

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

# Créer un repository avec GIT et GITHUB sous Windows

<b>1. PREREQUIS</b>	<b>1</b>
<b>2. CREER UN REPOSITORY : 9 ETAPES</b>	<b>2</b>
2.1. ETAPE 1 → CREER UN NOUVEAU REPOSITORY SUR GITHUB	2
2.2. ETAPE 2 → CREER UN NOUVEAU DOSSIER VIDE EN LOCAL ET L'INITIALISER	4
2.3. ETAPE 3 → GENERER DES FICHIERS DANS LE DEPOT LOCAL	5
2.4. ETAPE 4 → RENOMMER LA BRANCHE MASTER EN MAIN	6
2.5. ETAPE 5 → INDEXER LES FICHIERS	7
2.6. ETAPE 6 → CAPTURER UN INSTANTANE DES CHANGEMENTS ACTUELLEMENT INDEXES	8
2.7. ETAPE 7 → CREER LE LIEN POUR RELIER LE DEPOT LOCAL AU DEPOT DISTANT SITUE SUR GITHUB	9
2.8. ETAPE 8 → ENVOYER LE COMMIT REALISE EN LOCAL SUR LE DEPOT DISTANT SITUE SUR GITHUB	9
2.8.1. Connexion en SSH : to do	11
2.9. ETAPE 6 → CREER ET AJOUTER DES FICHIERS DANS LE PROJET EN LOCAL	11
2.10. ETAPE 7 → « STAGER » LES CHANGEMENTS APPORTES AU PROJET : GIT ADD	11
2.11. ETAPE 8 → CAPTURER UN INSTANTANE DES CHANGEMENTS ACTUELLEMENT STAGES DU PROJET : GIT COMMIT	11
2.11.1. Avec git commit	11
2.11.2. Avec git commit -m	12
2.12. ETAPE 9 → ENVOYER LE CONTENU DU DEPOT LOCAL DANS LE REPOSITORY DISTANT SITUE SUR GITHUB : GIT PUSH	12
2.12.1. Si erreur	13
2.12.2. Comment éviter l'erreur à l'avenir	14
<b>3. LIENS</b>	<b>14</b>
3.1. DEPOT GITHUB DU PROJET :	14

## 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)<sup>1</sup>.
- 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)<sup>2</sup> Git.
- **Configurer Git** sur votre machine Windows

<sup>1</sup> <https://github.com>

<sup>2</sup> <https://git-scm.com/downloads>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

