

 	Créer un repository avec GIT et GITHUB sous Windows	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

Créer un repository avec GIT et GITHUB sous Windows

1. PREREQUIS	2
2. CREER UN REPOSITORY : 9 ETAPES	2
2.1. ETAPE 1→ CREER UN NOUVEAU REPOSITORY SUR GITHUB	2
2.2. ETAPE 2→ CREER UN NOUVEAU DOSSIER EN LOCAL PORTANT LE MEME NOM QUE LE REPOSITORY SUR GITHUB	4
2.2.1. Lancer VS Code	4
2.2.2. Créer le fichier README.md	5
2.3. ETAPE 3→ TAPER LA COMMANDE GIT INIT POUR INITIALISER LE PROJET	6
2.4. ETAPE 4 → RENOMMER LA BRANCHE MASTER EN MAIN	7
2.5. ETAPE 5 → CREER LE LIEN ENTRE LE PROJET LOCAL ET LE PROJET DISTANT SITUE SUR GITHUB	8
2.5.1. Connexion en https (par défaut)	8
2.5.2. Connexion en SSH : to do	9
2.6. ETAPE 6 → CREER ET AJOUTER DES FICHIERS DANS LE PROJET EN LOCAL	9
2.7. ETAPE 7 → « STAGER » LES CHANGEMENTS APPORTES AU PROJET : GIT ADD	9
2.8. ETAPE 8 →CAPTURER UN INSTANTANE DES CHANGEMENTS ACTUELLEMENT STAGES DU PROJET : GIT COMMIT	9
2.8.1. Avec git commit	9
2.8.2. Avec git commit -m	10
2.9. ETAPE 9 → ENVOYER LE CONTENU DU DEPOT LOCAL DANS LE REPOSITORY DISTANT SITUE SUR GITHUB : GIT PUSH	10
2.9.1. Si erreur	11
2.9.2. Comment éviter l'erreur à l'avenir	12
3. AVEC CHATGPT : 4 ETAPES + 1	12
3.1.1. Prérequis	12
3.1.2. Étape 1 : Créer un nouveau repository sur GitHub	13
3.1.3. Étape 2 : Cloner le repository sur votre machine locale	13
3.1.4. Étape 3 : Ajouter des fichiers et effectuer un commit	13
3.1.5. Étape 4 : Pousser les changements vers GitHub	13
3.1.6. Résumé des commandes Git	13
4. YOU.COM & GPT-4O : 4 ÉTAPES	14
4.1. ETAPE 1→CREER UN REPOSITORY LOCAL AVEC GIT	14
4.1.1. Initialiser un nouveau repository Git	14
4.1.2. Ajouter des fichiers au repository local	14
4.1.3. Faire un commit initial	14
4.2. ETAPE2→CREER UN REPOSITORY SUR GITHUB	15
4.2.1. Accéder à GitHub	15
4.2.2. Créer un nouveau repository	15
4.2.3. Configurer le repository	15
4.2.4. Initialiser le repository avec un README (optionnel)	15
4.2.5. Créer le repository	15
4.3. ETAPE3→LIER LE REPOSITORY LOCAL A GITHUB	15
4.3.1. Ajouter l'URL du repository distant	15
4.4. ETAPE4→POUSSER LES CHANGEMENTS VERS GITHUB	15
5. OC	16

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

6. LINKS-----16

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.

