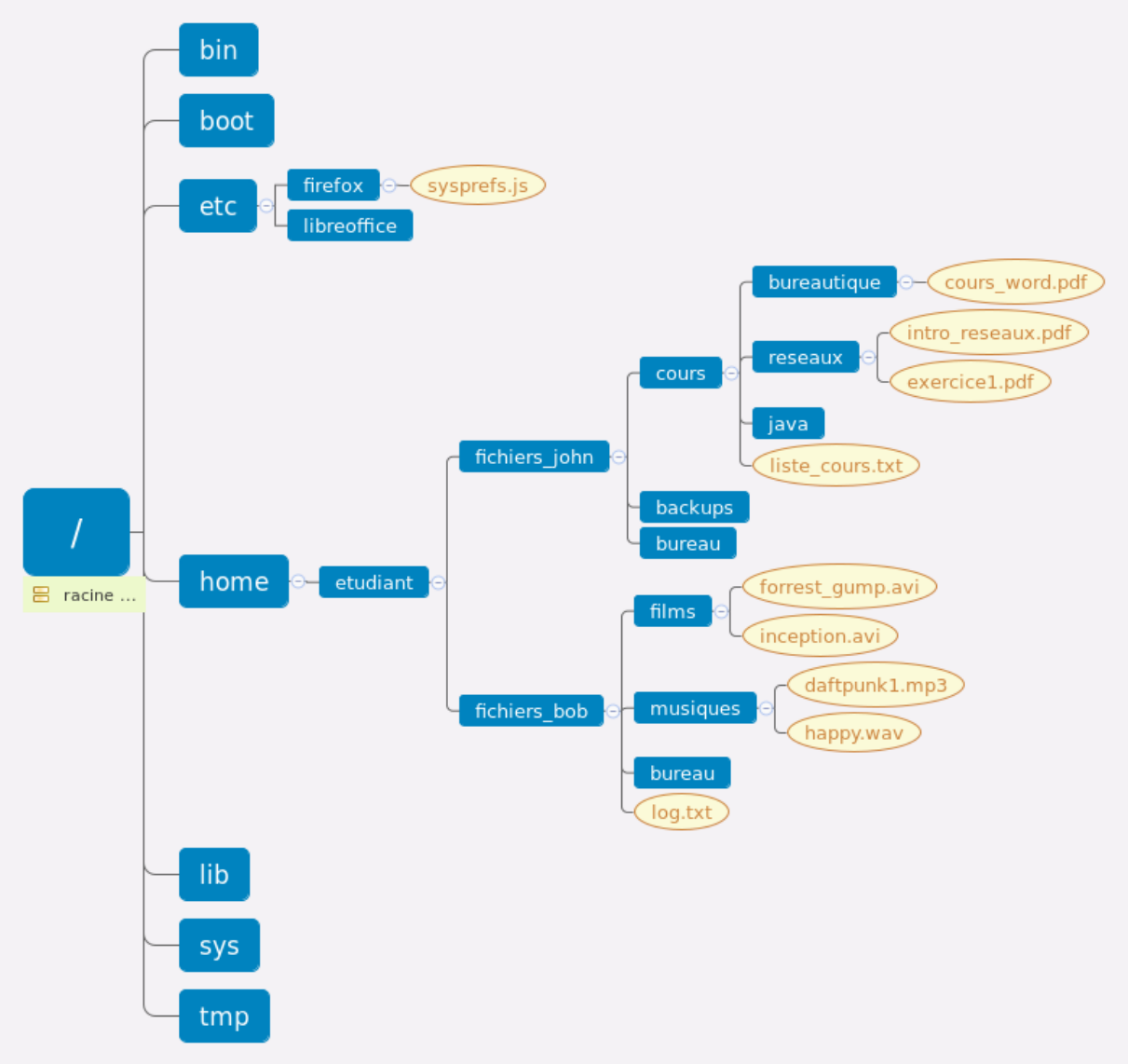
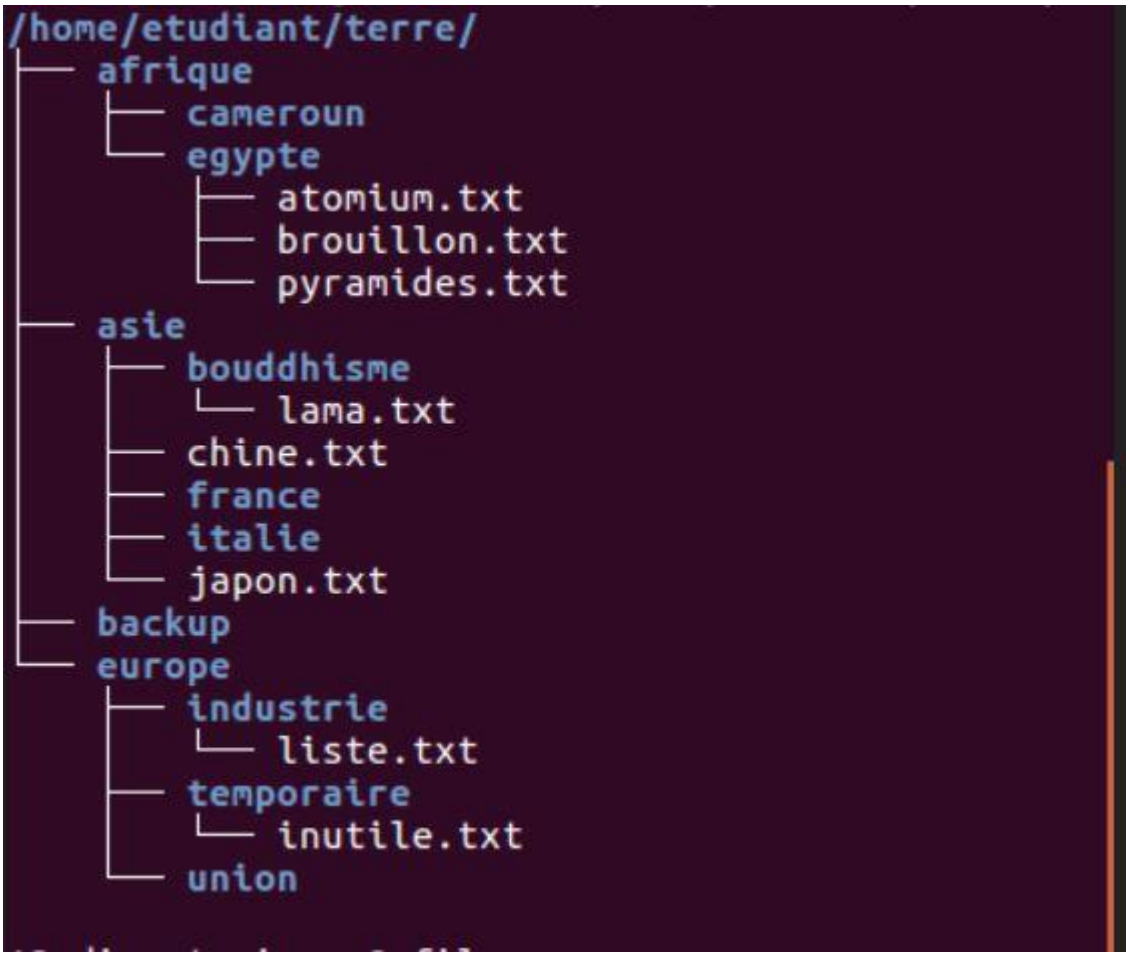
Exercice Bash et Git + théorie

