



Génie logiciel : GIT

Karën Fort

karen.fort@univ-lorraine.fr / <https://members.loria.fr/KFort>

Quelques sources d'inspiration

- ▶ <https://git-scm.com/book/fr/v2> (en CC BY-NC-SA)
- ▶ <https://www.w3schools.com/git/>
- ▶ <https://ohshitgit.com/>
- ▶ <https://people.irisa.fr/Anthony.Baire/git/git-for-beginners-handout.pdf>

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

Bonnes pratiques

Pour finir

Exercice : préparation

L'un-e d'entre vous :

- ▶ crée un fichier Etudiants.csv qui contient les titres de colonnes informations suivants, séparés par des virgules :
 - ▶ Numéro étudiant
 - ▶ Nom
 - ▶ Prénom
 - ▶ Apprenti ?
 - ▶ Niveau en Git
- ▶ Git :
 - ▶ add
 - ▶ commit -m "Création du fichier Etudiants"
 - ▶ push

Exercice : premier fichier collectif sur Git

Chacun·e d'entre vous doit maintenant remplir ce fichier !

Chacun récupère les ajouts des autres : git pull

```
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_git/m1_miage_2024-25$ git pull
Username for 'https://gitlab.univ-lorraine.fr': fort5
Password for 'https://fort5@gitlab.univ-lorraine.fr':
Déjà à jour.
```

- ▶ on récupère tout ce qui est nouveau sur le serveur (d'où la connexion)
- ▶ c'est fini !

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

Bonnes pratiques

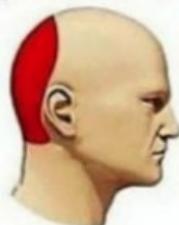
Pour finir

Types of Headaches

Migraine



Hypertension



Stress



Solving Git
Merge conflicts



[imgflip.com](https://i.imgflip.com/)

<https://i.imgflip.com/471xbc.jpg>

Et maintenant on fait quoi ?

```
karen@Delbo:~$ cd /home/karen/Documents/Semagramme/ra-2024
karen@Delbo:~/Documents/Semagramme/ra-2024$ git pull
Username for 'https://gitlab.inria.fr': fortkare
Password for 'https://fortkare@gitlab.inria.fr':
avertissement : redirection vers https://gitlab.inria.fr/semagramme/ra-group/ra-2024.git/
remote: Enumerating objects: 282, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 282 (delta 13), reused 0 (delta 0), pack-reused 252 (from 1)
Réception d'objets: 100% (282/282), 858.76 Kio | 11.93 Mio/s, fait.
Résolution des deltas: 100% (43/43), fait.
Depuis https://gitlab.inria.fr/semagramme/ra-group/ra-2024
   d86bec..f981152  git-dev    -> origin/git-dev
   e372b47..f981152  master    -> origin/master
Mise à jour d86bec..f981152
erreur : Vos modifications locales aux fichiers suivants seraient écrasées par la fusion :
    12_02-BIB-READONLY-year-publications.bib
Veuillez valider ou remiser vos modifications avant la fusion.
Abandon
```

Que s'est-il passé ?

- ▶ Git pull m'informe que mes :
"modifications locales aux fichiers [...] seraient écrasées par la fusion"
 - ▶ Git pull est en fait une combinaison de commandes :
 - ▶ Git fetch : récupère tous les changements d'une branche ou d'un dépôt
 - ▶ Git merge : fusionne la branche actuelle avec la branche spécifiée
- ici, la fusion n'a pas fonctionné, parce que je n'ai pas validé mes changements (git add + git commit + git push)

Les conflits de fusion (typiques) à plusieurs

```
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_
git/m1_miage_2024-25$ git merge new_branch_for_merge_conflict
Fusion automatique de FortKaren.csv
CONFLIT (contenu) : Conflit de fusion dans FortKaren.csv
La fusion automatique a échoué ; réglez les conflits et validez le
résultat.
```

```
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_
git/m1_miage_2024-25$ git status
Sur la branche main
Votre branche est en avance sur 'origin/main' de 2 commits.
  (utilisez "git push" pour publier vos commits locaux)

Vous avez des chemins non fusionnés.
  (réglez les conflits puis lancez "git commit")
  (utilisez "git merge --abort" pour annuler la fusion)

Chemins non fusionnés :
  (utilisez "git add <fichier>..." pour marquer comme résolu)
      modifié des deux côtés : FortKaren.csv

aucune modification n'a été ajoutée à la validation (utilisez "git
add" ou "git commit -a")
```

Comment Git fusionne les fichiers ?

