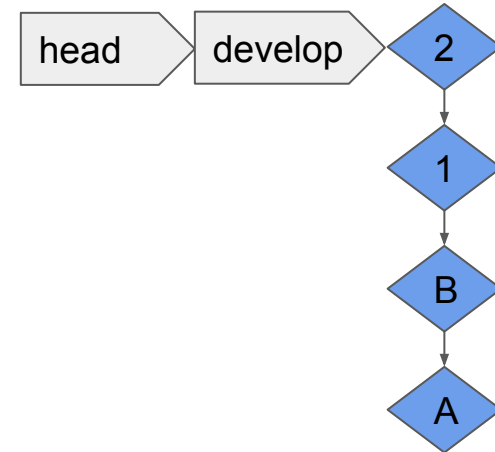
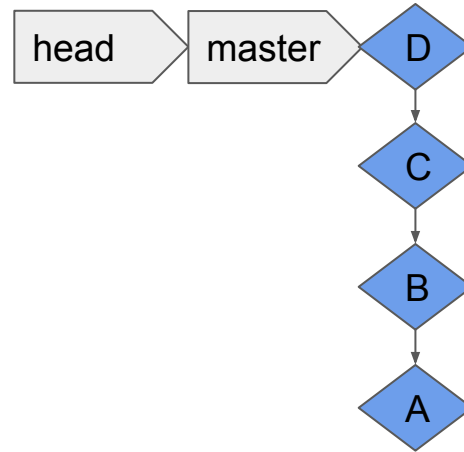
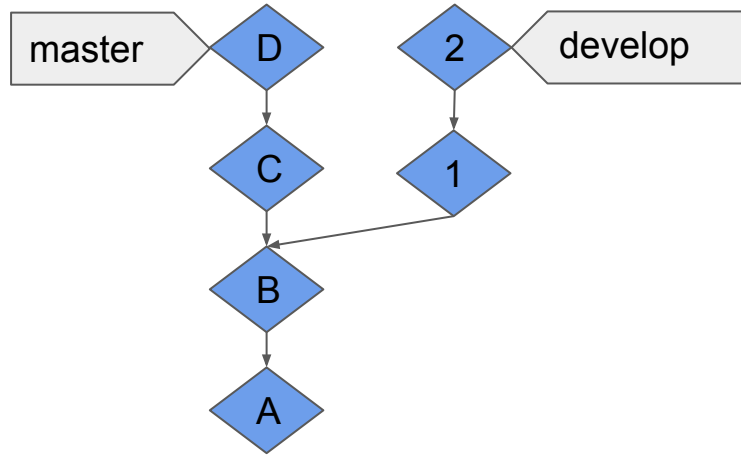


Formation Git

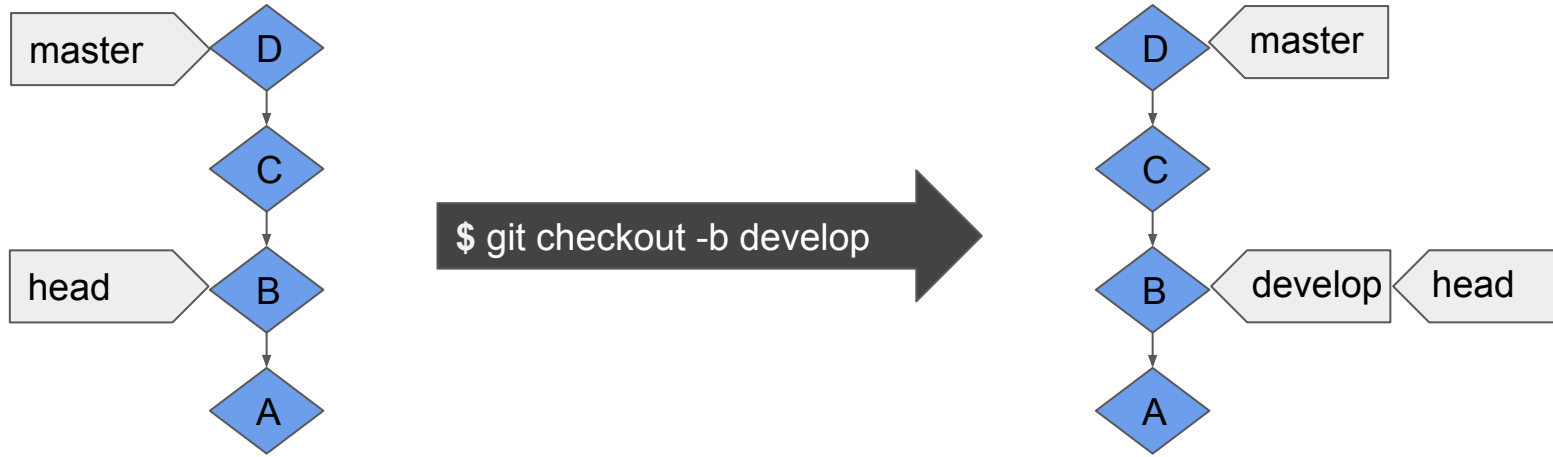
VI - Les branches



Le système de branches



Créer une branche



```
$ git branch [branch_name]  
$ git checkout [branch_name]
```

