

La puissance de git en ligne

# GitHub?

**GitHub** est un service en ligne pour héberger son *repository*

# Plateformes en ligne

**GitHub**

La plus répandue,  
sert de portfolio  
aux développeurs

**GitLab**

Principale alternative,  
peut être hébergé sur  
des serveurs privés

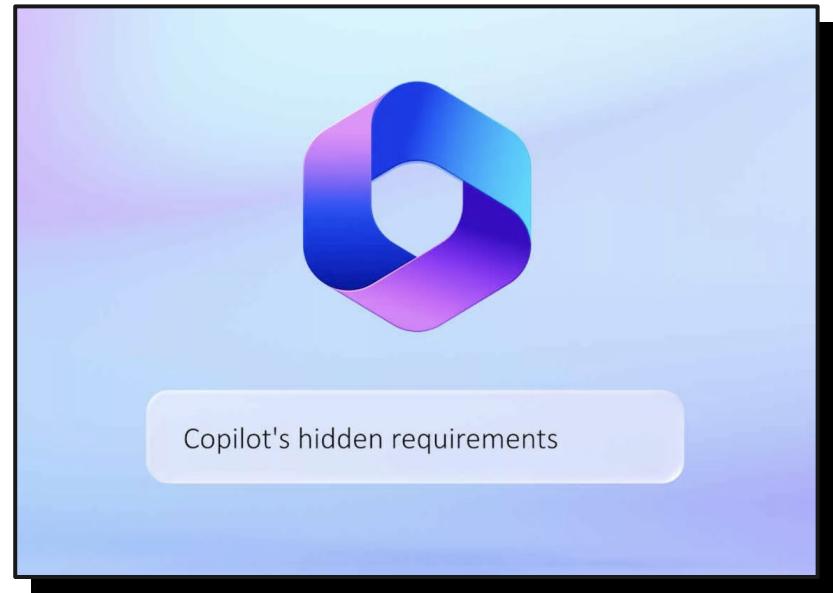
**Bitbucket**

Version développée  
par Atlassian, il en  
existe plein d'autres

...

# Prérequis licence étudiant Github

1. Ajouter son adresse mail de l'École  
[github.com/settings/emails](https://github.com/settings/emails)
2. Remplir son profil GitHub  
[github.com/settings/profile](https://github.com/settings/profile)
3. Customiser son **README**  
[github.com/<your-username>](https://github.com/<your-username>)
4. Renseigner ses infos bancaires  
[github.com/settings/billing/payment information](https://github.com/settings/billing/payment_information)
5. Activer la 2 factors Authentication
  - a. Par [SMS](#)
  - b. Puis avec l'[app mobile](#)



# Obtenir licence étudiant Github

1. Renseigner son institution  
[github.com/settings/education/benefits](https://github.com/settings/education/benefits)
2. Prendre en photo sa carte étudiant
3. Attendre la validation
4. Installer le plugin copilot sur son IDE
5. Profiter de copilot gratuitement !



Home / Benefits application

## Access free GitHub Education benefits

Complete the fields below to unlock tools and resources for your educational journey

Select your role in education \*

Teacher    Student    School

Enhance your tech skills with real-world tools

STUDENT  
FREE GitHub Pro while you are a student

STUDENT  
Valuable GitHub Student Developer Pack partner offers

STUDENT  
GitHub Campus Expert training for qualified applicants

# git / github

git

Logiciel de gestion de versions

Outil local sur sa machine  
pour modifier ses *repositories*

Ensemble de commandes  
disponibles dans le terminal

GitHub

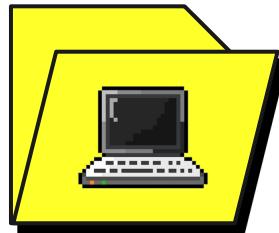
Service d'hébergement en  
ligne

Service en ligne pour stocker  
et partager ses *repositories*

Interface web pour visualiser  
et collaborer sur des *repos*

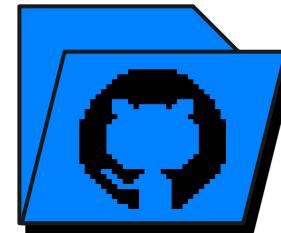
VS

# Dépôt local ou distant



**Local**

Dossier contenant le *repo*,  
situé sur votre ordinateur



**Remote**

Version en ligne du *repo*,  
à la manière d'un *drive*

# *Remote repository?*

Version du *repository* (aussi désignée par **origin**) en ligne qui permet de centraliser l'historique et les modifications. Public ou privé, plusieurs collaborateurs peuvent y contribuer, avec différents droits.



