



Gestion des versions de code avec Git

**Charles Tholliez
Simon Hay**

**Formation de formateurs
M2 GPhy - 2017**

A large blue diagonal graphic element that starts from the top right and extends towards the bottom left, creating a triangular shape on the right side of the slide.

1.

Logiciel de gestion
de version :

Kézako ?

Quand on fait un projet informatique...



« Qui a modifié le fichier X, il marchait bien avant et maintenant il provoque des bugs ! » ;



« Qui a ajouté cette ligne de code dans ce fichier ? Elle ne sert à rien ! » ;



« À quoi servent ces nouveaux fichiers et qui les a ajoutés au code du projet ? » ;

Avantages et objectifs d'un logiciel de gestion de version



résolution des problèmes listés précédemment ;



suivre l'évolution d'un code source : Git est capable de dire qui a écrit chaque ligne de code de chaque fichier et dans quel but ;



travailler à plusieurs : Git est capable d'assembler (de fusionner) leurs modifications et d'éviter que le travail d'une de ces personnes ne soit écrasé.

Différents

logiciels de gestion de versions

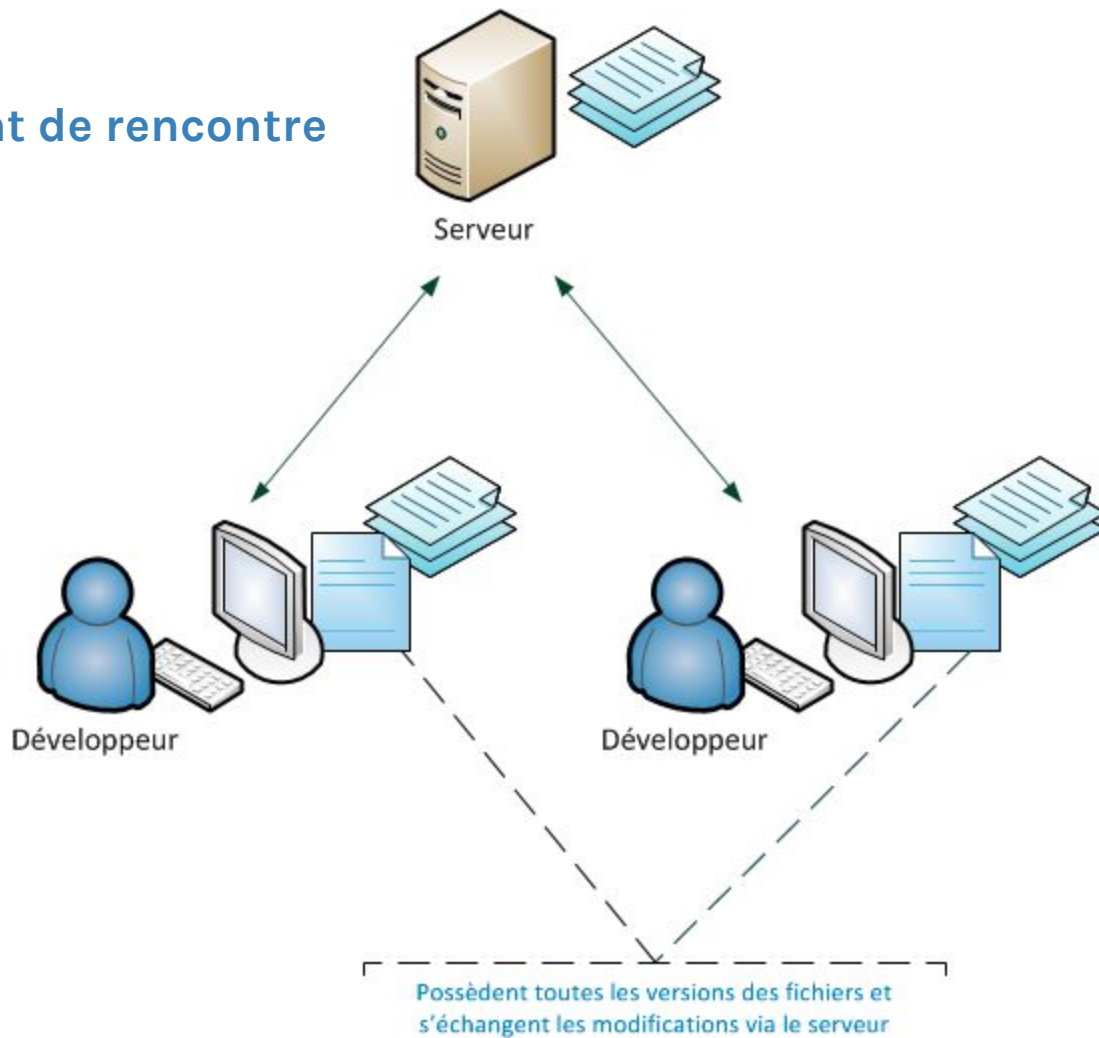
SUBVERSION®



2016 :
12 M d'utilisateurs
(#1 dans le monde)

Structure de Git

Point de rencontre





À savoir
sur Git...

Des interfaces graphiques...

À l'origine...



```
MINGW64/c/git-workspace
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ start test
test.txt      test2.txt      test-simon.txt
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ start test.txt
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ git add test.txt
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ git stus
git: 'stus' is not a git command. See 'git --help'.
The most similar command is
  status
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test.txt
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$ git gui
Simon_Hay@SimonHay-HP MINGW64 /c/git-workspace (master)
$
```

- En standard avec Git

git gui

- Service d'hébergement web



A thick, solid green diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from the solid green background on the right.

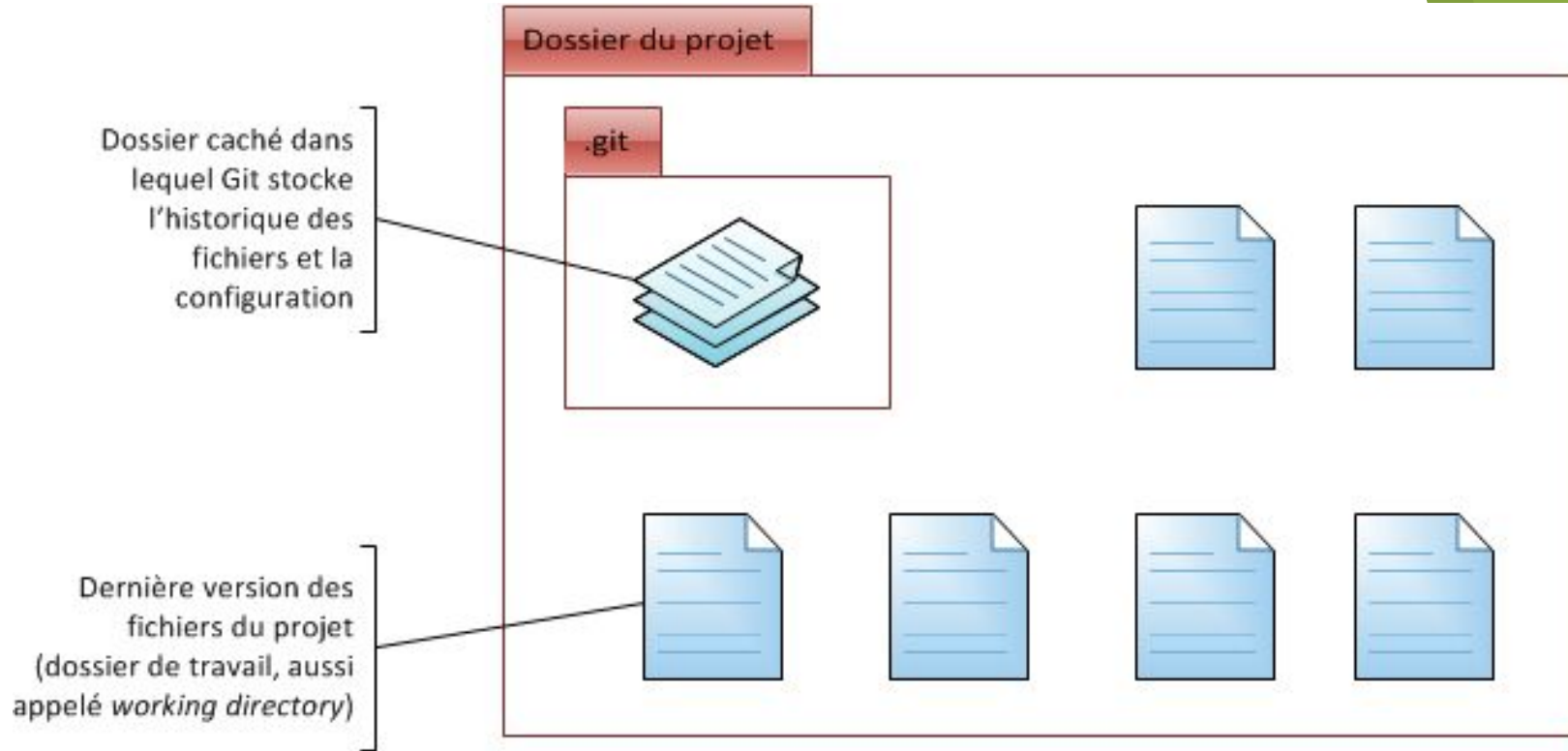
2.

