Préface

Git est un programme gestionnaire de versions permettant de construire et maintenir un historique de toutes les manipulations effectuées sur un dépôt de données.

Ce type de programme est particulièrement utile dans le cadre de projet de groupe puisqu'il permet :

- à plusieurs personnes de travailler simultanément sur un projet,
- de créer et fusionner des branches à un projet,
- de visualiser l'évolution d'un projet et
- de comparer les différentes versions d'un projet.

Installation

Plusieurs outils et services sont offerts pour la gestion et l'utilisation de dépôts de données Git : http://git-scm.com

Configurations

Plusieurs paramètres de l'environnement Git sont configurables.

Identification

Afin de pouvoir identifier les manipulations effectuées sur des dépôts de données Git :

```
git config --global user.name ‘Prénom Nom”
git config --global user.email ‘courriel@hote.com”
```

Coloration

```
git config --global color.ui true
```

SSH

L'une des méthodes utilisées par Git pour communiquer avec les serveurs de dépôts de données est le protocole SSH. Ce dernier permet une connexion sécurisée utilisant un système de chiffrement à clés publique / privée.

Afin de générer des clés RSA utilisables par le protocole SSH :

```
ssh-keygen -t rsa -C ‘courriel@hote.com”
```

Afin de démarrer le gestionnaire de clés :

```
eval $(ssh-agent -s)
```

Afin d'ajouter une clé privée au gestionnaire :

```
ssh-add NomClePrivee
```

Les clés publiques devront être transmises aux serveurs de dépôts de données Git.

Dépôt de données

Un dépôt de données Git n'est rien d'autre qu'une structure, dans le dossier caché « .git », contenant l'historique des manipulations de données du dépôt.

Initialisation

Afin d'initialiser un dépôt de données Git :

```
git init
```

Clonage

Afin de récupérer un dépôt de données Git depuis un serveur distant :

```
git clone utilisateur@hote.com:Dossier/DepotDonnees.git
```

Suite à un clonage, l'adresse du dépôt de données utilisée est nommée « origin ».

Mise à jour locale

Afin de mettre à jour un dépôt de données Git local depuis un serveur de dépôts de données Git distant :

```
git pull origin Branche
```

Mise à jour distante

Afin de mettre à jour un dépôt de données Git distant depuis un dépôt de données Git local :

```
git push origin Branche
```

Statut

Afin de consulter le statut actuel d'un dépôt de données Git :

```
git status -s
```

?? : Inconu du dépôt de données
A : Ajouter au dépôt de données
M : Modifier au dépôt de données
D : Retirer du dépôt de données

Ajouter

Afin d'ajouter des dossiers et fichiers à un dépôt de données Git :

```
git add Fichiers
```

Ignorer

Il est possible d'ignorer certains fichiers d'un dépôt de données Git local en les inscrivant dans un fichier « .gitignore ».

Déplacer

Afin de déplacer des dossiers et fichiers d'un dépôt de données Git :

```
git mv Source Destination
```

Retirer

Afin de retirer des dossiers et fichiers d'un dépôt de données Git :

```
git rm Fichiers
```

Versions

Les différentes versions d'un dépôt de données sont délimitées par les enregistrements commis.

Commettre

Afin d'enregistrer l'état actuel du dépôt de données :

```
git commit -am "Message"
```

Afin d'amender le dernier enregistrement :

```
git commit --amend -m "Message"
```

Journal

Afin d'afficher la liste des enregistrements :

```
git log
```

Comparer

Afin d'afficher les manipulations effectuées entre deux versions :

```
git diff IDEnregistrement1 IDEnregistrement2
```

Retour

Afin d'annuler les manipulations d'une version :

```
git revert IDEnregistrement
```

Branches

Il est possible de diviser un dépôt de données Git en plusieurs branches qui pourront, plus tard, être fusionnées. Par défaut, la branche principale d'un dépôt de données est nommée « master ».

Lister

Afin d'afficher toutes les branches :

```
git branch -a
```

Embrancher

Afin de changer de branche :

```
git checkout NomBranche
```

Créer

Afin de créer une branche :

```
git checkout -b NomBranche
```

Supprimer

Afin de supprimer une branche localement :

```
git branch -d NomBranche
```

Et, afin de supprimer la branche sur le serveur de dépôt de données :

```
git push origin :NomBranche
```

Fusionner

Afin de fusionner une branche :

```
git merge NomBranche
```