## VOS REPOSITOIRES

Page d'accueil

## RECHERCHE

Dashboard

Segolene-Albouy

Top repositories



Find a repository...

faouinti/vhs

Evarin/discover-demo

Segolene-Albouy/ChoucrouteMedievale

jnorindr/extractorAPI

Segolene-Albouy/XML-TEI\_M2TNAH

faouinti/similarity

Segolene-Albouy/Traktor-NML-to-Rekordbox-XML

Show more

Your teams

Find a team...

Chartes-TNAH/tnahbox

Chartes-TNAH/m2-2018

Type to search

Home

Send feedback

Filter

rsimon created a repository  
43 minutes ago

rsimon/diga-connectors

Annotorious Connector Plugin demo for the DiGA project.  
TypeScript

Star

ggerganov/llama.cpp released  
7 hours ago

Sponsor

b3668

2

Trending repositories · See more

Zeyi-Lin/HivisionIDPhotos

HivisionIDPhotos: a lightweight and efficient AI ID photos tools. 一个轻量级的AI证件照制作算法。

Python

6.2k

Star

lobehub/lobe-chat

Star

Latest changes

2 days ago  
Enterprise audit logs can be streamed to two endpoints [Private Beta]

2 days ago  
GitHub Actions: arm64 Linux and Windows runners are now generally available

Last week  
Add repository permissions to custom organization roles

Last week  
Unkey is now a GitHub secret scanning partner

View changelog →

Explore repositories

gethugothemes / liva-hugo

Liva is a personal blog template powered by Hugo.

274

HTML

wzhouxiff / RestoreFormer

[CVPR 2022] RestoreFormer: High-Quality Blind Face Restoration from Undegraded Key-Value Pairs

# Page de profil

Overview Repositories 37 Projects Packages Stars 83



## Ségalène Albouy

Ségalène-Albouy · she/her

Computer vision for digital humanities  
Teacher @ École des chartes Research  
@ École des Ponts

Edit profile

40 followers · 37 following



Paris

<https://discover-demo.enpc.fr>

<https://orcid.org/0009-0008-0998-978X>

@AlbouySegolene

Segolene-Albouy / README.md

- 👋 Hi, I'm @Segolene-Albouy
- 🛠 I build tools that enables researchers in social sciences to use computer vision algorithms
- 🧑‍🏫 I teach to Masters students Git, XML TEI, Machine Learning, Dataviz & Digital Humanities
- 💻 I'm interested in sauerkraut and process automation

Pinned Order updated.

Customize your pins

vayvi/HDV Public

Historical Diagram Vectorization

Jupyter Notebook ⭐ 11

XML-TEI\_M2TNAH Public

Ressources et présentations pour le cours en XML-TEI des M2 TNAH de l'École des chartes

HTML ⭐ 8 🏷 8

kanon Public

Forked from ALFA-project-erc/kanon

Python

iiif-downloader Public

Script for automatic image retrieval from IIIF manifests

Python ⭐ 1

Traktor-NML-to-Rekordbox-XML Public

Python script to convert your Traktor playlists to a Rekordbox format (with Hot Cues and Loops)

Python ⭐ 8 🏷 1

ChoucrouteMedievale Public

Simple website update for medieval party

JavaScript ⭐ 2

920 contributions in the last year

Contribution settings ▾

2024

# Page de repository

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)



GIT-M2TNAH

Public

Pin

Unwatch 1

Fork 0

Star 0



main



1 Branch



0 Tags



Go to file



t



+

Code

About



Cours de Git et bonnes pratiques de développement pour les étudiants de Master 2 de l'École des chartes

Readme

MIT license

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

## GIT-M2TNAH

Cours de Git et bonnes pratiques de développement pour les étudiants de Master 2 de l'École des chartes

L enroulages

# Danger zone



Code Issues Pull requests Actions Projects Wiki Security Insights Settings

## General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

## General



## Danger Zone

### Change repository visibility

This repository is currently public.

Change visibility

### Disable branch protection rules

Disable branch protection rules enforcement and APIs

Disable branch protection rules

### Transfer ownership

Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

### Archive this repository

Mark this repository as archived and read-only.

