

 	Créer un repository avec GIT et GITHUB sous Windows	Réalisé le :	12/11/2024
		Modifié le :	30/11/2024

Nombre de pages :
13

Créer un repository avec GIT et GITHUB sous Windows

1. PREREQUIS	1
2. CREER UN REPOSITORY EN 8 ETAPES	1
2.1. ETAPE 1 → CREER UN NOUVEAU REPOSITORY SUR GITHUB	1
2.1.1. Le fichier README	2
2.1.2. Le dossier .gitignore	3
2.2. ETAPE 2 → CREER UN NOUVEAU DOSSIER VIDE EN LOCAL ET L'INITIALISER	6
2.3. ETAPE 3 → GENERER DES FICHIERS DANS LE DEPOT LOCAL	7
2.4. ETAPE 4 → RENOMMER LA BRANCHE MASTER EN MAIN	8
2.5. ETAPE 5 → INDEXER LES FICHIERS	9
2.6. ETAPE 6 → CAPTURER UN INSTANTANE DES CHANGEMENTS ACTUELLEMENT INDEXES	10
2.7. ETAPE 7 → CREER LE LIEN POUR RELIER LE DEPOT LOCAL AU DEPOT DISTANT SITUE SUR GITHUB	11
2.8. ETAPE 8 → ENVOYER LE COMMIT REALISE EN LOCAL SUR LE DEPOT DISTANT SITUE SUR GITHUB	11
3. RESUME SCHEMATIQUE	12

1. Prérequis

Créer un repository avec Git et GitHub est une étape essentielle, dans le processus de développement, pour gérer et partager le code source de ses projets.

Voici un guide étape par étape qui indique les points essentiels.

Avant de pouvoir créer un repository sur GitHub, il faut satisfaire à certaines conditions.

- Créez un compte GitHub si ce n'est pas déjà fait. Inscrivez-vous sur [GitHub](https://github.com)¹.
- Réaliser l'installation de Git sur votre machine pour cela il faut :
 - Disposer des **droits administrateur** sur votre machine.
 - [Téléchargez](https://git-scm.com/downloads)² Git.
- **Configurer Git** sur votre machine Windows

Après avoir remplis ces prérequis , on peut rentrer dans le vif du sujet : créer un dépôt sur GitHub pour y déposer son code source.

2. Créer un repository en 8 étapes

2.1. Etape 1 → Créer un nouveau repository sur GitHub

Aller sur github.com et la première opération consiste à créer un nouveau repository qui contiendra le code du projet.