- ▶ les fichiers textuels sont fusionnés ligne par ligne
- ▶ les fichiers binaires génèrent systématiquement un conflit et doivent être gérés manuellement

Les marques de fusion dans les fichiers

```
<<<<<< HEAD|
:Fort,Karën,9
: Ajout de contenu pour générer des pbs depuis la branche main
: =====
: Ajout de contenu pour générer des pbs
: >>>>>> new_branch_for_merge_conflict
```

- ▶ ce qui est au-dessus du séparateur ===== vient de la branche main
- ▶ ce qui est au-dessous vient d'une autre branche

Résoudre les conflits de fusion

- ▶ "Nettoyer" le fichier
- ▶ `git add`
- ▶ `git commit`
- ▶ `git status`

```
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_
git/m1_miage_2024-25$ git add .
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_
git/m1_miage_2024-25$ git commit -m "nettoyage de fichier effectué"
[main e130e23] nettoyage de fichier effectué
karen@Delbo:~/Documents/Cours/CoursPR_Nancy/M1_GenieLogiciel/Cours_
git/m1_miage_2024-25$ git status
Sur la branche main
Votre branche est en avance sur 'origin/main' de 4 commits.
  (utilisez "git push" pour publier vos commits locaux)

rien à valider, la copie de travail est propre
```

Résoudre les conflits de fusion : version "j'en ai marre"

- ▶ renommer son fichier (local)
- ▶ `git pull`
- ▶ diff entre son fichier renommé et le fichier du dépôt
- ▶ modifications du fichier du dépôt
- ▶ `git add`
- ▶ `git commit`

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

Bonnes pratiques

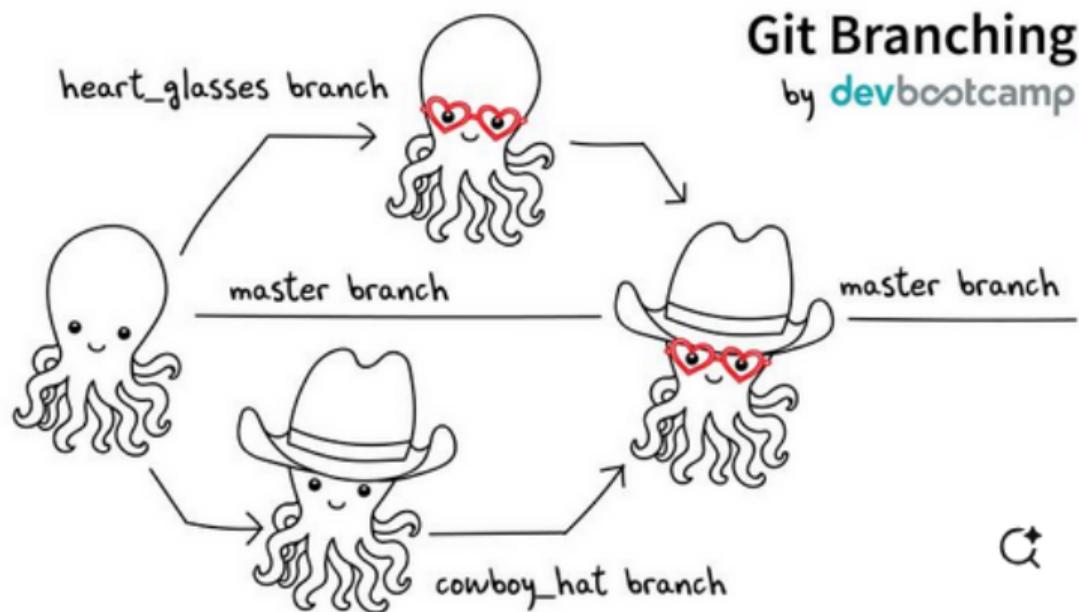
Pour finir

Définition superficielle

Branche

Dans Git, une branche est une version parallèle au dépôt principal (main ou master).