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

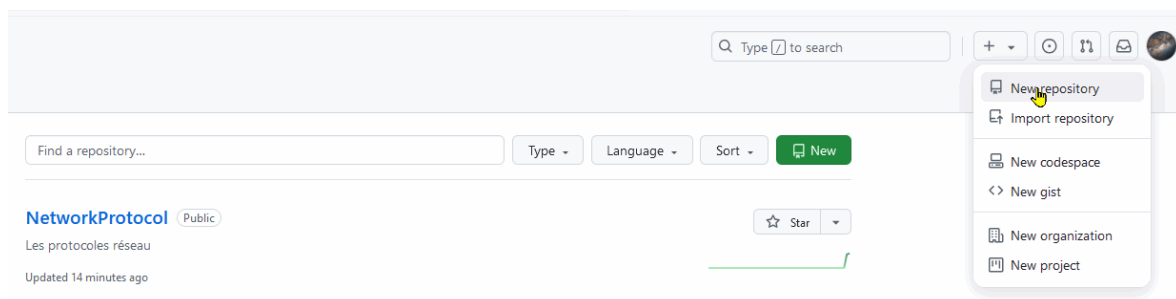
## 2. Créer un repository : 9 étapes

### 2.1. Etape 1→ Créer un nouveau repository sur GitHub

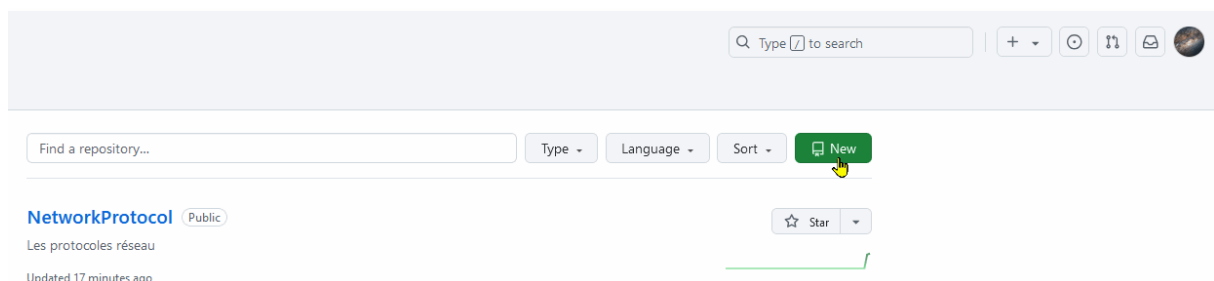
Aller sur [github.com](https://github.com) et la première opération consiste à créer un nouveau repository qui contiendra le code du projet.

On a deux manières de 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.


### 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 (\*).*

 	<b>Créer un repository avec GIT et GITHUB sous Windows</b>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

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

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

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

Quick setup — if you've done this kind of thing before

☐ Set up in Desktop
or
☐ HTTPS
☐ SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

---

...or create a new repository on the command line

```

echo "# mySite" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Djamelikra/mySite.git
git push -u origin main

```

---

...or push an existing repository from the command line

```

git remote add origin git@github.com:Djamelikra/mySite.git
git branch -M main
git push -u origin main

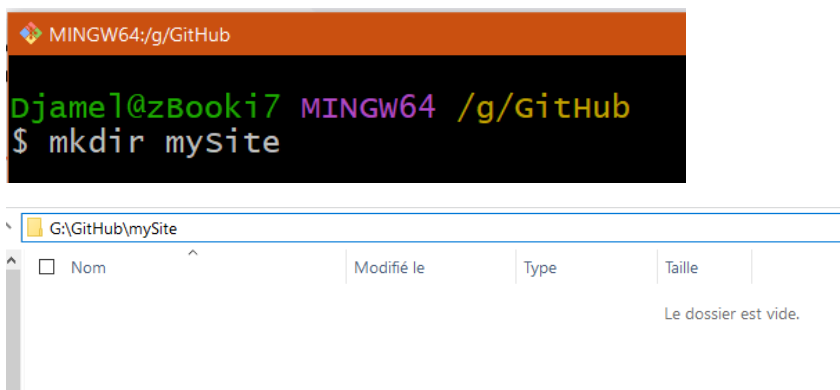
```

► <https://openclassrooms.com/fr/courses/1603881-creez-votre-site-web-avec-html5-et-css3/8061253-tirez-un-maximum-de-ce-cours>

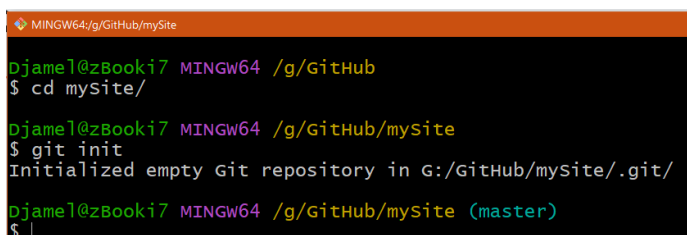
<https://openclassrooms.com/fr/courses/1603881-creez-votre-site-web-avec-html5-et-css3/8061253-tirez-un-maximum-de-ce-cours>

## 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.



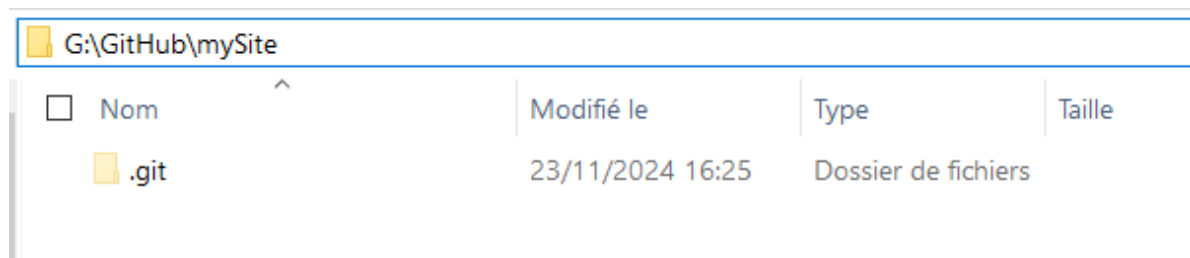
Se déplacer dans le répertoire de travail et l'initialiser avec la commande : **git init**