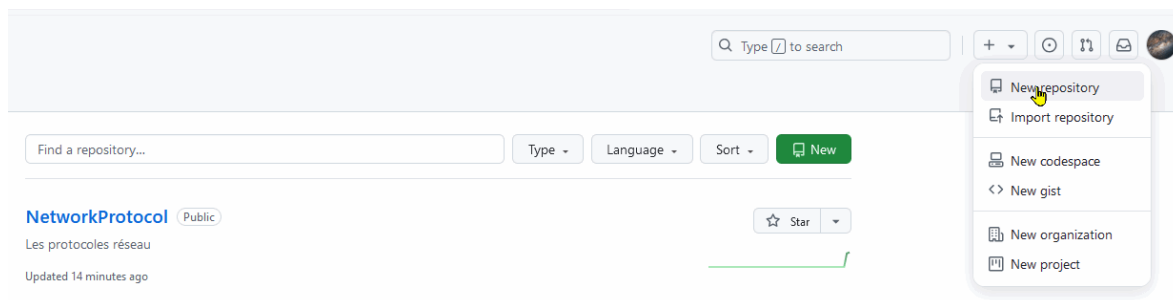
¹ <https://github.com>

² <https://git-scm.com/downloads>

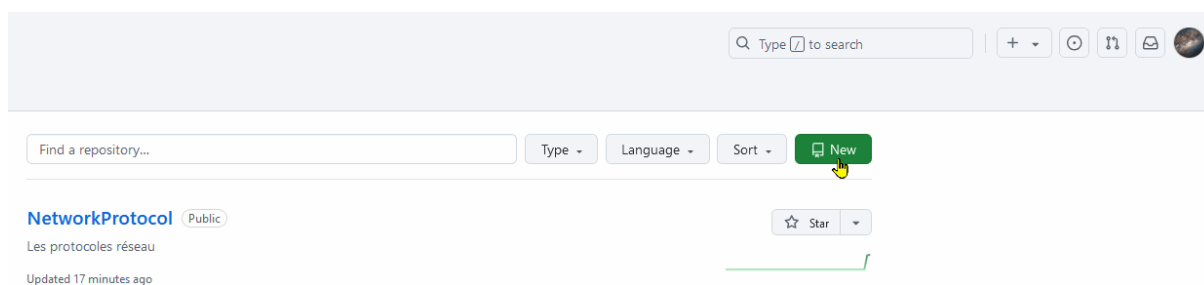
	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

Sur la plateforme GitHub, on dispose de deux manières pour créer un nouveau dépôt :

La première :



La seconde :



Ce nouveau dépôt s'appellera : mySite

Il sera Public.

Pas de fichier README pour l'instant ; il sera ajouté plus tard.

⚠ Ne pas ajouter de fichier README ni de .gitignore ! Pour éviter les risques de conflits lors des synchronisations.

Voici quelques explications sur ces deux éléments :

2.1.1. Le fichier README

Dans la sphère GitHub, le fichier README est comme une porte d'entrée vers le projet.

En effet, c'est un document écrit en [Markdown](https://fr.wikipedia.org/wiki/Markdown)³, qui se trouve à la racine du dépôt et qui fournit une description concise et claire du projet.

Le nom "README" vient de l'anglais "read me" (lis-moi), ce qui résume bien son but : être lu en premier par toute personne qui découvre le projet.

Le fichier README remplit plusieurs fonctions essentielles :

³ <https://fr.wikipedia.org/wiki/Markdown>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

- **Première impression** : C'est souvent la première chose que les gens voient lorsqu'ils visitent votre dépôt. Un README bien écrit donne une bonne impression et incite les autres à s'intéresser à votre projet.
- **Documentation essentielle** : Il fournit une documentation de base indispensable pour comprendre et utiliser votre projet.
- **Collaboration** : Il facilite la collaboration en permettant à d'autres développeurs de comprendre rapidement votre code et de contribuer au projet.
- **Référencement⁴** : Les moteurs de recherche peuvent indexer le contenu de votre README, ce qui peut aider de nouvelles personnes à découvrir votre projet.

2.1.2. Le dossier .gitignore

Le fichier .gitignore est un fichier texte **placé à la racine d'un dépôt Git**. Il sert à indiquer à Git quels fichiers ou dossiers il doit **ignorer** lors de l'ajout (add) ou du suivi (commit) de modifications.

En d'autres termes, les fichiers listés dans ce fichier ne seront pas inclus dans votre historique de version.

Pourquoi utiliser un fichier .gitignore?

- **Confidentialité** : Exclure les fichiers contenant des informations sensibles, comme des mots de passe ou des clés d'API.
- **Optimisation de l'espace** : Ignorer les fichiers volumineux ou générés automatiquement qui ne doivent pas être versionnés (comme les fichiers de compilation, les logs, etc.).
- **Propreté du dépôt** : Maintenir le dépôt net en n'incluant que le code source et les fichiers nécessaires au projet.

Le fichier .gitignore utilise une syntaxe simple:

- Les lignes commençant par un # sont des commentaires.
- Les lignes vides : Ignorées par Git.