1. 1. Vous allez créer un dossier "LinuxBash", en suite vous aller "Jour1, Jour2, Jour3, Jour4 et Jour5" avec une seule ligne de commande. Vous allez voir les permissions que vous avez dans chaque dossier.
2. [12:08]
Solution 1 : mkdir -p LinuxBash/{Jour1,Jour2,Jour3,Jour4,Jour5}
3. [12:10]
Solution 2 : mkdir -p LinuxBash/Jour{1..5}



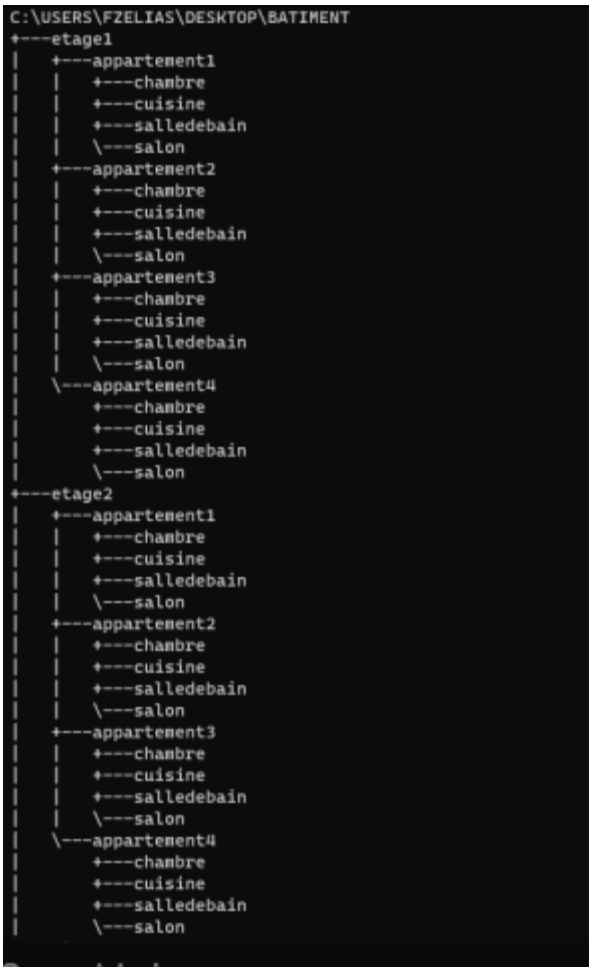
etudiant/{fichiers_john/{cours/{bureautique,reseaux,java},backups,bureau},fichiers_bob/{films,musiques,bureau}} && touch