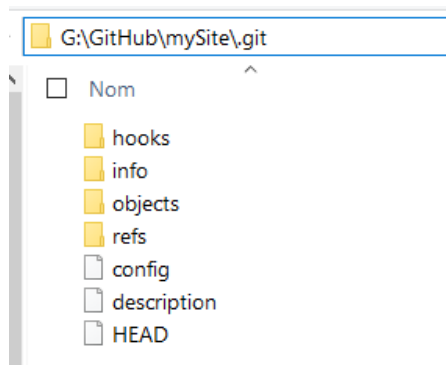
	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

⚠ 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 :



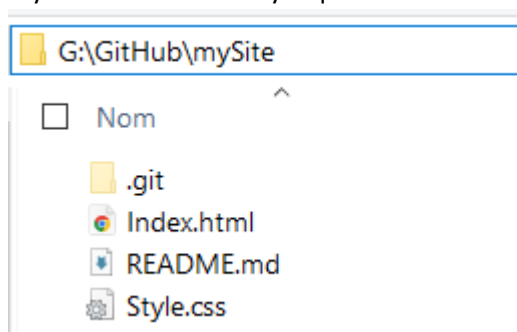
📁 Ce dossier caché `.git` contient toutes les informations nécessaires du projet comme le contrôle de version et toutes les informations sur les commits, l'adresse de dépôt distant, etc. Tous sont présents dans ce dossier. Il contient également un journal qui stocke l'historique, les "logs", les branches...



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

On va ouvrir [VS Code](https://code.visualstudio.com/)<sup>3</sup> pour générer trois fichiers :

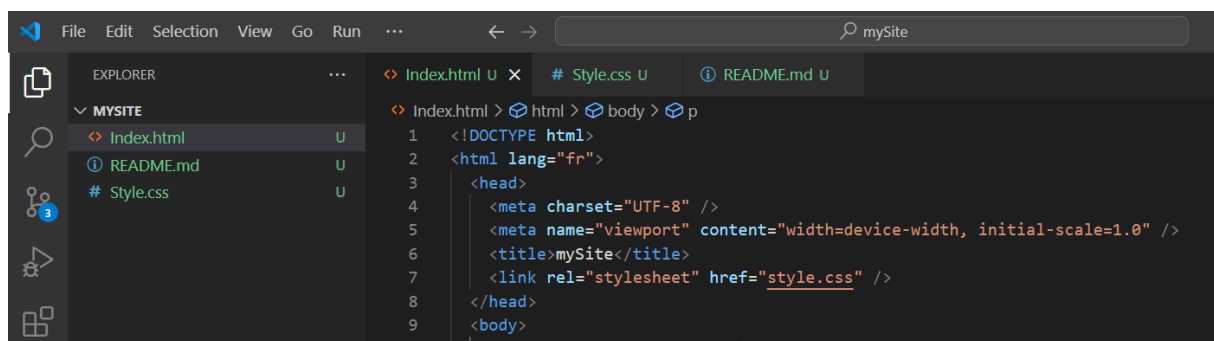
1. README qui porte une extension `.md` pour : Markdown<sup>4</sup> : qui sert à présenter le projet.
2. Index.html : contenu du site.
3. Style.css : fichier de style qui sera relié au fichier précédent.