Les branches de Git : vue d'artiste



<https://fr.pinterest.com/pin/git-branching--244953667210946459/>

Les branches dans Git, en vrai

*"Une branche dans Git est simplement un **pointeur** léger et déplaçable vers un [...] commit. La branche par défaut dans Git s'appelle master. Au fur et à mesure des validations, la branche master pointe vers le dernier des commits réalisés. À chaque validation, le pointeur de la branche master avance automatiquement."*

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Les branches dans Git, en vrai



*"Une branche dans Git est simplement un **pointeur** léger et déplaçable vers un [...] commit. La branche par défaut dans Git s'appelle master. Au fur et à mesure des validations, la branche master pointe vers le dernier des commits réalisés. À chaque validation, le pointeur de la branche master avance automatiquement."*

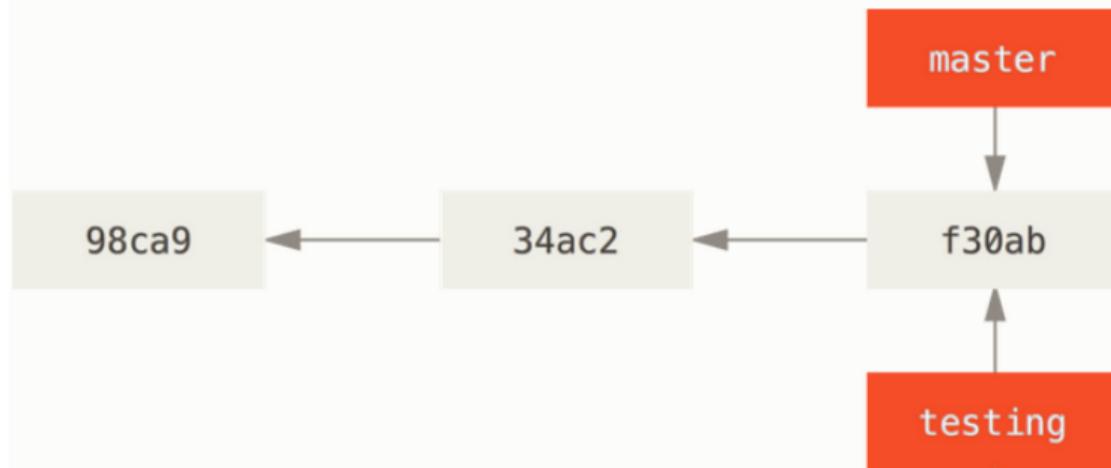
<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Créer une nouvelle branche

Ici, *testing* :

```
$ git branch testing
```

Cela crée un nouveau pointeur vers le commit courant.



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Où sommes-nous ?

Ici (*HEAD*, pointeur spécial) :



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Basculer entre les branches (de *master* à *testing*)

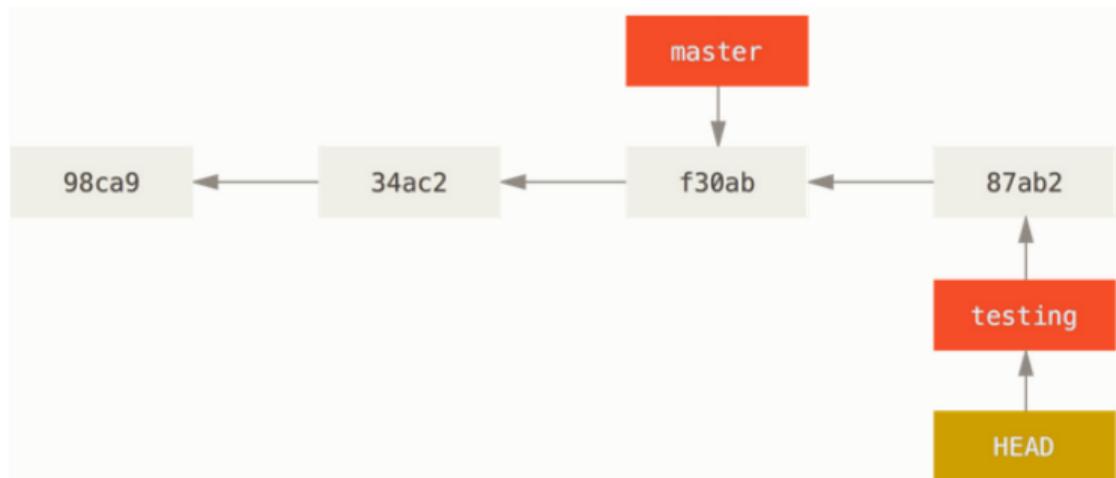
```
$ git checkout testing
```

Cela déplace HEAD pour le faire pointer vers la branche `testing`.