Commencer à
travailler avec **Git**

Deux solutions...

- création d'un dépôt vide pour commencer un nouveau projet.
- clone d'un dépôt existant, c'est à dire récupération de tout l'historique des changements d'un projet pour pouvoir travailler dessus.

Un dépôt ?



À vous de jouer !

<https://github.com/HellCarlito/Formation>

- doc : TP + Commandes utiles au TP + Slides avec commentaires
- README : liens de téléchargement de Git
- Installez git
- Faites l'exercice 1

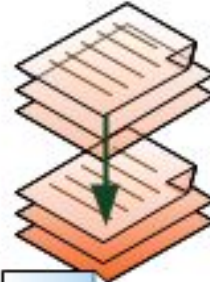
Enregistrer le travail en local : **Commit**

**Pas
d'interaction
avec le serveur**



Serveur

Nouvelles
modifications



`git commit`

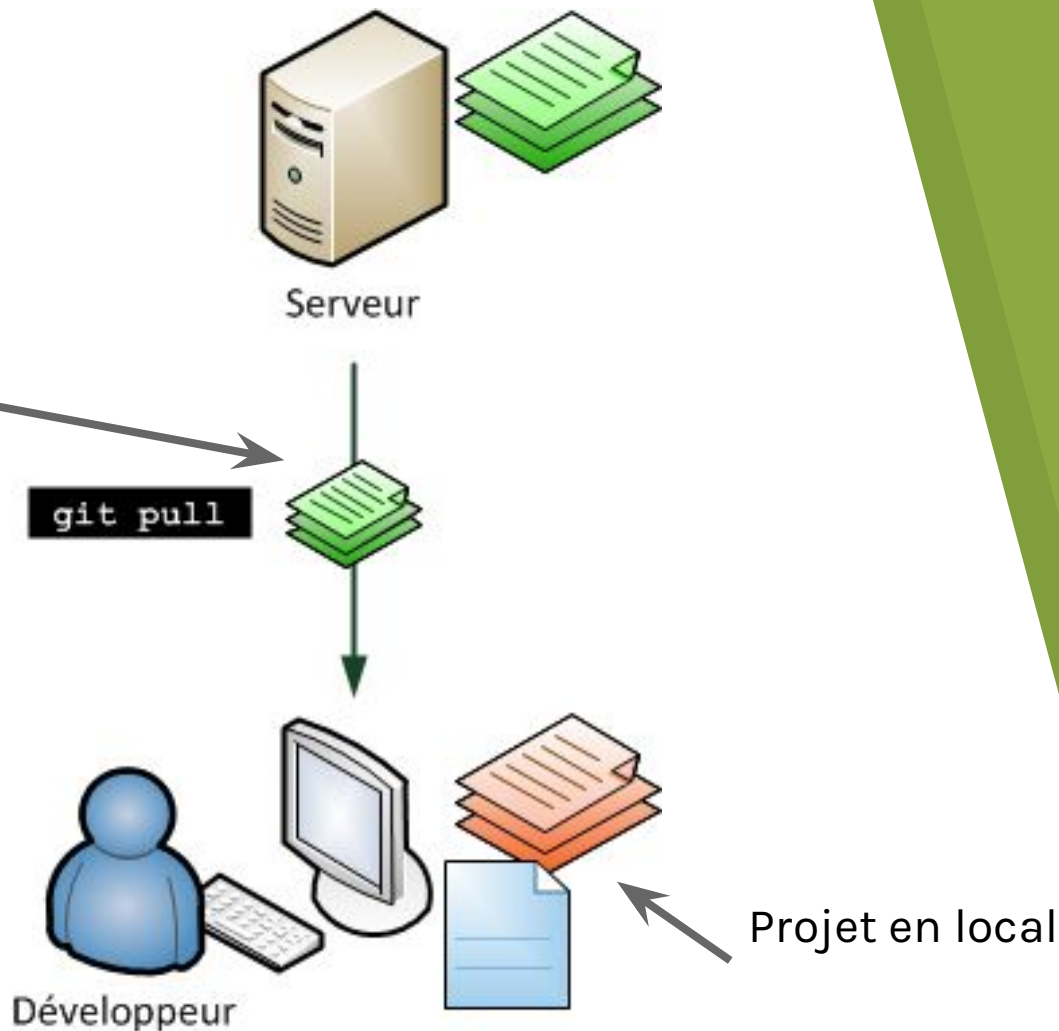
Projet en local



Développeur

Récupérer les commits des autres : **Pull**

