



Archivage et principe GIT

582-121-HU - Cours 04 - 11 septembre 2025

Retour sur la semaine



Plan de leçons

- Accueil et présences
- Retour semaine
- L'archivage
- Le principe GIT

- Les lieux de sauvegardes
- Exercice en équipe

- Retour d'exercice
- Explication du devoir



L'archivage





L'archivage

N.f. - **Archives**

Collection de documents anciens, classés à des fins historiques.

L'archivage sert à conserver les anciens documents et projets, à des fins de modification ou récupération potentiels.

Selon le dictionnaire LeRobert



Pourquoi archiver des projets?

Bien qu'il existe de multitudes de raisons pour archiver des projets, il est possible d'en noter deux en particulier:

Le besoin de modification futur:

Peut-être aurez-vous à modifier l'image de marque d'une compagnie, ou encore à faire une vidéo promotionnel pour les 50 ans d'une entreprise, ou encore simplement mettre à jours le site web d'un nouveau client à partir d'éléments existants.

Peu importe le type de travail, avoir des traces des projets passés (ainsi que les fichiers sources) peut vous offrir une grande marge de manoeuvre dans votre travail.

La récupération d'un projet effacé ou corrompu:

Malheur! Un client à oublié de renouveler son site web, les affiches promotionnels d'une entreprise ont été endommagés ou encore la vidéo que vous avez fait il y a 5 ans à été corrompu!

Avoir une version archivé vous permet de récupérer votre travail pour la remise en ligne, la réimpression ou encore une nouvelle publication. Bien qu'un travailleur puisse être terminé, en garder une trace ou une sauvegarde reste important.



La nomenclature dans l'archivage

Nous avons discuté un peu de l'importance de la nomenclature, de l'appellation et de l'organisation de fichiers dans un document, mais cette pratique est aussi importante lorsque l'on parle d'archivage.

Retrouver les documents nécessaires:

Vous n'êtes peut-être pas les premiers ni les derniers à toucher à un projets. De même manière que vous aimeriez vous retrouver dans les documents d'un autre, il est important que d'autres puissent se retrouver dans vos documents.

Assurer une bonne connection entre les différents éléments d'un projets:

Bien qu'un projets peut avoir été remis avec une bonne nomenclature, il est aussi important que les éléments soient archivés avec cette nomenclature. Un site web ou une vidéo pourraient avoir été créés avec un fichier appelé asdf.mp4, mais être incapable de le trouver car le fichier a été renommé par un nom beaucoup plus simple lors de l'archivage.



La documentation dans l'archivage

Il est possible qu'un projet vous soit attribué sans résumé ou directives. L'ajout ou la lecture de certaines documentations peuvent alors aider à la compréhension du projet:

Un document "lisez-moi":

Bien que ce type de document soit souvent oublié ou simplement ignorés, ils comportent souvent beaucoup d'information importante relative aux projets. Que ce soit au niveau du contexte des projets, du contenu des dossiers ou encore de méthodes d'installation ou de manipulations des documents.

Un document "mandat":

Un mandat aide à donner un contexte au projet et à son existence, chose qui peut parfois être perdue ou mal comprise avec le temps. Avoir un document ou une version du mandat dans les archives aide à comprendre quel était le but du projet.



Rappel!

Au moins 3 endroits
de sauvegarde!

Rien n'est sans risque!

Une panne internet? Une panne de
réseau? Un disque dur à l'eau? Tout est
possible! Assure-toi d'avoir une
sauvegarde de secours sous la main pour
éviter les pépins!

Principe GIT





GIT*

Global Information Tracker / Suivi
d'information globale

GIT est un principe utilisé pour faire le suivi d'information de projets. Avec ce principe, il est possible de travailler en collaboration sur un projet, peu importe où nous sommes.

Le principe utilise un type de suivi qui garde en mémoire les versions antérieures, permettant ainsi de retrouver d'anciennes versions d'éléments de projets.



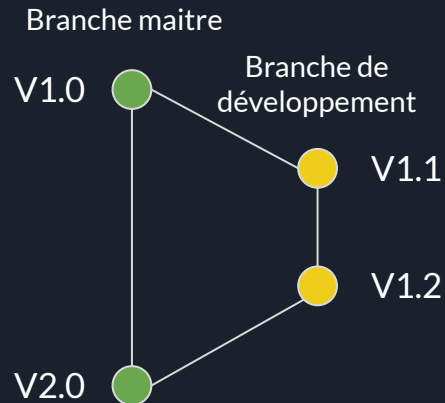
Un outil merveilleux

Git est un outils qui permet à plusieurs personnes de travailler sur un même projets, et de garder un historique de changement, le tout dans un lieu centralisé. Il est possible d'effectuer des manipulations de types:

- **Branching/Ramification** (créer une version parallèle du projet)
- **Pull/Tirage** (Télécharger une version du projets en cours)
- **Push/Pousser** (Mettre en ligne les changement effectué du projets sur notre branche)
- **Merge/Joindre** (Mettre en lignes les changements de notre branche sur la version maître du projet.)