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

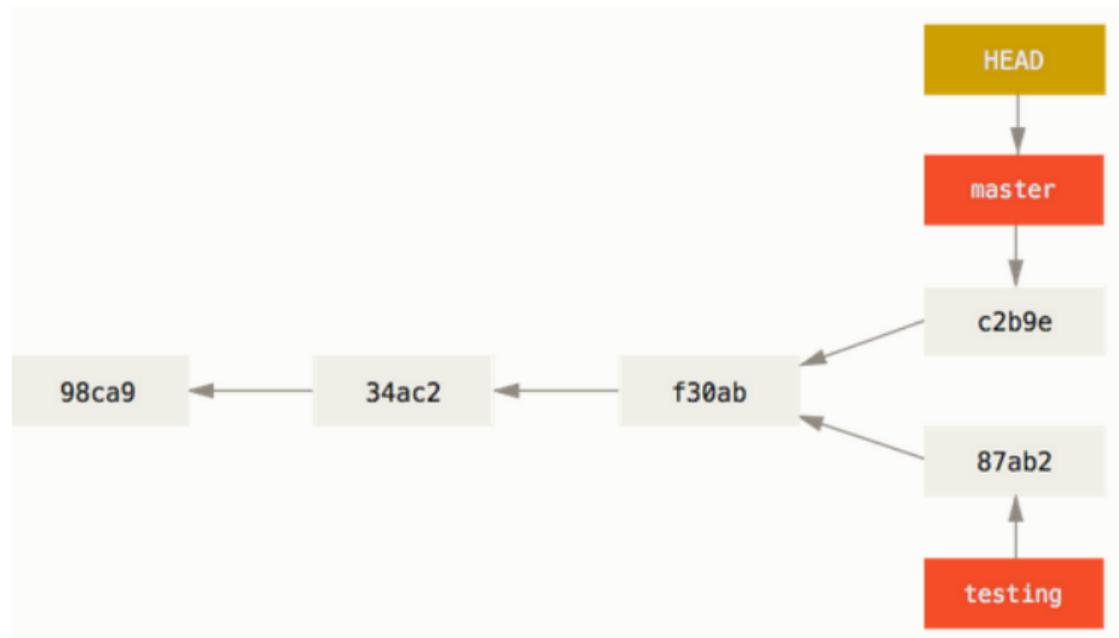
Et si on fait un commit sur la branche où on est (*testing*) ?



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Que s'est-il passé ?

Détaillez les différentes étapes pour en arriver là



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

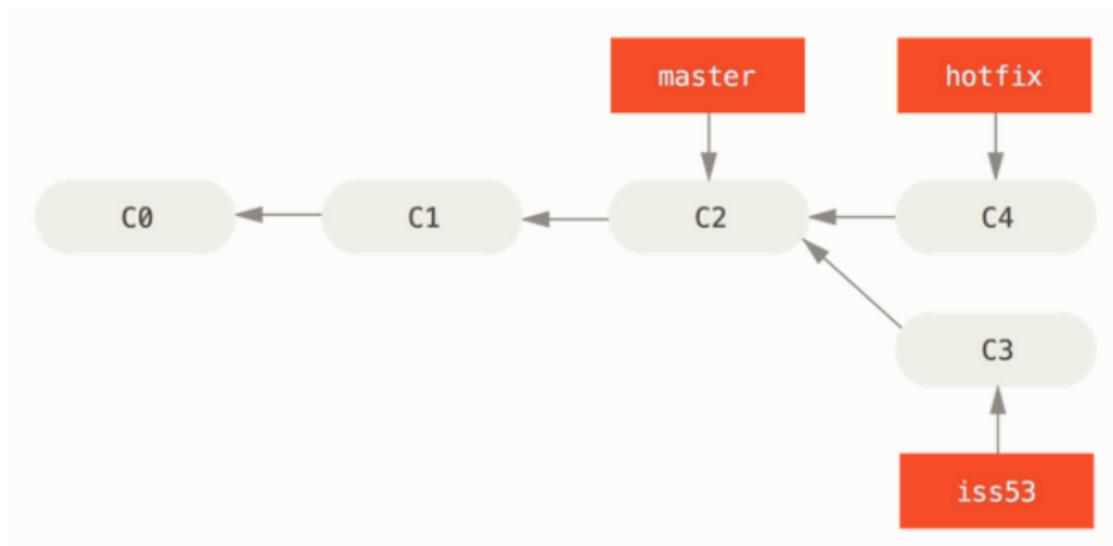
Gestion des branches sur un dépôt distant

Bonnes pratiques

Pour finir

Que s'est-il passé ?