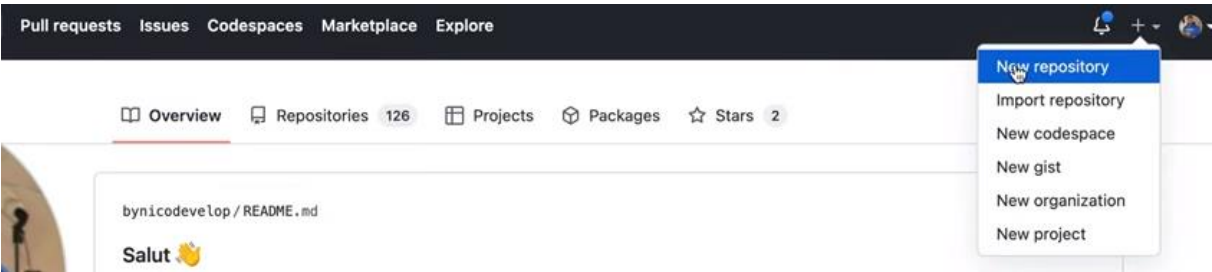
- Créez un compte GitHub si ce n'est pas déjà fait. [Inscrivez-vous sur GitHub ici](#).
- Réaliser l’installation de Git sur votre machine pour cela il faut :
 - Avoir Un accès internet.
 - [Téléchargez Git ici](#).
 - Disposer des droits administrateur sur votre machine.
- Configurer Git sur Windows : <https://chatgpt.com/share/74212c96-f594-4f07-a7ba-6ba1eddf6c47>

../..

2. Créer un repository : 9 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.




 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *

 bynicodevelop


/

Great repository names are short and memorable. Need inspiration? How about **verbose-octo-fiesta**?

Description (optional)


☒

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.



 bynicodevelop


/

com.nicodevelop.tutogit


✓

Great repository names are short and memorable. Need inspiration? How about **verbose-octo-fiesta**?

Description (optional)

☒

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:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license

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

License: None

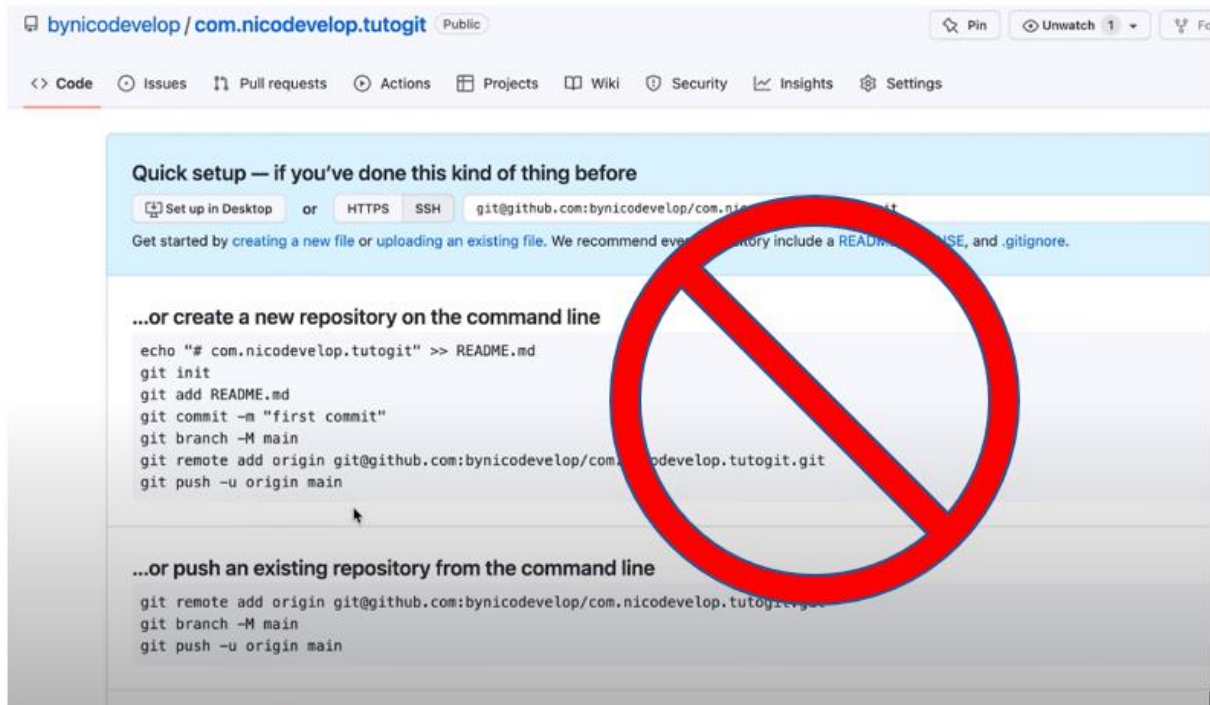
ⓘ You are creating a public repository in your personal account.

