

Contrôle de versions: Git

Exercice 1. Création et initialisation du repository local

- a) Créez un dossier nommé **TPgit** dans votre dossier personnel (ou dans un sous-dossier de votre dossier personnel). Le dossier TPgit ne doit *absolument pas* être placé dans le dossier que vous avez utilisé pour le dernier TP sur SVN.
- b) Dès le dossier TPgit, écrivez **git status**. Git vous informera que le dossier n'est pas un repository. Cela veut dire que git n'a pas encore créé le repository local.
- c) Vous êtes maintenant prêts à créer votre premier repository: tapez **git init** et vérifiez que le repository a été bien créé avec la commande **git status**.

Exercice 2. Utilisation du repository local

- a) Sauvegardez dans le répertoire TPgit les fichiers `login.info` et `groupe.info` que vous avez utilisé pour le TP sur Subversion.
- b) Essayez de faire un commit. Vérifiez que les fichiers ne sont pas sous contrôle de version avec **git status**.
- b) Mettez les fichiers sous contrôle de version (**git add <nom_fichier>**) et faites commit (**git commit**). Vous pouvez aussi faire les deux opérations avec une seule commande avec **git commit -a**.
- c) Vérifiez que la branche sur laquelle vous êtes en train de travailler c'est bien la branche "master" (rappel: **git branch**). Maintenant c'est l'heure de créer une nouvelle branche. Créez une branche appelée "testing".
- d) Maintenant changez de branche (rappel: **git checkout <nom_branche>**) vers "testing". Copiez ici les fichiers `login.info` et `groupe.info`. Modifiez `groupe.info` en y rajoutant une ligne "testing" à la fin. Ouvrez le fichier de `login.info` et changez l'ordre des informations.
- e) Vérifiez l'état du repository (encore une fois, **git status**). Mettez les fichiers que vous venez de rajouter sous contrôle de version. Faites commit. Vérifiez l'état du repository.
- f) Changez maintenant de branche vers "master". Vérifiez le contenu du repository. Changez encore vers "testing". Vérifiez le contenu du repository (vous allez voir que les fichiers changent sous vos nez).

g) Maintenant notre testing est à point, c'est l'heure de faire une fusion de branches (merge). Faites le merge entre master et testing (se positionner dans la branche master et exécuter `git merge testing`). Vérifiez le résultat du merge (et le contenu des fichiers).