

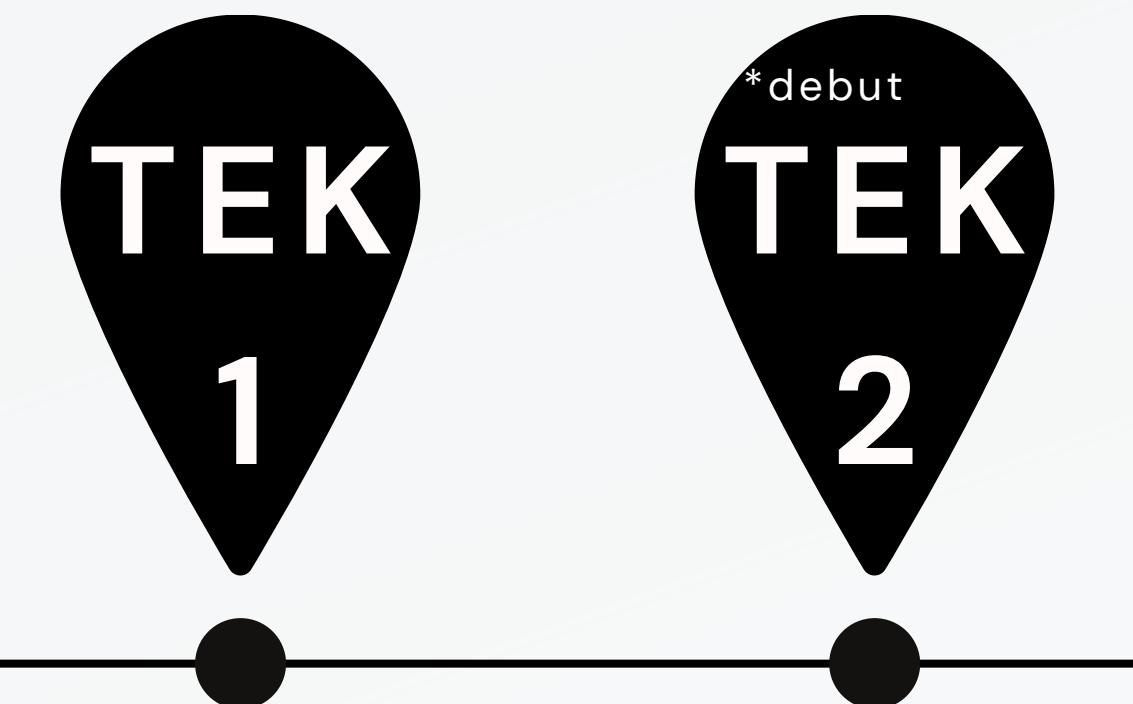


GitHub

HOW TO **GITHUB**

POUR UN MEILLEUR WORKFLOW DE PROJETS

CE WORKSHOP VISE



SOMMAIRE

01

RAPPEL GITHUB

02

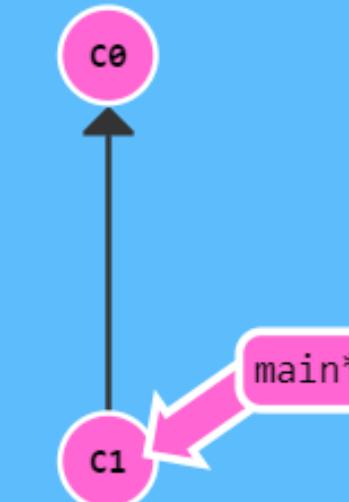
L'IDÉAL D'UN TRAVAIL EN GROUPE

03

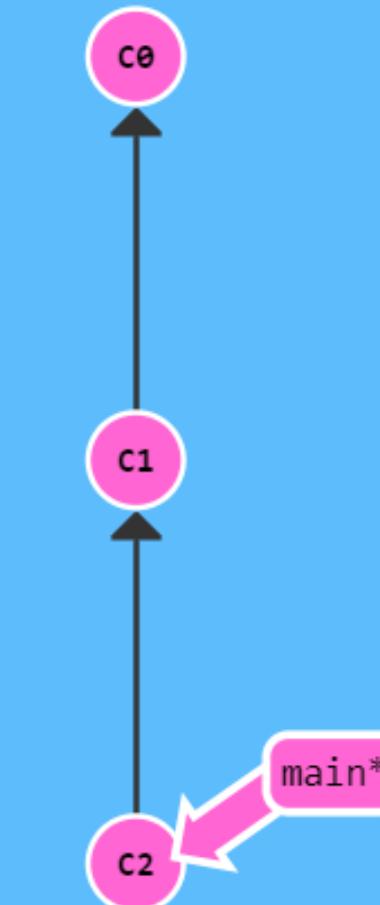
PRATIQUE



C1 et C0 représentent les commits
main* lui représente la branche principale



Chaque commit, qui a été push la branche
main s'update avec les modifications
apportés au dernier commit