=

```
$ git checkout -b [branch_name]
```

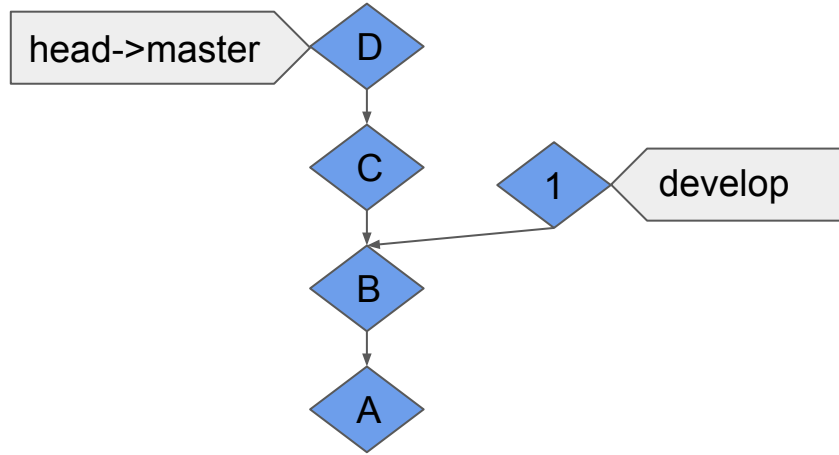
```
$ git branch [branch_name]  
$ git switch [branch_name]
```

=

```
$ git switch -c [branch_name]
```

git 2.24.0

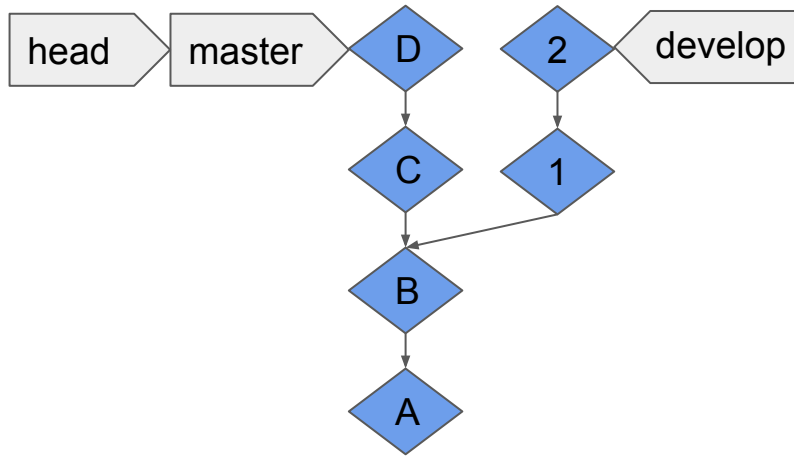
Lister les branches locales



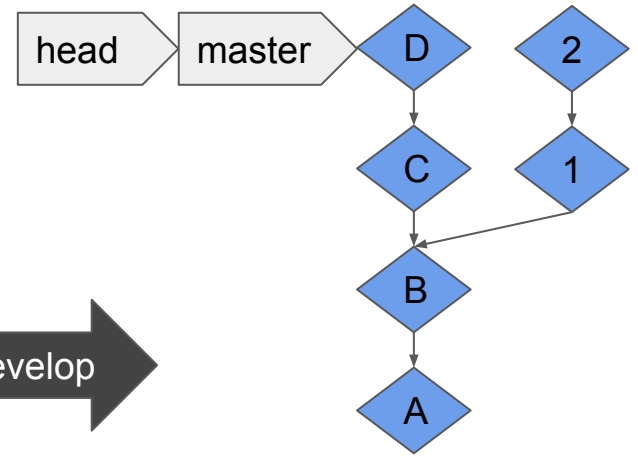
```
$ git branch  
* master  
develop
```

```
$ git branch // affiche les branches locales
```

Supprimer une branche



\$ git branch -d develop



\$ git branch -d [branche]



