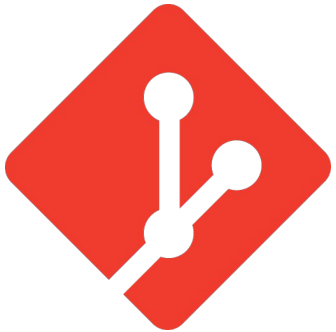



Aa

Objectifs

- Comprendre comment **réaliser une fusion**
- Résoudre les **problèmes liés à la fusion**

Git pull, une commande qui nous veut du bien

Lorsque plusieurs personnes collaborent sur un même projet, elles modifient constamment les fichiers.

Imaginons que l'on passe une semaine à perfectionner le pied de page de notre application web. Pendant ce temps, un de nos collègues a modifié la présentation des textes de la page.

Dans un monde sans Git, pour harmoniser votre travail, nous devons récupérer une copie de son fichier `index.html` et comparer les différences avec notre version, copier les lignes de code correspondantes de la nouvelle présentation du texte et les coller dans notre fichier.

C'est une opération longue (dans la vie réelle, il y a souvent beaucoup plus de changements) et risquée, car nous ne serions peut-être pas en mesure de repérer tous les changements effectués par notre collègue.

Heureusement, **Git peut faire ce travail pour nous !**

Git pull, une commande qui nous veut du bien

C'est ce qu'on appelle une **fusion** (merging).

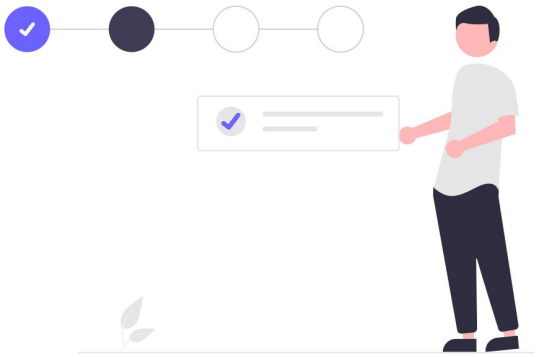
Elle est réalisée par la commande **"git pull"** (il y a d'autres façons de le faire, mais c'est suffisant pour l'instant), qui nous permet de **mettre à jour nos fichiers locaux avec les fichiers distants** (par exemple, sur le repo GitHub).

Dans la plupart des cas, cette *fusion* est **automatique** : Git localise de lui-même les différences entre nos fichiers locaux et leurs homologues distants et les intègre dans tes fichiers locaux.

Conflits de fusion

Cette vidéo présente un exemple de conflit et sa résolution

Pratiquons !



- Créer un nouveau dépôt sur GitHub, en cochant "Initialize this repository with a README"
- **Toujours dans GitHub,** écrire quelques lignes de texte dans le fichier README.md.
- Cloner le repo
- **Sur GitHub,** écrire "**REMOTE**" sur la première ligne du README.md et faire un commit pour ce changement
- Localement (sur notre ordinateur), écrire “**LOCAL**” sur la première ligne du README.md et faire un commit pour ce changement
- Dans notre terminal, faire un **"git pull"** → un joli petit conflit devrait apparaître
- Dans notre éditeur, résoudre le conflit dans le README.md (choisir de garder "LOCAL")
- Faire un commit pour ce changement
- Faire de nouveau “git pull” : git dira que nous sommes déjà à jour. Le conflit est réglé !
- Envoyer les modifications au repo distant en faisant un **git push origin main**
- Dans GitHub, ouvrir le README.md et s’assurer que "**LOCAL**" est maintenant sur la première ligne.