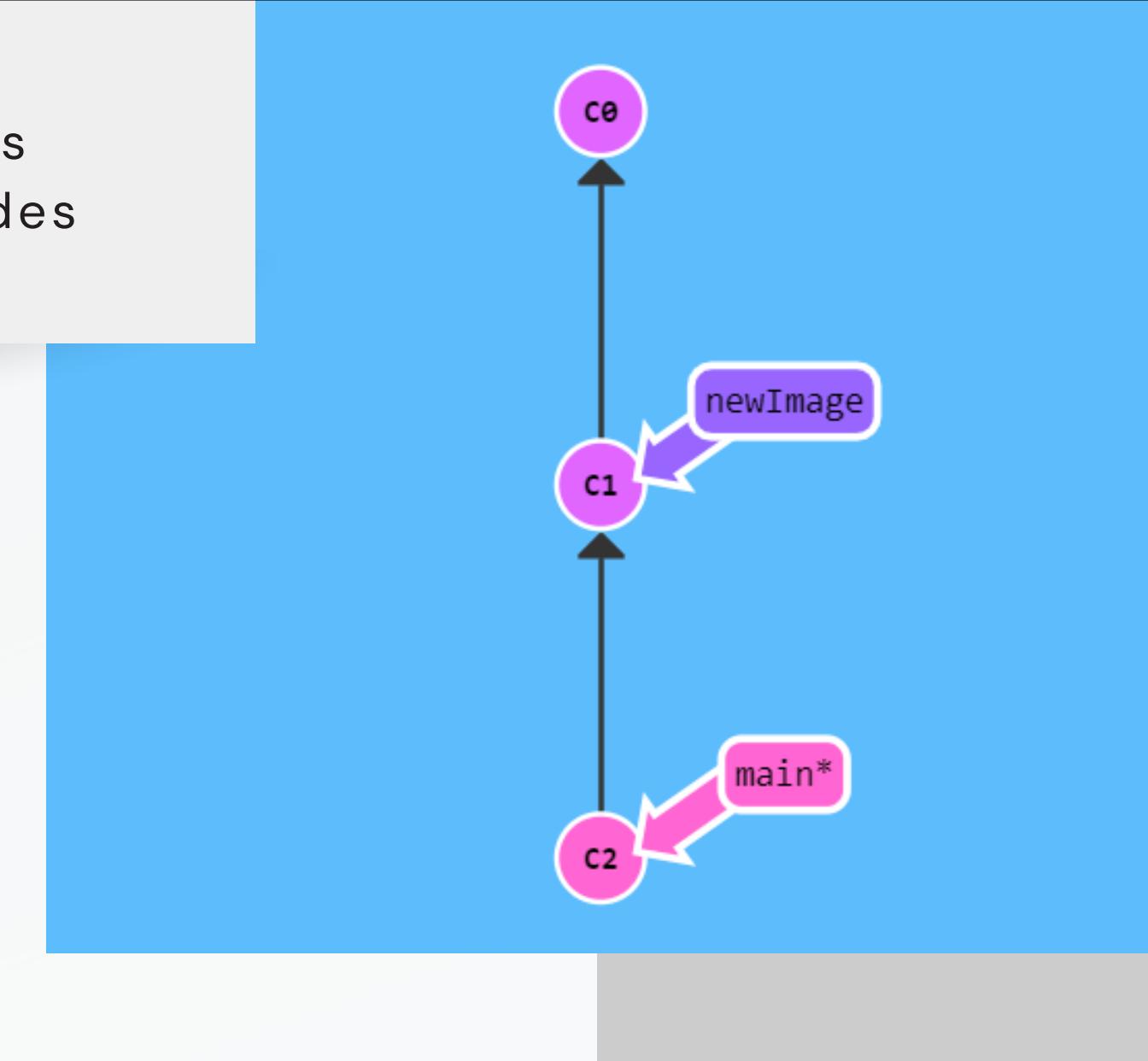
