Installation et configuration GIT

Allez sur le <u>site de Git</u> pour choisir sur quel système d'exploitation vous souhaitez le télécharger.

Downloads
🗰 macOS 🛛 📲 Windows
👌 Linux/Unix
Older releases are available and the Git source repository is on GitHub.

Ouvrez l'exécutable qui a été installé, choisissez ce qui vous intéresse et cliquez sur 'next' jusqu'à 'install'. Si vous n'y connaissez pas grand chose, vous pouvez laisser toutes les options prédéfinies, elles conviennent bien.

Select Components Which components should be installed? Select the components should be installed? Select the components you want to install; dear the components you do not want to install. Click Next when you are ready to continue. Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git UFS (Large File Support) Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space.					
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. Additional icons On the Desktop Windows Explorer integration G it Bash Here G it GUI Here G it LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space.	📀 Git 2.39.1 Setup			×	
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue. Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git GUI Here Git GUI LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Associate .git* configuration files with the default text editor Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/	Select Components Which components should be installed?			<u>>></u>	
Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	Select the components you want to install; clear the components install. Click Next when you are ready to continue.	you do not	want to		
On the Desktop Windows Explorer integration Of the Bash Here Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space.	Additional icons			7	
Windows Explorer integration Sit Bash Here Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	On the Desktop				
Current selection requires at least 293,4 MB of disk space. Current selection requires at least 293,4 MB of disk space.	Windows Explorer integration				
Current selection requires at least 293,4 MB of disk space. Current selection requires at le	– 🗹 Git Bash Here		- 1		
Cit LFS (Large File Support) Cascolate .git* configuration files with the default text editor Associate .ght files to be run with Bash Check daily for Git for Windows updates Check daily for Git for Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	Git GUI Here		- 1		
Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	Git LFS (Large File Support)		- 1		
Associate .sh files to be run with Bash Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/	Associate .git* configuration files with the default text editor		- 1		
Check daily for Git for Windows updates (NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	Associate .sh files to be run with Bash		- 1		
(NEW!) Add a Git Bash Profile to Windows Terminal Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	Check daily for Git for Windows updates			1	
Current selection requires at least 293,4 MB of disk space. https://gitforwindows.org/ Only show new options Back Next Cancel	(NEW!) Add a Git Bash Profile to Windows Terminal				
Only show new options Back Next Cancel	Current selection requires at least 293,4 MB of disk space.				
Only show new options Back Next Cancel					
	Only show new options Back	Next	Car	icel	

Finalisez l'installation en appuyant sur 'Launch Git Bash', vous devriez tomber sur une fenêtre type CMD comme celle-ci.



Une fois ici, la première chose à faire est de configurer votre identité, pour cela vous allez entrer les commandes suivantes :

- git config –global user.name "ton nom github"
- git config –global user.email "tonMail@gitHub.com"

Si tu n'as pas de compte GitHub, je t'invite à aller en faire un, ce sera important pour la suite des procédures. Je t'invite à suivre ma fiche de procédure sur la <u>configuration de GitHub</u>.

Pour moi voilà ce que ça donne :



Tu peux vérifier que ta configuration a été prise en compte en tapant :

git config -list

user.email=nicolasdijoux@yahoo.com user.name=nicolasdijoux

Nous pouvons voir que notre configuration a été prise en compte.

Maintenant que Git a été installé et configuré, il faut créer un dépôt local pour pouvoir l'utiliser. Pour cela, il existe 2 solutions :

- Créer un dépôt local vide pour un nouveau projet.
- Cloner un dépôt distant afin de récupérer un projet publié sur GitHub.

Pour créer un dépôt local vide pour un nouveau projet, créez un dossier par exemple sur votre bureau qui contiendra tous les fichiers de votre projet. Puis y accéder depuis Git Bash avec la commande 'cd' pour 'change directory'.

Exemple avec un dossier qui s'appelle 'Projet' situé sur mon bureau.

J'utilise la commande cd pour aller vers le dossier.



Une fois dans le dossier du projet, j'effectue la commande qui permet d'initialiser un dépôt Git : 'git init'



Et voilà vous pouvez désormais versionner vos projets localement avec Git.

Cependant il est possible de travailler avec des projets distants, que ce soit les vôtres ou non. Et pour cela nous allons nous intéresser au deuxième point : Cloner un dépôt distant afin de récupérer un projet publié sur GitHub.

Pour cela, il suffit de récupérer l'URL d'un dépôt distant sur GitHub. Pour récupérer cet URL, il faut accéder à un projet sur GitHub puis cliquer sur Code et enfin copier le lien HTTPS.



Une fois l'URL copiée, retournez sur Git Bash, allez dans le dossier de votre choix qui contiendra le projet que vous voulez prendre, puis écrivez :

git clone https://github.com/url-du-depot-distant.git/

nicol@PC-Nico MINGW64 ~/Desktop/Projet (master)
<pre>\$ git clone https://github.com/nicolasdijoux/mon-site-vitrine.git</pre>
Cloning into 'mon-site-vitrine'
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 80 (delta 16), reused 74 (delta 13), pack-reused 0
Receiving objects: 100% (80/80), 359.38 KiB 2.10 MiB/s, done.
Resolving deltas: 100% (16/16), done.

Et voilà ! Le dépôt distant à bien été copié dans mon dossier Projet.

Bureau > Projet	> mon-site-vitrine
Nom	^
.git	
assets	
vendor	
区 index.html	

Spécificité des dépôts distants :

On utilisera 'pull' pour récupérer les modifications du dépôt distant et 'push' pour envoyer nos modifications au dépôt distant

Vous savez désormais comment installer et configurer Git pour l'utiliser localement ou avec GitHub et versionner des projets localement ou à distance.

Bonus :

Il est possible d'utiliser Git & GitHub directement inclus dans l'éditeur de code Visual Studio Code, ce qui permet un gain de temps considérable. Pour cela je t'invite à suivre mon <u>tutoriel sur l'utilisation des dépôts distants avec VSCode</u>.