Create repository

⚠ Ne pas ajouter de fichier README ni de gitignore !

Pour éviter les risques de conflits lors des synchronisations.

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024



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

2.2. Etape 2 → Créer un nouveau dossier en local portant le même nom que le repository sur GitHub

→ `git_tuto mkdir com.nicodevelop.tutogit`

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

```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto
$ mkdir com.nicodevelop.tutogit

Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto
$ ls
com.nicodevelop.tutogit/

Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto
$ cd com.nicodevelop.tutogit/

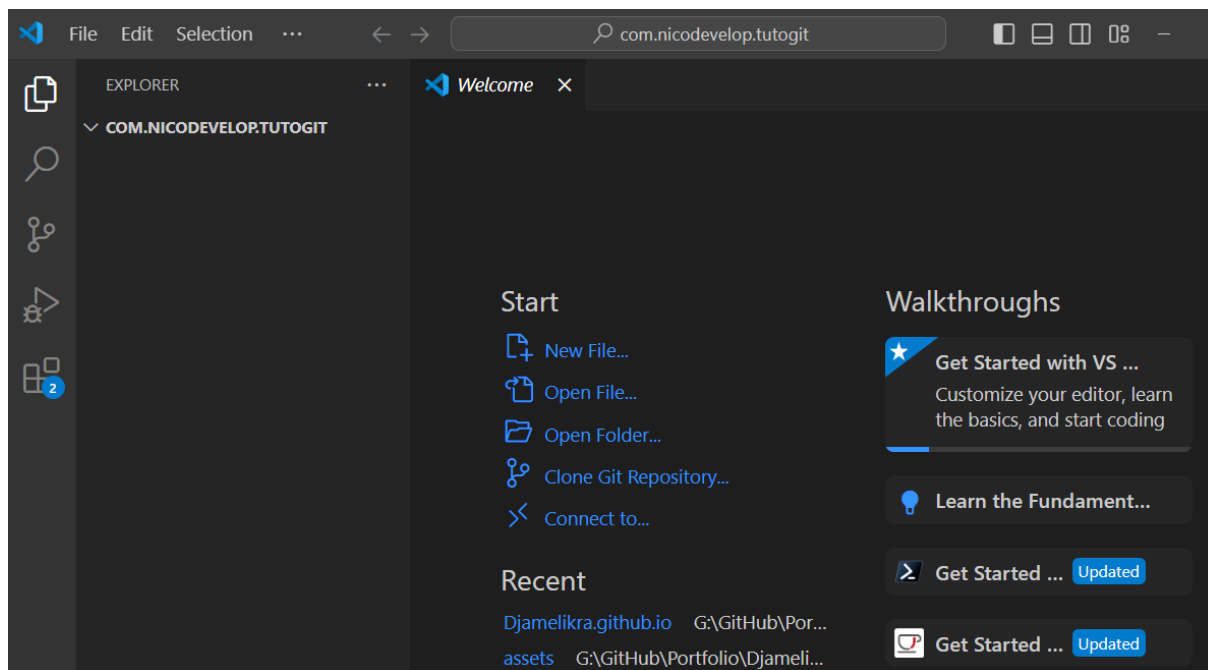
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit
$ code .
```

