

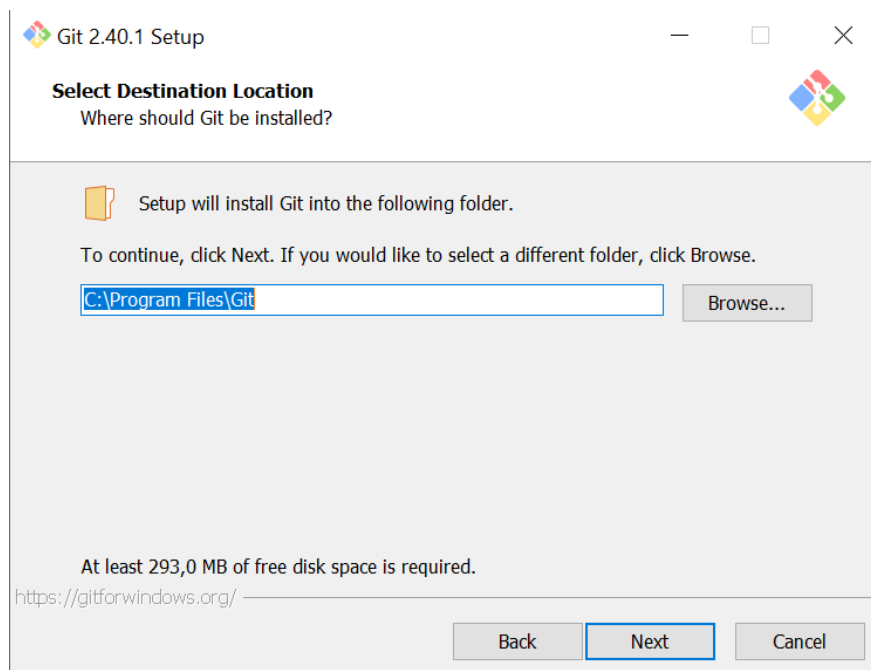
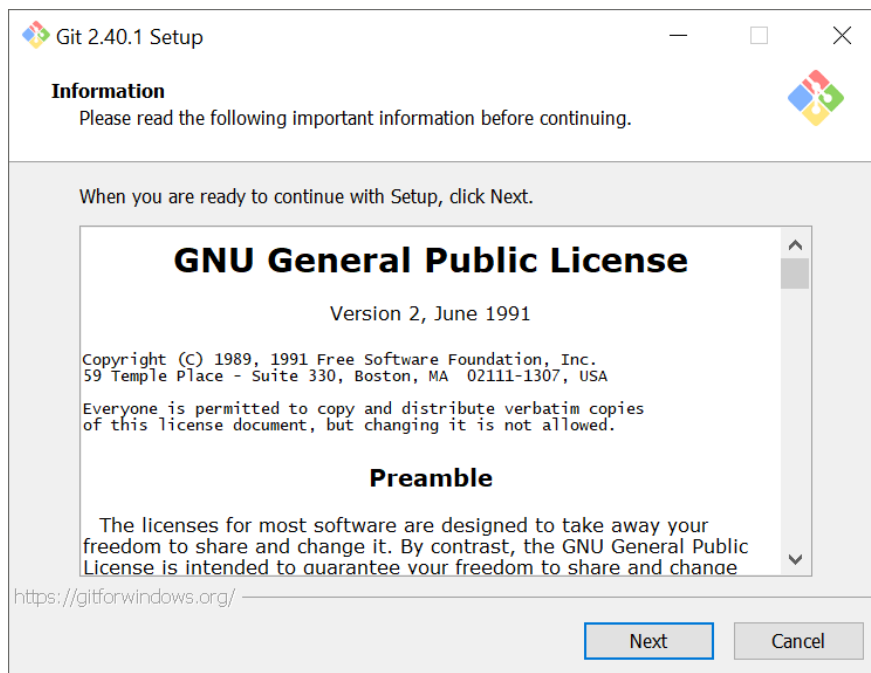
# Procédure installation et utilisation de Git

Git est un logiciel de contrôle de version open-source

Pour Télécharger Git sur Windows, aller sur ce lien :