Il ne faut pas être sur la branche à supprimer

Voir l'historique

```
* commit 8a16f0301c192603761047146df513de21b985c2 (HEAD -> master, tag: MON_SITE_V5)
Author: Arnaud Mercier <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 19:05:35 2018 +0200

    ajout médaille d'or

* commit bd88f16cf89d6cbc2bbd6f0776567fb915683a70 (origin/master)
Merge: 92df0a3 6d73307
Author: clone <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 17:47:54 2018 +0200

    merge de ma branche BRANCH A MERGE avec ma master

* commit 6d73307cb5c9581d802816ea455c73d631eb3385 (origin/BRANCH_A_MERGE)
Author: clone <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 17:20:01 2018 +0200

    ajout git log

* commit 5f0e4576815501bc3bce4707c5de4f45ea07a994 (BRANCH_A_MERGE)
Author: Arnaud Mercier <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 17:06:28 2018 +0200

    ajout git push

* commit d4d4451ec7cf1186e312c341d91a0a3d0fa44665
Author: Arnaud Mercier <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 17:04:28 2018 +0200

    ajout commande git pull

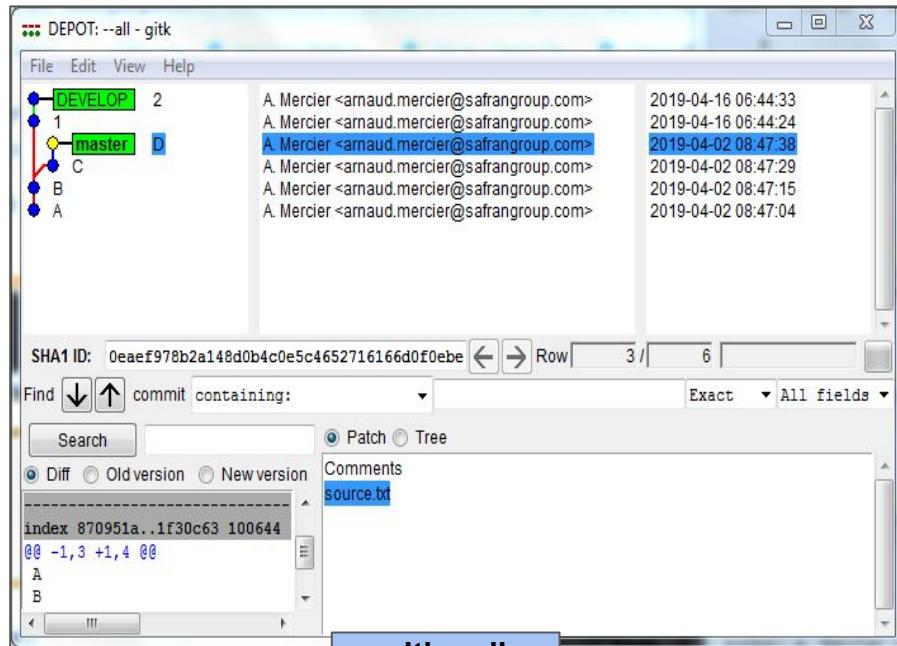
* commit 92df0e310f1bce1023e8aecdffe78d7a919dbba9
Author: Arnaud Mercier <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 17:04:28 2018 +0200

    ajout commande git pull

* commit e0a68a7debbe02994696bd0d04719b3e797cbe14 (tag: MON_SITE_V4)
Author: clone <arnaud.mercier@safrangroup.com>
Date: Wed Jun 13 16:16:56 2018 +0200

    ajout médaille argent
```

git log --graph --all



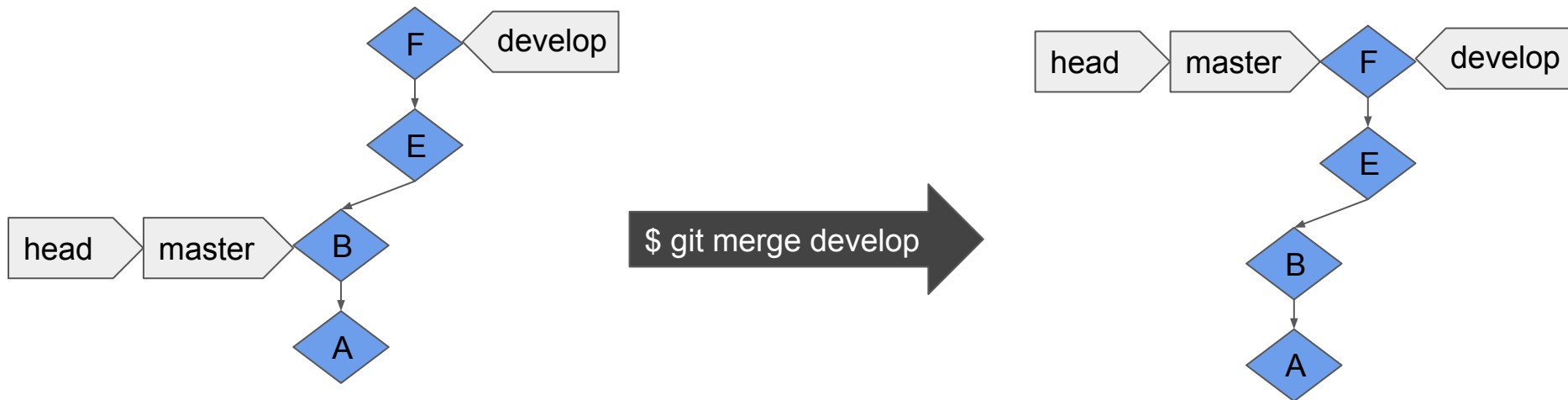
gitk --all

\$ git log [branch_name]