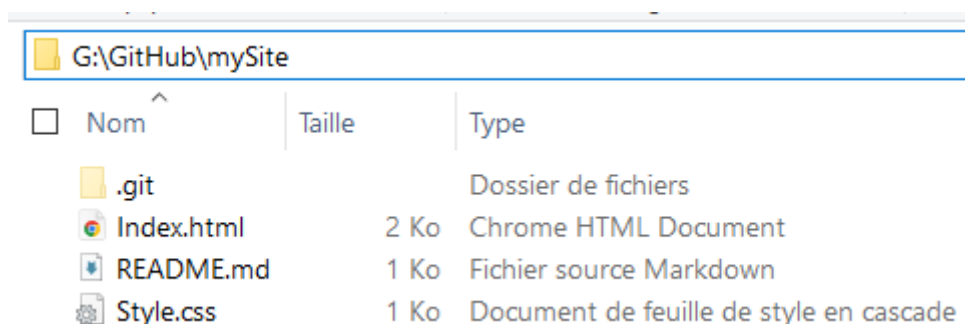
<sup>3</sup> <https://code.visualstudio.com/>

<sup>4</sup> <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 : 14		Modifié le :	26/11/2024

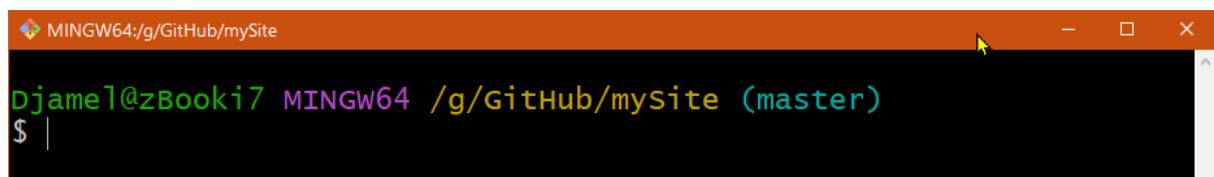


Dossier contenant les 3 fichiers :



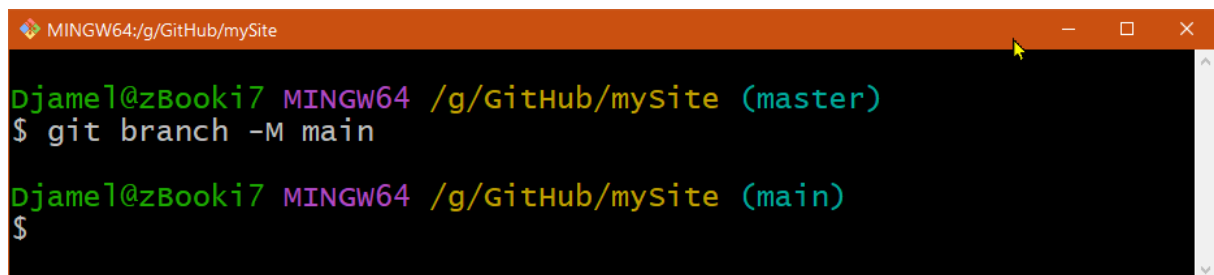
## 2.4. Etape 4 → Renommer la branche master en main

Dès le premier octobre 2020<sup>5</sup>, 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
```



<sup>5</sup> <https://github.com/github/renaming>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

## 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 indexer<sup>6</sup> 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)
$

```

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

<sup>6</sup> Ou bien : `git add -A`

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

```

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)
$

```



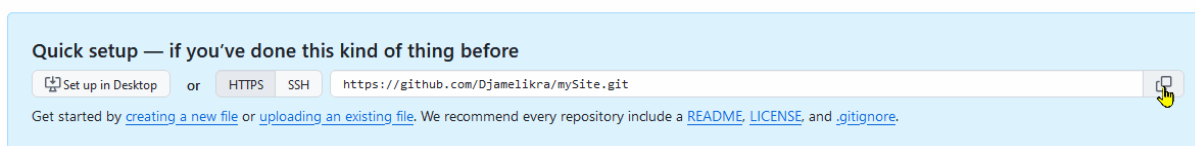
 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

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 :