Archive this repository

### Delete this repository

Once you delete a repository, there is no going back. Please be certain.

Delete this repository

# Fichiers spéciaux du *repository*



Liste de  
fichiers à  
ignorer par  
git

`.gitignore`

Fichier de  
présentation  
du *repo*

`README`

Licence  
précisant les  
conditions  
d'utilisation  
du *repo*

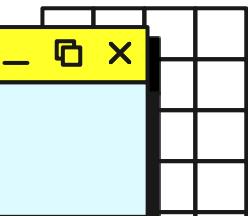
`LICENSE`





# .gitignore example

```
# Jupyter Notebook  
.ipynb_checkpoints  
  
# Environment variables  
.env  
  
# Virtual environments  
env/  
venv/  
  
# PyCharm  
.idea/
```



# README.md example

## # Installation

Step-by-step instructions on how to install the project and its dependencies.

1. Clone the repository:

```
```bash
git clone git@github.com:<username>/<repository>.git
cd <repository>
...```

```

2. Set up environment:

```
...```

```

# LICENSE

Sélection de la licence à la création du  
*repository* sur GitHub

[choosealicense.com](https://choosealicense.com)

## Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

Which of the following best describes your situation?



I need to work in a  
community.

Use the [license preferred by the  
community](#) you're contributing to or  
depending on. Your project will fit right in.

If you have a dependency that doesn't have  
a license, ask its maintainers to [add a  
license](#).



I want it simple and  
permissive.

The [MIT License](#) is short and to the point. It  
lets people do almost anything they want  
with your project, like making and  
distributing closed source versions.

[Babel](#), [.NET](#), and [Rails](#) use the MIT License.

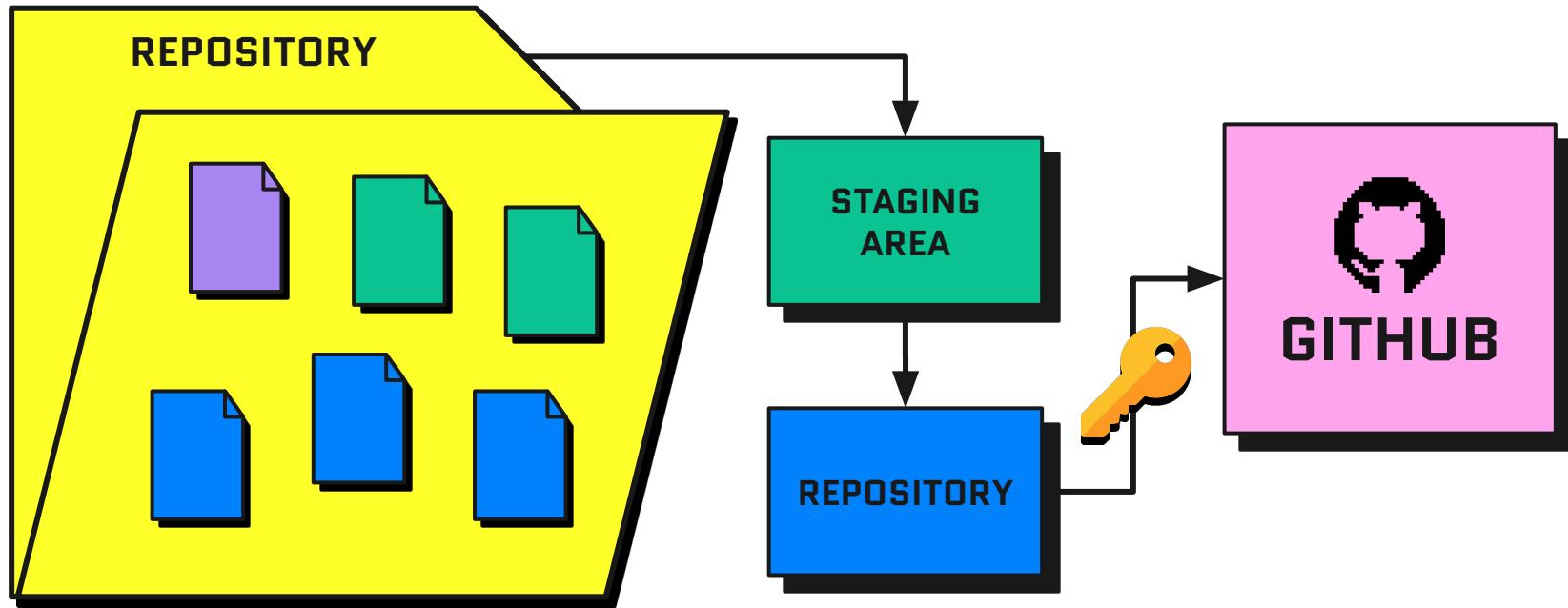


I care about sharing  
improvements.

The [GNU GPLv3](#) also lets people do almost  
anything they want with your project, except  
distributing closed source versions.

[Ansible](#), [Bash](#), and [GIMP](#) use the GNU  
GPLv3.

# Processus de publication





# Ajouter une clef SSH pour son ordinateur

1. Créer une clef SSH depuis le terminal