Exemple de fichier .gitignore :

⁴ Le SEO



Créer un repository avec GIT et GITHUB sous Windows

Réalisé le :

12/11/2024

Nombre de pages :
13

Modifié le :

30/11/2024

```
# Ignore les fichiers de configuration de l'éditeur
.vscode/
.idea/
*.swp

# Ignore les fichiers de compilation
*.class
*.o
*.out

# Ignore les fichiers de logs
*.log

# Ignore les fichiers spécifiques au système d'exploitation
.DS_Store
Thumbs.db
```


  Nombre de pages : 13	Créer un repository avec GIT et GITHUB sous Windows	Réalisé le :	12/11/2024
		Modifié le :	30/11/2024

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 *

 Djamelikra

/

Repository name *



mySite

 mySite is available.

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

Description (optional)

Site exemple

- ☒  **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

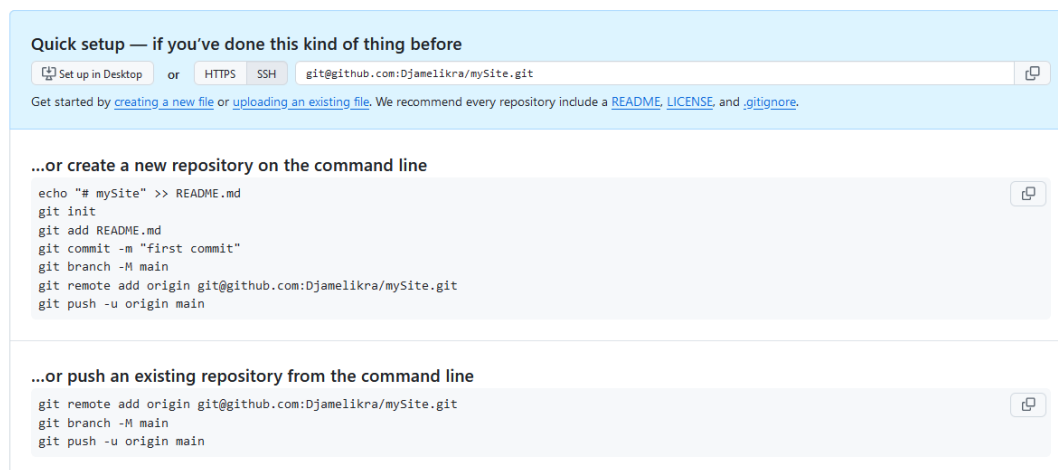
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

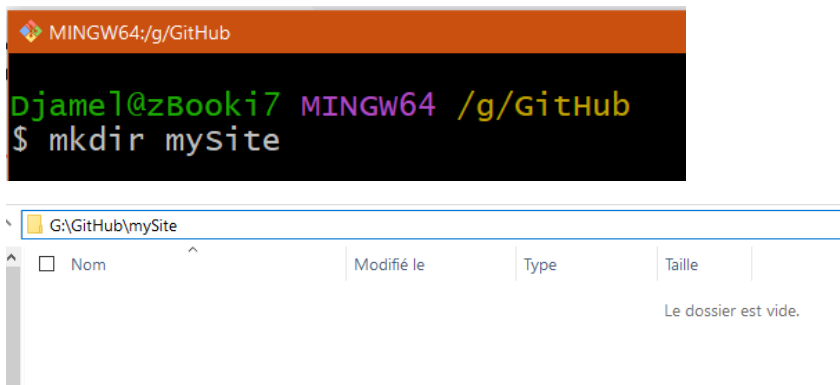
	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

⚠ Pour l'instant on n'utilise pas les commandes proposées par github !

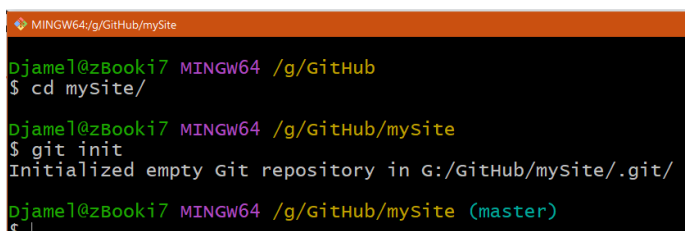


2.2. Etape 2 → Créer un nouveau dossier vide en local et l'initialiser

Il est conseillé de mettre le même nom pour le dossier en local et celui distant. Mais ce n'est pas obligatoire.



