

# Créer des branches avec Git/GitHub – Synthèse

❖ Étape / Thème	Résumé & explications	██ Commandes associées
❖ 1. Outils et notions de base	Git = système de gestion de versions local. GitHub/GitLab = plateformes collaboratives (hébergement, CI/CD).	—
❖ 2. Installation de Git (Windows)	1. Aller sur <a href="https://git-scm.com">https://git-scm.com</a> 2. Suivre la doc officielle (commandes, configuration) 3. Authentification à 2 facteurs recommandée 4. <b>Option:</b> installer Winget pour gérer les paquets Windows	—
✓ 3. Vérifier installation	Vérifie que Git est bien installé	- <code>git --version</code>
📁 4. Créer un dossier de travail local	Crée un dossier, se place dedans, puis visualise le contenu	- <code>mkdir git_tuto</code> - <code>cd git_tuto</code> - <code>ls</code>
🔗 5. Cloner un dépôt distant	Cloner un repo existant depuis GitHub (ou GitLab). Option : renommer le dossier à la volée.	- <code>git clone &lt;URL&gt;</code> - <code>git clone &lt;URL&gt; nouveau_nom</code>
🔒 6. Sécuriser la communication Git/GitHub (ou GitLab) avec SSH	**Optionnel mais recommandé** : utiliser une paire de clés SSH au lieu de HTTPS. Permet une connexion + rapide et sécurisée.	- <code>ssh-keygen -t ed25519 -C "ton@email.com"</code> - <code>cat ~/.ssh/Nom_clé_publique</code>  <i>Ajouter la clé sur GitHub &gt; Settings &gt; SSH &amp; GPG keys</i>
📦 7. Créer un dépôt local (init)	1. Initialise un dépôt Git local 2. Ajoute un fichier (README.md par ex.) 3. Prépare le commit initial	- <code>git init</code> - <code>touch README.md</code> - <code>git add README.md</code> - <code>git commit -m "Initial commit"</code>
🔗 8. Lier le dépôt local à GitHub	1. Créer un **nouveau repo vide sur GitHub** via le bouton  > New repository 2. Copier l'URL SSH ou HTTPS 3. Lier à Git local + push	- <code>git remote add origin &lt;URL&gt;</code> - <code>git branch -M main</code> - <code>git push -u origin main</code>
🚫 9. Fichiers à ignorer dans Git	Créer un fichier ` `.gitignore` pour exclure fichiers sensibles ou inutiles (ex: ` `.env` , ` `.DS_Store` , etc.)	- <code>touch .gitignore</code>
👉 10. Comprendre les branches	- ` `main` = branche principale - Chaque branche = nouvelle fonctionnalité isolée - Git Flow = schéma logique pour organisation du code - Fusion via ` `merge` une fois la fonctionnalité prête	—
🔍 11. Vérifier les branches existantes	Affiche la liste des branches locales existantes	- <code>git branch</code>
👉 12. Créer une nouvelle branche (` `develop` )	Création + bascule directe sur une nouvelle branche ` `develop` à partir de ` `main`	- <code>git checkout -b develop</code>
➕ 13. Ajouter un fichier (ex: README)	Préparation de fichiers pour le commit	- <code>git add README.md</code>
📝 14. Commit des modifications	Crée un commit clair pour la création de la branche	- <code>git commit -m "Ajout de la branche develop"</code>
👉 15. Pousser une branche vers le dépôt distant	Pousse la nouvelle branche ` `develop` vers GitHub	- <code>git push -u origin develop</code>

## Script Git/GitHub – Clonage, Création de Dépôt, Branche & Push

```
bash
# Étape 1 : Vérifier si Git est installé
git --version

# Étape 2 : Créer un dossier de travail et s'y rendre
mkdir git_tuto
cd git_tuto

# Étape 3 : Initialiser un dépôt Git local
git init

# Étape 4 : Créer un fichier README
echo "# Mon projet" > README.md

# Étape 5 : Ajouter le fichier au suivi Git
git add README.md

# Étape 6 : Commit initial
git commit -m "Initial commit avec README"

# Étape 7 : Créer un nouveau dépôt distant sur GitHub (via interface web)
# → Aller sur GitHub > "+" > New repository > Nom : git_tuto > Create

# Étape 8 : Lier le dépôt local au dépôt distant (remplacer par votre propre URL)
git remote add origin git@github.com:mon-utilisateur/git_tuto.git

# Étape 9 : Pousser le projet sur GitHub (branche principale)
git push -u origin main

# Étape 10 : Vérifier les branches existantes
git branch

# Étape 11 : Créer une nouvelle branche (ex: develop) et basculer dessus
git checkout -b develop

# Étape 12 : Ajouter un nouveau fichier dans la branche develop
echo "Nouvelles fonctionnalités en développement" > develop.txt
git add develop.txt
git commit -m "Ajout fichier develop.txt dans la branche develop"

# Étape 13 : Pousser la branche develop vers GitHub
git push -u origin develop

# Étape 14 : (Optionnel) Voir l'état actuel du projet
git status

# Étape 15 : Lister les branches locales
git branch

# Étape 16 : Lister les branches distantes
git branch -r
```

## Bonus : Générer et configurer une clé SSH (si pas encore fait)

bash

```
# Générer une paire de clés SSH (laisser le chemin par défaut)
ssh-keygen -t ed25519 -C "votre.email@example.com"

# Lancer l'agent SSH
eval "$(ssh-agent -s)"

# Ajouter la clé privée à l'agent
ssh-add ~/.ssh/id_ed25519

# Copier la clé publique
cat ~/.ssh/id_ed25519.pub
# ⌂ Puis aller sur GitHub > Settings > SSH and GPG Keys > New SSH Key > coller la clé
publique
```