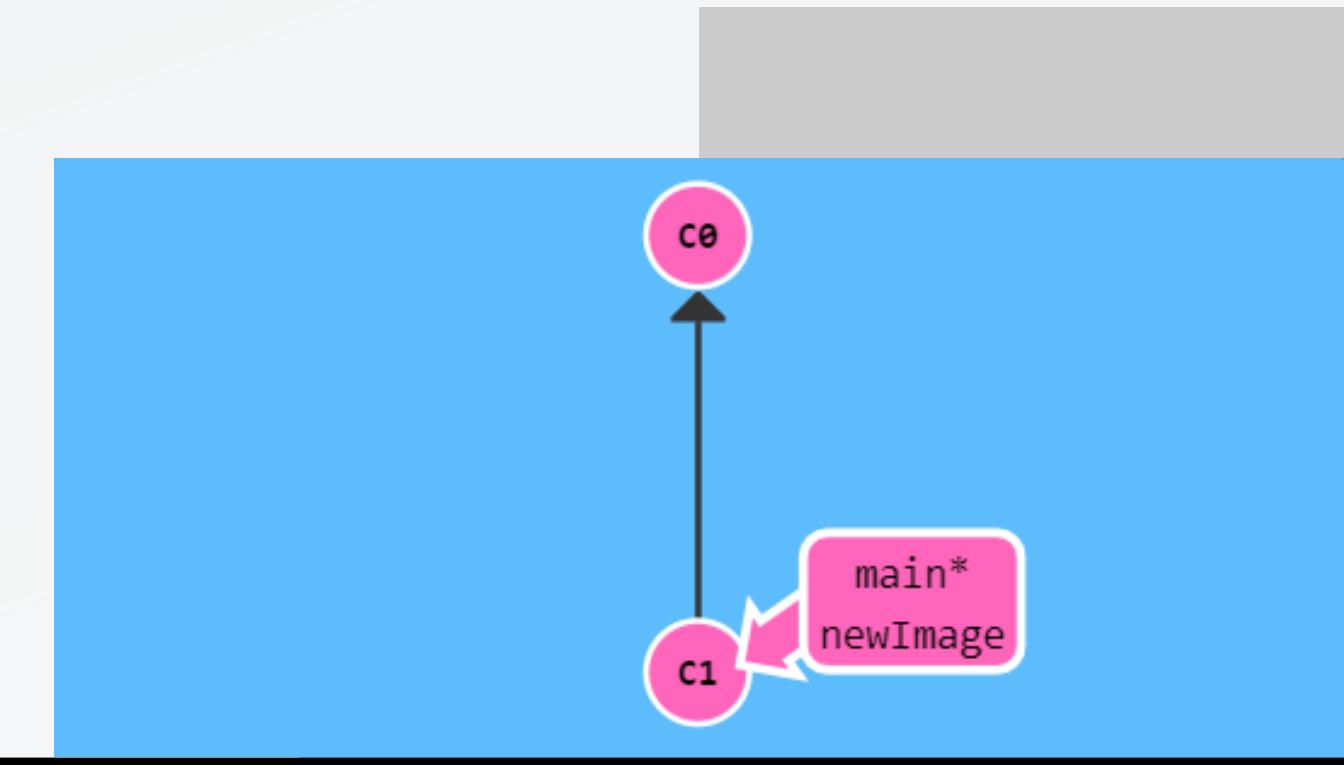