etudiant/{fichiers_john/cours/{bureautique/cours_word.pdf,reseaux/{intro_reseaux.pdf,exercice1.pdf},list_cours.txt},fichiers_bob/{films/{forrest_gump.avi,inception.avi},musiques/{daftpunk1.mp3,happy.wav},log.txt}}



mkdir -p etudiant/terre/{afrique/{cameroun,egypte},asie/{bouddhisme,france,italie},backup,europe/{industrie,temporaire,union}} && touch etudiant/terre/{afrique/egypte/{atomium.txt,brouillon.txt,pyramides.txt},asie/{chine.txt,japon.txt,bouddhisme/lama.txt},europe/{industrie/liste.txt,temporaire/inutile.txt}}

Vous allez créer un dernier exercice dans votre bureau appelé "batiment" Dans le batiment vous allez créer 2 étages, et dans chaque étage il y a 4 appartements Dans chaque appartement il y a une cuisine, un salon, un salle de bain et une salle de bain Puis vous devez créer des fichiers comme tv.txt, lit.txt, frigo.txt dans le appartement1 du etage1, et puis vous devrez les déplacer dans différents endroits avec la commande mv



mkdir -p batiment/etage{1..2}/appartement{1..4}/{salledebain,chambre,cuisine,salon}

Pour vous exercer vous pouvez : Créer un projet avec une sauvegarde contenant (index.html et style.css) et dans l'index.html une balise h1 contenant Bonjour Une sauvegarde ou le fichier style.css est lier au fichier html (balise Link) Une nouvelle sauvegarde avec un fichier contact.html ,index.html et style.css, profitons en pour supprimer le h1 contenant Bonjour du fichier index.html Une sauvegarde ou le fichier contact.html a été renommer contactme.html, il Une dernière étape consiste à aller dans la sauvegarde ou vous aviez encore le h1 avec Bonjour, copier le morceau de code, de revenir sur la dernière sauvegarde coller le morceau

de code dans index et dans contact et voila vous avez terminez ** Je sais bien que ce serait plus simple d'écrire la balise h1 soit même plus tôt que de retourner dans une ancienne sauvegarde pour copier cette balise , puis retourner dans la toute dernière et coller, mais ca fait parti du jeu de faire des choses comme ça pour mieux retenir