Changements effectués par d'autres personnes



Envoyer vos commits : **Push**

Envoi des modifications
préalablement
“enregistrées” en local



I WANT YOU TO



COMMIT PULL PUSH!

Annuler proprement un commit :

2 cas de figures :

- Commit “non-poussé” (enregistrement local) : reset
- Commit “poussé”, donc enregistré sur le serveur : revert

La gestions des **conflits**

= Modification de la même zone de code, en même temps

Git ne peut pas décider quelle est la modification à conserver

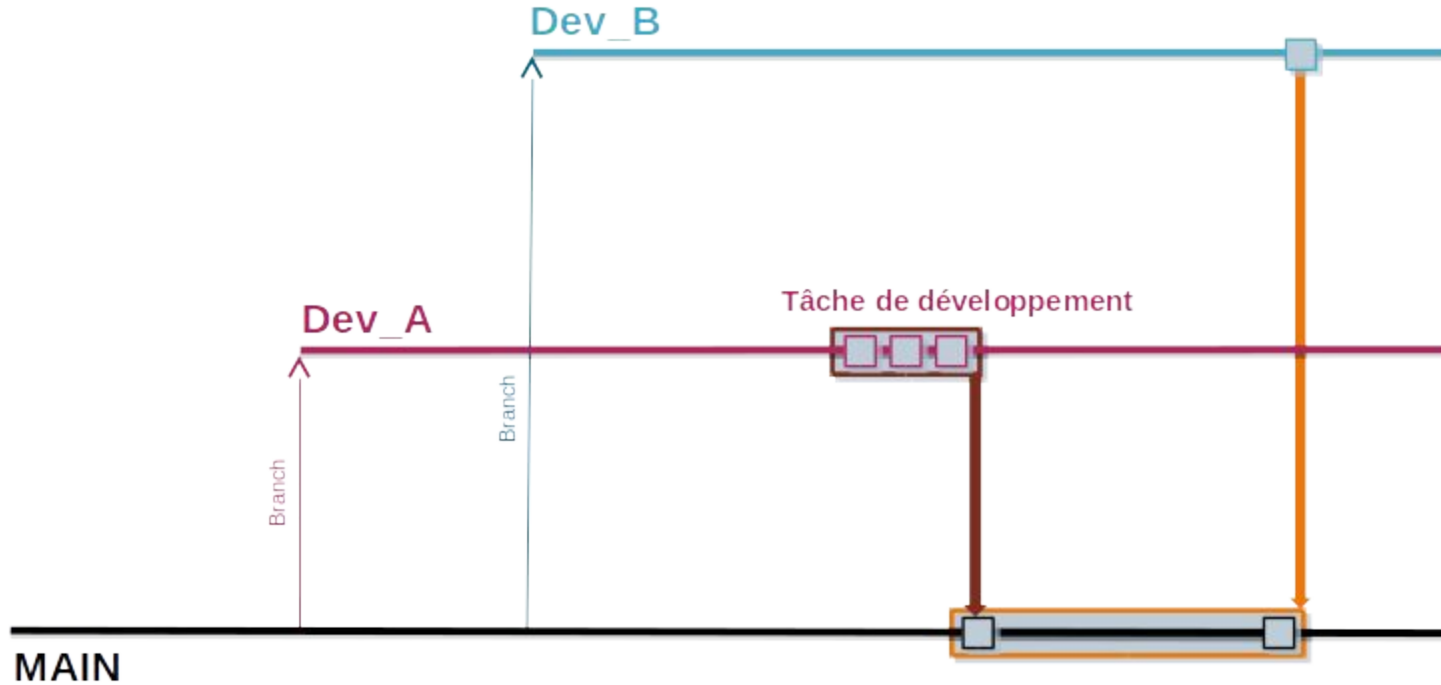
C'est à vous de décider manuellement quels sont les éléments à garder ou à supprimer

```
<<<<<< HEAD
J'écris sur cette ligne...
=====
J'écris sur cette ligne... Et moi aussi.
>>>>>> fee981d38f16d28082ec84daa1c61e56b2f16a29
```

