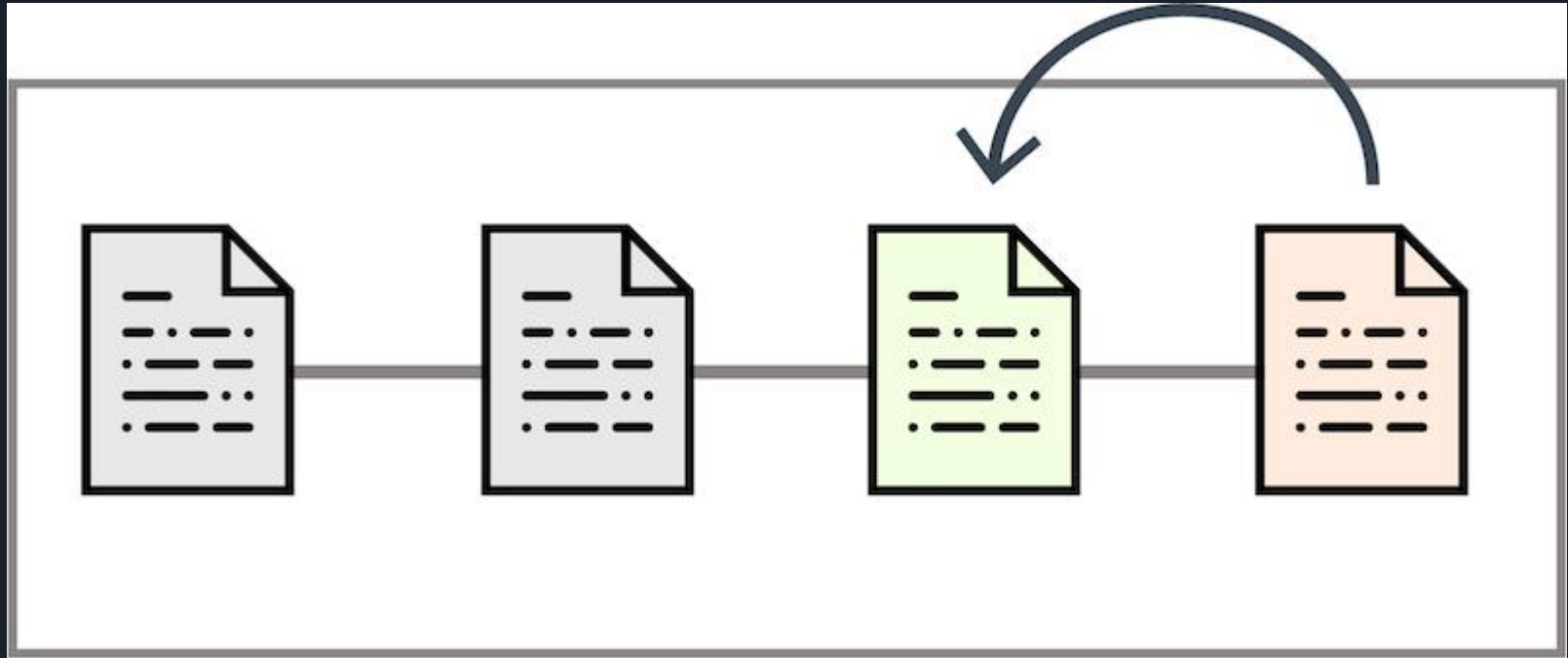




GIT & GITHUB

Théau Debureau

Git: Un gestionnaire de versions





Git: Un gestionnaire de versions

Conserve un historique des modifications et des versions des fichiers en gardant en mémoire:

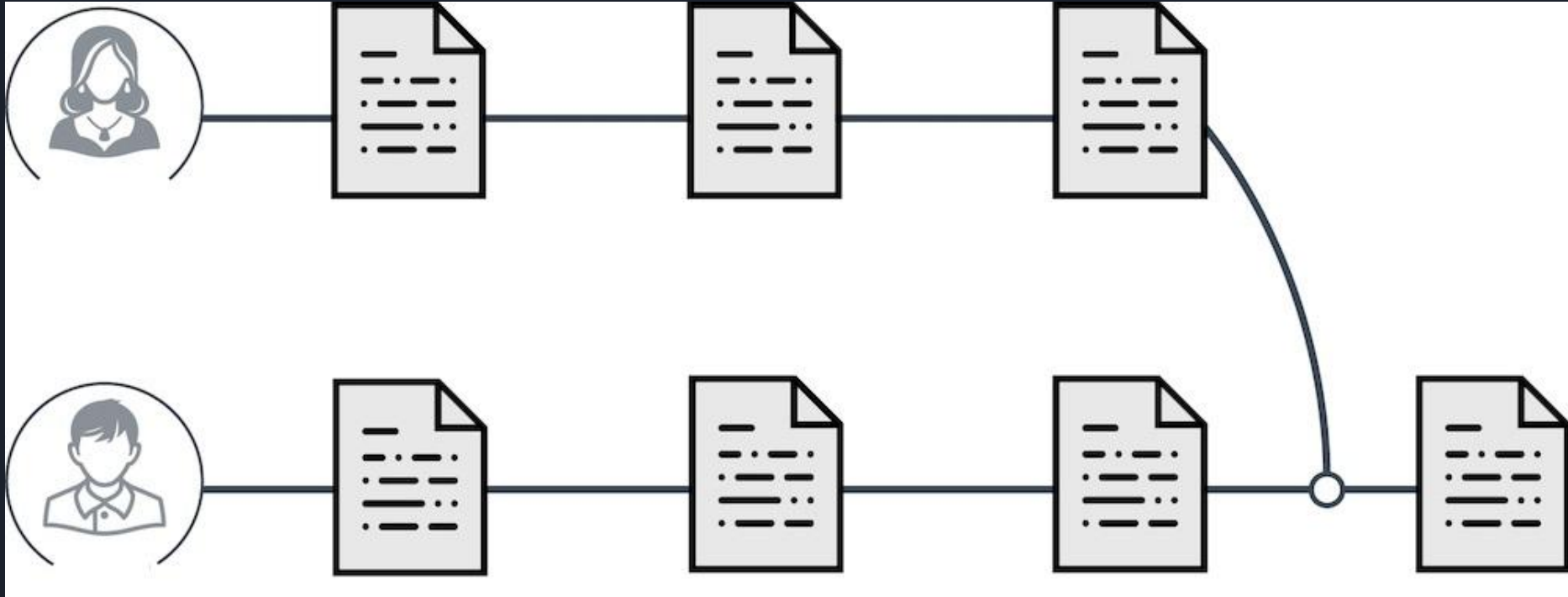
- chaque modification de chaque fichier
- pourquoi elle a eu lieu
- par qui cela a été réalisé



Les 3 grandes fonctionnalités de Git

- 1) Revenir à une version précédente de votre code en cas de problème
- 2) Suivre l'évolution d'un développement étape par étape
- 3) Améliorer la collaboration dans un projet en éliminant le risque de supprimer des modifications d'un membre de l'équipe

Git et le travail d'équipe






Git ou GitHub: quelle différence?

Git: gestionnaire de version local

Github: service en ligne pour héberger votre dépôt

Installer Git: *https://git-scm.com/*



git --distributed-is-the-new-centralized

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

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](#).

Downloads

[macOS](#) [Windows](#)

[Linux/Unix](#)

Older releases are available and the [Git source repository](#) is on GitHub.

Latest source Release
2.33.0
[Release Notes](#) (2021-08-16)

[Download for Mac](#)

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

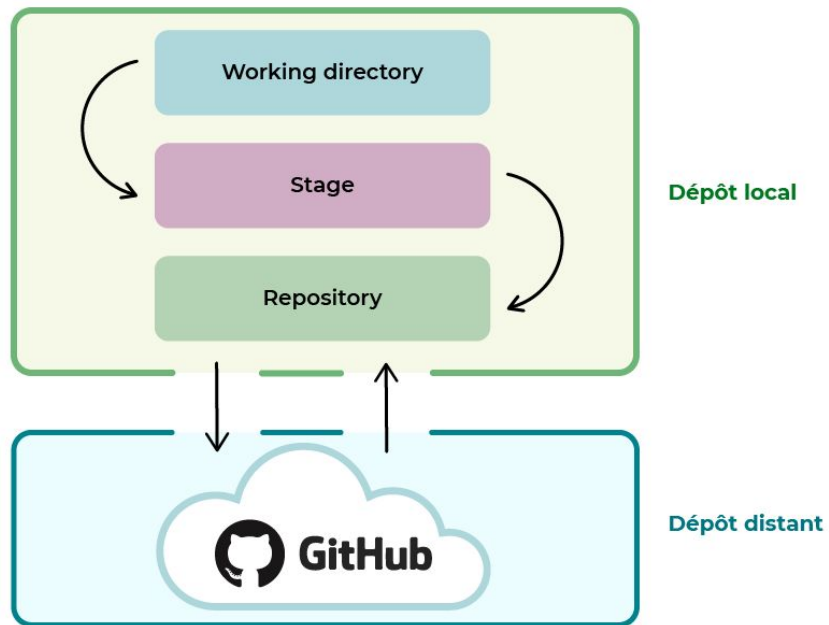


Git: Configurer votre identité

```
$ git config --global user.name "Prenom Nom"
```

```
$ git config --global user.email "moi@mail.com"
```


Git: travailler depuis un dépôt local



`git init`

`git add monfichier.ext`





`git commit -m "ajout du premier fichier"`

`git push -u origin main`

Github: création de compte



Github: création de repository


 Search or jump to... [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)   

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *


 Flackguy ▾

 /

Great repository names are short and memorable. Need inspiration? How about [curly-octo-garbanzo?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None** ▾

① You are creating a public repository in your personal account.