\$ git log --graph --all // voir sous forme de graph les branches

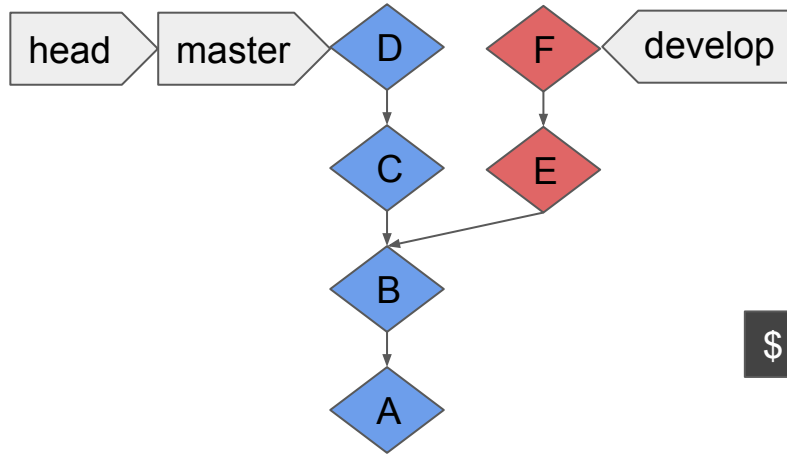
\$ git log --oneline --decorate --graph --all // super log

Faire un merge de branches: fast-forward

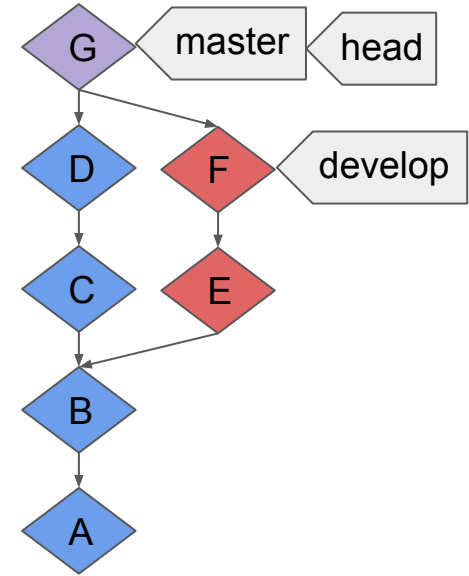


```
$ git checkout / switch [destination_branch]  
$ git merge [branche_a_merge]
```

