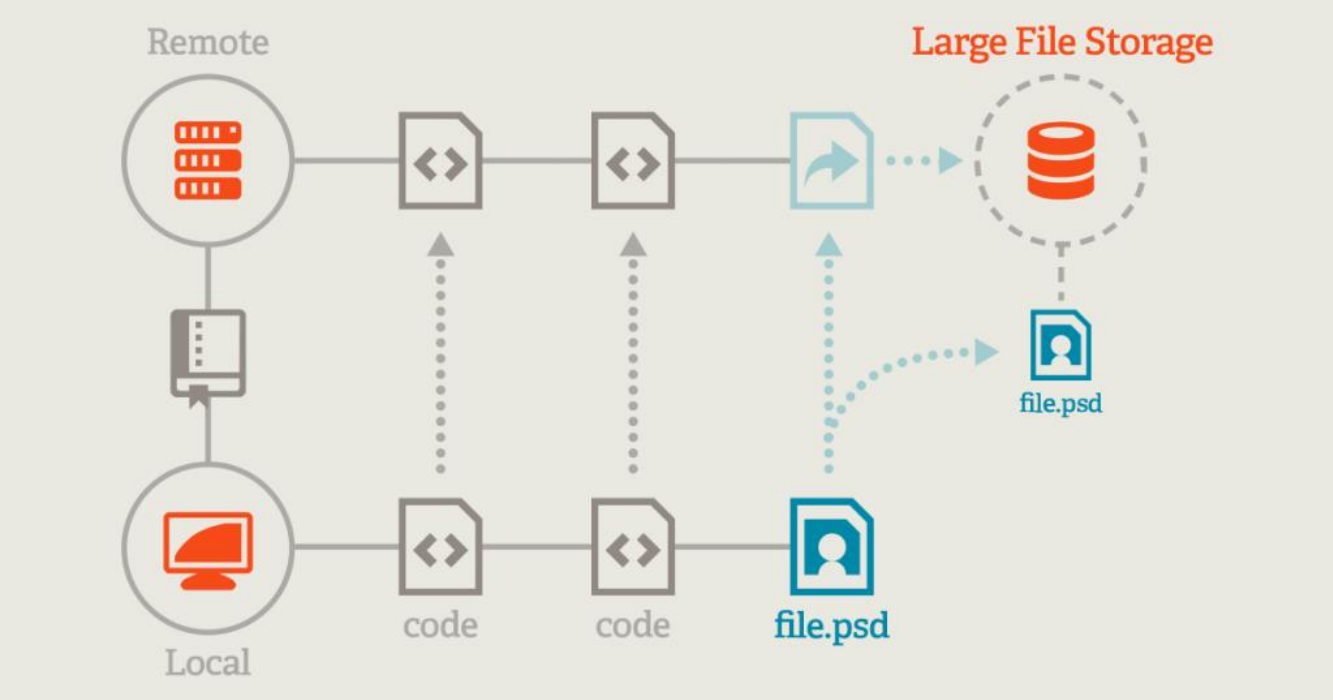
Autres commandes

git status	état du fichier
git diff	affiche les modifs avant commit

Commit son projet sur Github

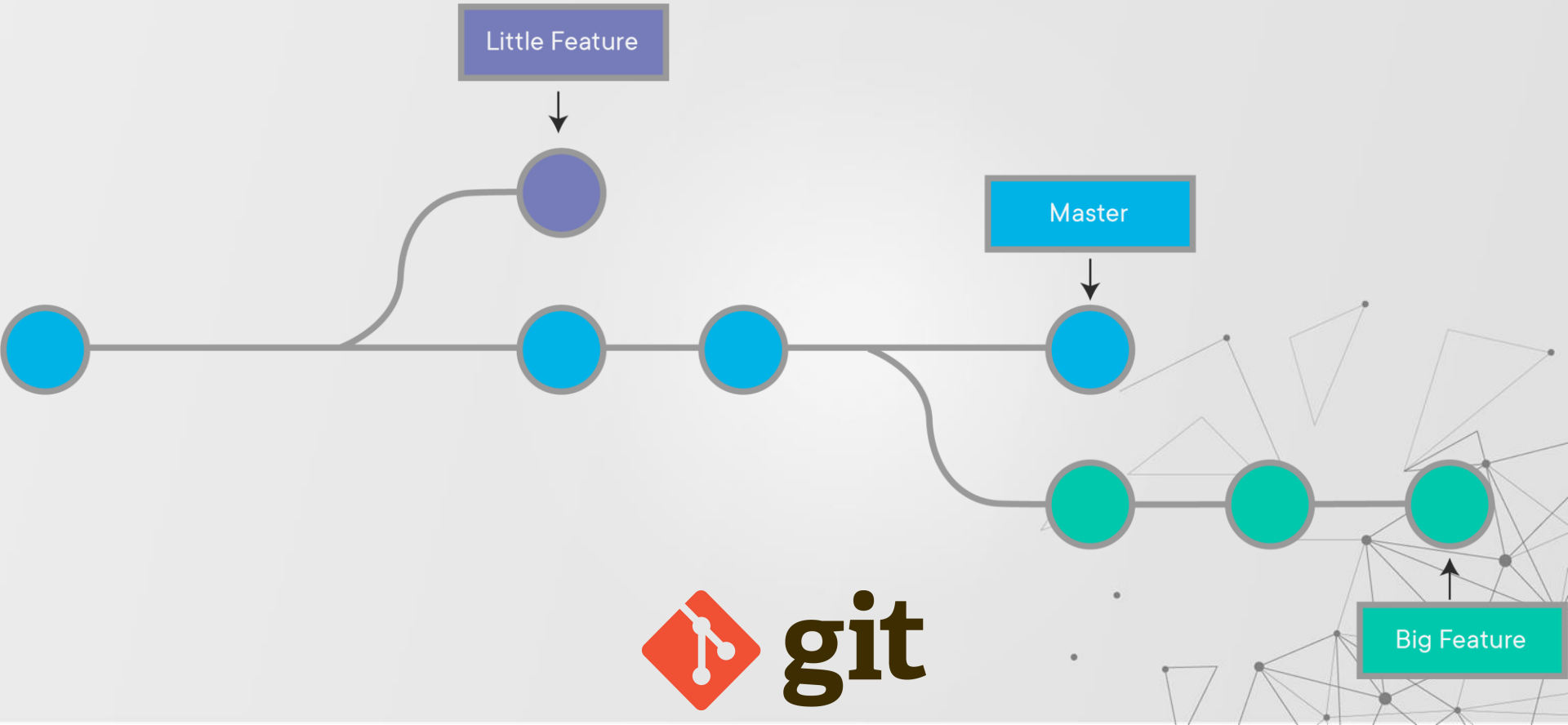
git add .
git commit -m "message"
git push -u origin main