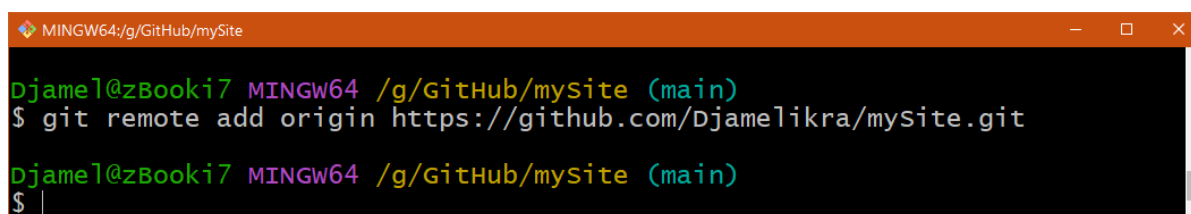
```
git remote add origin <url>
```

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
```



```

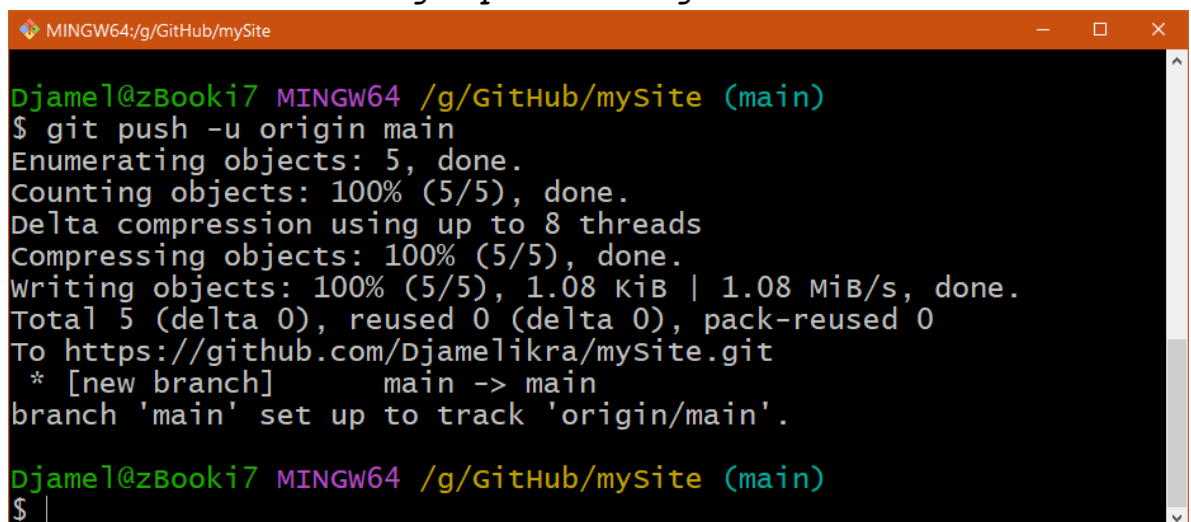
MINGW64:/g/GitHub/mySite
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git remote add origin https://github.com/Djamelikra/mySite.git
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$

```

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
```



```

MINGW64:/g/GitHub/mySite
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.08 KiB | 1.08 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Djamelikra/mySite.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (main)
$

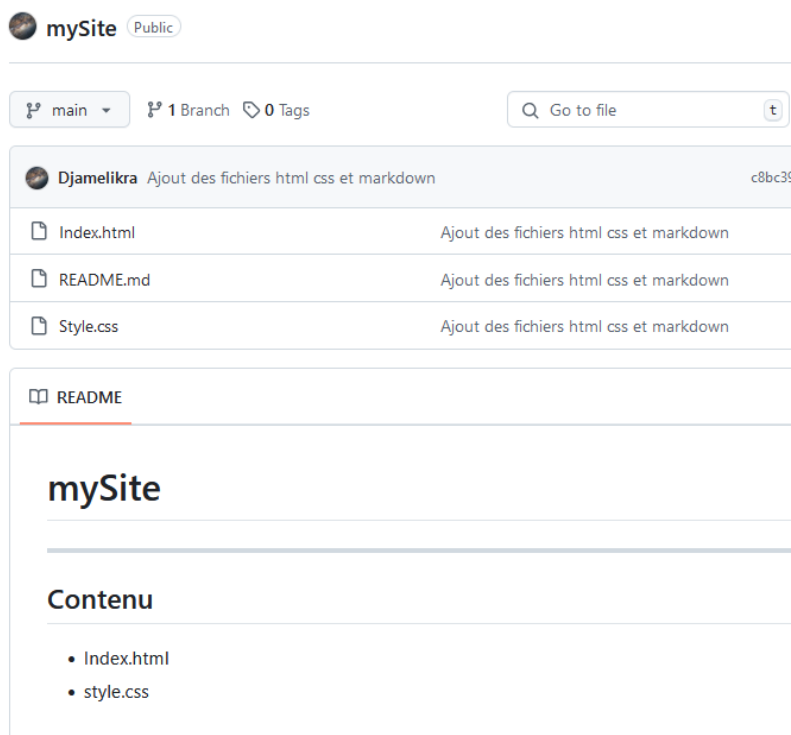
```