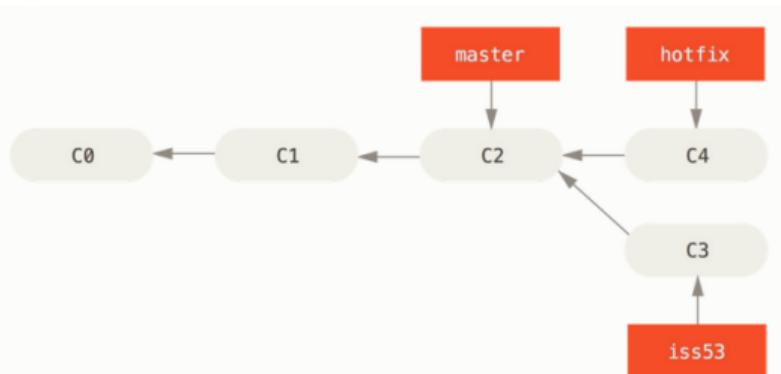
Détaillez les différentes étapes pour en arriver là



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Que s'est-il passé ?

- ▶ `git checkout -b iss53` (création d'une nouvelle branche `iss53`, sur laquelle on pointe)
- ▶ `git commit -a -m "ajout d'un pied de page [problème 53]"` (C3)
- ▶ `git checkout -b hotfix` (création d'une nouvelle branche `hotfix`, sur laquelle on pointe)
- ▶ `git commit -a -m "correction de l'adresse email incorrecte" [hotfix 1fb7853]`

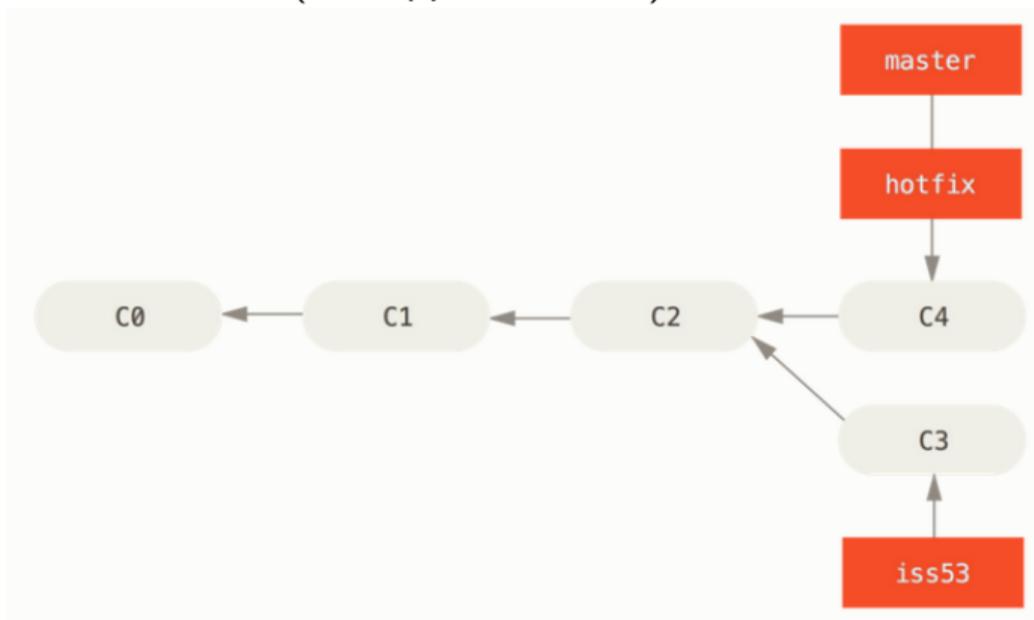


<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Fusion de hotfix avec la branche maîtresse (master)