Place à la pratique



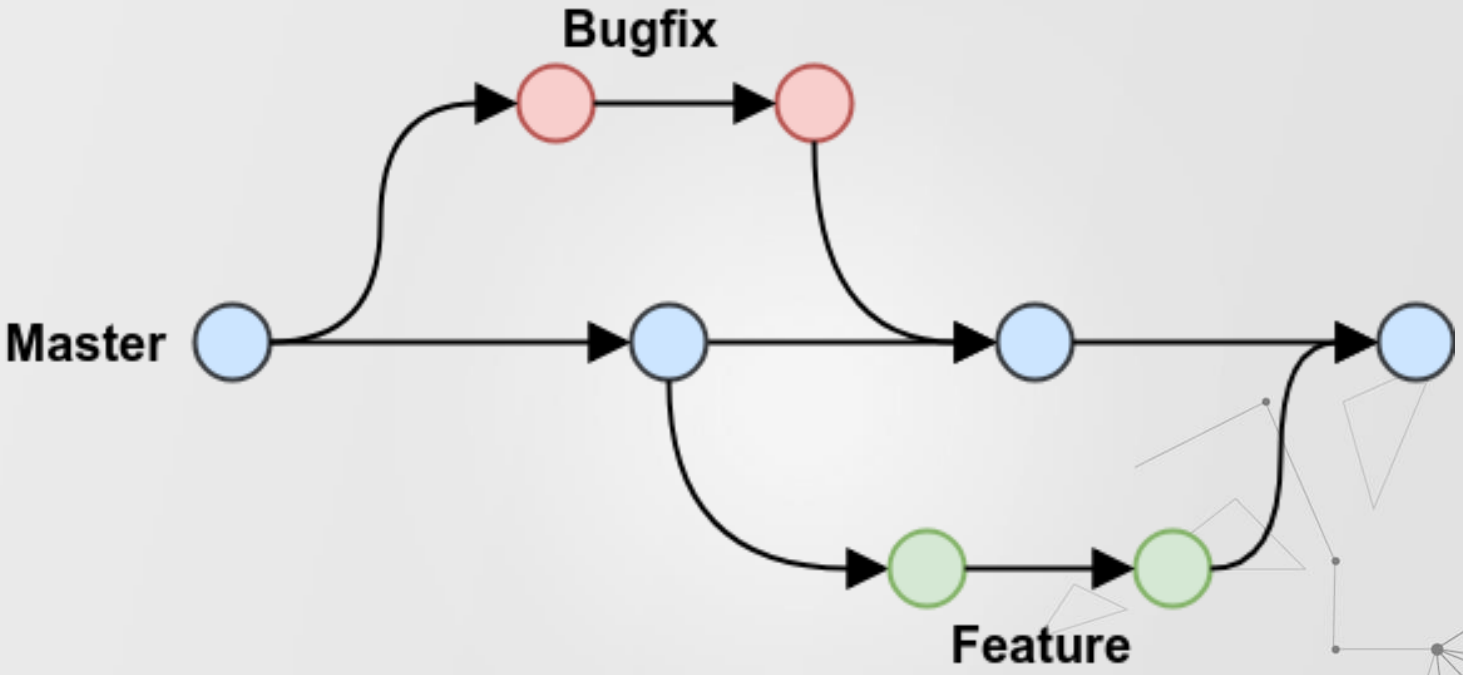
git

Notion de branche



git

Notion de branche



git

Commande GIT

Création d'une nouvelle branche

Git branch + nom de la branche

Ex: git branch corentin_feignant

Changer de branche

Passer de la branche master ou actuelle vers notre nouvelle branche

Git checkout + nom de la branche

Ex: git checkout corentin_feignant

Commande GIT

Créer un merge (une fusion)