```
ssh-keygen -t ed25519
```

```
cat ~/.ssh/id_ed25519.pub
```

2. GitHub > Settings > SSH and GPG keys > [New](#)
3. Coller la clef publique de votre terminal
4. Nommer la clef pour désigner votre ordinateur

# Créer un repository : <https://github.com/new>

The screenshot shows the GitHub interface for creating a new repository. At the top, there's a search bar with placeholder text 'Type / to search' and a toolbar with various icons. Below that, the main form starts with a title 'Create a new repository'. It includes fields for 'Owner \*' (set to 'Segolene-Albouy'), 'Repository name \*' (empty), and a 'Description (optional)' field (empty). There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there's a checkbox for 'Add a README file' which is unchecked. A note below says 'This is where you can write a long description for your project.' There's also a section for '.gitignore' with a dropdown menu set to 'None'. The bottom part of the form includes a 'Choose a license' section with a dropdown menu set to 'None'. A large blue callout bubble on the right contains the French text: 'En pratique, on crée son repo sur GitHub qu'on duplique en local, pas l'inverse'.

Create a new repository

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

Required fields are marked with an asterisk (\*).

Owner \* Repository name \*

Segolene-Albouy /

Great repository names are short and memorable. Need inspiration? How about [scaling-fortnight](#) ?

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:

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

Add .gitignore

.gitignore template: None

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

Choose a license

License: None

NOUVEAU

# Récupérer le lien d'un remote

1

2

Toujours sélectionner le lien SSH

GIT-M2TNAH Public

main · 1 Branch · 0 Tags

Segolene-Albouy [EXAMPLE] Add documentation to functions · c3bdb90 · 2 hours ago

.bashrc  
LICENSE  
README.md

README MIT license

Go to file Add file Code

Local Codespaces Copilot

Clone

HTTPS SSH GitHub CLI

Copy url to clipboard

git@github.com:Segolene-Albouy/GIT-M2TNAH.! Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

About

Cours de Git et bonnes pratiques de développement pour les étudiants de Master 2 de l'École des chartes

Readme  
MIT license  
Activity  
0 stars  
1 watching  
0 forks

Releases

No releases published  
Create a new release

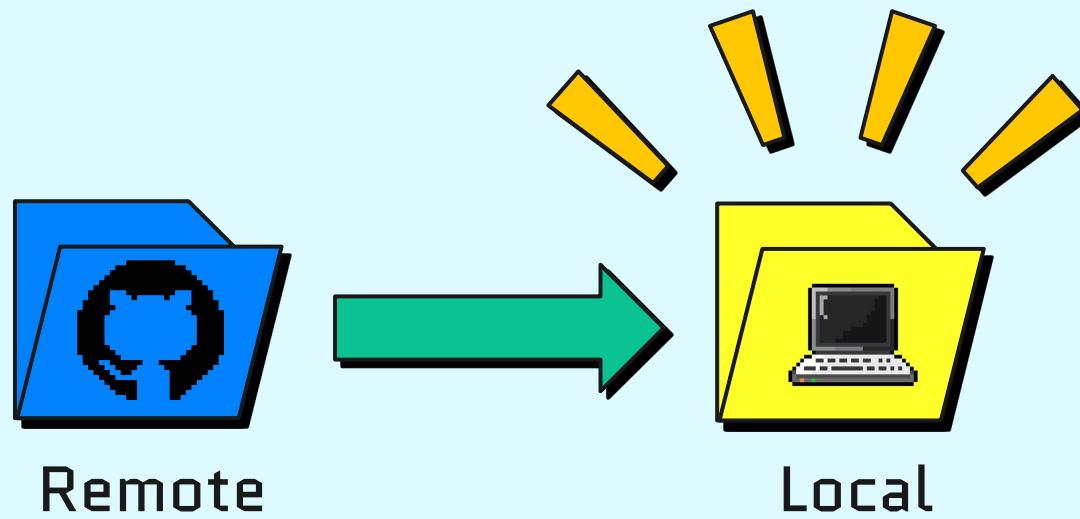
Packages

No packages published  
Publish your first package

Languages

Shell 100.0%

## COPIER UN REPOSITORY SUR GITHUB SUR SON ORDINATEUR



# Récupérer un remote en local



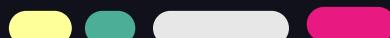
```
# Se déplacer à l'endroit où on veut créer le dossier  
cd Bureau/
```



