

Utiliser Git en ligne de commande

Voici une **page rapide** qui résume les commandes essentielles pour démarrer avec Git en ligne de commande. Elle peut te servir de mémo.

5TTR

6TTR

6TQ

 Découverte

Git : les commandes de base

1. Initialiser un dépôt

```
1 | git init
```

👉 Crée un nouveau dépôt Git dans le dossier courant. (Un dossier caché `.git` est créé, dont tu ne dois pas t'occuper)

2. Ajouter des fichiers à la zone de staging

```
1 | git add nom_du_fichier
```

ou pour tout ajouter :

```
1 | git add .
```

👉 Prépare les fichiers modifiés pour le prochain commit.

3. Enregistrer un commit

```
1 | git commit -m "Message clair"
```

👉 Sauvegarde les fichiers **ajoutés au staging** avec un message qui décrit les changements.

4. Envoyer vers GitHub (ou un autre dépôt distant)

```
1 | git push -u origin main
```

👉 Envoie les commits vers la branche `main` du dépôt distant `origin`.

- `-u` permet de créer le lien entre ta branche locale et la branche distante (à faire une seule fois).
- Après ça, un simple `git push` suffit.

5. Voir l'historique

```
1 | git log
```

👉 Affiche l'historique complet des commits (auteur, date, message).

Pour un affichage plus compact et visuel :

```
1 | git log --oneline --graph --all
```

Résumé visuel du flux

1. **Travailler** → modifier tes fichiers (working directory).
2. **Préparer** → `git add` (staging area).
3. **Sauvegarder** → `git commit` (repository local).
4. **Partager** → `git push` (repository distant).

👉 À retenir :

- `init` → je démarre un projet.
- `add` → je prépare les fichiers.
- `commit` → je sauvegarde.
- `push` → j'envoie sur GitHub.
- `log` → je consulte l'historique.

Parfait 👉 voici une fiche mémo **format A4** simple et claire pour tes élèves :

Mémo Git – Commandes de base

Commande	Rôle	Exemple
<code>git init</code>	Crée un nouveau dépôt Git dans le dossier courant.	<code>git init</code>
<code>git add fichier</code>	Ajoute un fichier à la zone de staging (préparation du commit).	<code>git add index.html</code>
<code>git add .</code>	Ajoute tous les fichiers modifiés au staging.	<code>git add .</code>
<code>git commit -m "message"</code>	Enregistre un snapshot (sauvegarde) des fichiers préparés.	<code>git commit -m "Ajout page contact"</code>
<code>git push -u origin main</code>	Envoie la branche locale sur GitHub (premier envoi, crée le lien).	<code>git push -u origin main</code>
<code>git push</code>	Envoie les nouveaux commits sur GitHub (si le lien existe déjà).	<code>git push</code>
<code>git log</code>	Affiche l'historique complet des commits.	<code>git log</code>
<code>git log --oneline --graph --all</code>	Affiche un historique simplifié et visuel.	<code>git log --oneline --graph --all</code>

Cycle de travail avec Git

1. **Modifier tes fichiers** → (working directory).
2. **Préparer** avec `git add` → (staging area).
3. **Enregistrer** avec `git commit` → (repository local).
4. **Partager** avec `git push` → (repository distant sur GitHub).

🔴 À retenir :

- `init` → créer un projet Git.
 - `add` → choisir les fichiers à sauvegarder.
 - `commit` → enregistrer une version.
 - `push` → envoyer vers GitHub.
 - `log` → voir l'historique.
-

Vidéo à la une à regarder sur Youtube:  <http://youtu.be/rE2zRhZdjFU>