<https://git-scm.com/download/win>

Puis sélectionnez la version x64



**Select Components**

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

<input type="checkbox"/>	Additional icons
<input type="checkbox"/>	On the Desktop
<input checked="" type="checkbox"/>	Windows Explorer integration
<input checked="" type="checkbox"/>	Git Bash Here
<input checked="" type="checkbox"/>	Git GUI Here
<input checked="" type="checkbox"/>	Git LFS (Large File Support)
<input checked="" type="checkbox"/>	Associate .git* configuration files with the default text editor
<input checked="" type="checkbox"/>	Associate .sh files to be run with Bash
<input type="checkbox"/>	Check daily for Git for Windows updates
<input type="checkbox"/>	(NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 292,9 MB of disk space.

<https://gitforwindows.org/>

Back

Next

Cancel

**Select Start Menu Folder**

Where should Setup place the program's shortcuts?



☰ Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

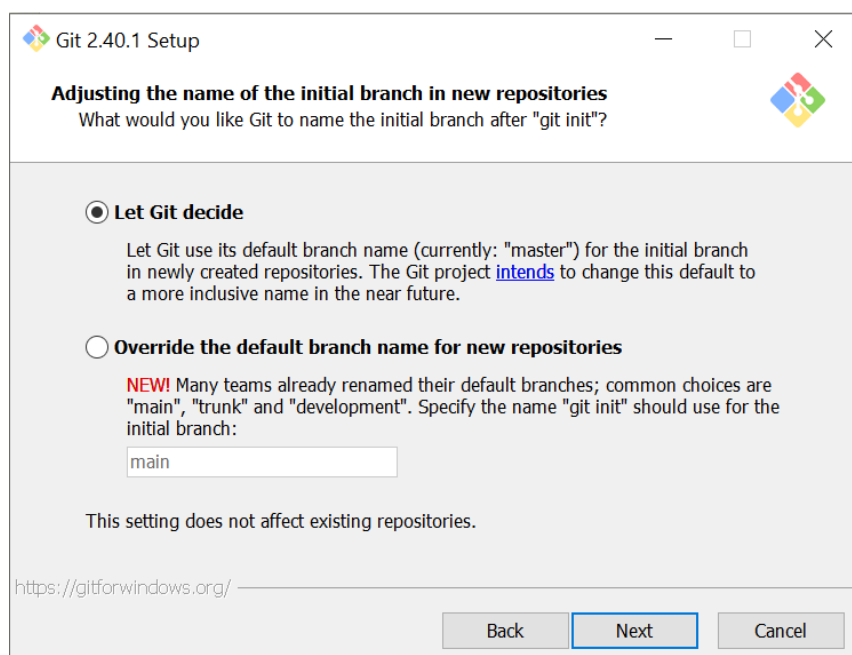
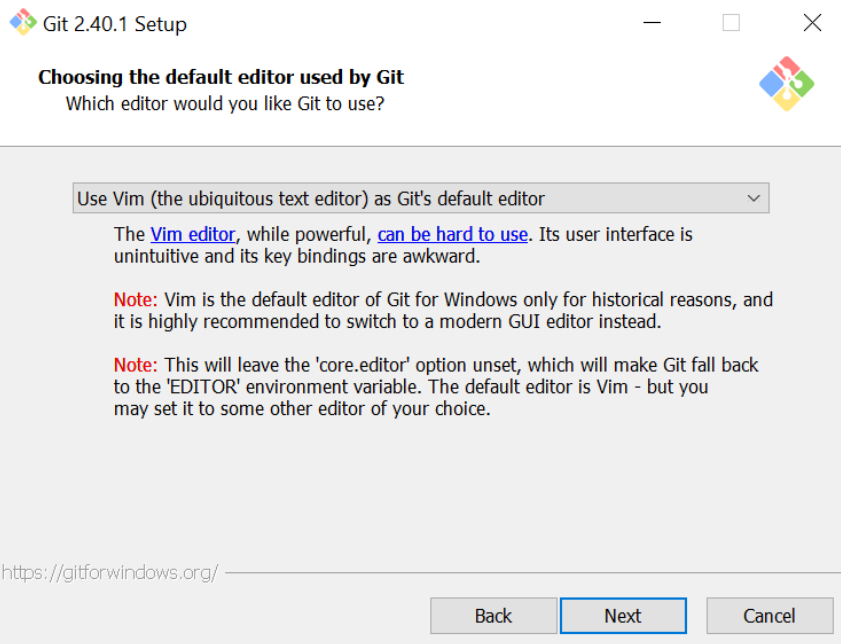
Browse...