*chaque commit est "child" de son "parent"
d'origine:
Ici, C1 est le "parent" de C2

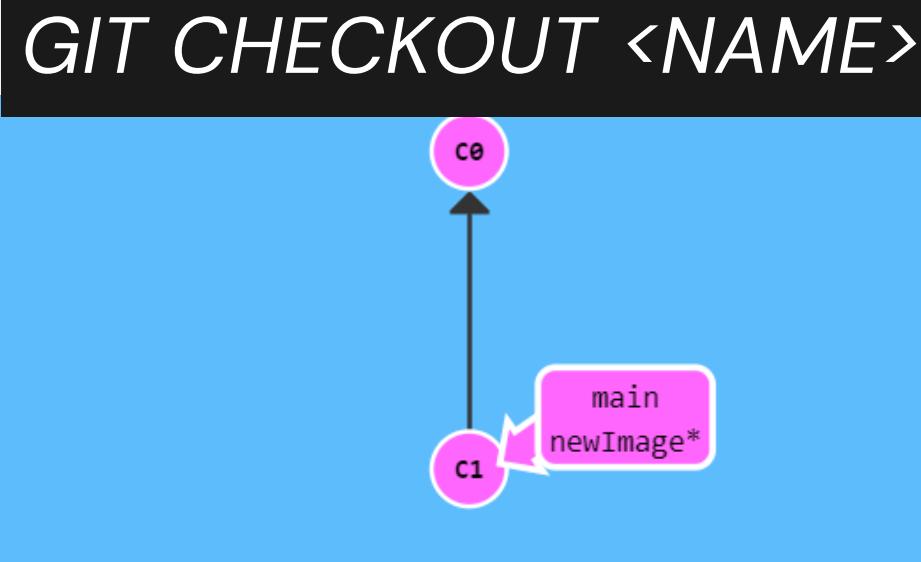
BRANCHING



Ce sont simplement des pointers à un commit spécifique rien de plus. C'est plus simple de diviser ton travail que d'avoir des "branches" surchargées.