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

2.2.1. Lancer VS Code

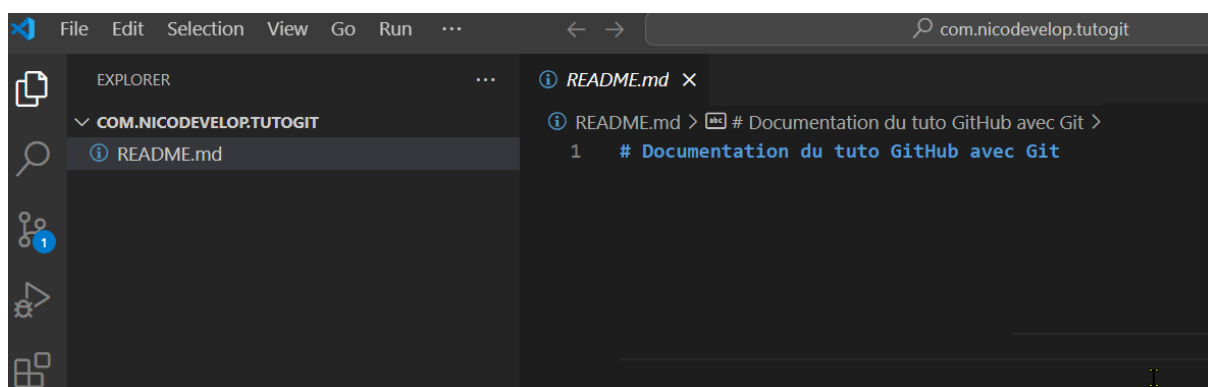
On va ouvrir VS Code pour créer le fichier README

```
djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit
$ code .
```



2.2.2. Créer le fichier README.md

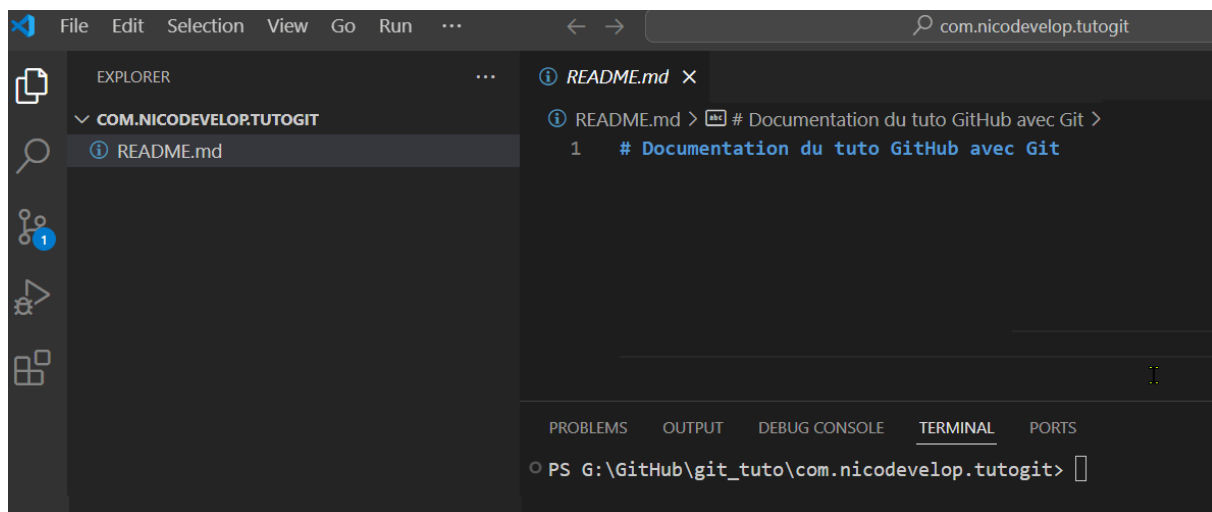
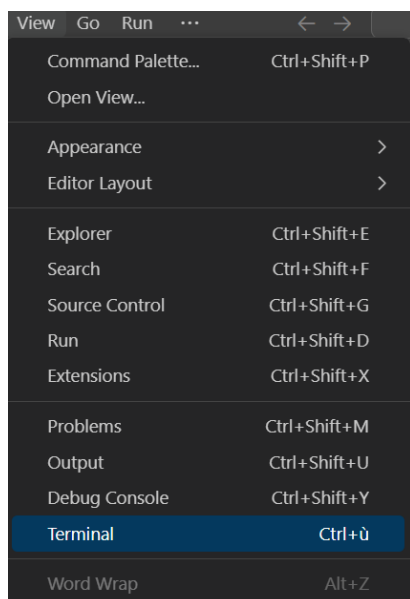
Le fichier README.md porte une extension .md pour : Markdown¹.