Se déplacer dans le répertoire de travail et l'initialiser avec la commande : **git init**




⚠ Il faut vérifier qu'on est bien placé dans le bon dossier avant de faire un `git init`








Une fois le projet initialisé grâce à la commande `git init`, un nouveau dossier caché `.git` apparaît dans le dossier du projet :

	Créer un repository avec GIT et GITHUB sous Windows	Réalisé le :	12/11/2024
		Modifié le :	30/11/2024

Nombre de pages :
13

G:\GitHub\mySite			
<input type="checkbox"/> Nom	Modifié le	Type	Taille
 .git	23/11/2024 16:25	Dossier de fichiers	





📁 Ce dossier caché **.git** contient toutes les informations nécessaires du projet comme le contrôle de version et toutes les informations sur les commits, l'adresse de dépôt distant, etc. Tous sont présents dans ce dossier. Il contient également un journal qui stocke l'historique, les "logs", les branches...

G:\GitHub\mySite\.git			
<input type="checkbox"/> Nom			
 hooks			
 info			
 objects			
 refs			
 config			
 description			
 HEAD			

2.3. Etape 3 → Générer des fichiers dans le dépôt local

On va ouvrir [VS Code](https://code.visualstudio.com/)⁵ pour générer trois fichiers :

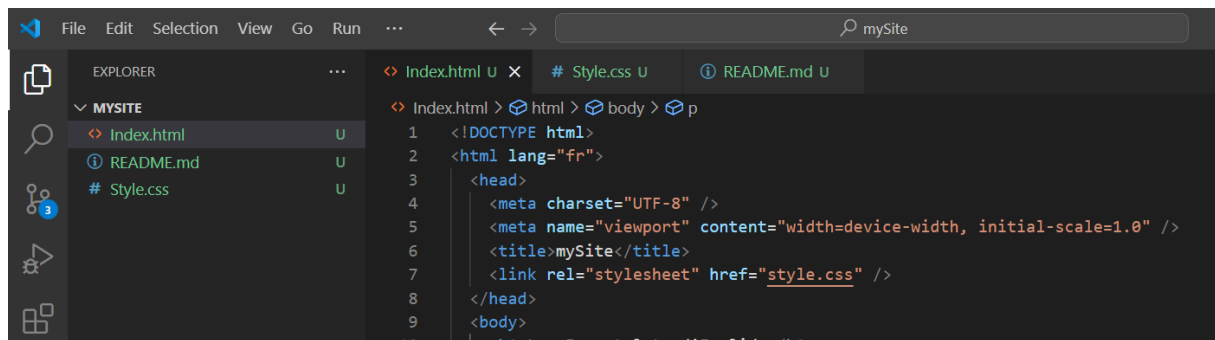
1. README qui porte une extension .md pour : Markdown⁶ : qui sert à présenter le projet.
2. Index.html : contenu du site.
3. Style.css : fichier de style qui sera relié au fichier précédent.

G:\GitHub\mySite			
<input type="checkbox"/> Nom			
 .git			
 Index.html			
 README.md			
 Style.css			

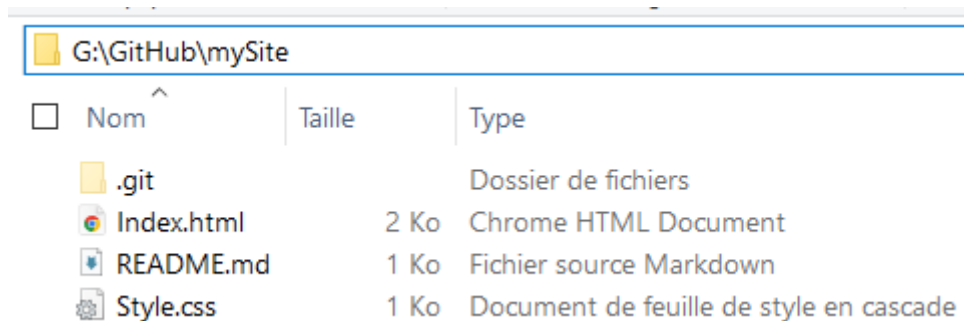
⁵ <https://code.visualstudio.com/>

⁶ <https://fr.wikipedia.org/wiki/Markdown>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

