

# GIT : TP d'initiation

**Objectif de l'exercice :** Ces exercices visent à vous familiariser avec les concepts clés de Git, notamment la création de dépôts, la gestion des commits, l'utilisation des branches et la fusion de ces branches.

**Remarque :**

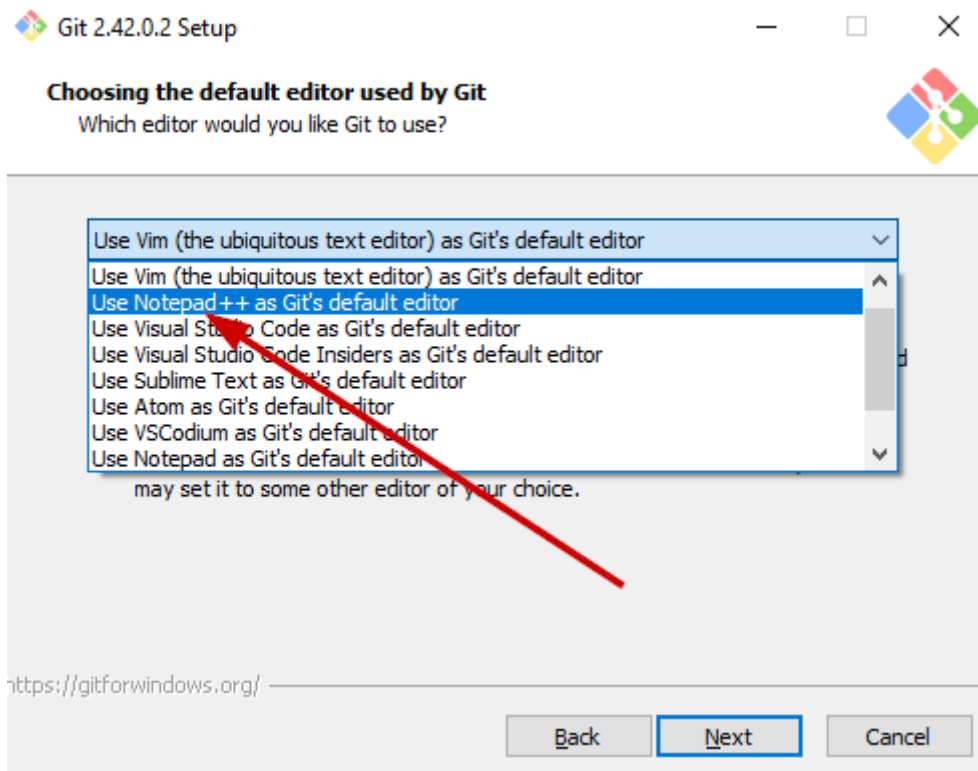
- Les commandes dans le bash de GIT sont celles de Linux (mkdir etc...)
- Répondez aux questions, et donnez les commandes GIT nécessaires

## I. Etape N° 1: Installation et Configuration de Git

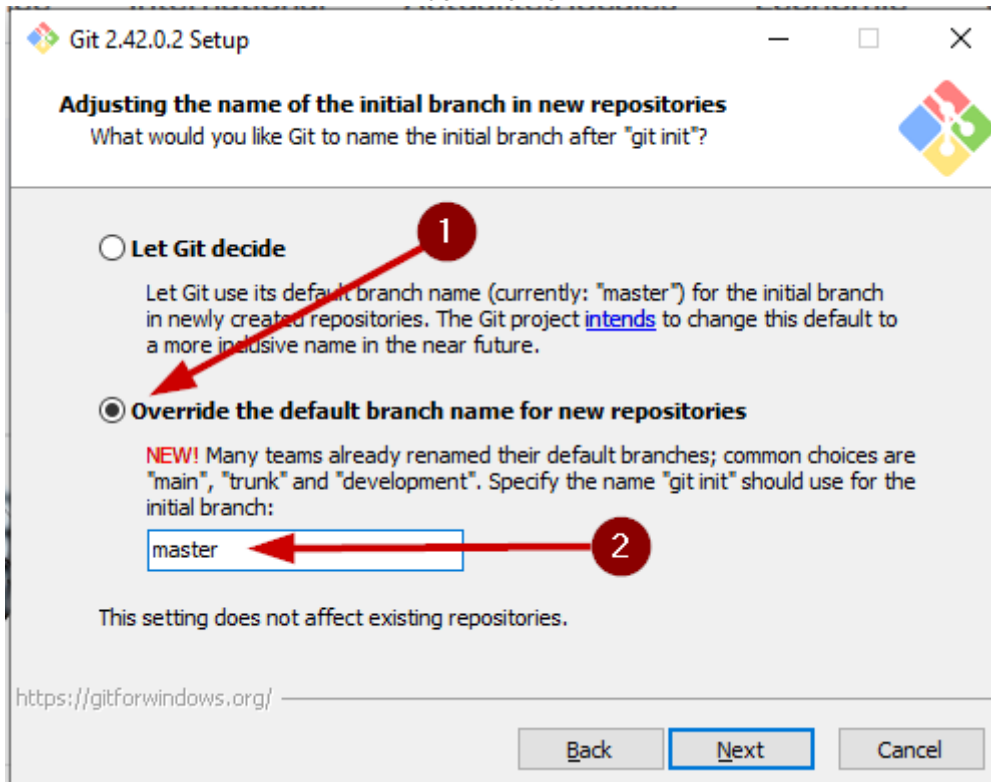
Étape N°1 : Visitez le site officiel de <https://git-scm.com/downloads> pour télécharger et installer Git (sauf pour les derniers mac)

**Remarque pour les utilisateur windows :**

Modifiez l'éditeur par défaut (Notepad++ ou Visual Code ...)



