

## 1. Synchroniser la branche develop locale avec le dépôt distant

git fetch origin && git checkout develop && git pull origin

- **Explication** : Cette commande met à jour la branche develop locale avec les dernières modifications depuis le dépôt distant. Cela garantit que vous commencez avec une version à jour.
- 

## 2. Passer à une branche de fonctionnalité

git checkout <nom-branche>

- **Explication** : Permet de passer à une branche où vous allez travailler. Si la branche n'existe pas encore localement, vous pouvez la créer et la relier au dépôt distant.
- 

## 3. Fusionner develop dans la branche de fonctionnalité

git merge develop

- **Explication** : Cela apporte les dernières modifications de develop dans votre branche actuelle. Vous travaillez ainsi avec une version à jour, réduisant les risques de conflits plus tard.
- 

**Note** : Tu fais ton travail, puis lorsque tu as fini, tu passes à ces commandes.

## 4. Ajouter et valider les modifications

git add .

git commit -m "Message décrivant votre travail"

- **Explication** :
    - git add . : Ajoute tous les fichiers modifiés ou nouveaux au suivi de Git.
    - git commit -m "Message" : Enregistre un commit avec un message décrivant les modifications.
- 

## 5. Pousser la branche de fonctionnalité

git push origin

- **Explication** : Envoie vos modifications locales vers le dépôt distant pour les partager avec l'équipe.
- 

## 6. Fusionner une branche de fonctionnalité dans develop

git checkout develop && git pull origin develop

git merge <nom-branche>

git push origin

- **Explication :**

- Passez à la branche develop pour intégrer vos modifications.
- Mettez à jour develop avec les dernières modifications depuis le dépôt distant.
- Fusionnez votre branche de fonctionnalité dans develop.
- Enfin, poussez develop à jour sur le dépôt distant.

**Note : Toutes ces étapes sont à faire chaque fois que tu travailles sur une branche ou que tu fais une modification du code.**