Git: gestion du système de branches

- git branch: liste les branches disponibles
- git branch nouvellebranche: crée une nouvelle branche
- git checkout nouvellebranche: permet de changer de branche
- git merge nouvellebranche: fusionne la nouvellebranche sur la branche actuelle



Git & Github: travailler avec un dépôt distant

- git clone url-depot-distant: récupère une copie du dépôt distant
- git remote add origin url-depot-distant: permet de se lier au dépôt distant
- git fetch: télécharge un contenu depuis un dépôt distant, souvent suivi d'un merge
- git pull: permet de fetch et merge en une commande



Git: corriger ses erreurs sur les branches

-git branch -M main: passe votre branche main comme branche principale

-git branch -d branchetest: supprime la branche “branchetest”

-git branch -D branchetest: supprimer la branche si des modifications ont été apportées



Git: les logs et historique de modifications

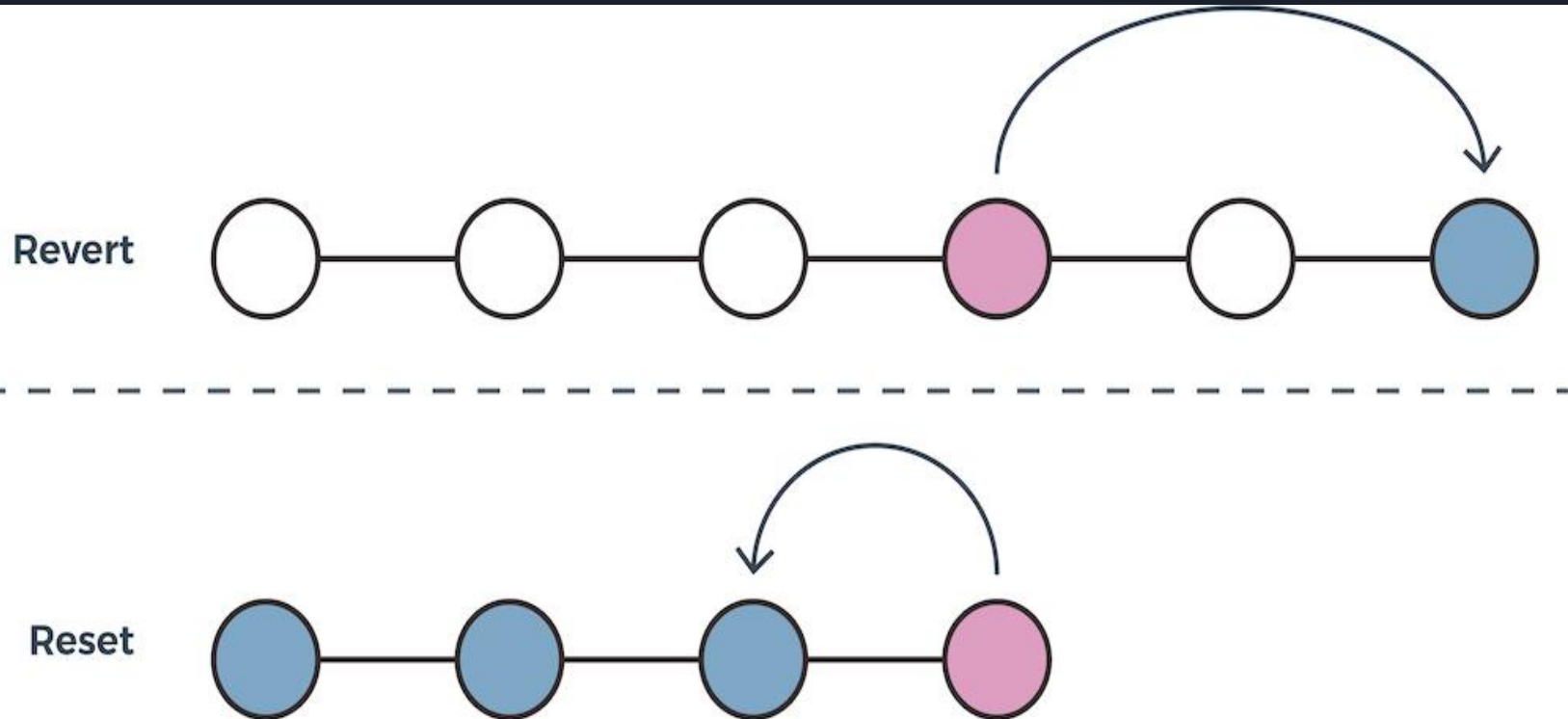
-git status: permet de voir dans quel état sont les fichiers modifiés

-git log: liste les commit par ordre chronologique inversé

-git blame fichier.ext: liste les modifications apportées sur un fichier

-git reflog: log tous les commits et actions effectuées en local

Git: revenir en arrière avec git reset & git revert





Git: corriger et agir sur ses commits

-git commit --amend -m "nouveau message": modifie le message du commit précédent

-git add fichier.ext & git commit --amend --no-edit: rajoute un fichier au dernier commit

-git cherry-pick commit1 commit2: sélectionne un ou plusieurs commit et les migre sur la main



Git & Github

A vos questions!