Faire un merge de branches: commit de merge

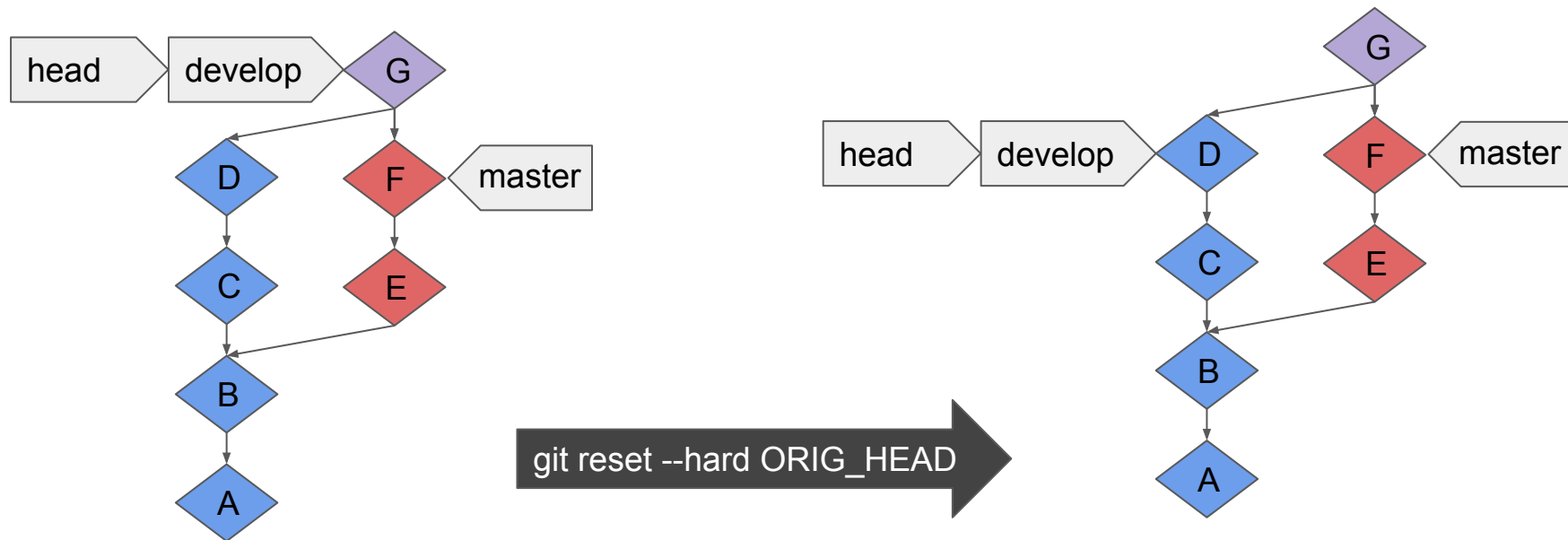


\$ git merge develop



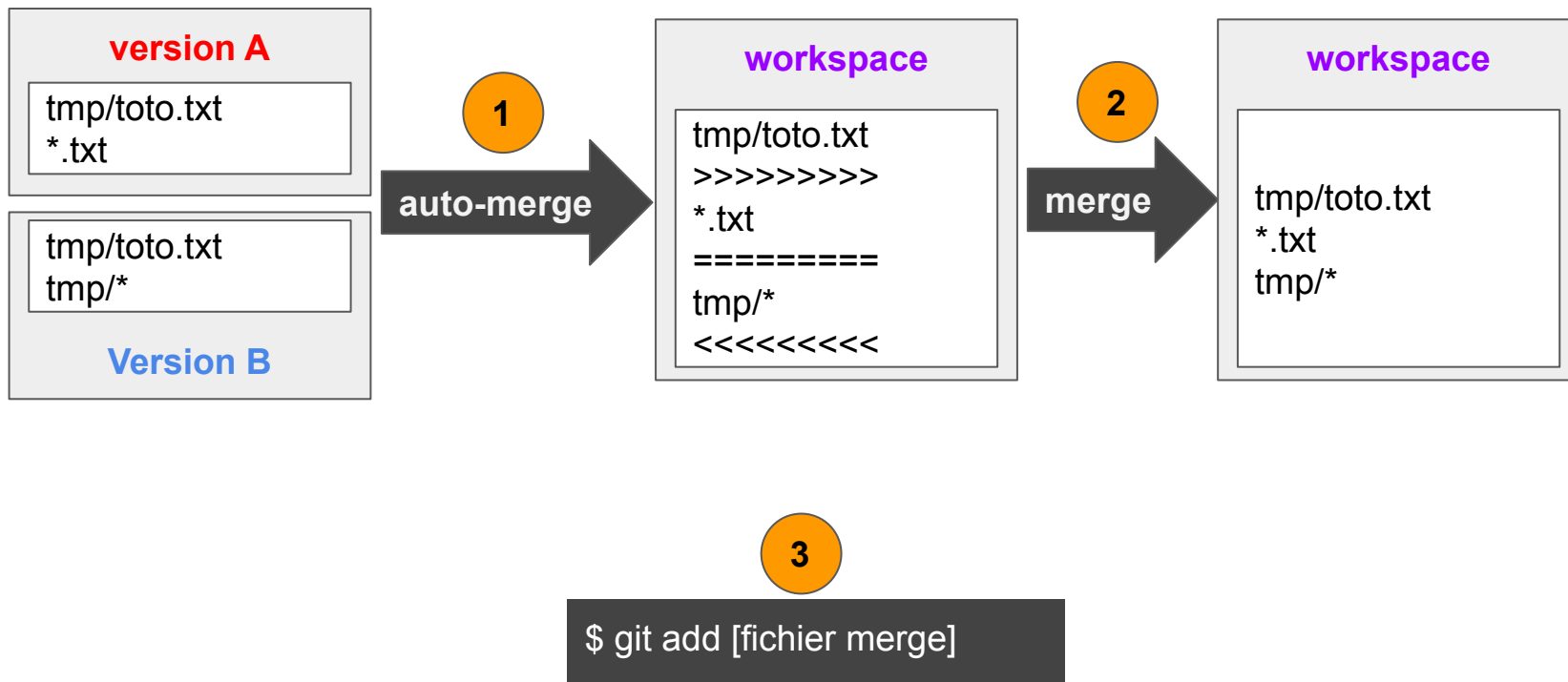
```
$ git checkout / switch [destination_branch]  
$ git merge [branche_a_merge]
```