u" remplace le paramètre "--set-upstream" lors de l'appel à la commande git push, sans aucun changement entre les deux paramètres<sup>7</sup>.

<sup>7</sup> <https://code-garage.fr/blog/a-quoi-sert-le-parametre-u-lors-d-un-git-push/>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

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.

```
../..
```

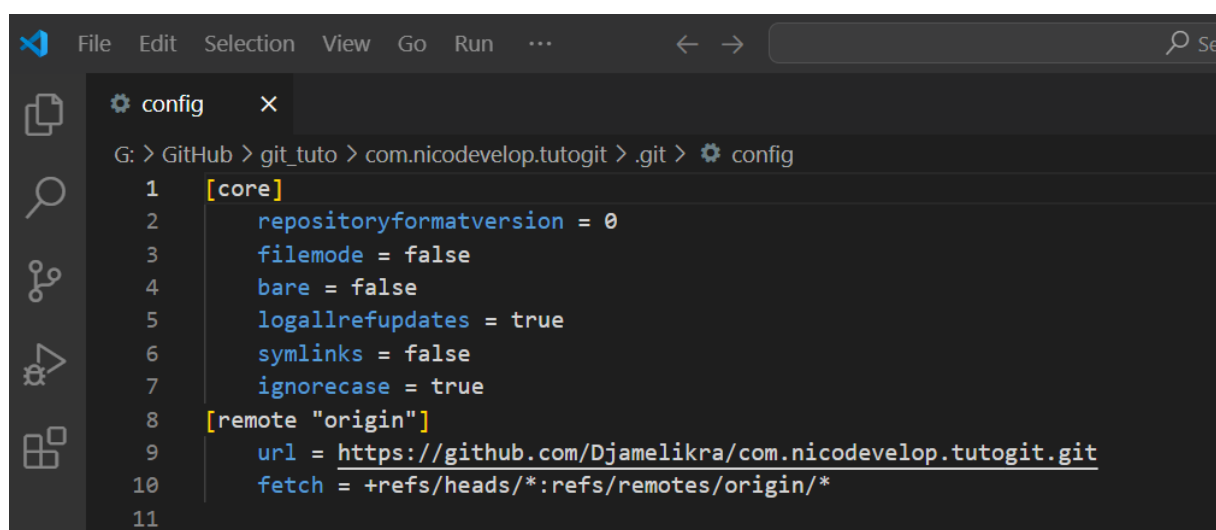
```
git remote add origin https://github.com/Djamelikra/com.nicodevelop.tutogit.git
```

```
Djamel@zBook17 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git remote add origin https://github.com/Djamelikra/com.nicodevelop.tutogit.git
```

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

En +:

```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ ls .git
HEAD config description hooks/ info/ objects/ refs/
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ code .git/config
```



```
1 [core]
2     repositoryformatversion = 0
3     filemode = false
4     bare = false
5     logallrefupdates = true
6     symlinks = false
7     ignorecase = true
8 [remote "origin"]
9     url = https://github.com/Djamelikra/com.nicodevelop.tutogit.git
10    fetch = +refs/heads/*:refs/remotes/origin/*
11
```

