

Bases Github


Github est une plateforme Web collaborative reposant sur Git, logiciel de gestion de versions.

Créer un repository

Un repository est un projet hébergé sur la plateforme, permettant de travailler à plusieurs.



La solution la plus simple, surtout pour les débutants, est de créer le repository sur github :

1. Créer au besoin un compte guthub si vous n'en avez pas ;
2. Allez dans **Repositories**, puis cliquez sur **New** :

 **Repositories** 112





Cocher la case Add a README file, pour que le système de fichier du repository existe (ce qui va permettre de le cloner ensuite).

Owner ^{*}  jcheron / Repository name ^{*} 

Great repository names are short and memorable. Need inspiration? How about [bug-free-potato?](#)

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.](#)

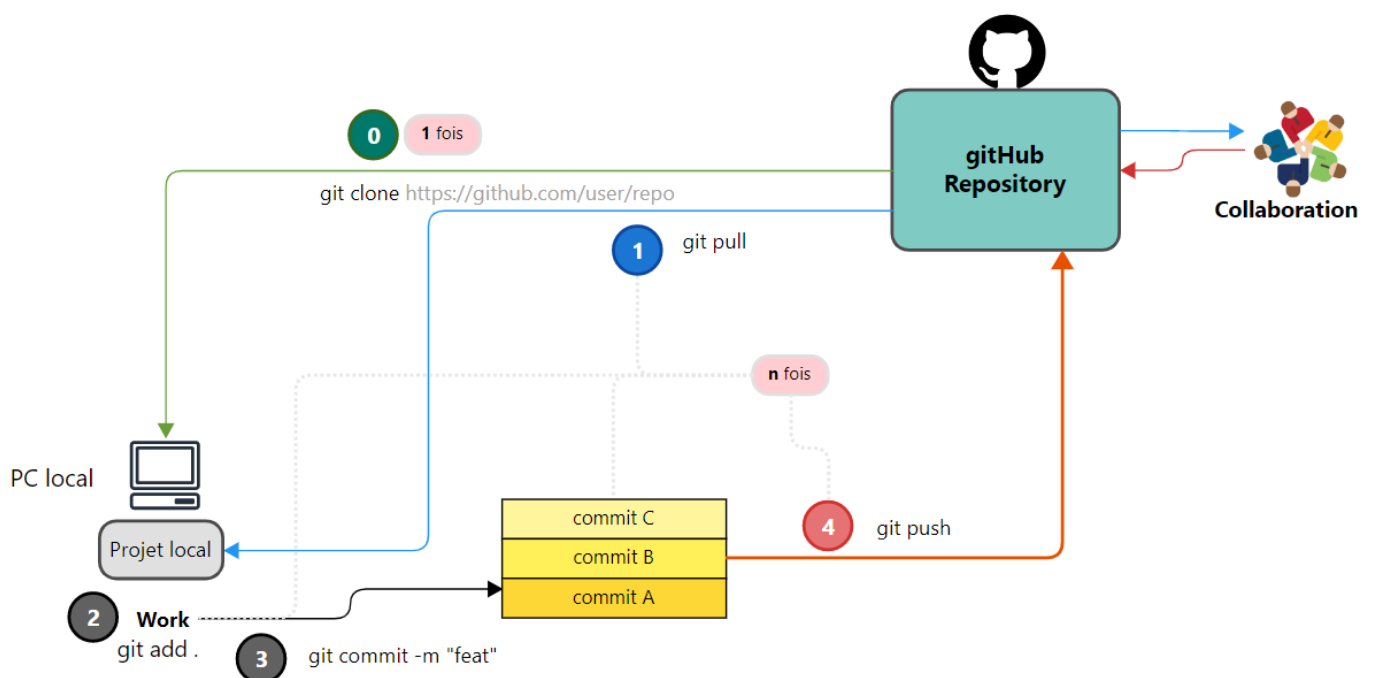
Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

[Create repository](#)

Principe



?900

Pré-requis

Paramètres Git locaux

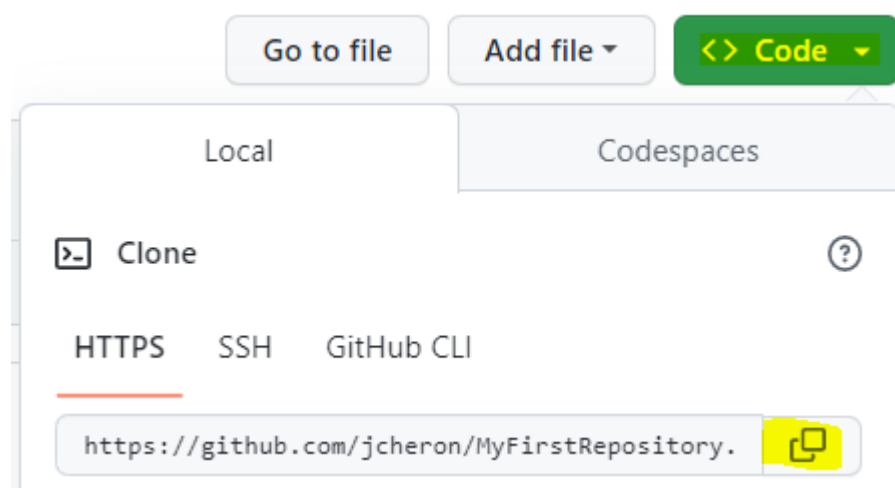
Git ayant besoin de vous identifier, [ajouter vos identifiants à la configuration locale de git](#)

Travailler en local

0 - Clone

Pour travailler en local sur le projet, il faut déjà en créer une copie en local : un clone :

Copier l'adresse du repository sur Github, avec le bouton **Code** :



```
git clone https://github.com/username/repositoryName.git
cd repositoryName
```

Ignorer des modifications

Créer le fichier **.gitignore** (il n'y a pas de commande git pour ça) Il va permettre d'ajouter les fichiers ou les dossiers à ne pas "tracker", en respectant le principe : un ligne par exclusion :

1 - Commit

- Vous avez travaillé et modifié des fichiers (ajout/modification/suppression)
- Vous pouvez créer un commit, qui représente votre progression (1 commit = 1 unité de code)

```
git status
```

```
git commit - m "feat: ajout de [NomFonctionnalite]"
```

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/cours/git-start?rev=1679505038>

Last update: **2023/03/22 18:10**

