

# 04 - INTRODUCTION RAPIDE À GIT

IPA 11x001 exercices - Université de Genève

---

Stéphane Nguyen

[https://github.com/Zenchiyu/11x001\\_tp](https://github.com/Zenchiyu/11x001_tp)

# CONTENTS

---

1. Git c'est quoi ? Et GitHub ?
2. Installation
3. Tutoriel
4. Tutoriel cont.
5. Liens complémentaires

- **Git**: système de gestion de versions décentralisé (Version Control System VCS) installé localement sur votre ordinateur.

Plutôt que d'ajouter un numéro à chaque nouvelle version d'un fichier, Git enregistre les différences (deltas) à chaque modification. Vous pouvez donc

- Suivre l'historique de votre projet & revenir facilement aux anciennes versions.
- Travailler en parallèle sur différentes versions de votre projet à travers des “branches”.
- **GitHub et Gitlab**: plateformes web permettant d'héberger des dépôts Git (repositories, pensez à projets). Elles offrent des fonctionnalités collaboratives dont:
  - suivi des problèmes (issues),
  - revue de code,
  - intégration continue/déploiement (CI/CD) (ex. couverture de tests)

- Git devrait déjà être installé dans Ubuntu dans WSL 2
- Idem pour **macOS**. Cependant, si pas installé: <https://git-scm.com/downloads/mac>
- Vous allez voir quelques commandes mais **vous n'aurez juste besoin de deux durant le semestre: git clone et git pull.**

- Nous allons vous montrer les prochain(e)s outils et commandes en direct:
  - Création d'un dépôt/projet GitHub
  - Composition minimale: README.md et LICENSE ([explication](#))
  - Fichier gitignore
  - `git init`
  - `git clone`
  - `git status`
  - `git add`
  - `git commit`
  - `git push` (avec set upstream)
  - `git pull`
  - `git diff`
  - `git log`

- Concepts en lien avec les “branches”:
  - `git checkout`
  - `git branch`
  - `git merge` et pull request sur GitHub
- Après avoir `git clone` le dépôt des exercices la première fois, à chaque fois que nous ferions des modifications, allez dans le dossier du dépôt et tapez donc `git pull` dans votre terminal !

Si vous le souhaitez, vous pouvez lire plus sur Git et GitHub.

Voici quelques liens intéressants de concepts dont certains n'ont pas été expliqué:

- [Explication de Git par Fireship](#)
- `git stash` (l'[explication de git stash ainsi que d'autres commandes](#))
- `git rebase` (l'[explication de merge vs rebase](#))