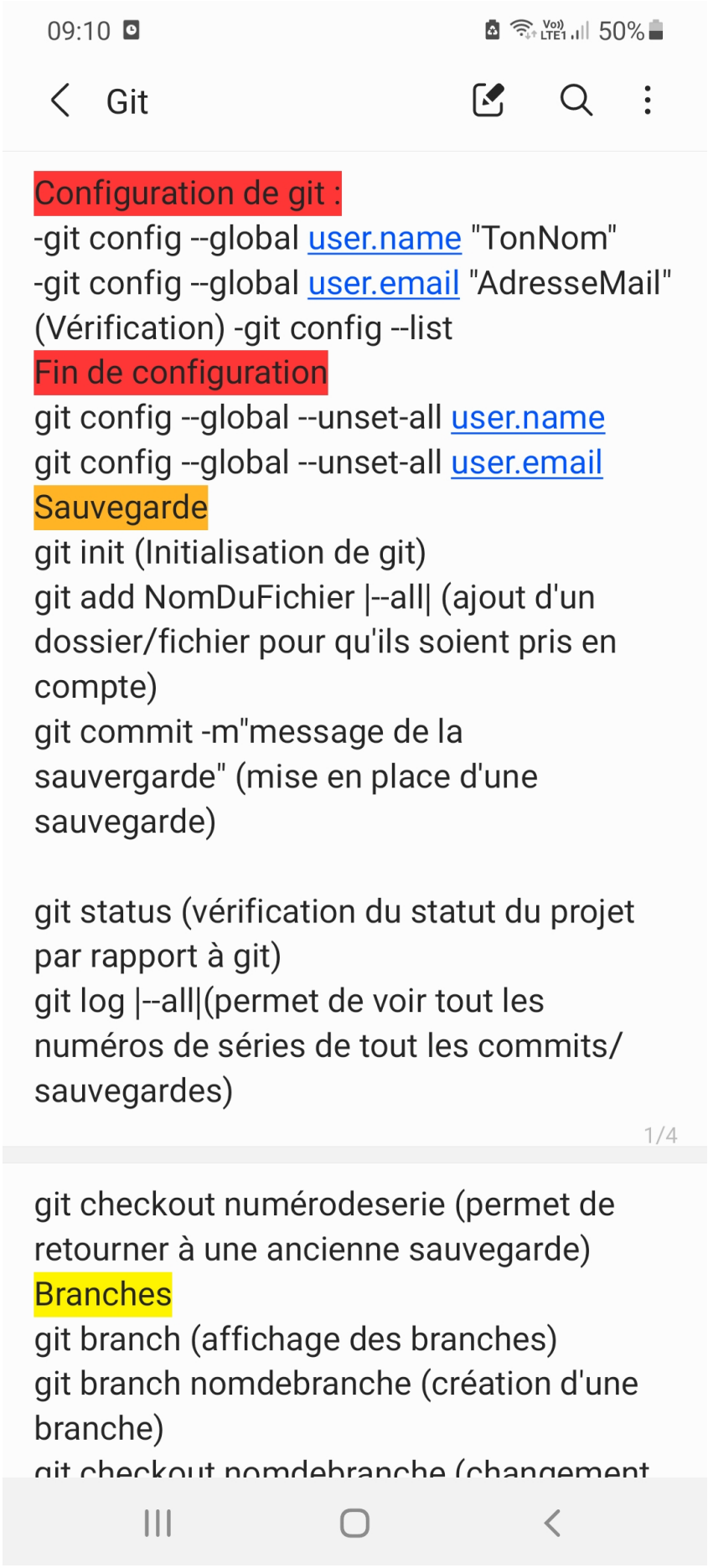
clonez ce dossier là https://github.com/nicolasprimo/gitJ3exo4

1) Créer un répository 2) Inviter ses collegues 3) Créer un projet avec juste un readme.md et l'envoyer sur le repo 4) Chaque membre du repo doit créer sa branche 5) Chaque membre doit créer un fichier html et/ou un fichier css/image etc et l'envoyer sur le repository sur la branche principale Tout le monde doit push et pull le projet de sorte a etre a jours et avoir tout le contenu du projet

2) Dans un nouveau projet avec un nouveau repository Chaque membre doit créer une page avec un menu menant vers les autres pages, les pages sont définis Accueil(index.html) Contact(contact.html) Creer également un fichier css pour votre page (ne pas utiliser style.css) Vous n'avez pas le droit de voir le code de l'autre et de discuter Il faut qu'a la fin une fois terminé, vous margez votre travail ensemble et que tout fonctionne, qu'on puisse aller d'une page à l'autre

1. Dans un nouveau projet et par deux, créer un site sur la thématique de votre choix, avec du html du css et de bootstrap, il faut faire des commîts à chaque évolution de votre site avec des messages claires. L'un d'entre vous doit créer un repository et chaquun travail sur sa branche. Présentez également une page où il y aura tous les exercices de cette semaine téléchargeable avec une petite explication
2. [09:04]
Projet avec minimum trois pages
3. [09:04]
Faites un travail soigné

3)


09:10
50%
Git
Configuration de git :
-git config --global user.name "TonNom"
-git config --global user.email "AdresseMail"
(Vérification) -git config --list
Fin de configuration
git config --global --unset-all user.name
git config --global --unset-all user.email
Sauvegarde
git init (Initialisation de git)
git add NomDuFichier |--all| (ajout d'un dossier/fichier pour qu'ils soient pris en compte)
git commit -m"message de la sauvergarde" (mise en place d'une sauvegarde)
git status (vérification du statut du projet par rapport à git)
git log |--all|(permet de voir tout les numéros de séries de tout les commits/ sauvegardes)
1/4
git checkout numérodeserie (permet de retourner à une ancienne sauvegarde)
Branches
git branch (affichage des branches)
git branch nomdebranche (création d'une branche)

4)

git checkout numérodeserie (permet de retourner à une ancienne sauvegarde)