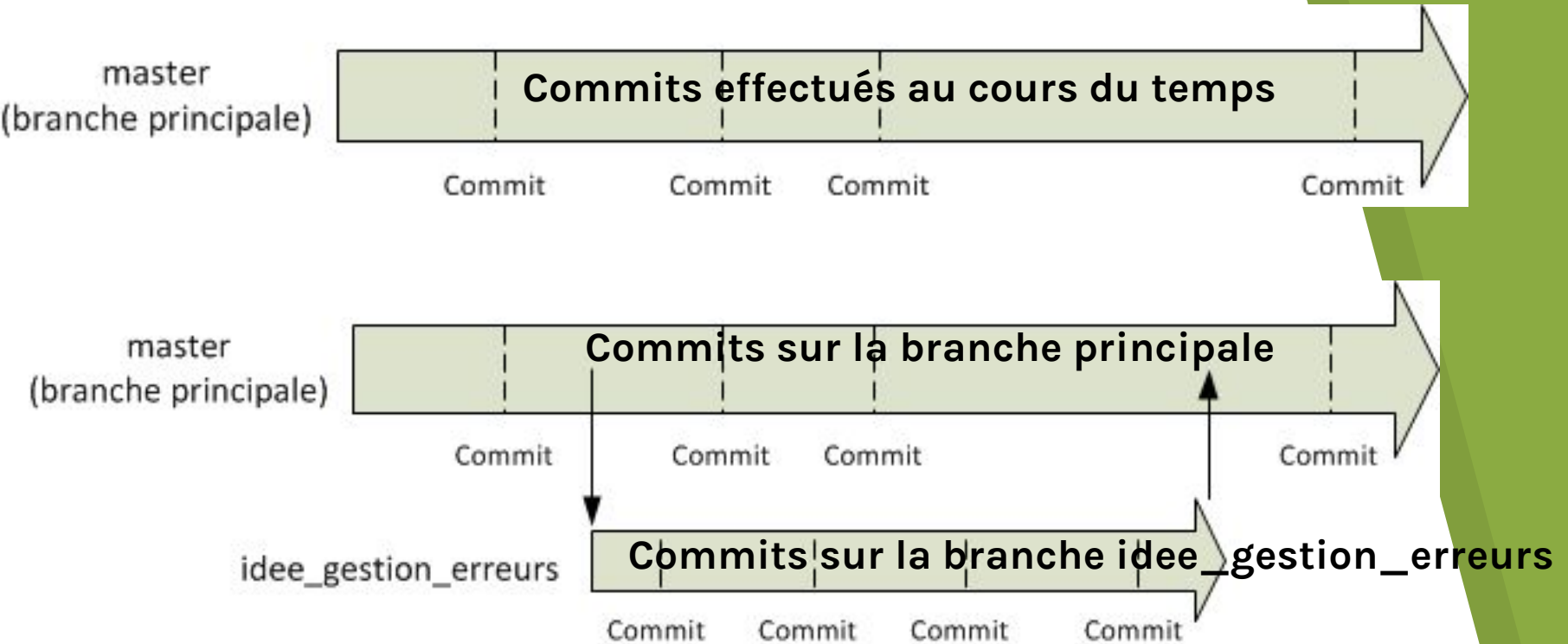
Travailler avec des **branches**

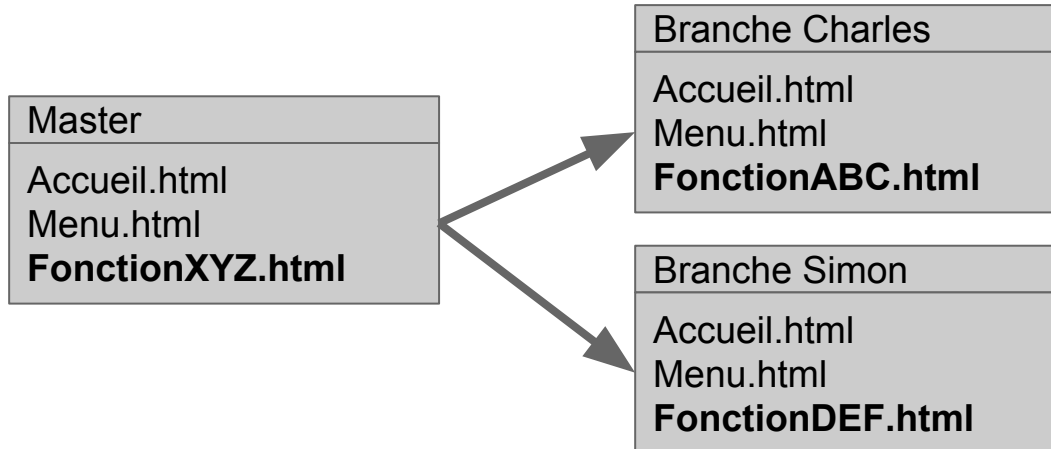
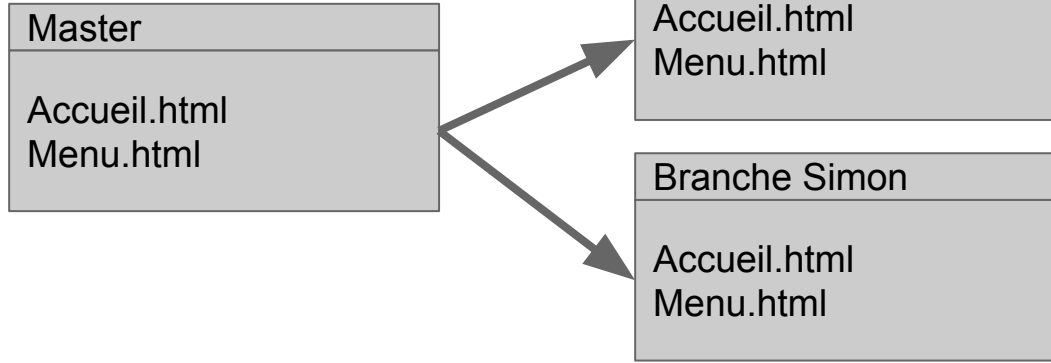
- ? « Ma modification sera-t-elle rapide ? » ;
- ? « Ma modification est-elle simple ? » ;
- ? « Ma modification nécessite-t-elle un seul commit ? » ;
- ? « Est-ce que je vois précisément comment faire ma modification d'un seul coup ? »

