

|   |  |              |            |
|---|--|--------------|------------|
|   | <h1>Créer un repository avec GIT et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

# 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 TROIS FICHIERS DANS LE DEPOT LOCAL   | 5         |
| 2.4. ETAPE 4 → RENOMMER LA BRANCHE MASTER EN MAIN   | 6         |
| 2.5. ETAPE 5 → CREER LE LIEN ENTRE LE PROJET LOCAL ET LE PROJET DISTANT SITUE SUR GITHUB                | 7         |
| 2.5.1. Connexion en https (par défaut)  | 7         |
| 2.5.2. Connexion en SSH : to do   | 8         |
| 2.6. ETAPE 6 → CREER ET AJOUTER DES FICHIERS DANS LE PROJET EN LOCAL                                    | 8         |
| 2.7. ETAPE 7 → « STAGER » LES CHANGEMENTS APPORTES AU PROJET : GIT ADD                                  | 8         |
| 2.8. ETAPE 8 → CAPTURER UN INSTANTANE DES CHANGEMENTS ACTUELLEMENT STAGES DU PROJET : GIT COMMIT        | 8         |
| 2.8.1. Avec git commit  | 8         |
| 2.8.2. Avec git commit -m   | 9         |
| 2.9. ETAPE 9 → ENVOYER LE CONTENU DU DEPOT LOCAL DANS LE REPOSITORY DISTANT SITUE SUR GITHUB : GIT PUSH | 9         |
| 2.9.1. Si erreur  | 10        |
| 2.9.2. Comment éviter l'erreur à l'avenir   | 11        |
| <b>3. UTILISER VISUAL STUDIO CODE</b>   | <b>11</b> |
| 3.1. CLONEZ ET UTILISEZ UN REFERENTIEL GITHUB DANS VISUAL STUDIO CODE                                   | 11        |
| 3.2. TRAVAILLER EN LOCAL SUR UN DEPOT GITHUB  | 11        |
| <b>4. CREER UN DEPOT GIT A PARTIR DE VISUAL STUDIO</b>  | <b>12</b> |

