

TP git

Préambule

L'objectif principal de ce TP est de travailler avec git :

- sur la gestion de branches en local
- à plusieurs en réseau

1 Les branches

1.1 Créer un projet git

- créer un répertoire dir
- dans ce répertoire, créer 2 fichiers file1.txt et file2.txt (non vides)
- initialiser le dépôt git (*git init*)
- propager en s'assurant que les 2 fichiers sont sur le dépôt (*git add, git commit*)

1.2 Travailler avec git en local

- modifier file1.txt
- ajouter file3.txt
- propager les modifications
- modifier file.txt
- propager les modifications
- voir l'historique des modifications

1.3 Créer une branche idee et travailler dessus

- créer une branche 'idee'
- sur cette branche :
 - modifier file1.txt
 - ajouter file4.txt
 - propager (sur cette branche)
 - voir l'historique

1.4 Travailler aussi sur la branche master

En parallèle, on travaille aussi sur le branche principale :

- sur la branche principale
 - modifier f2.txt
 - propager
 - créer file5.txt
 - propager
 - voir l'historique

1.5 Voir l'état de chaque branche

- voir l'état du projet sur chacune des branches, les différences, ...

1.6 Fusionner la branche idee sur la branche master

- Maintenant, on suppose que ce qui a été développé sur chacune des branches est à conserver, on va donc fusionner la branche idee sur la branche master
- S'assurer que les modifications réalisées sur les 2 branches ont été conservées
- Voir l'historique

2 Travailler à plusieurs en réseau

Cette partie doit être réalisé à plusieurs (au moins 3).

2.1 Création des projets : init, clone

- user1 :
 - télécharge tp_svn.tgz, le décompresse
 - mv tp_svn tp_git
 - initialise le projet (*git init, git add, git commit*)
- user2 :
 - clone le projet de user 1
- user3 :
 - clone le projet de user1

2.2 Chaque développeur travaille sur sa partie

- user1 : ajoute des prénoms (data/prenoms.txt)
- user2 : ajoute des mois (data/mois.txt)

- user3 : créé un nouveau fichier (jours.txt) dans lequel il ajoute les jours de la semaine

Chaque utilisateur propage en local ses modifications

2.3 Il est temps faire profiter les autres de son travail

- user3 a fini son travail, les autres vont récupérer ce qu'il a fait
 - user1 et user 1
 - récupèrent les modifications de user3 (*git pull*)
- user3 souhaite aussi avoir les modifications de user1

Quel est l'état des différents projets ?

Y a-t-il un projet intégrant les modifications de chacun ?

2.4 Continuer le développement à plusieurs