fast-forward

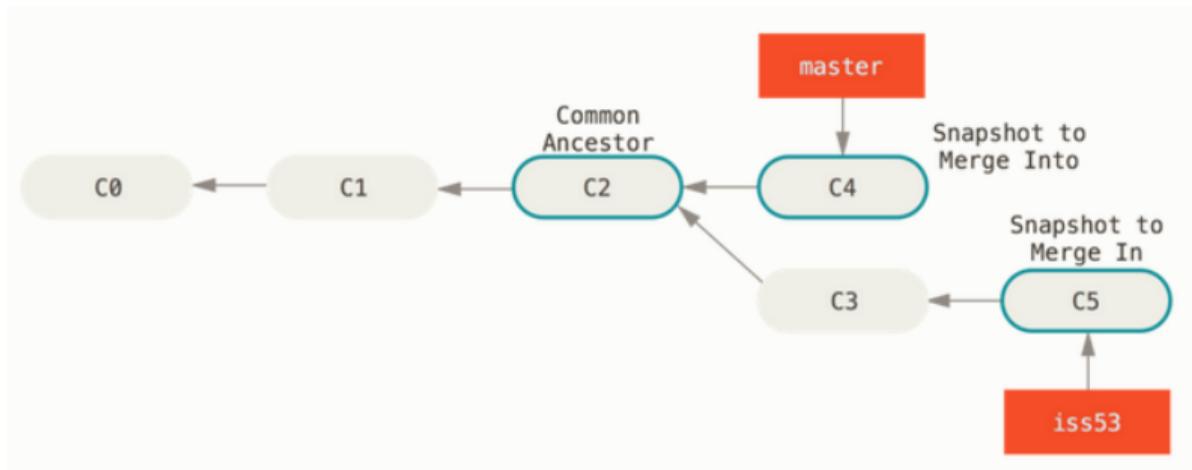
- ▶ `git checkout master` (on se met sur la branche master)
- ▶ `git merge hotfix` (on fusionne avec la branche hotfix)
- ▶ `git branch -d hotfix` (on supprime hotfix)



Fusion de iss53 avec la branche maîtresse après fix du problème iss53

Commit de fusion

- ▶ `git checkout master` (on se met sur la branche master)

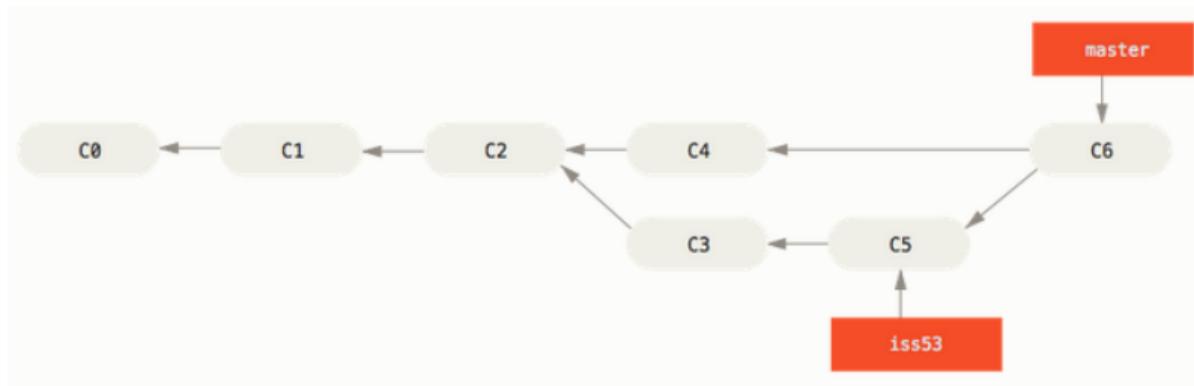


<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Fusion de iss53 avec la branche maîtresse après fix du problème iss53

Commit de fusion

- ▶ `git checkout master` (on se met sur la branche master)
- ▶ `git merge iss53` (on fusionne avec la branche iss53)



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Fusion de iss53 avec la branche maîtresse après fix du problème iss53

Commit de fusion

- ▶ `git checkout master` (on se met sur la branche master)
- ▶ `git merge iss53` (on fusionne avec la branche iss53)
- ▶ `git branch -d iss53` (on supprime iss53)

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Conflits de fusion : que faire ?