Ensuite il faudra ouvrir un terminal pour pouvoir y taper des commandes.

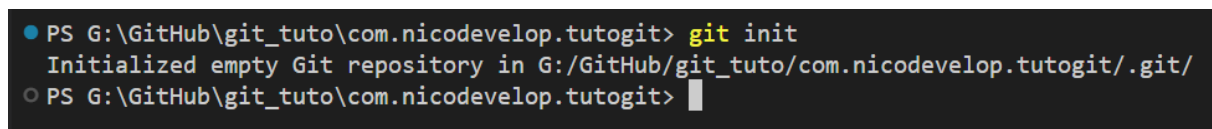
¹ <https://fr.wikipedia.org/wiki/Markdown>

	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024



2.3. Etape 3→ taper la commande git init pour initialiser le projet

`git init`



Une fois le projet initialisé grâce à la commande `git init`, un nouveau dossier `.git` apparait dans le dossier du projet à côté du fichier `README.md` :

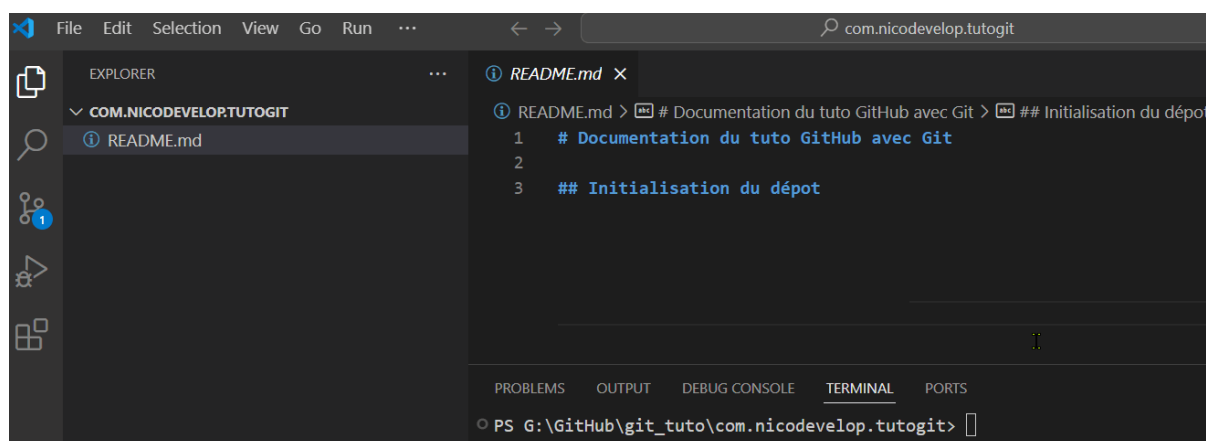
 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (master)
$ ls -al
total 8
drwxr-xr-x 1 Djamel 197121 0 Jun 15 17:26 ./
drwxr-xr-x 1 Djamel 197121 0 Jun 15 12:46 ../
drwxr-xr-x 1 Djamel 197121 0 Jun 15 17:26 .git/
```

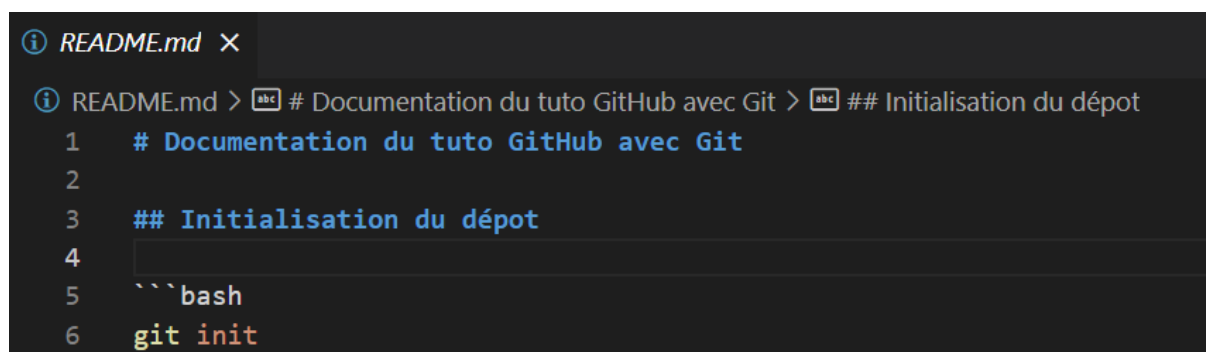
⚠ Il faut vérifier qu'on est bien placé dans le bon dossier avant de faire un `git init`

Rajouter dans le fichier markdown :