☐ Don't create a Start Menu folder<https://gitforwindows.org/>

Back

Next

Cancel



**Adjusting your PATH environment**

How would you like to use Git from the command line?

☐ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ **Git from the command line and also from 3rd-party software**

**(Recommended)** This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

☐ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.  
**Warning:** This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

Back

Next

Cancel

**Choosing the SSH executable**

Which Secure Shell client program would you like Git to use?

☒ **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

☐ **Use external OpenSSH**

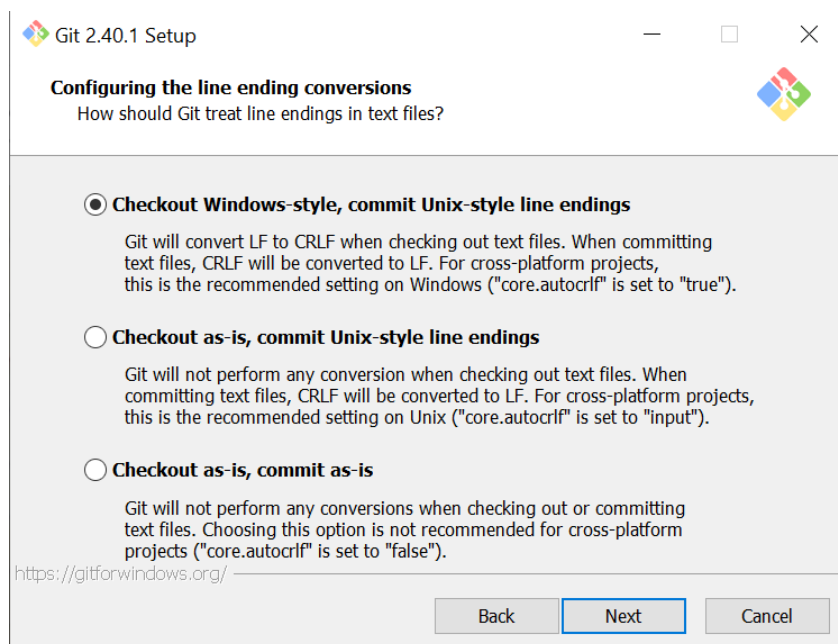
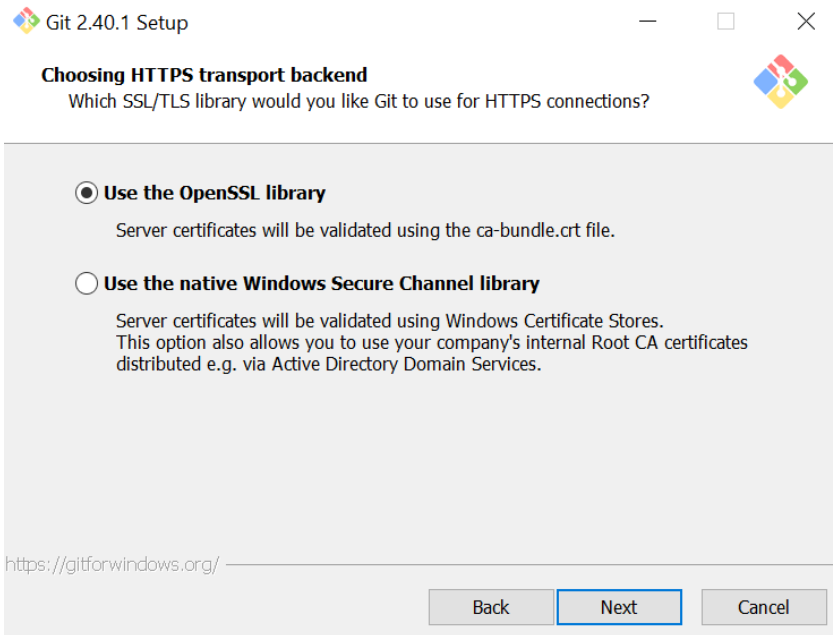
**NEW!** This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