Annuler un merge

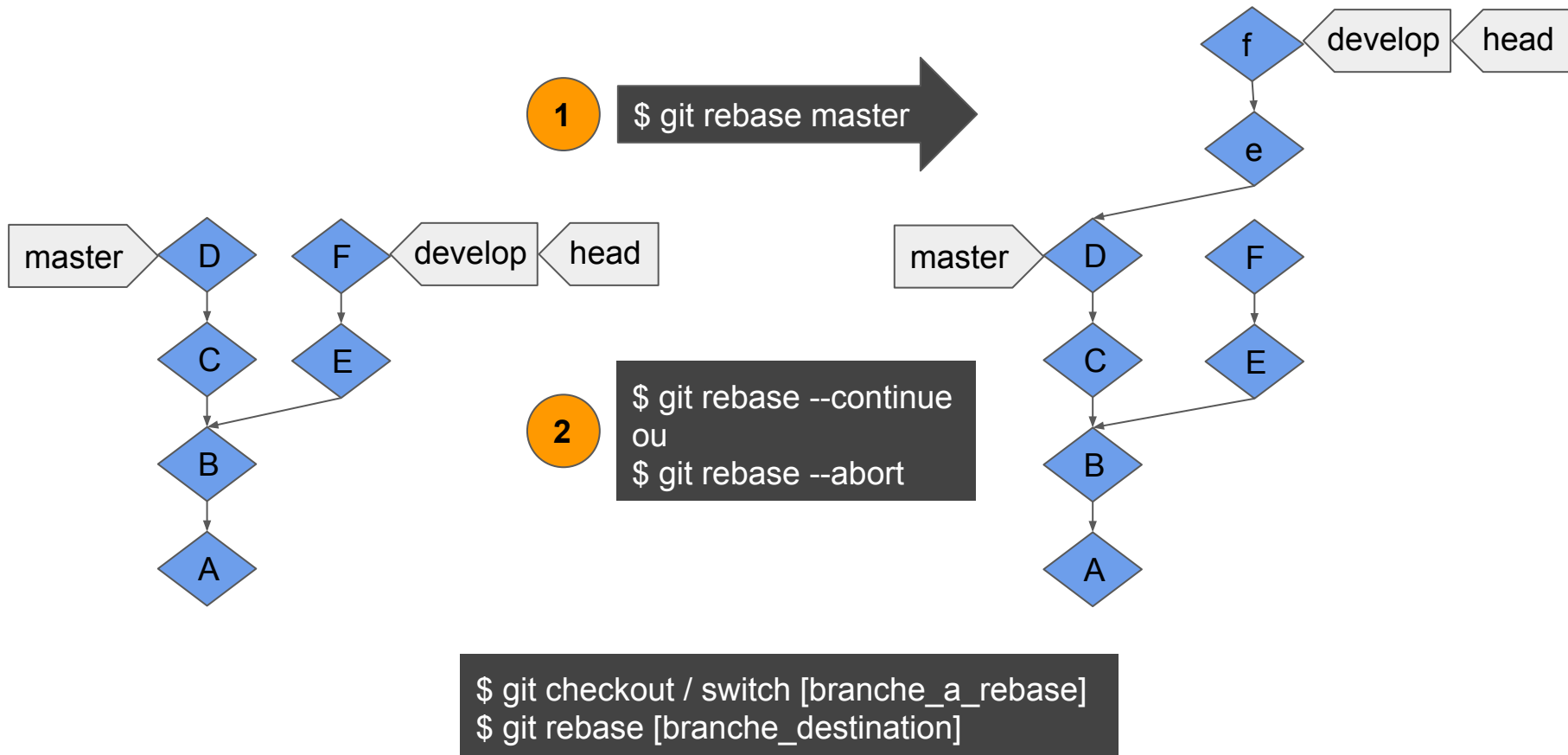


uniquement si aucunes grosse actions depuis le merge

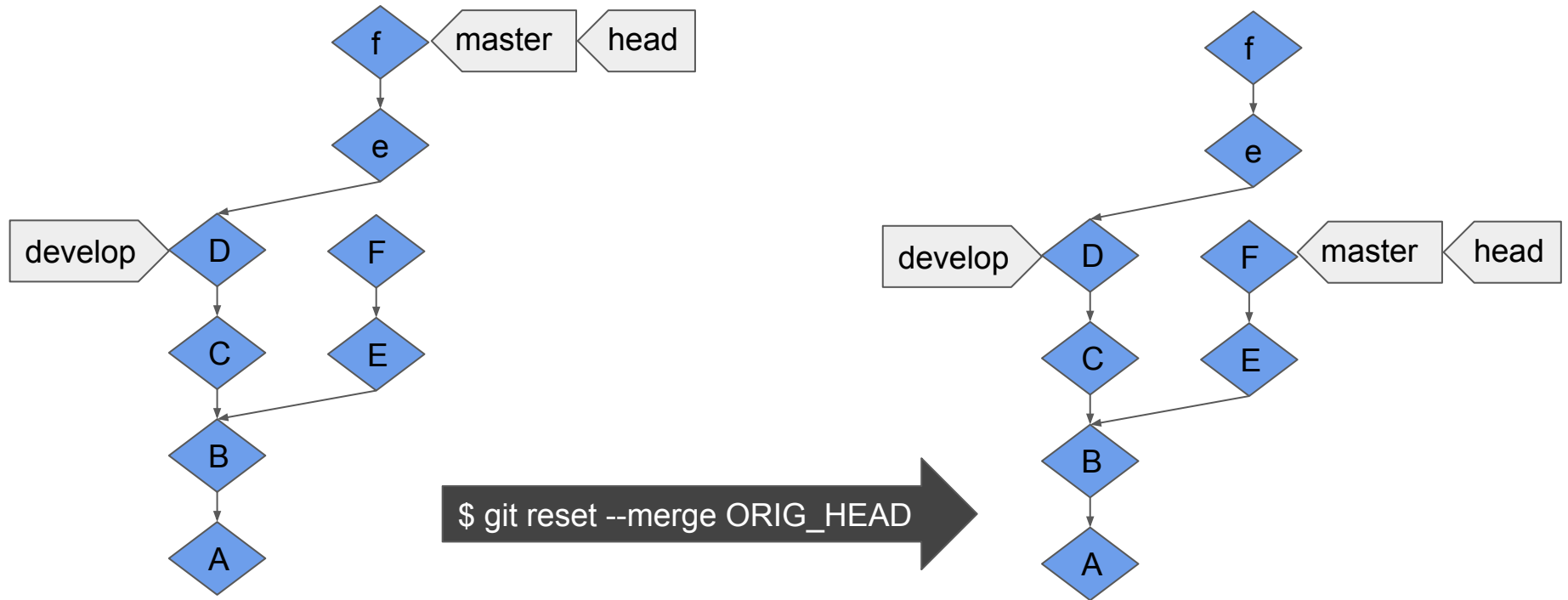
Résoudre un conflit



Faire un rebase d'une branche

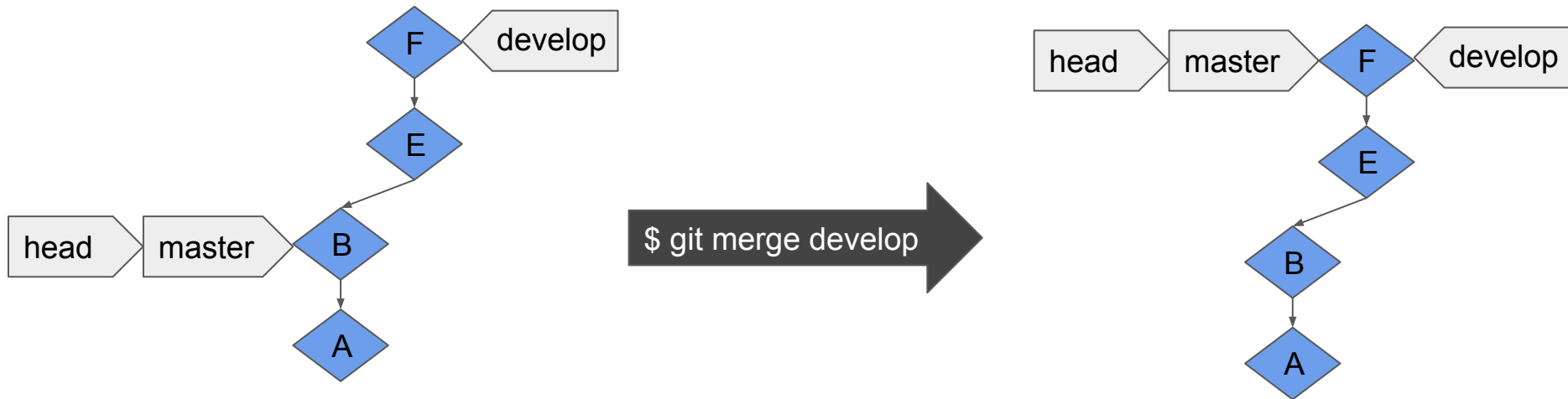


Annuler un rebase

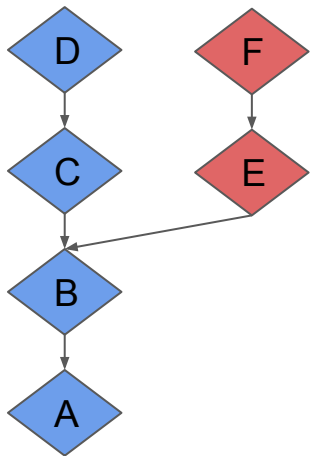