```
## Initialisation du dépôt
```



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 2020², tous les nouveaux dépôts que vous créerez utiliseront main plutôt que master pour désigner la branche par défaut », annonce GitHub. La

² <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-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 : 17		Modifié le :	18/11/2024

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`

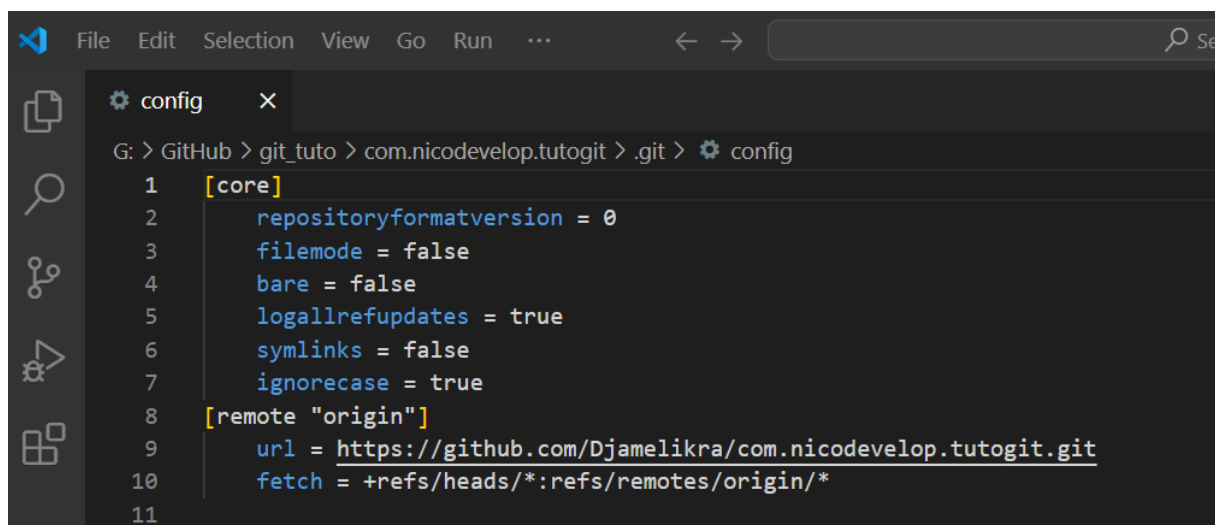
```
Djamel@zBooki7 MINGW64 /g/GitHub/git_tuto/com.nicodevelop.tutogit (main)
$ git remote add origin https://github.com/Djamelikra/com.nicodevelop.tutogit.git
```

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
```