- ▶ identifier le problème

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> iss53:index.html
```

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Conflits de fusion : que faire ?

- ▶ identifier le problème
- ▶ `git mergetool` (on utilise l'outil approprié pour résoudre le problème)
ou éditer le fichier puis `git add`

Conflits de fusion : que faire ?

- ▶ identifier le problème
- ▶ `git mergetool` (on utilise l'outil approprié pour résoudre le problème)
ou éditer le fichier puis `git add`
- ▶ `git commit` (on valide les modifications)

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

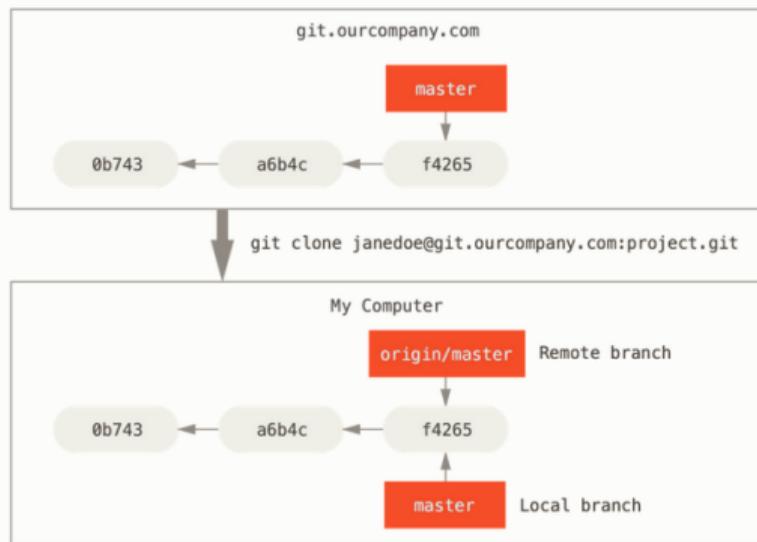
Bonnes pratiques

Pour finir

Remarque sur le nommage

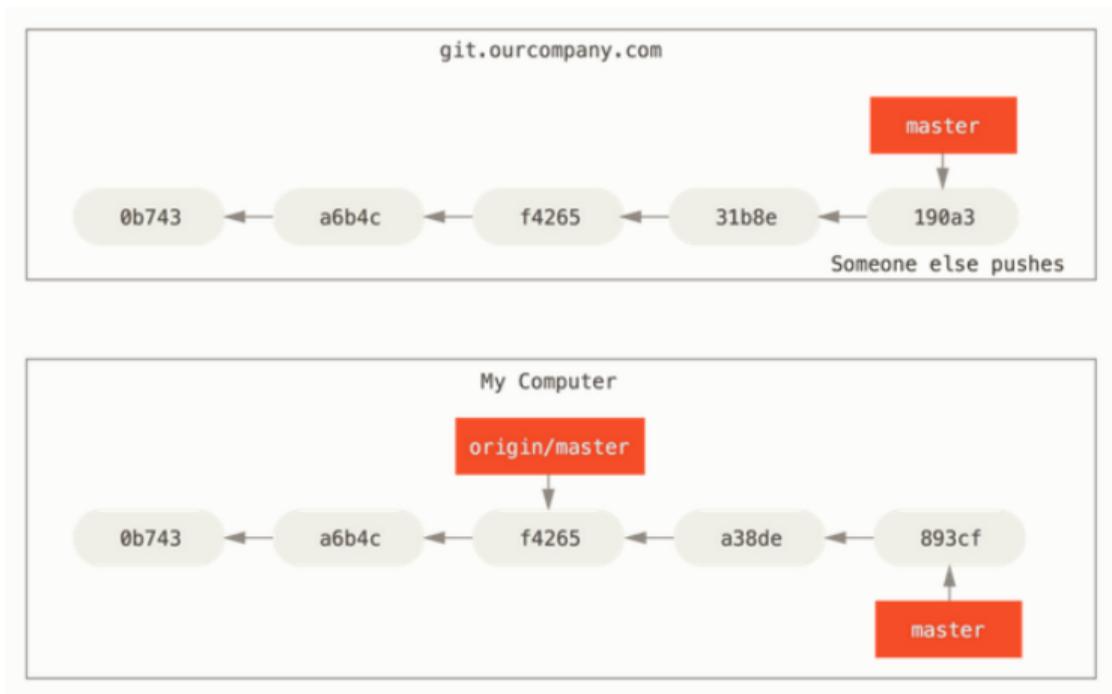
Quand on clone un dépôt distant, Git lui attribue le nom " **origin** ", donc :

- ▶ **master** fait référence à votre branche maîtresse (master) locale
- ▶ **remote/origin/master** fait référence à la branche maîtresse du dépôt distant



Que se passe-t-il quand on travaille à plusieurs ?

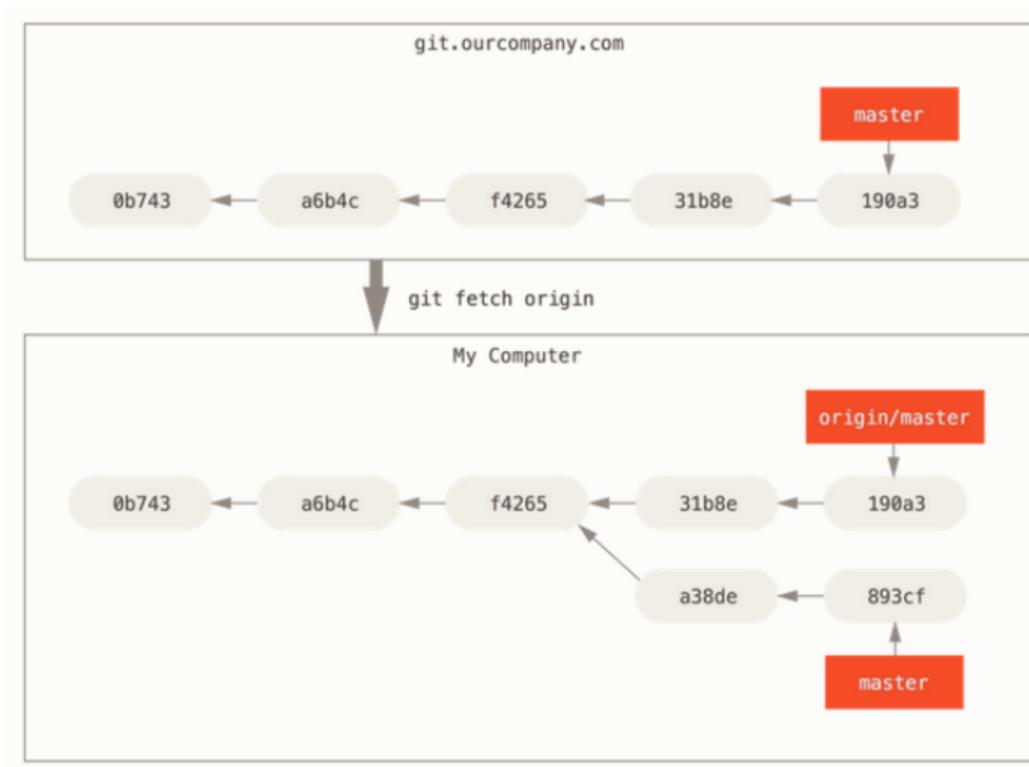
ça diverge



<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Branches-de-suivi-%c3%a0-distance>

Que se passe-t-il quand on travaille à plusieurs ?

se mettre à jour



Rappel : différence entre `git fetch` et `git pull`

?

Utilisation de GitLab : merge request

GitLab gère différents **droits** sur le dépôt

- ▶ quand vous voulez modifier des fichiers (push) en tant que développeur, vous devez demander la permission : merge request (avec une justification)
- ▶ les personnes disposant des droits afférents peuvent accepter votre demande (ou pas)

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

Bonnes pratiques

Pour finir

Bonnes pratiques GIT

- ▶ commit pas trop gros,

Bonnes pratiques GIT

- ▶ commit pas trop gros,
- ▶ commiter séparément les changements,

Bonnes pratiques GIT

- ▶ commit pas trop gros,
- ▶ commiter séparément les changements,
- ▶ messages utiles

Retour sur le dernier cours

Conflits de fusion

Branches

Fusionner des branches

Gestion des branches sur un dépôt distant

Bonnes pratiques

Pour finir

CQFR : Ce Qu'il Faut Retenir



- ▶ les conflits de fusion
- ▶ les branches