

## Partie I – Mise en place des structures

### Git init :

Contexte : Initialisation du projet avant d'écrire le code.

Commande utilisée : ``git init``

Effet produit : Création d'un dépôt Git local vide pour suivre l'évolution du projet.

```
PS C:\Users\slypo\Documents\TP VOITURE> git init
Initialized empty Git repository in C:/Users/slypo/Documents/TP VOITURE/.git/
PS C:\Users\slypo\Documents\TP VOITURE> █
```

### Git add. / git commit -m :

Contexte : utilise git commit a chaque nouvelle fonction que j'ajoute au code initial

```
PS C:\Users\slypo\Documents\TP VOITURE> git add .
PS C:\Users\slypo\Documents\TP VOITURE> git commit -m " feat : Structure + fonction Print info voiture"
[master (root-commit) e7e8b93] feat : Structure + fonction Print info voiture
1 file changed, 34 insertions(+)
create mode 100644 main.go
PS C:\Users\slypo\Documents\TP VOITURE> █
```

## Partie II – Méthodes et évolutions

### git status :

Contexte : Vérifier les fichiers modifiés avant un commit.

Commande ``git status``

Effet produit : Affiche les fichiers modifiés, ajoutés ou non suivis.

```
PS C:\Users\slypo\Documents\TP VOITURE> git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
PS C:\Users\slypo\Documents\TP VOITURE> █
```

### git diff :

Contexte : Voir les différences avant un commit.

Commande : ``git diff``

Effet produit : Affiche les changements ligne par ligne.

### **git log :**

Contexte : Voir l'historique des commits.

Commande : ``git log``

Effet produit Affiche tous les commits avec leurs messages, dates et auteurs.

```
PS C:\Users\slypo\Documents\TP VOITURE> git log
commit b04b1ff53f1239cf8e85639f640ac234769b060d (HEAD -> master)
Author: Slypot <sylvain.pottier@ynov.com>
Date: Tue Sep 23 12:05:15 2025 +0200

    Ajout item coffre et affichage infos voiture

commit 7e0f533fa55c40f203967a41ff1fa35dc310a209
Author: Slypot <sylvain.pottier@ynov.com>
Date: Tue Sep 23 12:03:56 2025 +0200
```

## **Partie III – Gestion des erreurs et publication**

### **git reset :**

Contexte : Retour à l'état du dernier commit après une erreur.

Commande : ``git reset --hard HEAD``

Effet produit : Toutes les modifications non sauvegardées sont perdues.

```
PS C:\Users\slypo\Documents\TP VOITURE> git reset --hard HEAD
HEAD is now at b04b1ff Ajout item coffre et affichage infos voiture
PS C:\Users\slypo\Documents\TP VOITURE> 
```

### **git revert :**

Contexte : Supprimer un commit problématique sans casser l'historique.

Commande : ``git revert <hash_du_commit>``

Effet produit : Crée un nouveau commit qui annule le commit ciblé.

### **git rebase -i :**

Contexte : Nettoyer l'historique avant de publier le projet.

Commande : ``git rebase -i HEAD~3``

Effet produit : Permet de modifier les messages de commit ou de fusionner des commits.

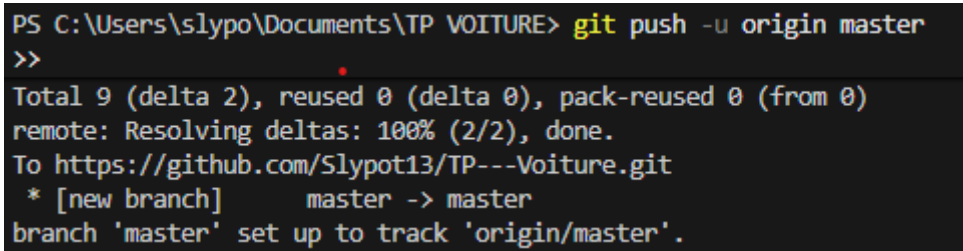
### **git remote add & git push :**

Contexte : Publication du projet terminé sur GitHub.

Commandes :

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

- ``git push -u origin master``



```
PS C:\Users\slypo\Documents\TP VOITURE> git push -u origin master
>>
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Slypot13/TP---Voiture.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Effet produit : Le dépôt local est maintenant lié à GitHub et le projet est visible en ligne.

*J'ai eu problème de chargeur donc je n'ai pas eu le temps de pouvoir avoir tout les screens, car j'avais quelques erreurs, mais pas le temps de tout refaire.*