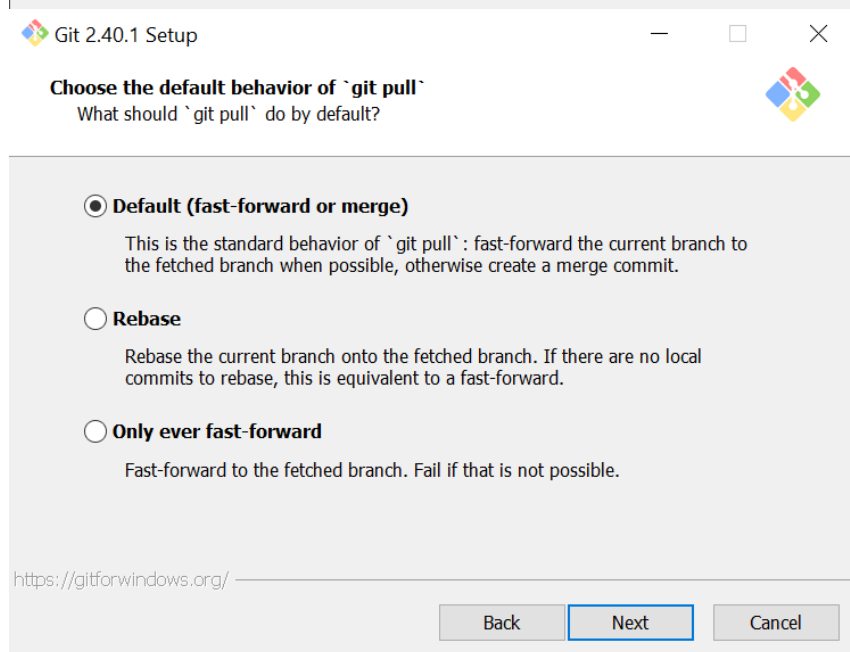
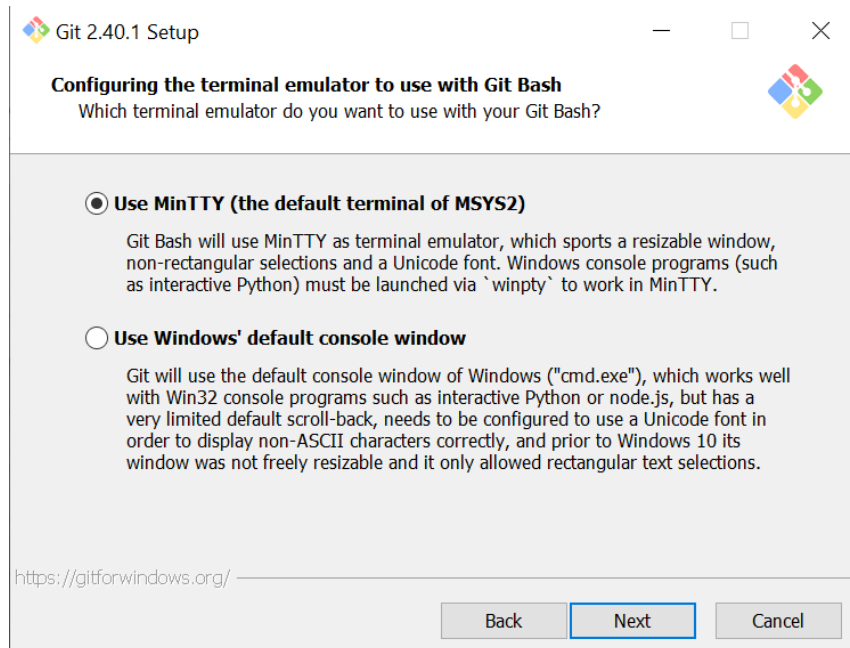
<https://gitforwindows.org/>