 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024



```

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

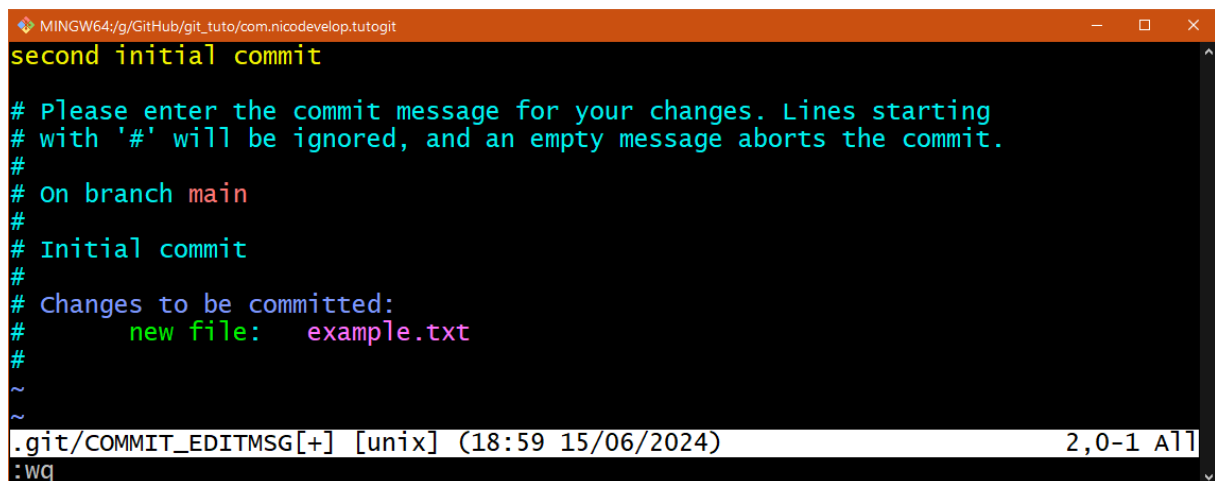
2.8.1. Avec git commit

git commit

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

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..."3
```

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**

³ 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 : 17		Modifié le :	18/11/2024

```

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 paramètres⁴.

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
```

⁴ <https://code-garage.fr/blog/a-quoi-sert-le-parametre-u-lors-d-un-git-push/>

	<h2>Créer un repository avec GIT et GITHUB sous Windows</h2>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

3.1.2. Étape 1 : Créer un nouveau repository sur GitHub

1. Connectez-vous à votre compte GitHub.
2. Cliquez sur le bouton **New repository** (ou sur le signe plus (+) en haut à droite, puis sélectionnez **New repository**).
3. Remplissez les champs nécessaires :
 - **Repository name** : Donnez un nom à votre repository.
 - **Description** (optionnel) : Ajoutez une description.
 - Sélectionnez **Public** ou **Private** selon vos préférences de visibilité.
 - Cochez l'option **Initialize this repository with a README** si vous souhaitez ajouter un fichier README initial.
4. Cliquez sur **Create repository**.

3.1.3. Étape 2 : Cloner le repository sur votre machine locale

1. Ouvrez votre terminal ou votre interface de ligne de commande.
2. Naviguez vers le répertoire où vous souhaitez cloner votre repository.
3. Utilisez la commande suivante pour cloner le repository (remplacez `YOUR-USERNAME` et `YOUR-REPOSITORY` par votre nom d'utilisateur GitHub et le nom de votre repository) :

```
git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY.git
```

4. Accédez au répertoire cloné :

```
cd YOUR-REPOSITORY
```

3.1.4. Étape 3 : Ajouter des fichiers et effectuer un commit

1. Ajoutez ou modifiez des fichiers dans le répertoire cloné.
2. Utilisez les commandes Git suivantes pour ajouter ces fichiers et effectuer un commit :

```
git add .
git commit -m "Initial commit"
```

3.1.5. Étape 4 : Pousser les changements vers GitHub

1. Utilisez la commande suivante pour pousser vos commits vers le repository distant sur GitHub :

```
git push origin main
```

Remarque : La branche principale par défaut peut être `main` ou `master`, selon la configuration de votre repository. Vérifiez et utilisez le bon nom de branche.

3.1.6. Résumé des commandes Git

Cloner le repository depuis GitHub

```
git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY.git
```

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

Naviguer vers le répertoire cloné
cd YOUR-REPOSITORY

Ajouter des fichiers et les commiter
git add .
git commit -m "Initial commit"

Pousser les changements vers GitHub
git push origin main

4. You.com & GPT-4o : 4 étapes

4.1. Etape 1→Créer un repository local avec Git

4.1.1. Initialiser un nouveau repository Git

Ouvrez votre terminal et naviguez vers le répertoire où vous souhaitez créer votre projet.
Ensuite, exécutez la commande suivante :

```
git init
```

Cette commande crée un sous-répertoire `.git` dans votre répertoire de projet, qui contient tous les fichiers nécessaires pour le contrôle de version.