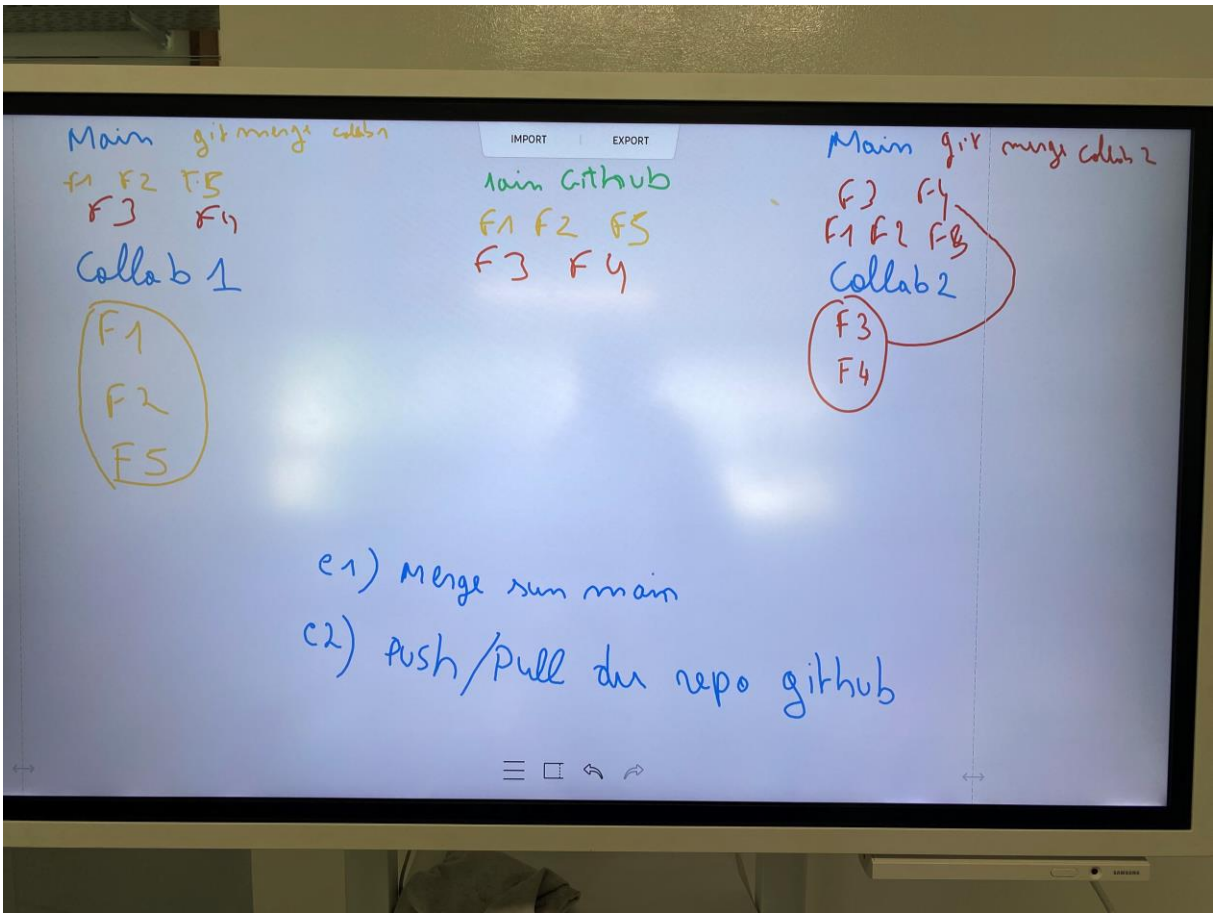
Branches

git branch (affichage des branches)
git branch nomdebranche (création d'une branche)
git checkout nomdebranche (changement de branche)
git branch -d nomdebranche (suppression de la branche)
git merce nomdebranche (apporte les modifications de la branche citée à celle en cours)

Optimisation de la sauvegarde

git add . (Prends tout les dossiers présent)
git commit -a -m"message" (-a est un raccourci au add --all mais il faut au préalable avoir déjà enregistré les fichiers une premières fois dans un commit précédent pour que cela fonctionne)
git commit -am"message" (marche tout

aussi bien)
git commit --amend(prends en considération les nouveaux fichiers/ modifications en général et les envois au dernier commit)


Main git merge collab
F1 F2 F5
F3 F4
Collab 1
F1
F2
F5
IMPORT
EXPORT
Main Github
F1 F2 F5
F3 F4
Main git merge collab 2
F3 F4
F1 F2 F5
Collab 2
F3
F4
c1) Merge sur main
c2) Push/Pull du repo github