Travailler avec des branches

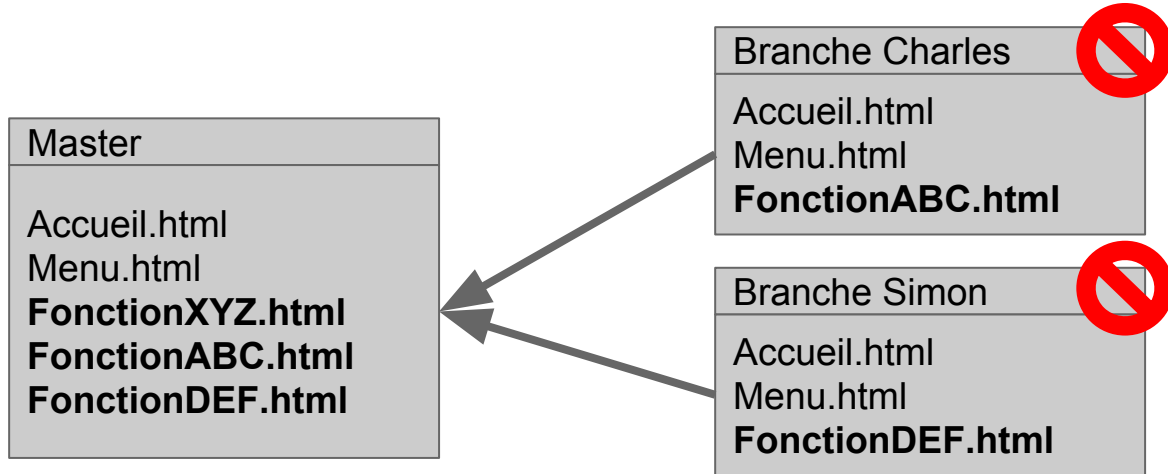


Travailler avec des **branches**



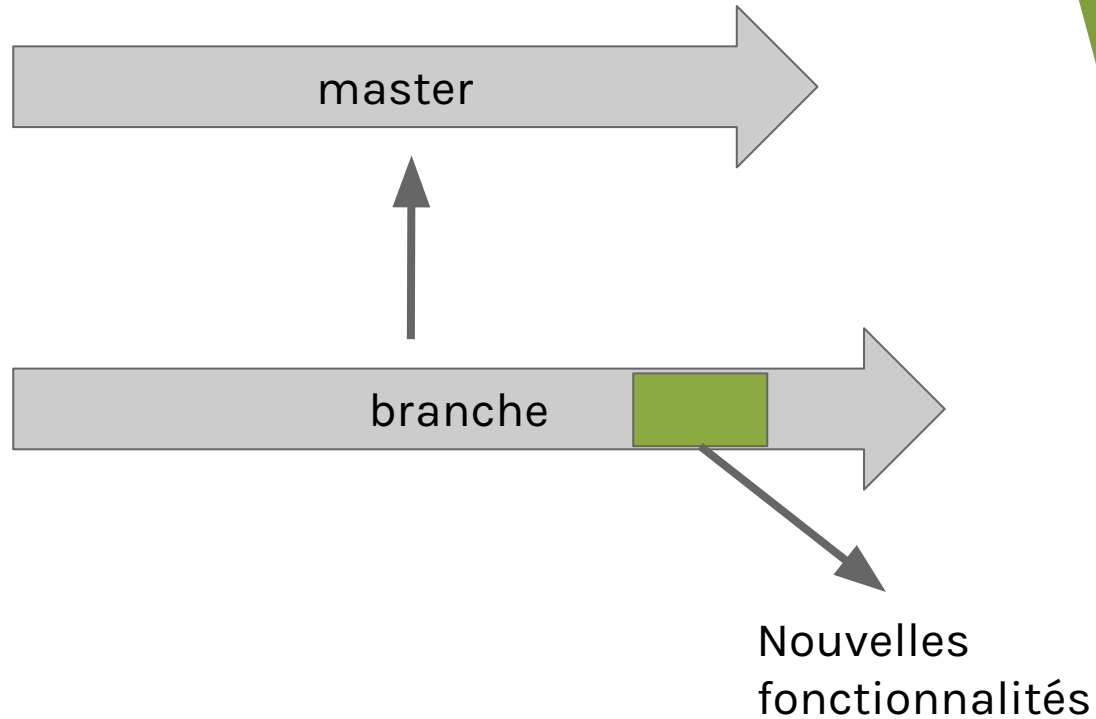


Travailler avec des branches



Merge

Fusionner les changements : **Merge**



BRATISLA BOYS



STACH STACH

LE TITRE RÉVÉLÉ par le



Mettre de côté : **Stash**

Situation :

- Modifications non-sauvegardées
- Vous voulez changer de branche ou faire un merge

→ Solution : Commit ? Annulation des modifications ?

Stash : met les modifications non-sauvegardées de côté dans un endroit dédié.

Pour restaurer les modifications : **stash apply**





Au travail !



Charles Tholliez
Simon Hay

Formation de formateurs
M2 GPhy - 2017