```
# Copier un repo distant sur son ordinateur  
git clone <url-repo-distant>
```



```
# De manière générale, pour créer un nouveau projet,  
# on le crée sur GitHub, puis le clone en local
```



```
# Cloner le repository du cours
```



```
git clone git@github.com:Segolene-Albouy/GIT-M2TNAH.git
```





OSSEKOUR !



# GitHub veut pas m'autoriser à cloner le *repo* ! 😭

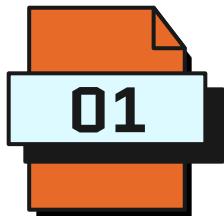
On commence par lire le message de son terminal :

```
git@github.com: Permission denied (publickey).  
fatal: Could not read from remote repository.
```

Please make sure you have the correct access  
rights and the repository exists

# Permission denied

L'erreur peut être due à :



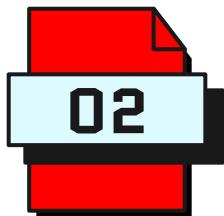
**COMPTE GIT LOCAL  
MAL CONFIGURÉ**

```
# Vérifier dans sa config locale que user & email  
# correspondent bien à ceux du compte GitHub  
git config --list
```

```
# Si ce n'est pas le cas, corriger avec  
git config --global user.name "<github-user>"  
git config --global user.email "<github-email>"  
# (! retirer les chevrons)
```

# **Permission denied**

L'erreur peut être due à :



**CLEF SSH  
DÉFECTUEUSE**