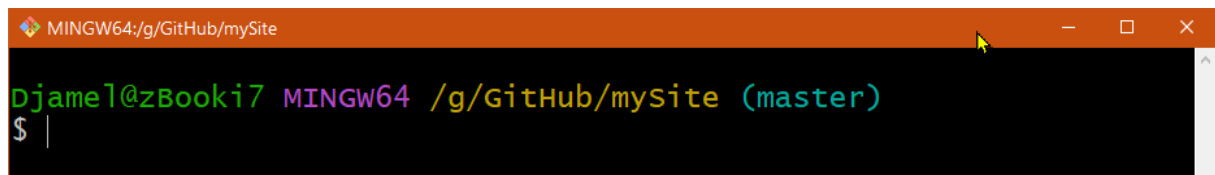


Dossier contenant les 3 fichiers :



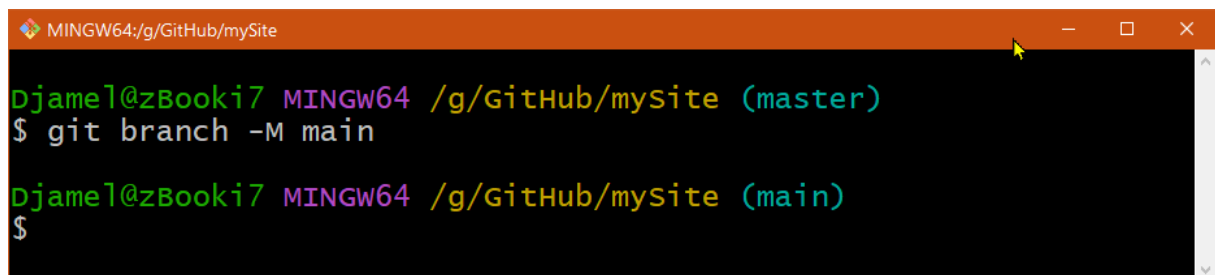
2.4. Etape 4 → Renommer la branche master en main

Dès le premier octobre 2020⁷, tous les nouveaux dépôts que vous créerez utiliseront `main` plutôt que `master` pour désigner la branche par défaut », annonce GitHub. La mesure cible également les dépôts existants qui s'appuient sur le terme `master` pour désigner la branche principale.



Pour renommer la branche **master** en **main**, il faut taper la commande :

```
git branch -M main
```



⁷ <https://github.com/github/renaming>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

2.5. Etape 5 → Indexer les fichiers

On va vérifier le contenu du dossier de travail avec la commande : **git status** :

```

MINGW64:~/g/GitHub/mySite
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Index.html
        README.md
        style.css

nothing added to commit but untracked files present (use "git add
" to track)
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ |

```

Nous avons les 3 fichiers qui sont « untracked » : donc non encore indexés ; pour y remédier il faut taper la commande : **git add** . pour indexer⁸ les 3 fichiers à la fois.

```

MINGW64:~/g/GitHub/mySite
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Index.html
        README.md
        style.css

nothing added to commit but untracked files present (use "git add
" to track)
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git add .

Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$

```

⁸ Ou bien : `git add -A`

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

Si on fait un **git status**, on voit que les fichiers passent en vert ce qui confirme leur indexation :

```

MINGW64:~/g/GitHub/mySite
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Index.html
  README.md
  style.css

nothing added to commit but untracked files present (use "git add
" to track)

Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git add .

Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   Index.html
  new file:   README.md
  new file:   style.css

Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$