Il semble qu'il y ait un problème de génération de la branche principale, aussi cochez (1) et mettre *master* comme nom



Étape N°2 : Vérification : Ouvrez un terminal ou une commande (pour windows Clic droit )

Tapez git --version

Étape N°3 : Création d'un alias pour notepad++



Par exemple si votre logiciel a pour chemine complet :

"C:\Program Files\Notepad++\notepad++.exe"

Alors la création d'alias (raccourci) pour l'utiliser dans git est :

alias editeur="/C/Program\ Files/Notepad++/notepad++.exe"

Remarque : si le dossier contenant le répertoire Notepad++ est « Program Files (x86) alors  
alias editeur='c/Program\ Files \ \(x86\)/Notepad++/notepad++.exe'

	GIT	
	TP D'INITIATION	

Étape N°4 : Maintenant dans bash GIT vous pouvez éditer n'importe quel fichier à l'aide de la commande *editeur* suivi du nom du fichier

Remarque : pour revenir dans le bash de GIT il faut fermer notepad++

---

## II. Étape 2: Configuration initiale

---

Étape N°5 : Configurez votre identité :

1. `git config --global user.name "Votre Nom"`
2. `git config --global user.email votre@email.com`

Étape N°6 : Vérifiez la bonne configuration avec `git config --list` (sortir en entrant la lettre q)

---

## III. Étape 3 : Création et gestion d'un dépôt local

---

Étape N°7 : Créez un nouveau dossier (`mkdir PremierProjetWindows`) et initialisez un dépôt Git :

Étape N°8 : Allez dans le dossier (commande `cd`) et créez un fichier `README.md` (`$ touch README.md`).  
Y ajouter du contenu (en utilisant l'alias)

Étape N°9 : Ajoutez puis committez ce fichier.

Étape N°10 : Vérification : Exécutez `git log`.

---

## IV. Étape 4: Exploration des branches

---

Étape N°11 : Créez une nouvelle branche : `git checkout -b DEV` et rappelez ce que fait exactement cette commande

Étape N°12 : Visualisez toutes les branches

Étape N°13 : Modifiez `README.md` (en ajoutant par exemple « Modifié dans la branche `feature1` ») et committez le changement (sans oublier de l'ajouter)

Étape N°14 : Vérifier avec un `git status`

Étape N°15 : Basculez vers la branche principale (MASTER), vérifiez avec un `git branch`

Étape N°16 : Visualisez à nouveau le fichier `README.md`, a-t-il été modifié ?

Étape N°17 : Vérification : exécutez `git log`

## V. Étape 4: Fusion, conflits et leur résolution

- Étape N°18 : Dans la branche principale, modifiez **README.md** à un endroit où **dev** l'a déjà modifié.
- Étape N°19 : Committez le changement mais cette fois en utilisant `git commit -a`.
- Étape N°20 : Quelles sont les différences entre `git commit -m` et `git commit -a` ?
- Étape N°21 : Essayez de fusionner **dev** dans la branche principale (avec `merge`). Résolvez les conflits manuellement en éditant le fichier. A la fin vérifiez avec un `git status`, ajoutez les fichiers puis un `commit`

## VI. Étape 5: Introduction aux dépôts distants avec GitHub

- Étape N°22 : Créez un compte sur GitHub si vous n'en avez pas.
- Étape N°23 : Créez un nouveau dépôt et liez-le à votre dépôt local.
- Étape N°24 : Poussez vos modifications vers le dépôt distant.
- Étape N°25 : Vérification : Sur le site web de GitHub, naviguez dans votre dépôt.

## VII. Étape 6: Clonage, Pull et Push



- Étape N°26 : Clonez un dépôt d'exemple depuis GitHub vers un nouveau dossier sur votre machine.
- Étape N°27 : Modifiez un fichier et poussez ces modifications vers GitHub.
- Étape N°28 : Simulez une modification "d'un collègue" en éditant un fichier directement depuis l'interface de GitHub.
- Étape N°29 : Dans votre dépôt local, récupérez (`pull`) ces modifications.
- Étape N°30 : Vérification : Comparez le contenu local avec le contenu sur GitHub.

## VIII. Aide-mémoire

- Étape N°31 : En utilisant le cours et vos recherches sur Internet, complétez ce tableau

Action	Commande Git
Initialiser un nouveau dépôt	

Action	Commande Git
Cloner un dépôt	
Ajouter des changements	
Commiter des changements	
Voir l'historique des commits	
Voir l'état des modifications	
Brancher	
Passer à une branche	
Créer et passer à une branche	
Fusionner une branche dans la branche courante	
Supprimer une branche	
Pousser les changements vers un dépôt distant	
Tirer les changements depuis un dépôt distant	
Configurer un dépôt distant	
Liste des dépôts distants	
Supprimer un dépôt distant	
Stasher des changements	
Appliquer des changements stashés	
Voir la liste des changements stashés	
Supprimer les changements stashés	
Réinitialiser la HEAD au dernier commit	
Réinitialiser un fichier à son état précédent	

	GIT	
	TP D'INITIATION	