### 2.8.1. Connexion en SSH : to do

CF. Fichier docx + pdf

## 2.9. Etape 6 → Créer et ajouter des fichiers dans le projet en local

### 2.10. Etape 7 → « Stager » les changements apportés au projet : git add

Avec : **git add**

### 2.11. Etape 8 → Capturer un instantané des changements actuellement stagés du projet : git commit

Ici aussi il y a deux options : avec git commit et git commit -m

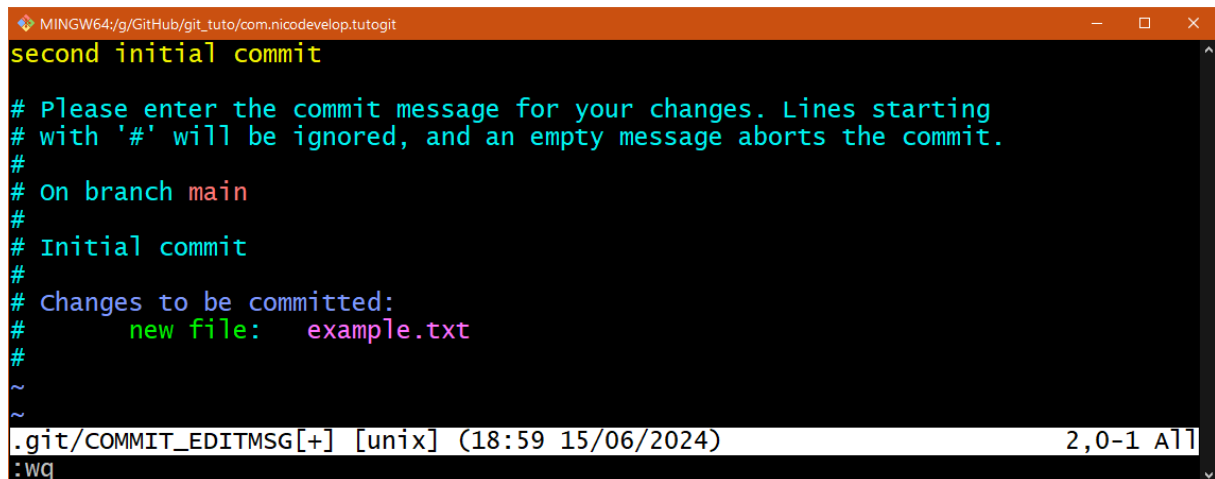
#### 2.11.1. Avec git commit

```
git commit
```

Avec cette syntaxe l'éditeur de texte s'ouvre dans le terminal.

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git commit
```



```

MINGW64: g/GitHub/git_tuto/com.nicodevelop.tutogit
second initial commit

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
#
# Initial commit
#
# Changes to be committed:
#   new file:   example.txt
#
~
~
.git/COMMIT_EDITMSG[+] [unix] (18:59 15/06/2024) 2,0-1 All
:wq

```

**Esc + :wq + Enter**

**(:q ! → pour quitter et annuler le commit)**

```

[main (root-commit) 312a376] second initial commit
1 file changed, 2 insertions(+)
create mode 100644 example.txt
hint: waiting for your editor to close the file...
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$

```

## 2.11.2. Avec git commit -m

```
git commit -m "texte..."8
```

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.

## 2.12. Etape 9 → Envoyer le contenu du dépôt local dans le repository distant situé sur GitHub : git push

Avec : **git push -u origin main**

<sup>8</sup> Code ascii : " : Alt 0147 et " :Alt 0148

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 14		Modifié le :	26/11/2024

```

MINGW64:/g/GitHub/git_tuto/com.nicodevelop.tutogit
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 273 bytes | 273.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Djamelikra/com.nicodevelop.tutogit.git
   c5df8be..f604d0f  main -> main
branch 'main' set up to track 'origin/main'.

Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$

```

"-u" remplace le paramètre "--set-upstream" lors de l'appel à la commande git push, sans aucun changement entre les deux paramètres<sup>9</sup>.

### 2.12.1. Si erreur

Si on rencontre ce message d'erreur :...

```

Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git push origin main
To https://github.com/Djamelikra/com.nicodevelop.tutogit.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/Djamelikra/com.nicodevelop.tutogit.git'
hint: updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: see the 'Note about fast-forwards' in 'git push --help' for details.

Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git pull origin main --allow-unrelated-histories

```

... il faut taper la commande :

```
$ git pull origin main --allow-unrelated-histories
```

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