```
# Vérifier la config SSH avec GitHub
ssh -T git@github.com

# Si cela renvoie une erreur,
# recréer une paire de clefs SSH
ssh-keygen -t ed25519

# Afficher le contenu de la clef publique
cat ~/.ssh/id_ed25519.pub

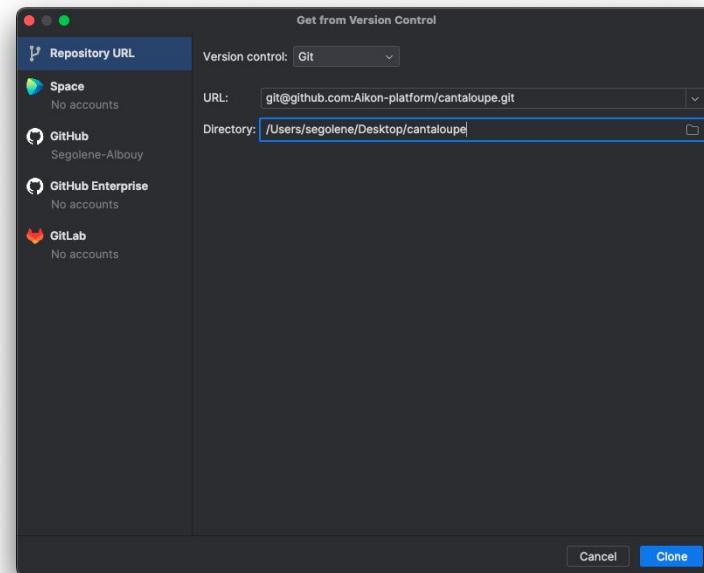
# Copier la clef publique et la coller sur GitHub
```

# Cloner un projet Git sur son IDE

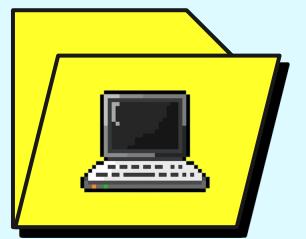
Copier l'URL SSH du repository

VSCode 🍎

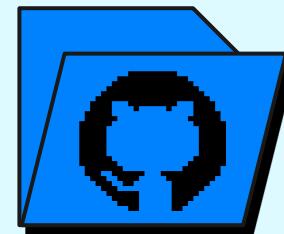
1. Ouvrir la *Command palette*  
`Ctrl+Shift+P`
2. Rechercher "Git clone"
3. Connexion à GitHub
4. Sélection du *repository*
5. Choix de l'emplacement
6. Ajout au *Workspace*



## DÉPOSER SON CODE SUR LE DÉPÔT DISTANT

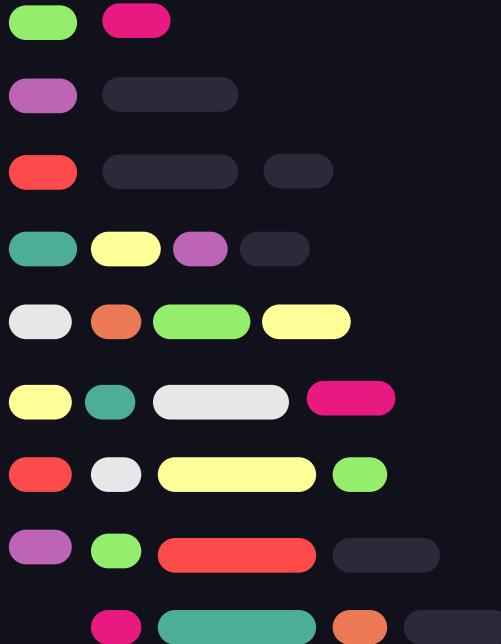


Local



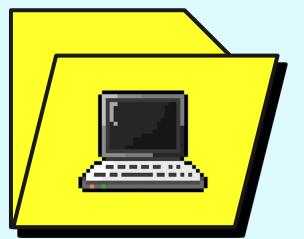
Remote

# Publier son code

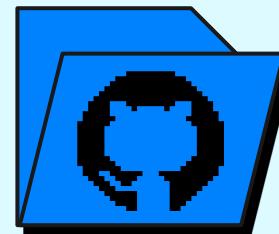