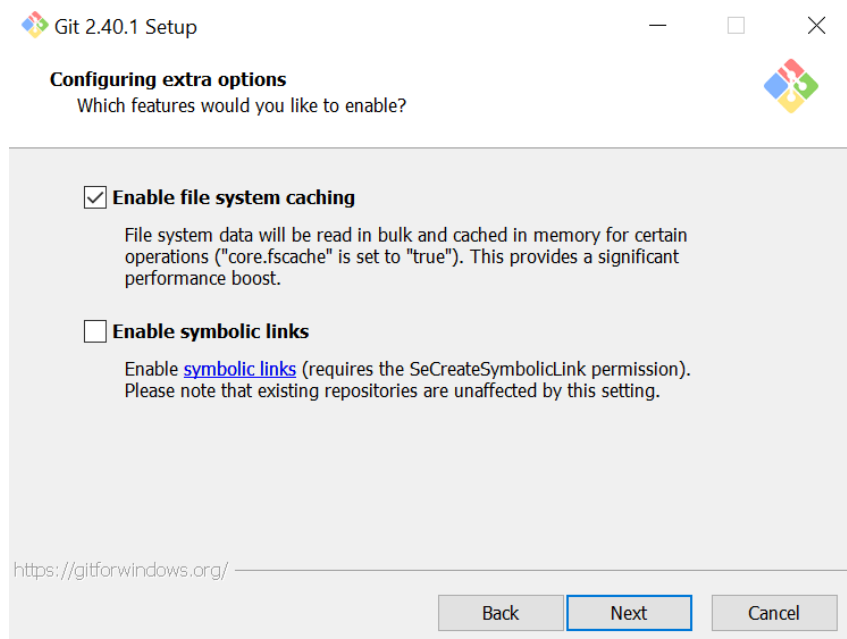
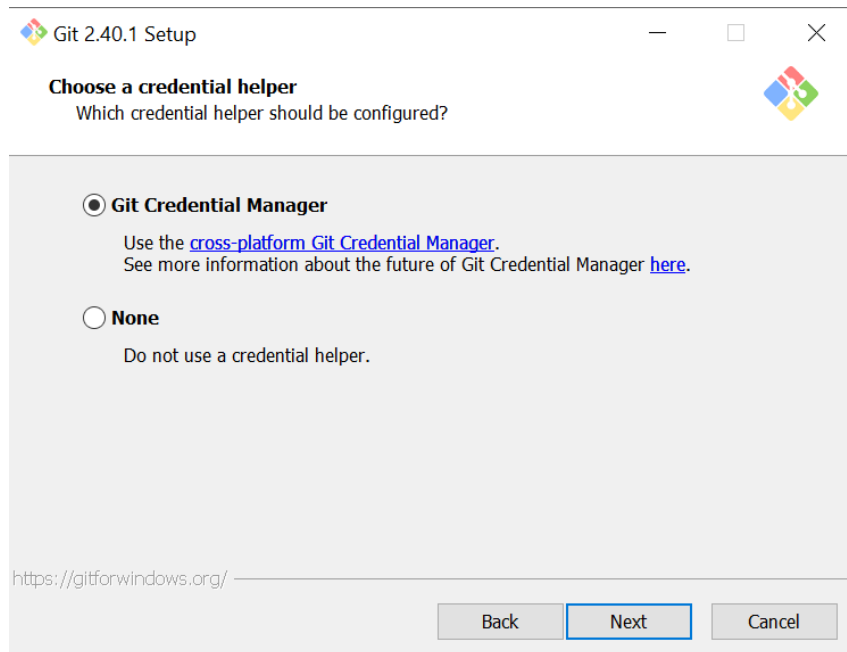
Back