DÉPLACEMENT

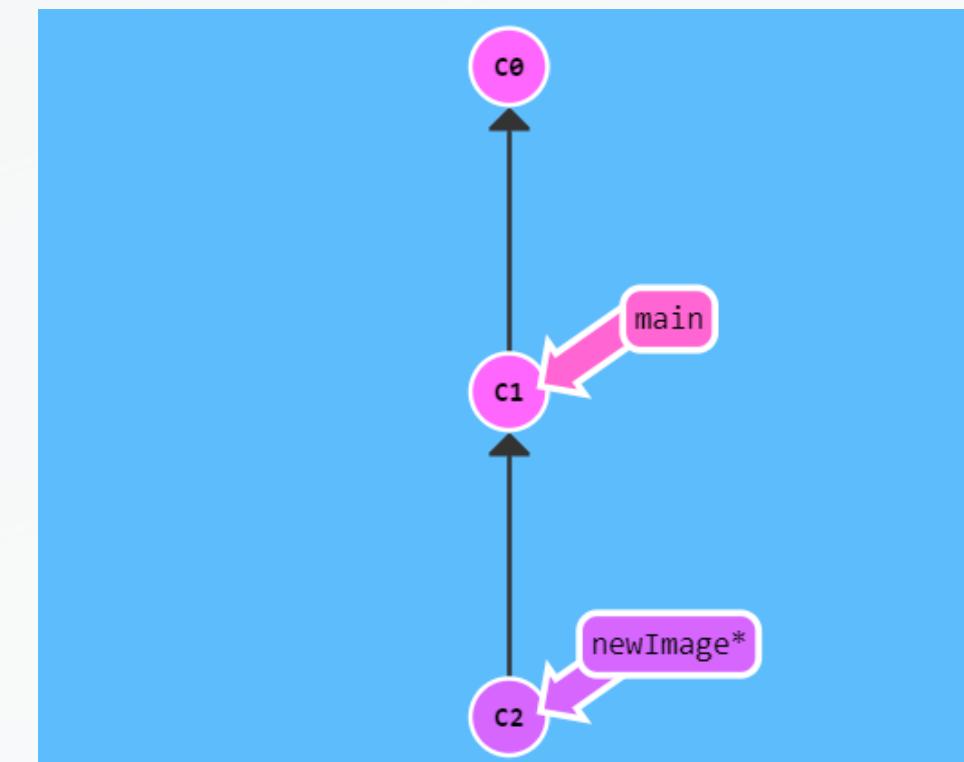


GIT CHECKOUT <BranchName>
vous permet de vous déplacer parmi vos branches

Pro tips, si vous faites la commande au-dessus avec un `-b`, vous créerez donc une branche en vous y déplaçant en même temps ;)

COMMIT DEPUIS NOUVELLE BRANCH

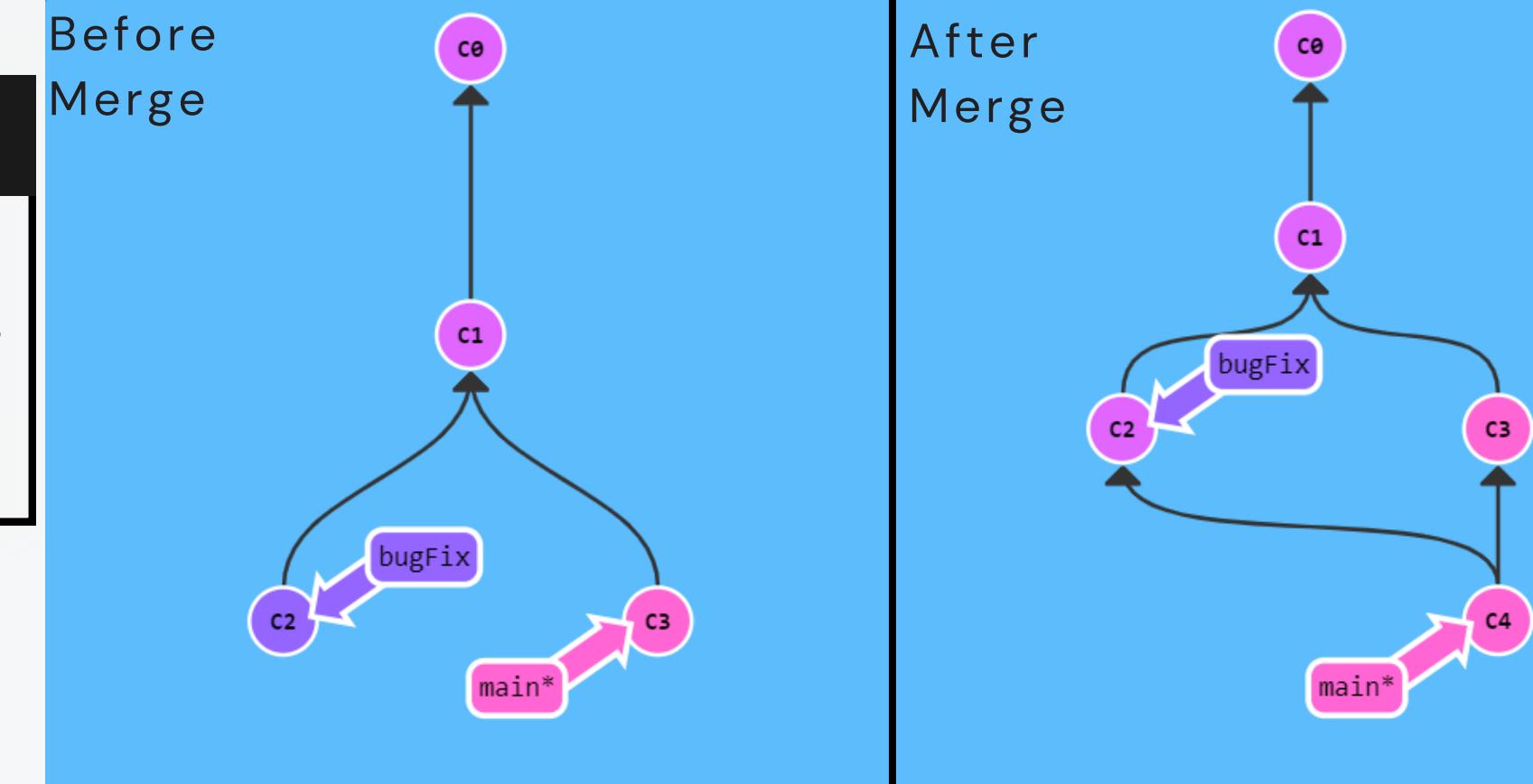
Dans Git, version 2.23, une nouvelle commande "git switch" a été créée pour éventuellement remplacer git checkout. Nous n'y toucherons pas aujourd'hui, car il est considéré "expérimental".



