

Organisation d'un projet AVEC GITHUB

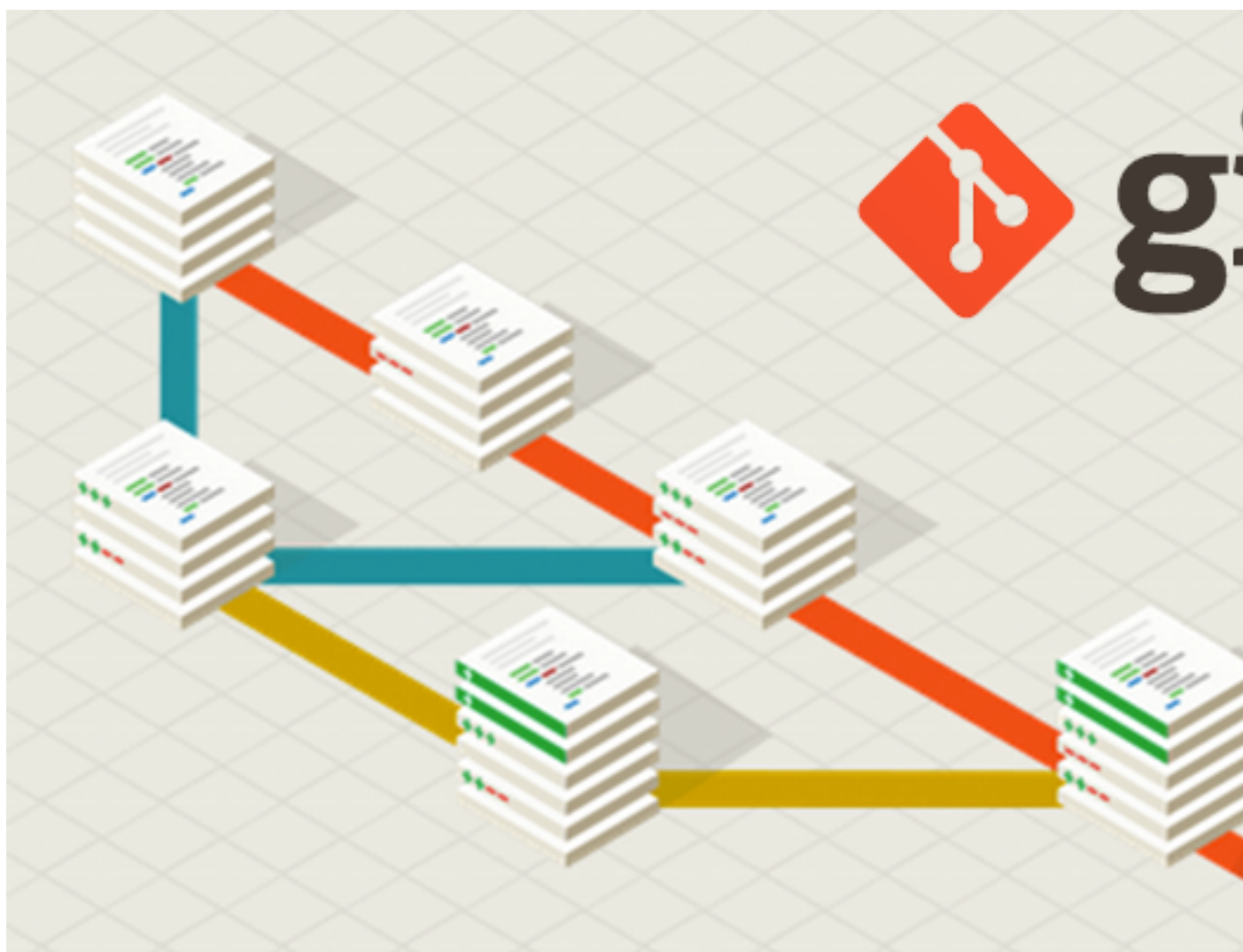
2017-10

0.1 Objectifs

Nous avons maintenant besoin de vos recommandations au sujet de l'organisation de notre projet d'édition des archives de l'Oulipo. Comme vous le savez maintenant, le projet consiste à éditer un ensemble de compte-rendus de réunions de l'Ouvroir de littérature potentielle. Les promoteurs du projet considèrent que cet ensemble constitue un tout cohérent. Les documents que vous avez déjà transcrits appartenant à cet ensemble.