uniquement si aucunes autre actions depuis le merge

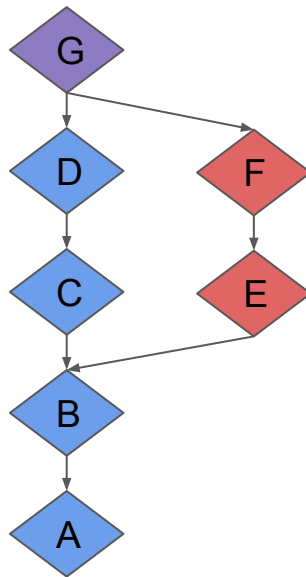
Merge une branche rebase



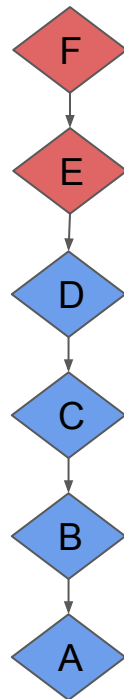
Merge Vs Rebase



Origin

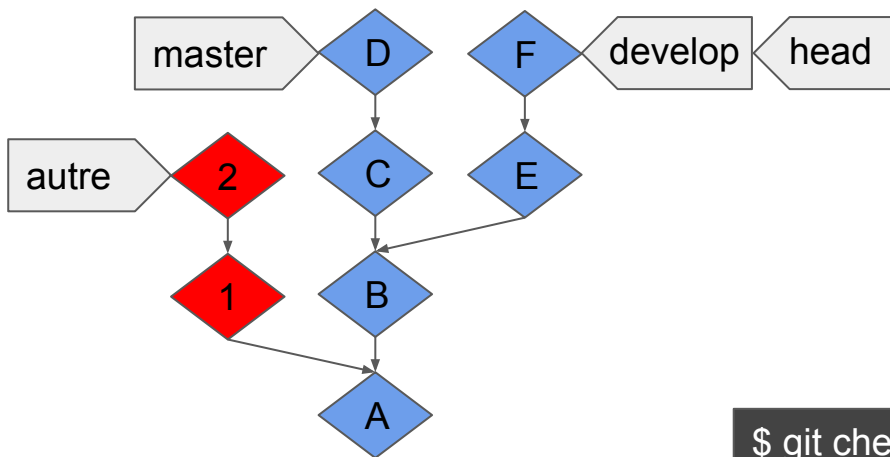


Merge

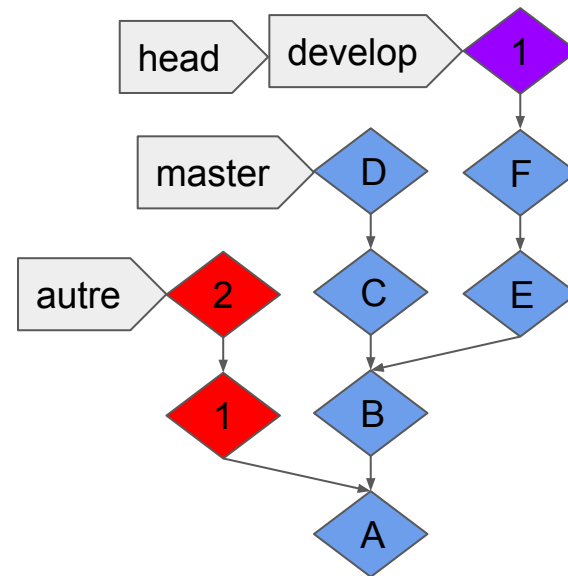


Rebase

Cherry pick



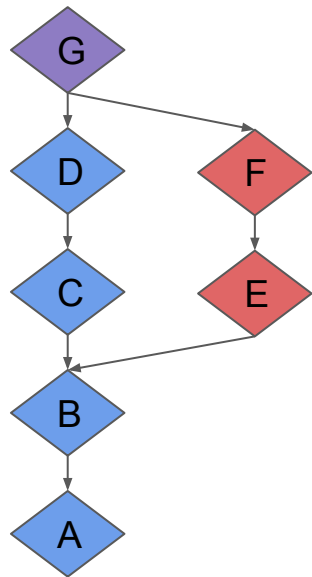
\$ git cherry-pick 1



\$ git cherry-pick [sha1 / ref]

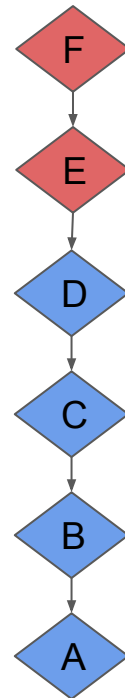
Bilan

merge



```
$ git checkout / switch <destination_branch>  
$ git merge <branche_a_merge>
```

rebase



```
$ git checkout / switch <branche_a_rebase>  
$ git rebase <destination_rebase>
```