“MERGING BRANCHES”

GIT MERGE <NAME1>

```
GIT MERGE <BranchName>
// <BranchName> ne doit pas être la branche sur laquelle vous
vous trouvez, ici, c'est "git merge bugFix" car nous nous
trouvons sur main
```



dans la seconde image, vous
pouvez voir que main inclue mtm
toute l'arborescence en lui, alors
que bugFix lui seulement C0, C1, C2

BIENTÔT FINI

Maintenant que tu maîtrises git localement. Tu vas apprendre à utiliser git avec ton équipe. Et interagir avec un repo public

80%



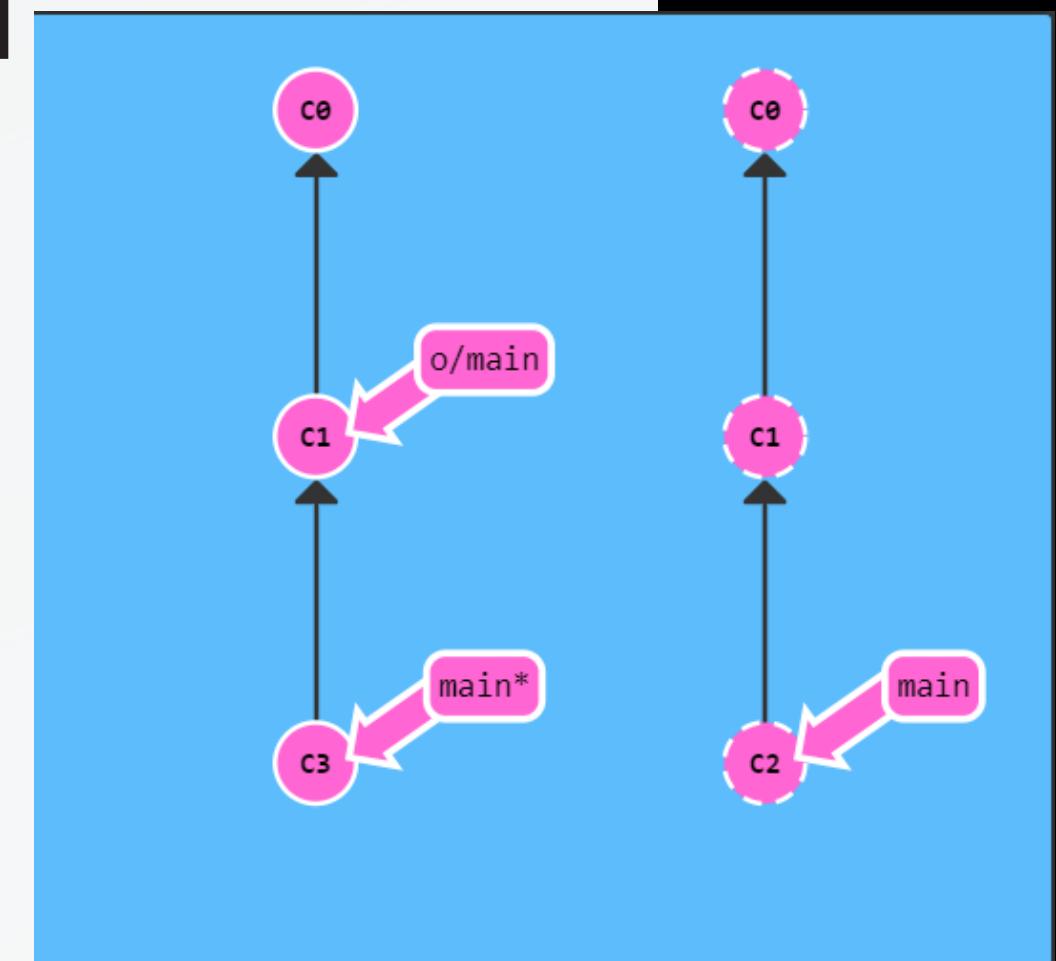
PUSH AMBIGUITY

- La colonne de gauche représente votre repo local
- La colonne de droite représente votre repo epitech