Branching/Ramification

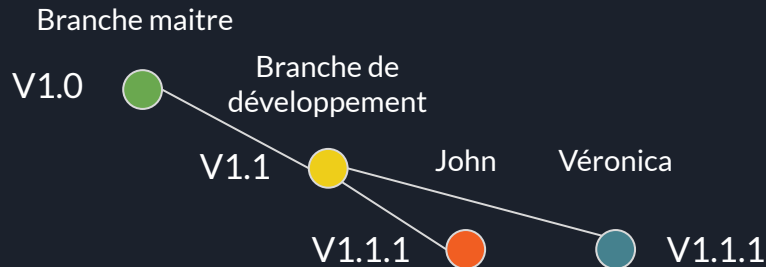
La ramification permet à l'utilisateur de créer une version parallèle du projets, sur lequel il pourra travailler et effectuer des changements. Cette branche permettra à plusieurs utilisateurs de faire des changement, tout en s'assurant que la base maître reste la même. Les changements seront fait sur la branche, puis testé et approuvé avant d'être envoyé à la branche maître.



Pull/Tirage

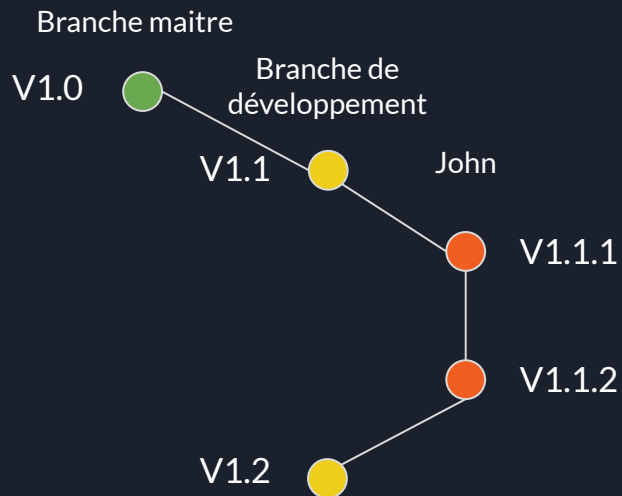
L'action de retirer une version du projet vous permet alors de modifier cette dernière pour la retourner au projet une fois terminé. Plusieurs personnes peuvent retirer une version du projets pour la modifier, de manière simultanée.

Pour une bonne pratique, il est recommandé de ne pas travailler sur les mêmes parties d'un projets, mais sur différentes parties. (Exemples, 2 personnes qui modifient la même ligne de code risque d'être difficile lorsque viendra temps de joindre les versions, mais avoir une personne qui travaille sur le code et une autre sur les images peut être très bénéfique).



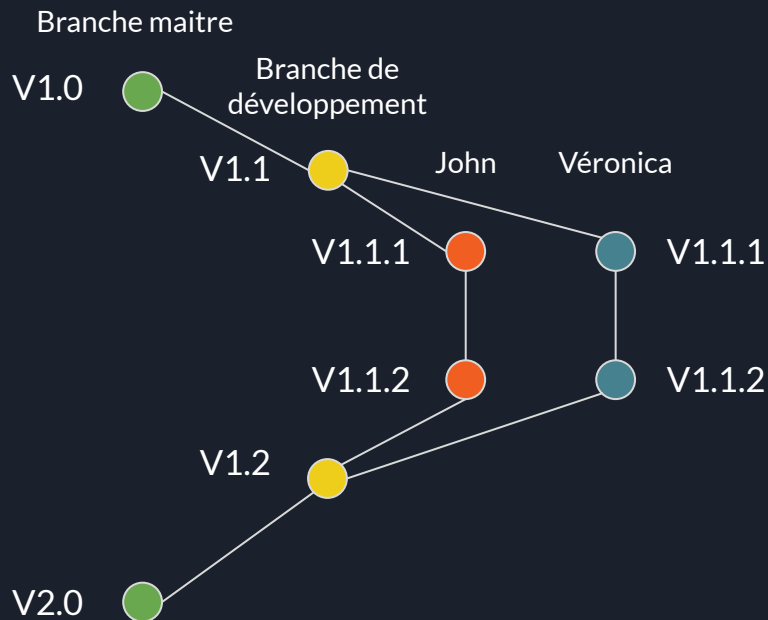
Push/Pousser

Après avoir apporté des modifications au projets, il est possible pour un utilisateur de “pousser” leurs changements, pour que d’autres puissent y avoir accès lorsqu’ils font un retrait. Les changements sont en effets, sans toutefois être joint à la branche maître.