```

2.6. Etape 6 → Capturer un instantané des changements actuellement indexés

En fait il s'agit de **générer une nouvelle version des fichiers** avec la commande : **git commit**

Dont la syntaxe est :

```
git commit -m "Message descriptif et concis"
```

Avec cette syntaxe l'éditeur de texte ne s'ouvre pas dans le terminal et on intègre le texte du commit directement dans la commande.

```

MINGW64:~/g/GitHub/mySite
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git commit -m "Ajout des fichiers html css et markdown"
[main (root-commit) c8bc398] Ajout des fichiers html css et markdown
3 files changed, 64 insertions(+)
create mode 100644 Index.html
create mode 100644 README.md
create mode 100644 style.css

Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$

```

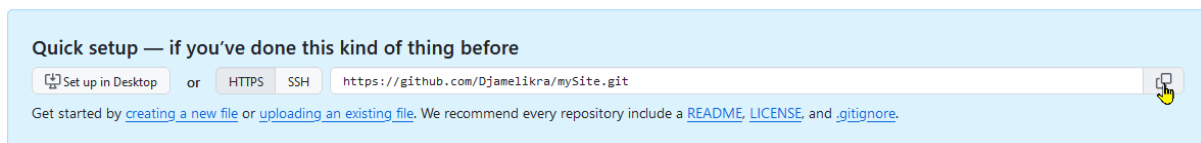
	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

2.7. Etape 7 → Créer le lien pour relier le dépôt local au dépôt distant situé sur GitHub
A ce niveau il y a deux types de connexion avec des protocoles différents : https et SSH.

La connexion en https est utilisée par défaut, sa syntaxe est :

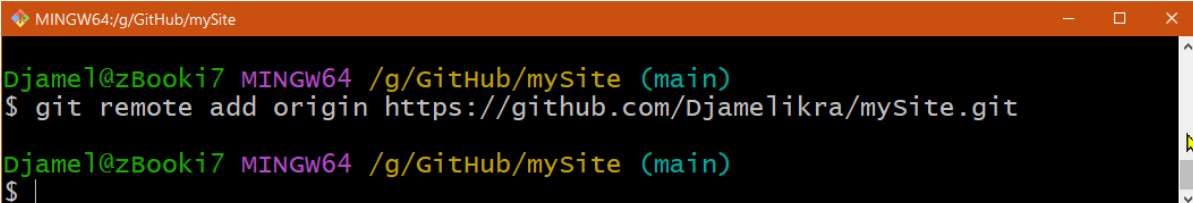
```
git remote add origin <url>
```

Pour avoir l'url il faut copier le lien (ici en https) distant du dépôt GitHub



et intégrer l'url dans la commande :

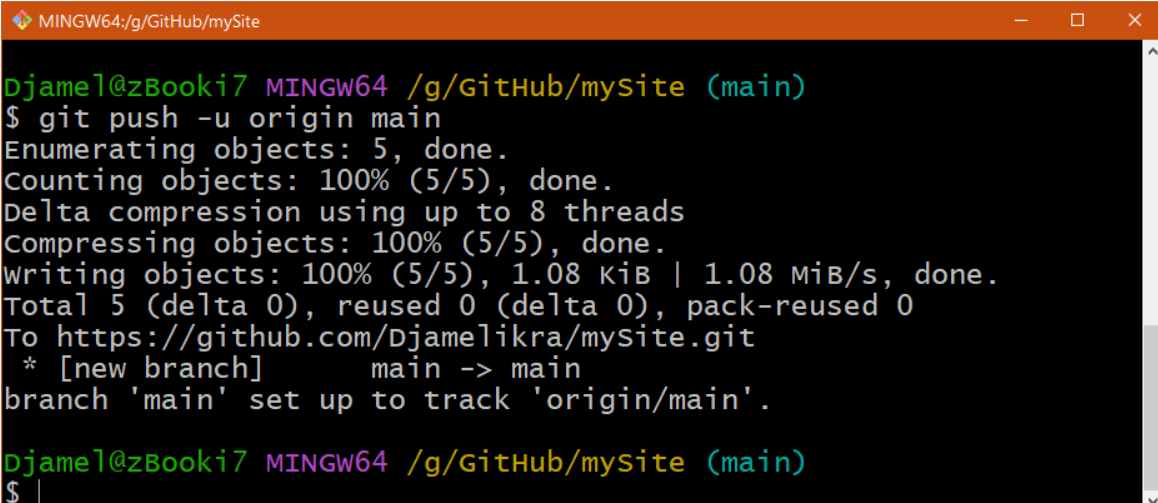
```
git remote add origin https://github.com/Djamelikra/mySite.git
```