```
# déposer son code local sur GitHub  
git push
```

## RÉCUPÉRER DU CODE DEPUIS LE DÉPÔT DISTANT



Local



Remote

# Récupérer du code



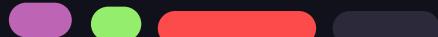
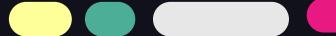
```
# récupérer du code depuis GitHub  
git pull
```



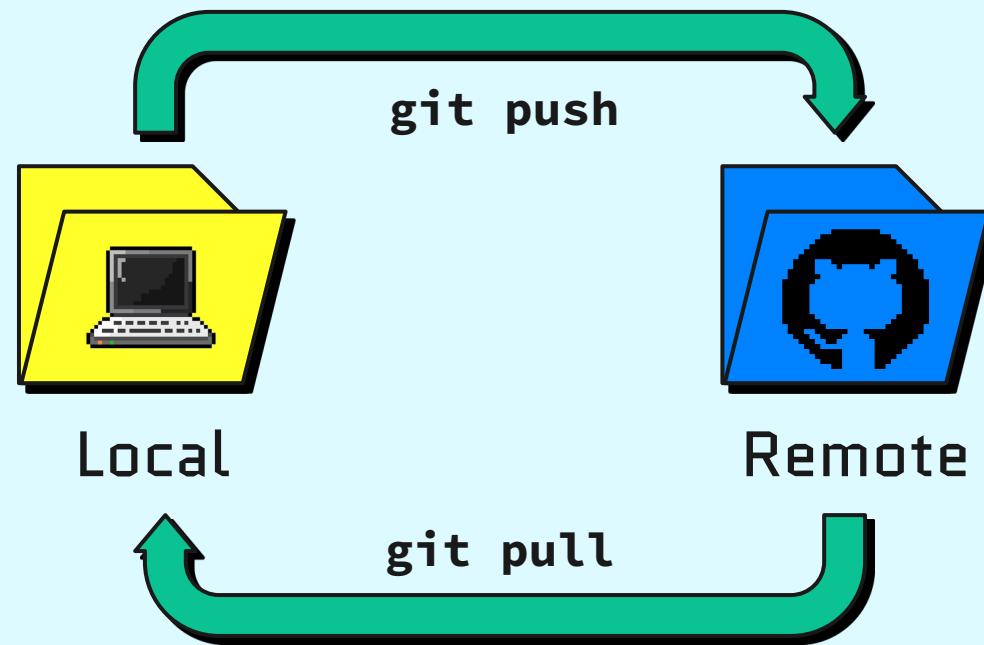
```
# git pull est une forme de fusion de branches  
# puisqu'on réunit le contenu d'une branche  
# distante avec une branche locale
```



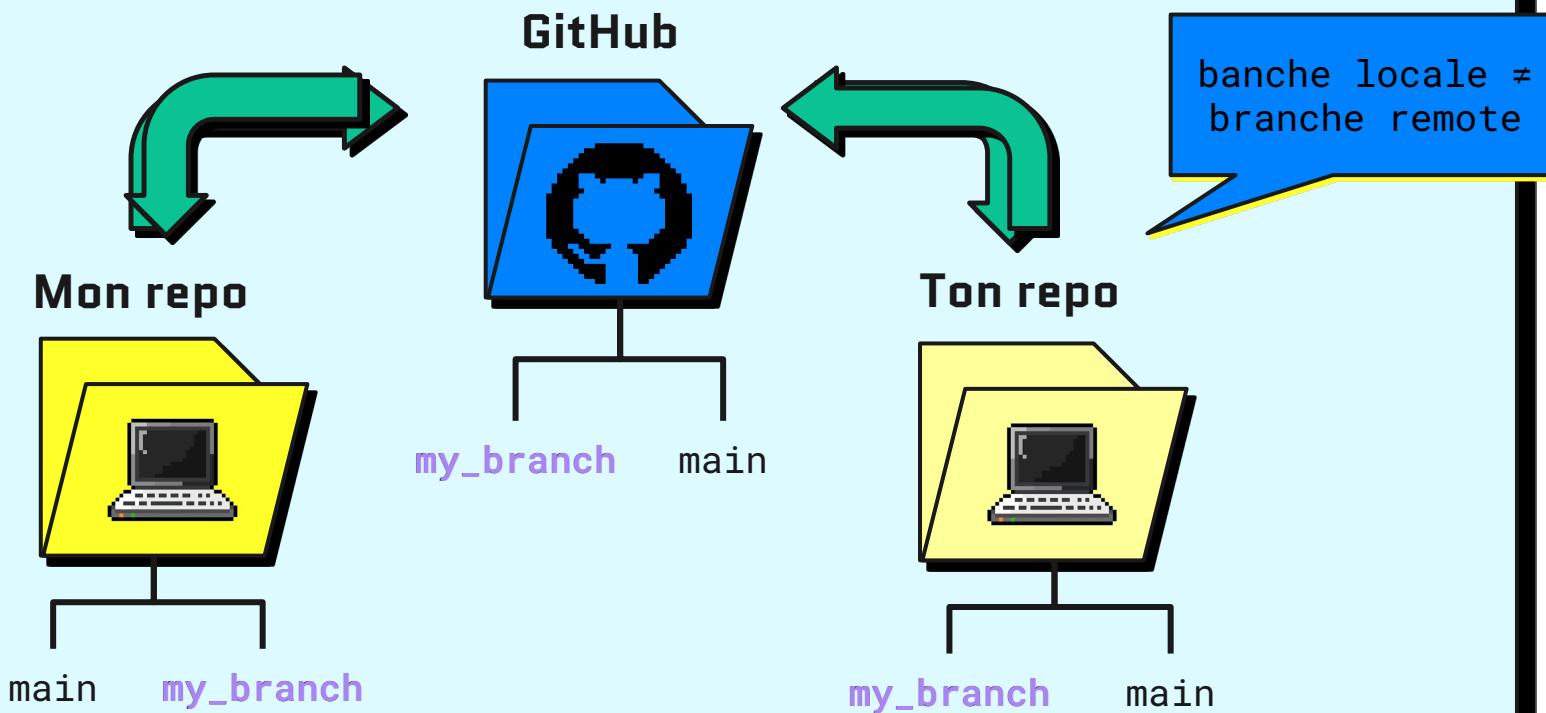
```
# récupérer du code depuis GitHub sans fusion  
git fetch
```



## SYNCHRONISATION DU REPOSITORY



## DUPLICATION DES BRANCHES



# Ajouter un collaborateur



Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Integrations

## Who has access



### Public repository

This repository is public and visible to anyone

Manage

### PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage



### DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

## Manage access



You haven't invited any collaborators yet

Add people

Les collaborateurs doivent ensuite accepter l'invitation par email

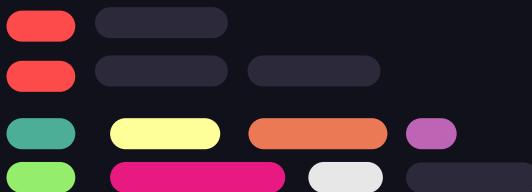


# Exercice 4



04 { *par deux*

Collaborer sur un  
*repository*





# { .. Partie 1

## Création

L'un·e crée un  
*repo* sur GitHub  
(avec README)

## Commit

Depuis VSCode,  
chacun effectue  
des commits sur  
la branche **main**



## Collab

Il ajoute l'autre  
dans les  
collaborateurs

## Push

Chacun push ses  
commits sur le  
*repo* distant

## Clone

Chacun clone le  
*repository* sur  
son ordinateur

## Quoi ?

Tenter de  
comprendre ce qui  
se passe





OSSEKOUR !

L'autre a réussi à  
pusher mais moi ça  
marche paaaas ! 😭

On commence par lire le message de son terminal :