Git merge + nom de la branche à rapatrier

Ex: git merge corentin_feignant

Démonstration
Mieux comprendre les merge

https://onlywei.github.io/explain-git-with-d3/

Conflits de merge Git

On va souvent rencontrer des conflits de merge avec Git, il n'y a pas de réponse universelle votre principal ami reste Google…

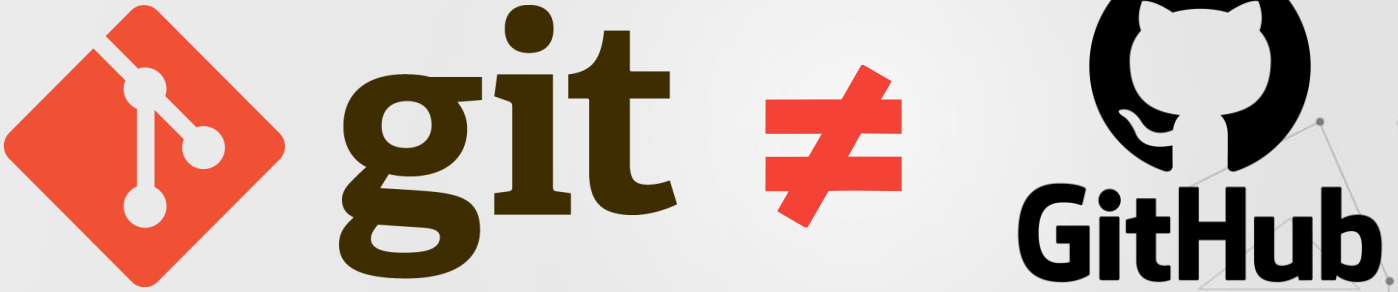
Sinon Atlassian nous fournit une documentation assez claire pour la gestion des erreurs de merge.

https://www.atlassian.com/fr/git/tutorials/using-branches/merge-conflicts

Dépôt distant

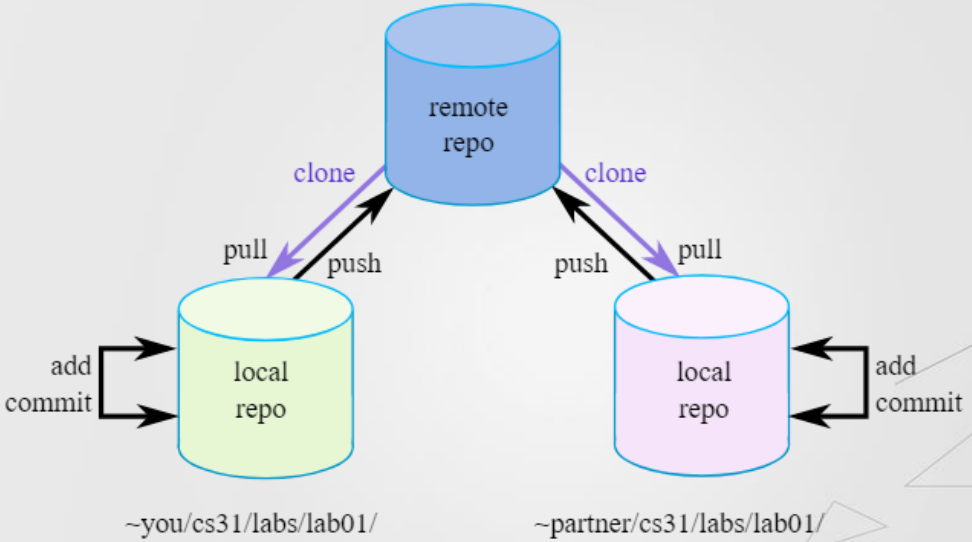
Pour configurer un repository (dépôt) distant

Git remote add + URL de votre repository


git
≠
GitHub

Remote

Comment marche le remote?



Pull / clone

Effectuer un pull sur un repository (dépôt distant)

Git pull <remote> + nom de la branche

Effectuer un clone sur un repository (*très utile pour récupérer un projet*)

Git clone + url du dépôt git

Git Application

Bon j'ai pensé encore à notre ami Corentin qui a la flemme de retenir toutes les commandes et de les exécuter à chaque fois… Bon heureusement qu'il existe des outils permettant de le faire en GUI…

On va utiliser et tester dans notre situation une application très répandue qui est GitKraken:

https://www.gitkraken.com/download

GitKraken

En apprendre plus sur Git

Vidéo sur Git:

https://grafikart.fr/formations/git

Vidéo sur GitKraken:

https://grafikart.fr/tutoriels/gitkraken-749