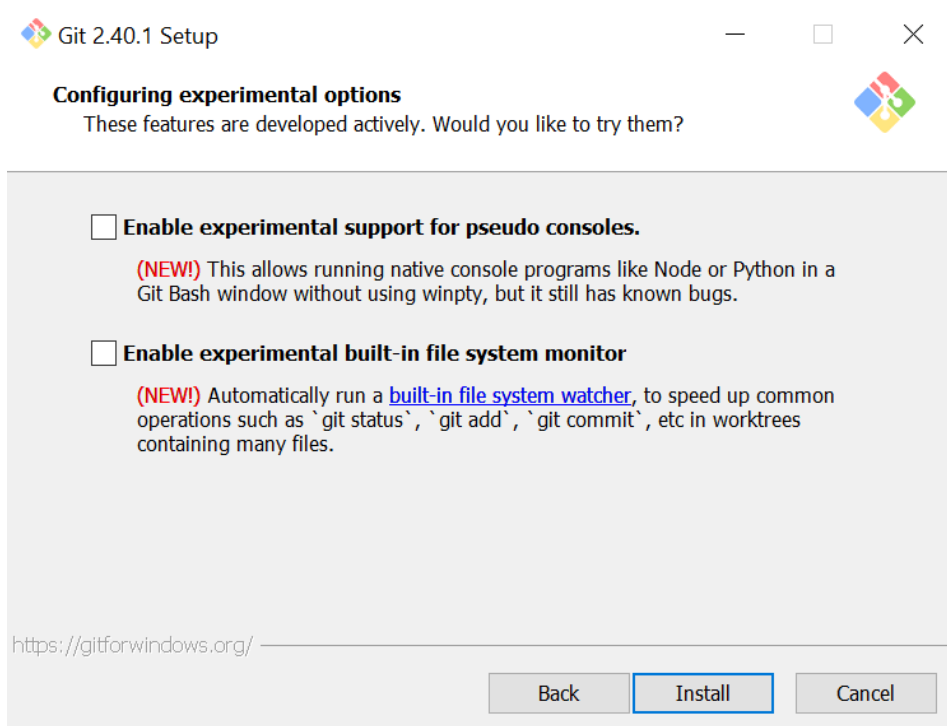
Next

Cancel

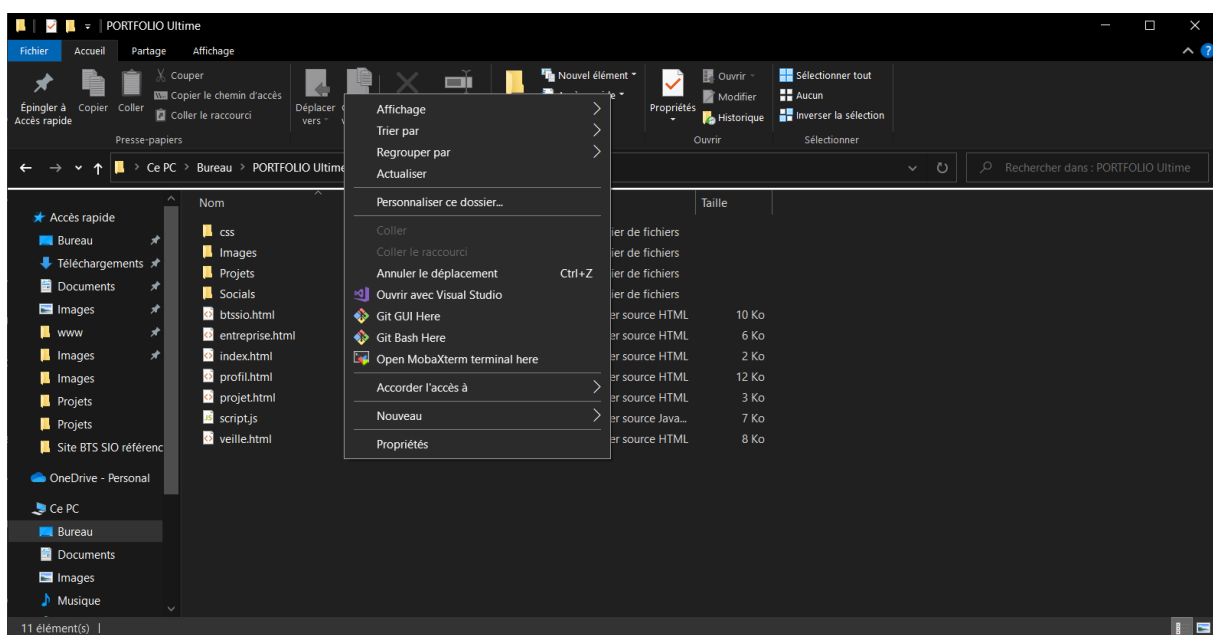






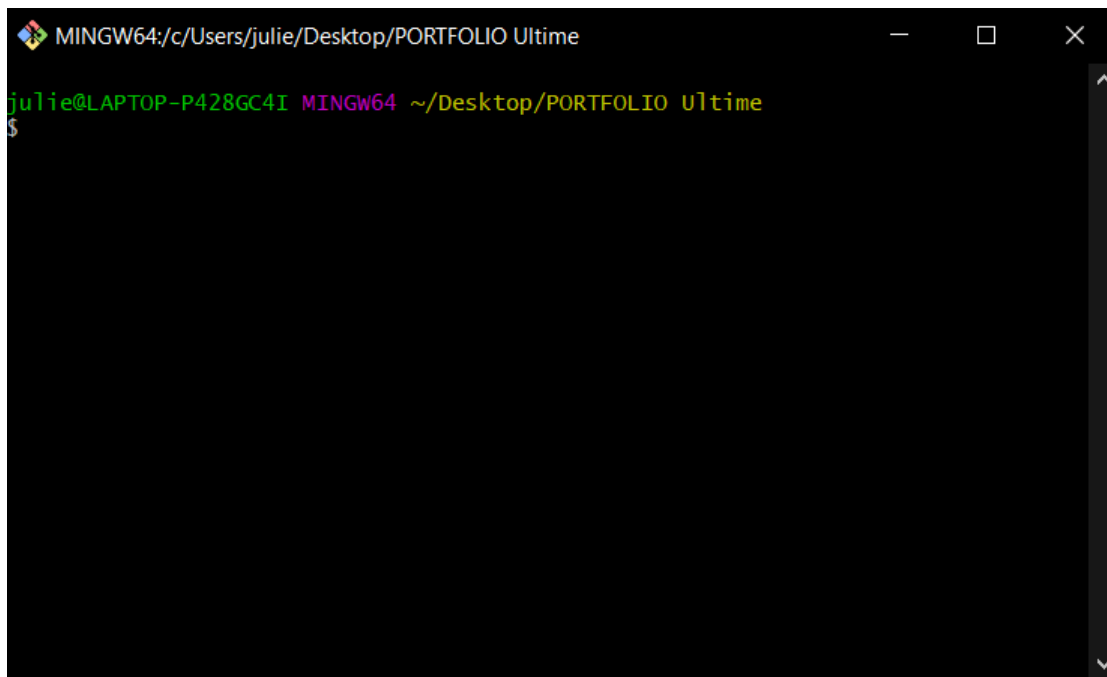


« Git bash here » à l'endroit où l'on veut tout mettre dans un repo distant sur github



Une fenêtre Git Bash s'ouvrira alors



A screenshot of a MINGW64 terminal window. The title bar at the top reads "MINGW64:/c/Users/julie/Desktop/PORTFOLIO Ultime". The terminal content shows the prompt "julie@LAPTOP-P428GC4I MINGW64 ~/Desktop/PORTFOLIO Ultime" followed by a dollar sign "\$" on the next line, indicating it is ready for input. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Quelques commandes basiques mais très utiles concernant git :

Pour cloner un dépôt Git, utilisez la commande **git clone <url>** où <url> est l'URL du dépôt que vous souhaitez cloner.

Pour ajouter des fichiers à votre zone de staging, utilisez la commande **git add <nom-du-fichier>** ou **git add .** pour ajouter tous les fichiers modifiés à votre zone de staging.

Pour créer un commit, utilisez la commande **git commit -m "message du commit"** où "message du commit" est une description courte des modifications apportées. Vous pouvez également utiliser **git commit** pour ouvrir un éditeur de texte et ajouter une description plus détaillée.

Pour récupérer les dernières modifications depuis le dépôt distant, utilisez la commande **git pull**.

Pour pousser vos modifications vers le dépôt distant, utilisez la commande **git push**.

Pour vérifier l'état de votre dépôt Git, utilisez la commande **git status**. Cette commande vous indiquera quels fichiers ont été modifiés, quels fichiers ont été ajoutés à la zone de staging, etc.

Pour afficher l'historique des commits, utilisez la commande **git log**. Cette commande vous montrera les informations sur chaque commit, y compris les auteurs, les messages de commit, les dates et les heures, les identifiants de commit, etc. Il existe de nombreuses autres commandes Git que vous pouvez utiliser avec Git Bash.

Pour en savoir plus, vous pouvez consulter la documentation Git officielle ou utiliser la commande **git --help** pour afficher une liste des commandes Git disponibles.