```
To github.com:utilisateur/nom-du-repo.git
! [rejected] <local> -> <remote> (fetch first)
error: failed to push some refs to '<remote>'
...
hint: (e.g., 'git pull ...') before pushing again.
```

# Condition pour pusher

Il n'est pas possible de pusher ses modifications sans avoir auparavant intégré les commits déjà publiés sur le *remote*

Pour mettre à jour sa version du code et pouvoir pusher, il faut d'abord pull le dépôt distant

```
To <remote> ! [rejected] <b> -> <b> (fetch first)
error: failed to push some refs to '<remote>'

...
$ git pull
```



OSSEKOUR !

- □ ×

# GitHub veut pas m'autoriser à *pusher!* 😭

On commence par lire le message de son terminal :

```
remote: Support for password authentication  
was removed on August 13, 2021. Please use a  
personal access token instead.
```

```
fatal: Authentication failed for  
'https://github.com/username/repo.git/'
```

# Authentication failed

Lorsqu'on utilise le **lien HTTPS** pour cloner un *repository*, il n'est plus possible de publier son code.

Il faut alors changer le *remote* pour utiliser **lien SSH** du *repository*

```
# Modifier l'URL du remote pour le lien SSH  
git remote set-url origin git@github.com:ssh.git
```

# Configurer pull

```
warning: Pulling without specifying how to reconcile divergent branches is  
discouraged. You can squelch this message by running one of the following  
commands sometime before your first pull:
```

```
# git pull opère un merge des branches locale et remote  
git config --global pull.rebase false # ← choisir celle-là
```

```
# git pull opère un rebase des branches locale et remote  
git config --global pull.rebase true
```

```
# git pull opère un merge des branches locale et remote seulement si aucun conflit  
git config --global pull.ff only # fast-forward only
```



# { .. Partie 2

## Pull

*Pull* de la branche **main** locale avec les modifs remote

## Branche

Chacun crée une branche à partir de **main**

## Modif

Chacun modifie les mêmes lignes sur un fichier déjà existant

## Commit

Ajout et sauvegarde des modifications

## Push

Chacun *push* sa branche sur le repo distant

## Quoi ?

Tenter de comprendre ce qui se passe



# **Pusher une nouvelle branche**

Lorsqu'une nouvelle branche est créée en local, il faut configurer une branche distante sur laquelle pusher

```
$ git push  
fatal : La branche courante <branch> n'a pas de  
branche amont.
```

Pour pousser la branche courante et définir la distante comme amont, utilisez

```
git push --set-upstream origin <branch>
```



# { .. Partie 3

## Pull

Récupérer la  
branche de  
l'autre en local

## Switch

Changer de branche  
pour aller sur  
l'autre

## Merge

Merger sa branche  
dans celle de  
l'autre

## Conflit

Ouvrir le fichier  
avec conflit sur  
l'IDE

## Résolution

Modifier le  
fichier pour ne  
conserver qu'une  
version des modifs

## Fusion

Finir le merge et  
publier le code  
sur GitHub