Comment pouvons nous les organiser avec la TEI pour, tout en traduisant cette organisation, assurer une méthode de travail adaptée pour continuer le travail d'édition qui implique plusieurs personnes.

0.2 Quelques mots sur GitHub



0.2.1 Le paradigme du 'versioning'

diff et wikipédia

Code par nature mouvant, obsolescence

Corrections et versions. Développement de système pour garder la trace des changements. Crucial pour le développement informatique.

Plusieurs protocoles, SVN Subversion, etc. Git.

0.2.2 Le protocole Git

Finalelement assez stupide, une manière de

- stocker des arbres de contenus un peu comme un système de fichiers
- un commit crée un nouvel arbre accessible par son nom
- on peut changer le répertoire pour recréer n'importe quel arbre

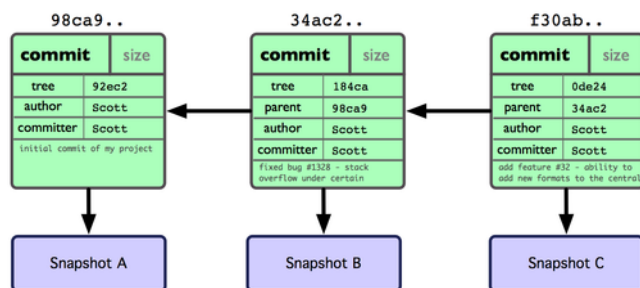
Conçu pour gérer le développement du noyau Linux (Linux Kernel)

- très efficient
- prend en charge les développements non linéaires gracieusement
- un système distribué

0.2.3 Ce qui différencie Git

- Commit, Sorte de clichés de l'historique.
- Git branching model
- Rebase
- Système distribué

0.3 Le commit



0.3.1 repo

Répertoire dans lequel suit les fichiers

0.3.2 Idée du Snapshots

- manière dont garde la trace de l'historique du code
- enregistrement de ce que contiennent tous les fichiers à un moment donné
- possibilité de revenir en arrière

0.3.3 Commit

Acte de créer un snapshot. Un projet consiste en une série de commit.

Trois éléments d'information

- informations sur les changements de fichier
- une référence au commit précédent (le parent-
- un hash code qui le désigne

0.4 Pourquoi utiliser un système de gestion de version ?

- *Sécuriser* votre travail d'encodage
- Assurer la *traçabilité* des fichiers
- Organiser la *collaboration* distante

En somme, c'est une bonne pratique !

0.5 Notre répertoire GitHub

<https://github.com/tei-fr/formationEnc2017-10>

- Consultez l'historique
- Voir les contributeurs

1 Instruction pour partager les fichiers de l'édition

- Dans le terminal rendez-vous dans votre dossier de la formation formationEnc2017-10
- Nous avons créé un répertoire destiné à l'édition intitulé `oulipoEd`
- Placez-y vos fichiers édités
- Poussez vos documents avec la séquence de commandes suivante

```
git add edition/*  
# ajouter tous les fichiers du répertoire à l'index git commit  
-m"ajout des fichiers d'édition" # création du commit git push #  
envoi des fichiers vers les serveur
```
- Consultez le répertoire et constatez le versement du travail des autres collaborateurs ('git pull')

2 Ajouter votre fichier au corpus

- Consultez le fichier `oulipoCorpus.xml` que nous avons ajouté au répertoire
- Ajoutez votre fichier au corpus en suivant l'exemple compris dans le fichier

2.1 Définir la place des autres documents

- Où mettre les images
- Index des personnes, lieux etc. `data`
- Transcriptions éditées `sources`
- Ajouts scientifiques (bibliographie, textes d'introduction etc.) `edition`
- Documentation, schema, ODD `meta`