4.1.2. Ajouter des fichiers au repository local

Ajouter les fichiers que vous souhaitez suivre avec Git en utilisant la commande `git add` :

```
git add .
```

ou bien

```
git add -A
```

Cette commande ajoute **tous les fichiers** du répertoire courant à l'index de Git.
On dit aussi « stager » les fichiers.

4.1.3. Faire un commit initial

Enregistrer les fichiers ajoutés dans l'historique de Git avec un commit :

```
git commit -m "Initial commit"
```

Cette commande **enregistre un instantané** de vos fichiers dans l'historique de Git avec un message de commit descriptif.

 	<h1>Créer un repository avec GIT et GITHUB sous Windows</h1>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

4.2. Etape2→Créer un repository sur GitHub

4.2.1. Accéder à GitHub

Connectez-vous à votre compte GitHub. Si vous n'avez pas encore de compte, créez-en un sur github.com.

4.2.2. Créer un nouveau repository

Cliquez sur le bouton "+" dans le coin supérieur droit de la page GitHub et sélectionnez "New repository".

4.2.3. Configurer le repository

Remplissez les détails de votre nouveau repository :

- **Nom du repository** : Choisissez un nom unique pour votre projet.
- **Description** (optionnel) : Ajoutez une brève description de votre projet.
- **Visibilité** : Choisissez entre un repository public ou privé.

4.2.4. Initialiser le repository avec un README (optionnel)

Vous pouvez cocher l'option pour initialiser le repository avec un fichier README, ce qui est utile pour décrire votre projet.

4.2.5. Créer le repository

Cliquez sur le bouton "Create repository" pour finaliser la création de votre repository sur GitHub.

4.3. Etape3→Lier le repository local à GitHub

4.3.1. Ajouter l'URL du repository distant

Retournez à votre terminal et ajoutez l'URL de votre nouveau repository GitHub comme remote :

```
git remote add origin https://github.com/votre-utilisateur/votre-repository.git
```

Remplacez `votre-utilisateur` et `votre-repository` par votre nom d'utilisateur GitHub et le nom de votre repository respectivement.

4.4. Etape4→Pousser les changements vers GitHub

Envoyez vos commits locaux vers le repository GitHub avec la commande suivante :

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

Cette commande pousse les changements de la branche `main` vers GitHub et définit `origin` comme le remote par défaut pour les futures commandes `git push`.

 	<h2>Créer un repository avec GIT et GITHUB sous Windows</h2>	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024

Et voilà ! Vous avez maintenant un repository Git local lié à un repository GitHub. Vous pouvez continuer à travailler sur votre projet localement et pousser vos changements vers GitHub pour les sauvegarder et les partager avec d'autres.

5. OC

<https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github/7165721-installez-git-sur-votre-ordinateur>

Maintenant que vous avez configuré les paramètres de base, vous pouvez **créer votre fameux dépôt local**. Pour ce faire, **deux solutions sont possibles** :

- **créer un dépôt local** vide pour accueillir un nouveau projet : la procédure est expliquée ci-dessous ;
- **cloner un dépôt distant**, c'est-à-dire rapatrier l'historique d'un dépôt distant en local. Nous aborderons cette méthode dans la deuxième partie du cours.

6. Links

https://www.youtube.com/watch?v=85lbuuFIXG4&list=PLdKVEanRftb-H1p8ar5gt7EBPixFIVt-y&index=5&ab_channel=NicoDevelop

<https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github/7165721-installez-git-sur-votre-ordinateur>

	Créer un repository avec GIT et GITHUB sous Windows	Réalisé le :	12/11/2024
Nombre de pages : 17		Modifié le :	18/11/2024