## 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 :<br>12   |  | Modifié le : | 24/11/2024 |

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

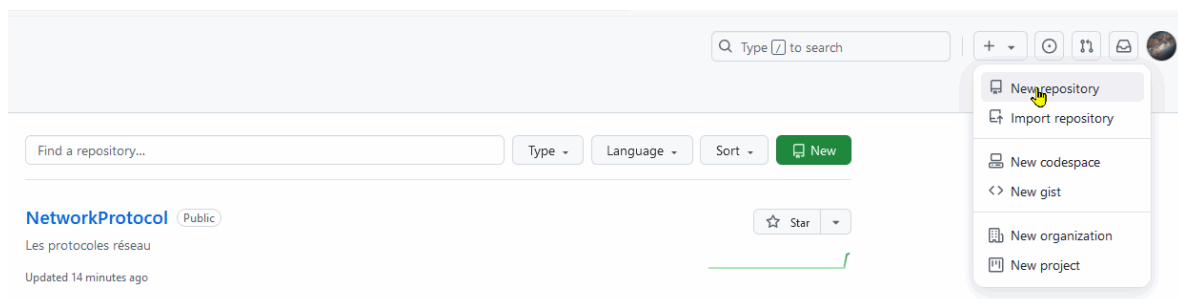
## 2. Créer un repository : 9 étapes

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

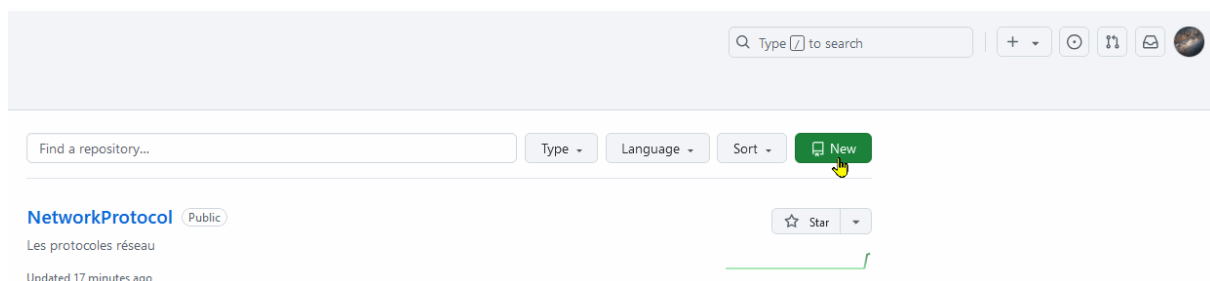
Aller sur [github.com](https://github.com) et la première opération consiste à créer un nouveau repository qui contiendra le code du projet.

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<br/>et GITHUB sous Windows</b> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

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 :<br>12   |  | Modifié le : | 24/11/2024 |

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.

MINGW64:/g/GitHub

```

Djamel@zBooki7 MINGW64 /g/GitHub
$ mkdir mySite

```

G:\GitHub\mySite

| Nom                  | Modifié le | Type | Taille |
|----------------------|------------|------|--------|
| Le dossier est vide. |            |      |        |

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

MINGW64:/g/GitHub/mySite

```

Djamel@zBooki7 MINGW64 /g/GitHub
$ cd mySite/

Djamel@zBooki7 MINGW64 /g/GitHub/mySite
$ git init
Initialized empty Git repository in G:/GitHub/mySite/.git/

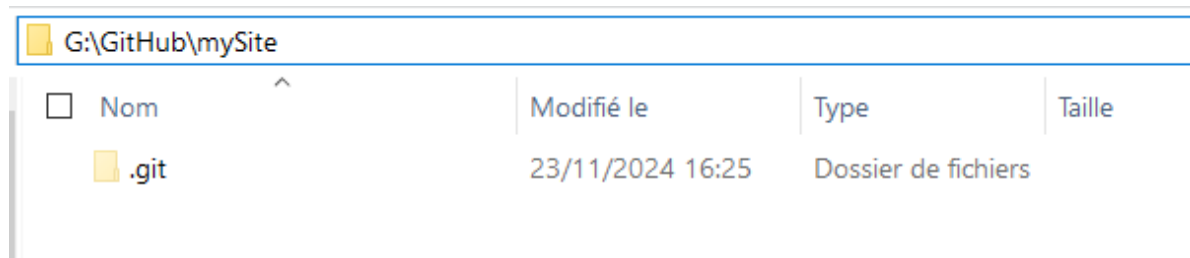
Djamel@zBooki7 MINGW64 /g/GitHub/mySite (master)
$

```

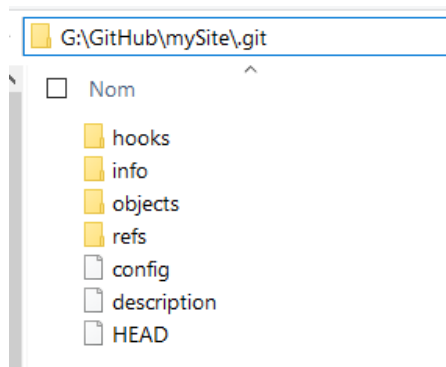
|   |  |              |            |
|---|--|--------------|------------|
|  | <h1>Créer un repository avec GIT et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

⚠ 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 :



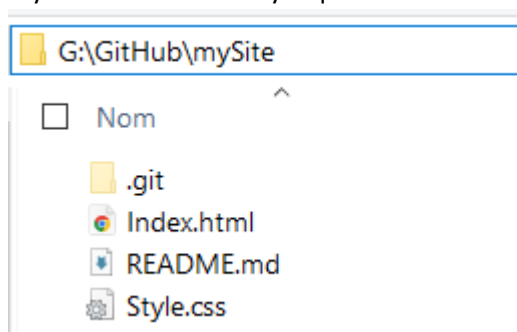
📁 Ce dossier caché `.git` contient toutes les informations nécessaires du projet comme le contrôle de version et toutes les informations sur les commits, l'adresse de dépôt distant, etc. Tous sont présents dans ce dossier. Il contient également un journal qui stocke l'historique, les "logs", les branches...