Ici la commande git push ne fonctionnerait pas, car le repo epitech a reçu un commit de vos coéquipiers, comment résoudre ce problème?

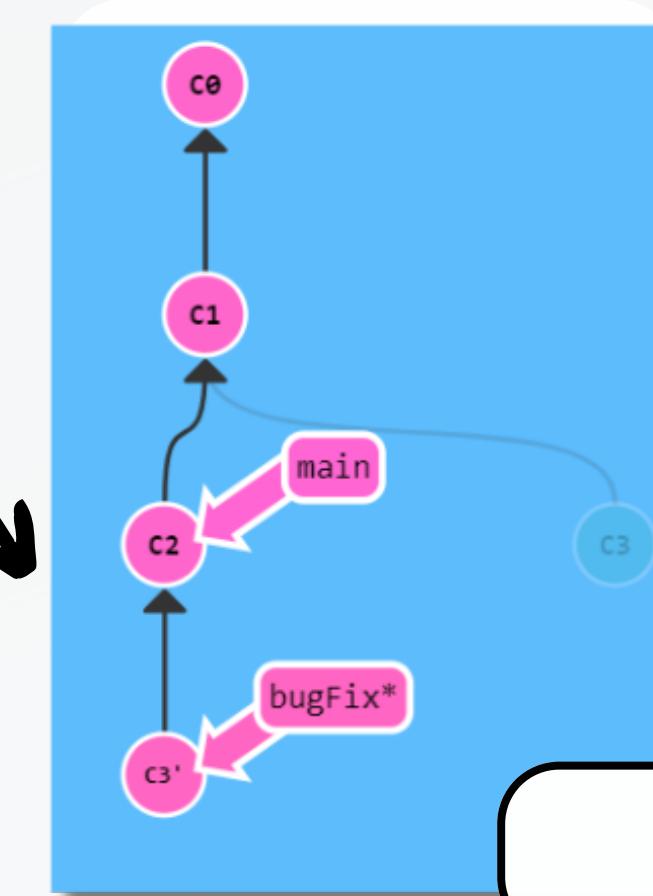
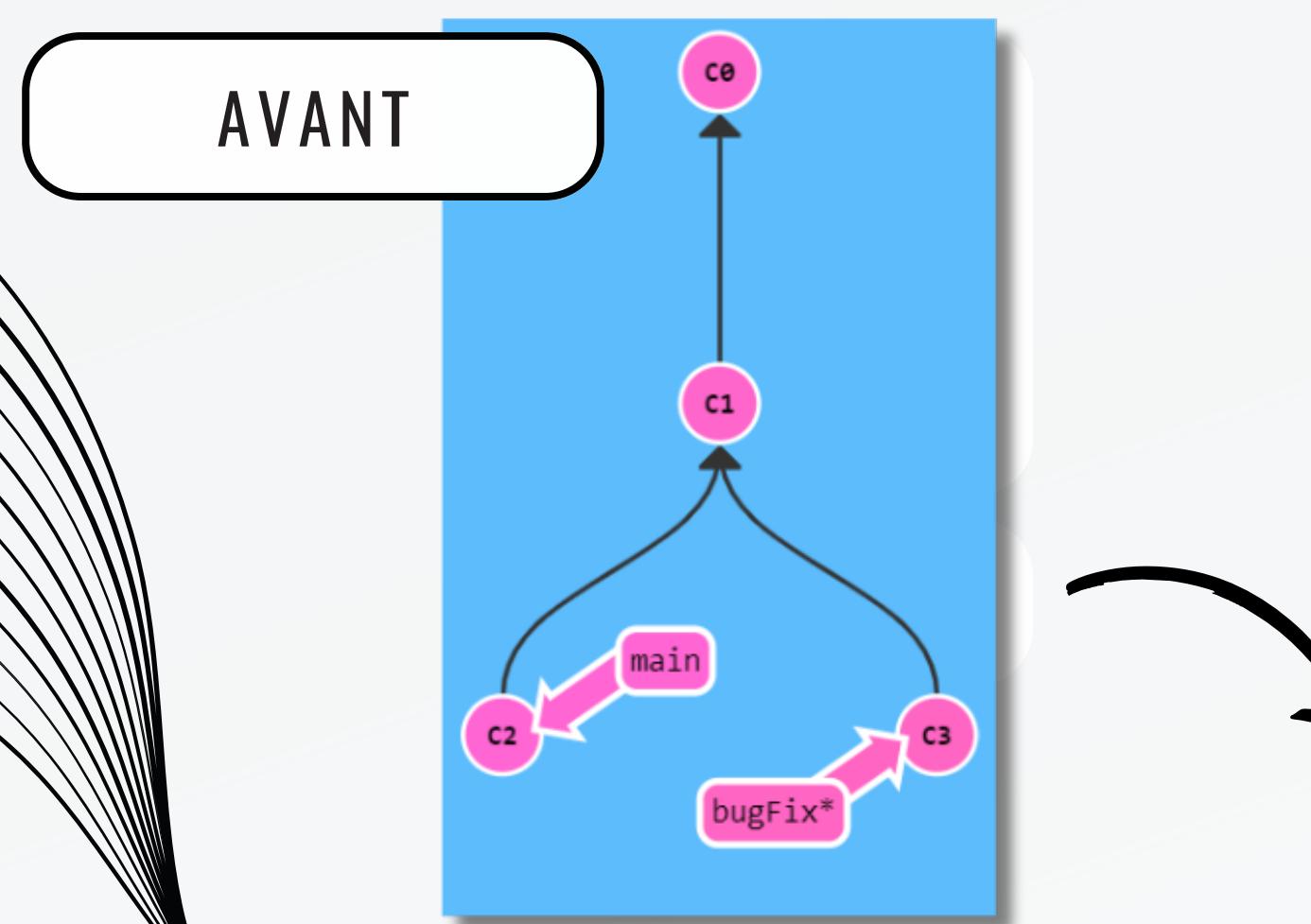
git pull --rebase

C3 est basé de C1, maintenant que C2 existe le push est annulé



BASIC REBASE

une fois sur la branch que vous voulez prio, utilisez “git rebase <name>”



APRÈS

Rebase est
essentiellement un
merge mais plus propre,
il permet de garder une
seule ligne de commits
à la place d'une énorme
arborescence

L'HEURE DE PRATIQUER

Je vous invite à jeter un coup d'œil a ces commandes ici.



01
02
03

CHERRY-PICK
FETCH
--SET-UPSTREAM

Et de surtout poncer ce site :
<https://learngitbranching.js.org>

LA STRAT



Faites un binôme ou trinôme,
suivez ensuite les
instructions du PDF dans le
repo git!

ÉTAPE N°1



Vous n'êtes pas obligé de
suivre le pdf, l'objectif du
workshop est de vous faire
manipuler git et de faire vos
propres recherches, afin de
maîtriser l'outil

ÉTAPE N°2



Demandez de l'aide, je ne
sais pas si je serais en
mesure de pouvoir vous
aider, mais peut-être
essayer de vous aiguiller
vers une recherche plus
pertinente

ÉTAPE N°3

APRÈS LE PDF:

<https://learngitbranching.js.org>

MERCI POUR L'ÉCOUTE

A vous de jouer!

