

## TP Git

- En ligne de commande :

- 1) Créer un dépôt Git « TpCLI » avec un fichier de code.
- 2) Modifier ce fichier et ajouter un nouveau fichier puis vérifier ces modifications via Git.
- 3) Indexer ces fichiers puis commit avec le message « Ajout d'un fichier et modification d'un fichier ».
- 4) Afficher l'historique.
- 5) Créer une nouvelle branche (vérifier la création avec une commande), modifier un des fichiers et commit ce changement. Ensuite, fusionner cette branche vers la branche principale tout en vérifiant au fur et à mesure l'historique pour confirmer les changements.

- Avec GitLab et Fork :

- 6) Créer un dépôt Git sur GitLab « TpGUI » puis cloner sur Fork.
- 7) Avec Fork, ajouter 1 fichier (fichier1) et le déposer dans le dépôt distant.
- 8) Ajouter un Gitignore afin de ne pas prendre en compte les fichiers texte.
- 9) Ajouter un fichier texte dans le dépôt local et vérifier que celui-ci n'est pas considéré par Git.
- 10) Avec GitLab, créer une nouvelle issue « Nouveau fichier à ajouter » et la relier à une nouvelle branche Feature/Modif1 et une merge request. Puis vérifier la bonne création sur Fork.
- 11) Avec Fork, se placer sur cette branche, ajouter un second fichier puis envoyer sur le dépôt distant.
- 12) Avec GitLab, effectuer le processus de fusion de la branche vers la branche principale. Vérifier sur Fork.
- 13) Avec GitLab, créer une nouvelle issue « Modification de commentaires » et la relier à une nouvelle branche Feature/Modif2 et une merge request. Puis vérifier la bonne création sur Fork.

- 14) Avec Fork, se placer sur cette branche, ajouter un commentaire « Nouveau commentaire » sur la première ligne du fichier fichier1 puis envoyer sur le dépôt distant.
- 15) Avec Fork, se placer sur la branche principale et ajouter une ligne vide au début du fichier fichier1 puis envoyer sur le dépôt distant
- 16) Avec GitLab, effectuer le processus de fusion de la branche vers la branche principale. Pourquoi cela ne fonctionne pas ? Essayer de le faire via Fork. Est-ce possible ?
- 17) En faisant le bon choix, via Fork faire le nécessaire pour fusionner la branche et créer un tag sur ce dernier commit. Vérifiez ensuite sur GitLab.
- 18) Avec GitLab, créer une nouvelle issue « Nouvelle fonctionnalité » et la relier à une nouvelle branche Feature/Modif3 et une merge request. Puis vérifier la bonne création sur Fork.
- 19) Avec Fork, sur la branche principale, effectuer un nouveau commit et envoyer le sur le dépôt distant.
- 20) Le but étant de faire cette modification sur la branche Feature/Modif3, sans modifier le fichier et en utilisant exclusivement Fork, annuler ce changement sur cette branche pour le mettre ensuite sur la branche Feature/Modif3.
- 21) Fusionner cette branche dans la branche principale et créer un nouveau tag. Vérifier sur GitLab/Fork.
- 22) Effectuer un nouveau clone dans un autre emplacement du PC (avec Visual Studio par exemple) et depuis ce dépôt local, effectuer et envoyer sur le dépôt distant une modification.
- 23) En repassant sur Fork et le premier dépôt local, vérifier que cette dernière modification n'est pas visible puis effectuer les actions nécessaires pour récupérer cette modification dans ce dépôt.
- 24) Créer des modifications différentes sur un même fichier dans les 2 dépôts puis envoyer les modifications du premier dépôt sur le dépôt distant tout en gardant en local les modifications du second dépôt (sans faire de commit). Sur le second dépôt, récupérer les modifications déposées par le premier dépôt. Est-ce possible ? Quelle est la solution pour récupérer ces modifications sans faire de commit en gardant ces modifications ? L'appliquer et vérifier.