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

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

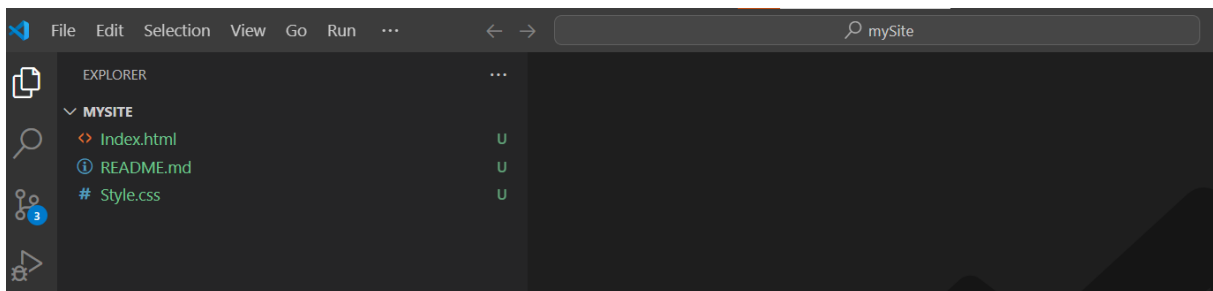
1. README qui porte une extension `.md` pour : Markdown<sup>4</sup> : qui sert à présenter le projet.
2. Index.html : contenu du site.
3. Style.css : fichier de style qui sera relié au fichier précédent.



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

<sup>4</sup> <https://fr.wikipedia.org/wiki/Markdown>

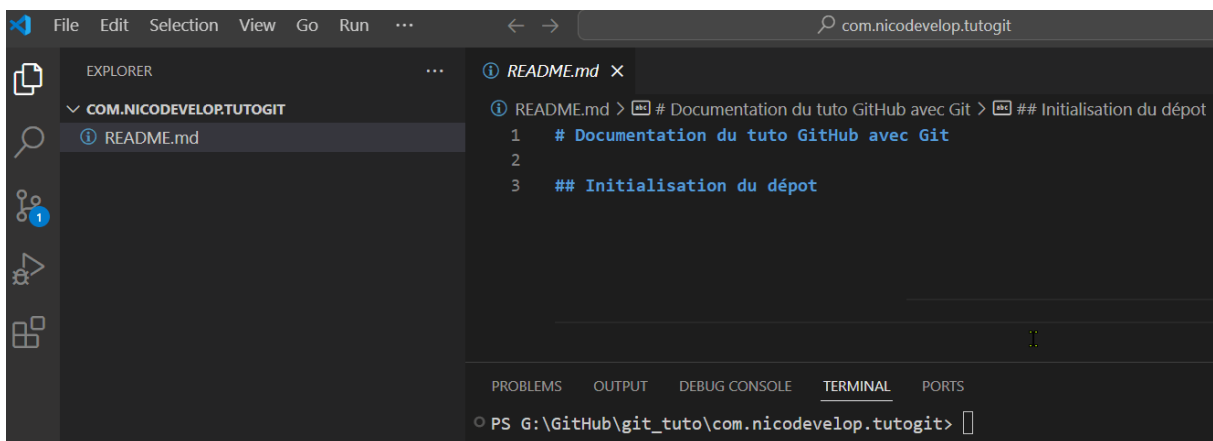
|   |  |              |            |
|---|--|--------------|------------|
|  | <h1>Créer un repository avec GIT et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |



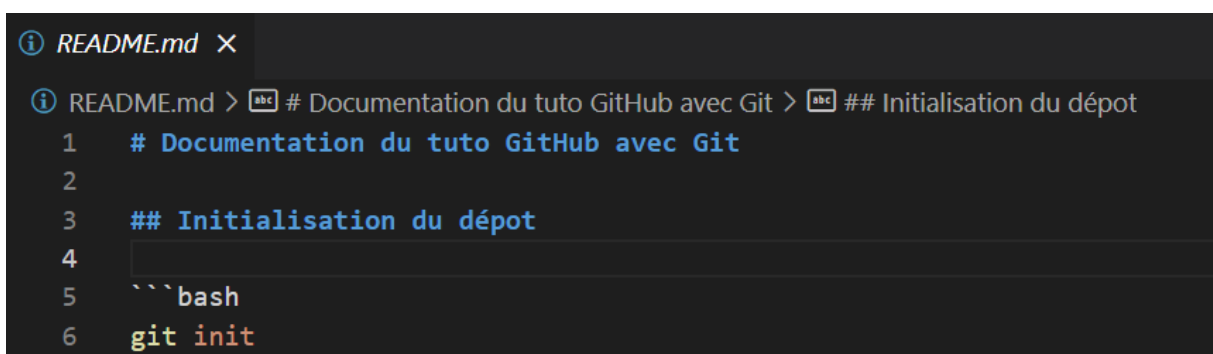
Dossier contenant les 3 fichiers :



Rajouter dans le fichier markdown :



En parallèle, des commandes tapées, on met à jour la documentation dans le fichier README.md .



## 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

<sup>5</sup> <https://www.developpez.com/actu/308996/GitHub-va-desormais-faire-usage-du-terme-main-en-lieu-et-place-de-master-pour-designer-la-branche-par-defaut-des-projets-et-ainsi-lutter-contre-les-stereotypes-raciaux-sur-la->

|   |  |              |            |
|---|--|--------------|------------|
|   | <h1>Créer un repository avec GIT<br/>et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

mesure cible également les dépôts existants qui s'appuient sur le terme `master` pour désigner la branche principale.

Commande : `git branch -M main`

```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (master)
$ git branch -M main

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

2.5. Etape 5 → Créer le lien entre le projet local et le projet distant situé sur GitHub  
A ce niveau il y a deux types de connexion : https et SSH.

2.5.1. Connexion en https (par défaut)

En utilisant la commande :

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

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

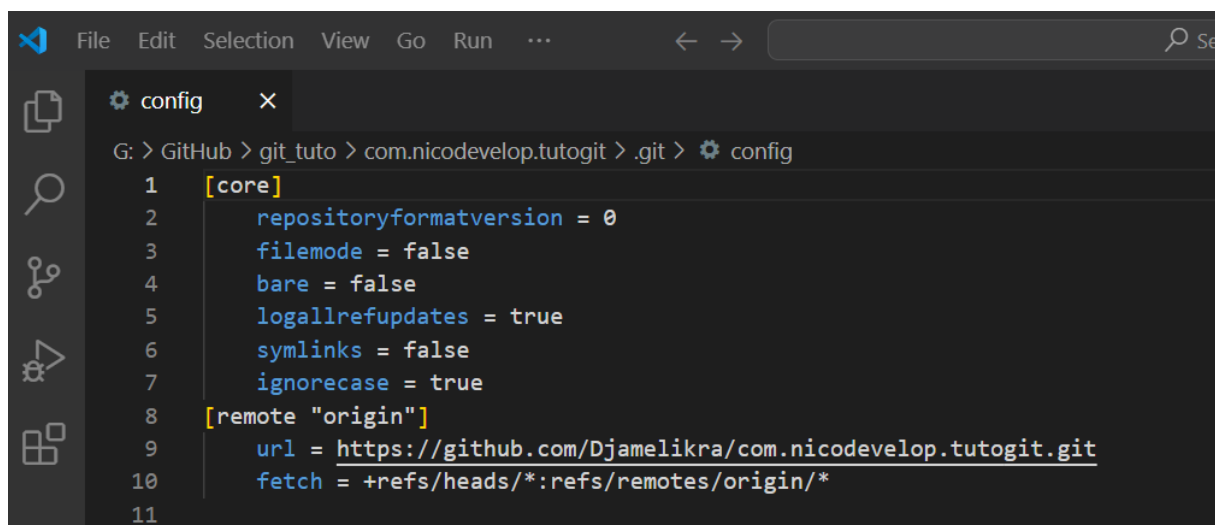
En +:

```
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
```

[plateforme/#:~:text=%C2%AB%20D%C3%A8s%20le%20premier%20octobre%202020,pour%20d%C3%A9signer%20la%20branche%20principale.](#)

|   |  |              |            |
|---|--|--------------|------------|
|   | <h1>Créer un repository avec GIT et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |



```

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

```

## 2.5.2. Connexion en SSH : to do

CF. Fichier docx + pdf

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

```

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

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

```

```

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

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    example.txt

```

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

Avec : **git add**

```

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

```

## 2.8. 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

### 2.8.1. Avec git commit

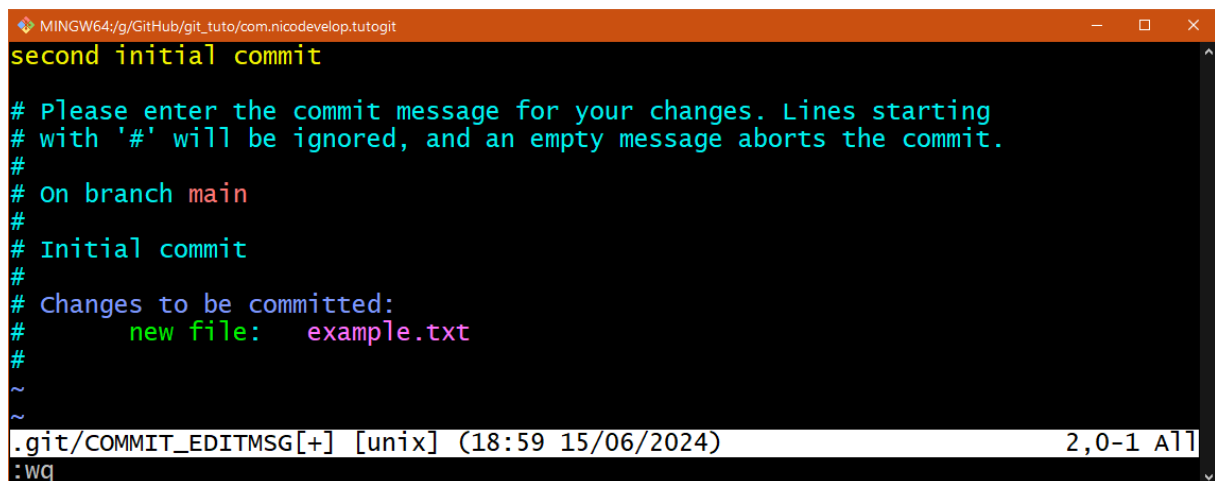
git commit



|   |  |              |            |
|---|--|--------------|------------|
|   | <h1>Créer un repository avec GIT et GITHUB sous Windows</h1> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

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

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



**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)
$
```

## 2.8.2. Avec git commit -m

```
git commit -m "texte..."6
```

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.

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

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

<sup>6</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 :<br>12   |  | Modifié le : | 24/11/2024 |

```

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>7</sup>.

### 2.9.1. 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>7</sup> <https://code-garage.fr/blog/a-quoi-sert-le-parametre-u-lors-d-un-git-push/>



|   |  |              |            |
|---|--|--------------|------------|
|   | <b>Créer un repository avec GIT<br/>et GITHUB sous Windows</b> | Réalisé le : | 12/11/2024 |
| Nombre de pages :<br>12   |  | Modifié le : | 24/11/2024 |

#### 4. Créer un dépôt Git à partir de Visual Studio

<https://learn.microsoft.com/fr-fr/visualstudio/version-control/git-create-repository?view=vs-2022>