

git

Alexis Nasr alexis.nasr@univ-amu.fr

31 juillet 2020



Système de gestion de version décentralisé

- Logiciel permettant de stocker un ensemble de fichiers sur un serveur
- Conserve la chronologie des modifications effectuées
- Permet de retrouver les différentes versions des fichiers
- Permet de travailler à plusieurs sur un projet sans trop se marcher sur les pieds et sans s'envoyer de fichiers par mail !

- **Répertoire de travail** : ensemble de fichiers sur votre disque dur qui constituent votre projet.
- **Dépôt local** : historique de toutes les modifications apportées aux fichiers qui constituent le répertoire de travail. Ces informations sont stockées sur votre disque, dans le répertoire `.git`
- **Dépôt distant** : historique de toutes les modifications stockées sur le serveur.

- A partir d'un dépôt existant
`git clone [url]`
à la suite du clone, `git` associe au dépôt distant le nom `origin`
- A partir d'un répertoire de travail
`git init [nom-du-projet]`

- Ajout d'un fichier
`git add [fichier]`
- Elimination d'un fichier
`git rm [fichier]`
- Renommage d'un fichier
`git mv [ancien-nom] [nouveau-nom]`

Synchronisation du dépôt local et du dépôt distant

- Synchroniser le dépôt local à partir du dépôt distant
`git pull`
- Enregistrer les modifications dans le dépôt local
`git commit`
- Synchronisation du dépôt distant à partir du dépôt local
`git push`

- Affichage de l'état général du dépôt local

```
git status
```

affiche la liste des fichiers non suivis, des fichiers suivis dont la version locale et la version distante diffèrent

- Affichage de l'état d'un fichier

```
git diff [fichier]
```

affiche les différences entre la version locale d'un fichier et sa version distante

- Affichage de l'historique des versions

```
git log
```

Cycle de travail classique

- synchronisation du dépôt local
`git pull`
- création du fichier `nouveauFichier`
- ajout de `nouveauFichier` au dépôt local
`git add nouveauFichier`
`git commit -m "création de NouveauFichier"`
- modification de `ancienFichier`
`git add ancienFichier`
`git commit -m "modification de AncienFichier"`
- synchronisation du dépôt distant
`git push`

- Le dépôt `https://etulab.univ-amu.fr/nasr/programmation2` contient les morceaux de code que l'on vous fournit pour les différents TP
- Vous pouvez faire un **fork** de ce dépôt à partir de l'interface web
- A la suite du **fork**, un nouveau dépôt est créé, dont l'url est la suivante :
`https://etulab.univ-amu.fr/nomId/programmation2`
où `monId` est votre identifiant etulab
- Ce dépôt est une copie de dépôt d'origine
- Vous pouvez maintenant faire un clone de votre projet
`git clone https://etulab.univ-amu.fr/nomId/programmation2`

- Il est possible (et conseillé) de garder un lien entre votre nouveau dépôt et le dépôt d'origine :
- `git remote add base`
`https://etulab.univ-amu.fr/nasr/programmation2`
- `base` est le nom du nouveau dépôt distant
- vous pouvez visualiser les dépôts distants de votre dépôt local :
- `git remote -v`
- Pour récupérer les modifications du dépôt `base`, il faut faire :
- `git pull base master`