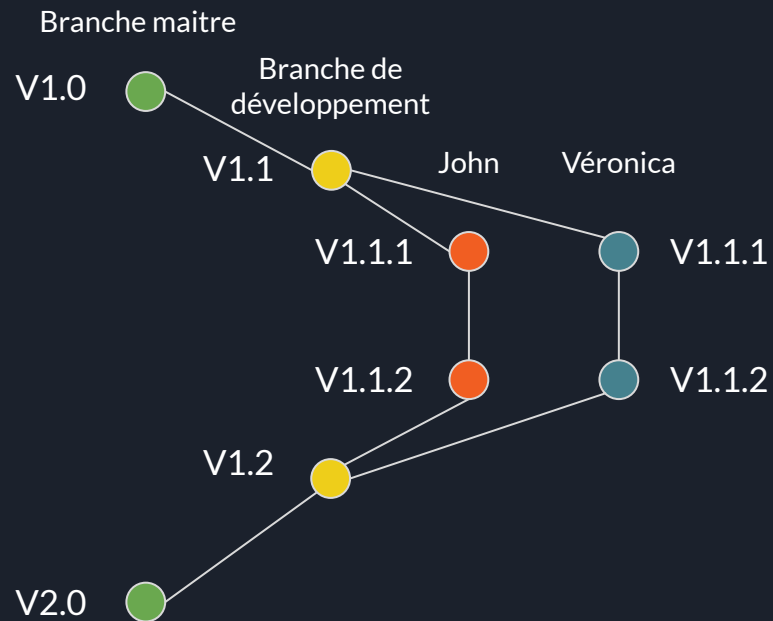


Merge/Joindre

Dans une situation où deux personnes touchent au même document, l'action de merge est donc nécessaire. Lors de cette action, les différents éléments d'un documents sont comparés, afin de vérifier les éléments les plus recent/qui sont nécessaires de garder. Cette action requiert plusieurs vérifications et parfois même des approbations de personnes tierces.



Donc...?



Exercise





Créer un répertoire, inviter des collaborateurs et pousser un document

Comment?

- En équipe de 3-4
- Utiliser votre compte GitHub
- Une personne créer un répertoire et invite ses camarades
- Chacuns met en ligne un documents sur le répertoire
- Vous avez 50 minutes

Quoi?

- Créer un répertoire sur GitHub comme montré en classe
- Une page appelé index.html contenant nom d'équipe et des liens vers les documents HTML de tous
- Chacques membres de l'équipe doit mettre en ligne un document html simple:
 - Un documents HTML simple incluant votre nom et un lien vers les documents de vos camarades.

Exemple:

- Index.html
- Laura_Secord.html
- Jean_Drapeau.html
- Yvan_Dubois.html
- Luc_Leduc.html

Démonstration: GitHub et Visual Studio Code





Utiliser GitHub avec Visual Studio Code

Il est possible de télécharger et de mettre en lignes des documents directement à partir de GitHub. Dans le cadre de ce cours, nous n'aurons malheureusement pas accès à l'utilisation du principe GIT localement sur les ordinateurs. Il est cependant très intéressant pour vous d'apprendre à utiliser GIT, mais aussi à installer les différentes composantes à la maison.

- Installation de GIT (Windows, FR): <https://www.youtube.com/watch?v=Ysv3RDGXwtY>
- Installation de GIT (Mac/Linux, FR): https://www.youtube.com/watch?v=g_l9W4FAfjI
- Installation de GIT et GitHub dans Visual Studio Code (Windows, EN)
<https://www.youtube.com/watch?v=PzYxpncAYD8>

Devoir(s)



Préparation pour l'examen:

- Réviser les documents de nomenclatures
- Pratiquer à renommer des documents

Expérimentation GitHub:

- Explorer les communautés GitHub
- Essayer des travaux d'équipes en faisant des répertoires GitHub