2.8. Etape 8 → Envoyer le commit réalisé en local sur le dépôt distant situé sur GitHub

Pour cela on utilise la commande **git push**, dont la syntaxe est la suivante :

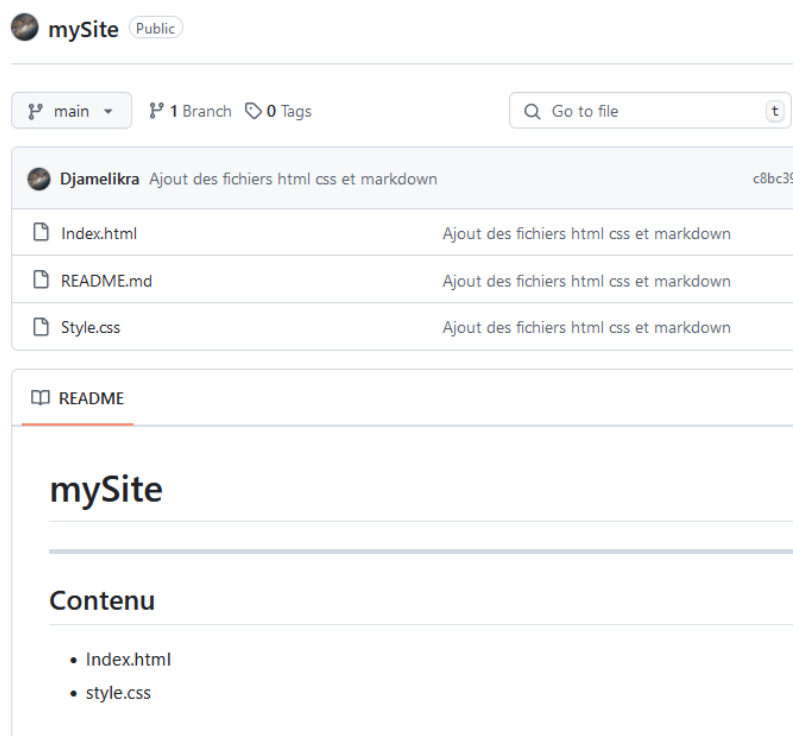
```
git push -u origin main
```



	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

"-u" remplace le paramètre "--set-upstream" lors de l'appel à la commande git push, sans aucun changement entre les deux paramètres⁹.

Il ne reste plus qu'à vérifier que tout s'est bien passé en allant sur le dépôt distant ; et en rafraichissant la page on retrouve bien nos 3 fichiers qui ont bien été transférés :



On retrouve, également, le texte du commit.

3. Résumé schématique

Git s'organise sur plusieurs niveaux. Pour faire simple, on peut découper ce fonctionnement en deux niveaux : du code stocké en local, sur votre machine, et en remote, sur un serveur¹⁰.

⁹ <https://code-garage.fr/blog/a-quoi-sert-le-parametre-u-lors-d-un-git-push/>

¹⁰ <https://blog.freelancerepublik.com/git-commandes-indispensables-developpeurs/>

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 13		Modifié le :	30/11/2024

