

# GIT

@GillesRoustan

JIT ?

ON DIT GIT !!!



# about:me

@GillesRoustan

about:vcs

Qu'est-ce que c'est ?

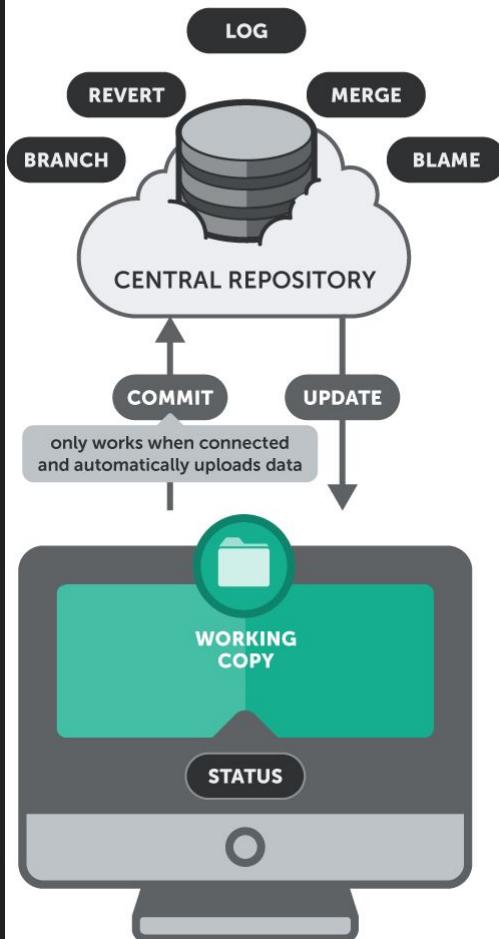
# Système de gestion de version

Pourquoi ?

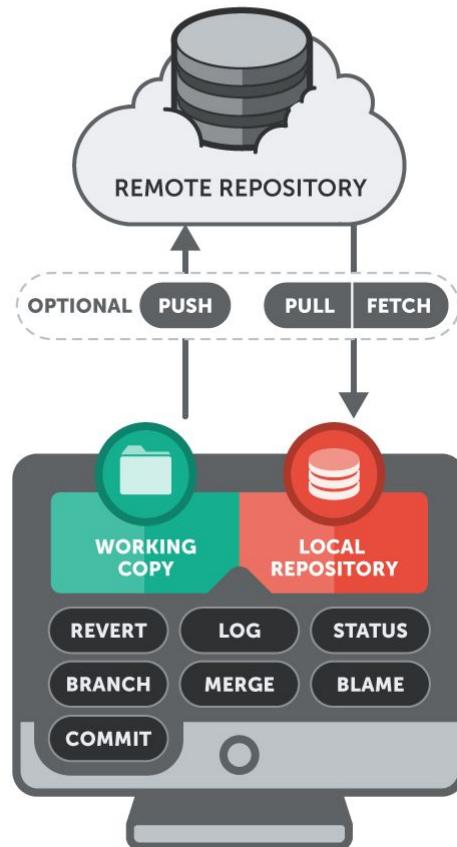
# Décentralisé

# DVCS

# SUBVERSION



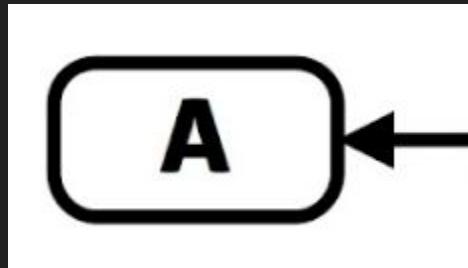
# GIT



source : <https://borntocode.fr/git-tutoriel-et-configuration-sur-le-gestionnaire-de-versions/>

# Vocabulaire

# commit



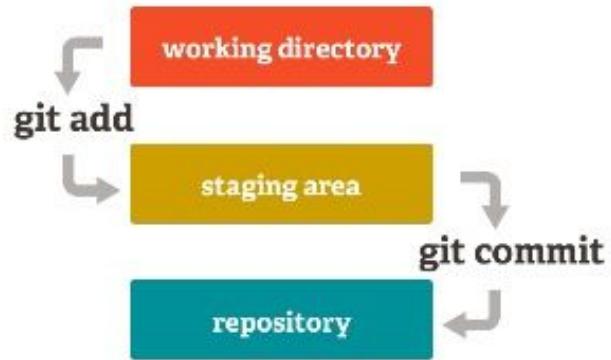
sha1

82dbc04f0d16950a7fcda3d  
e273025933a8e6ffd

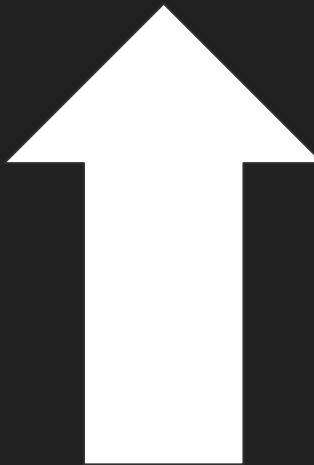
82dbc04

staging

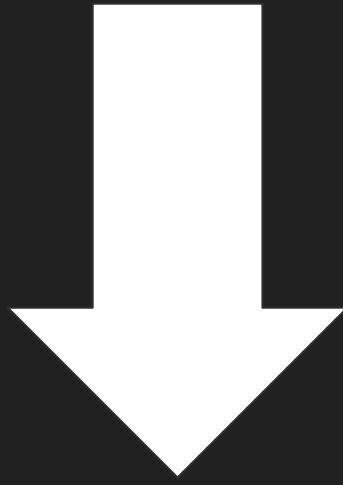
# Staging Area



push



pull



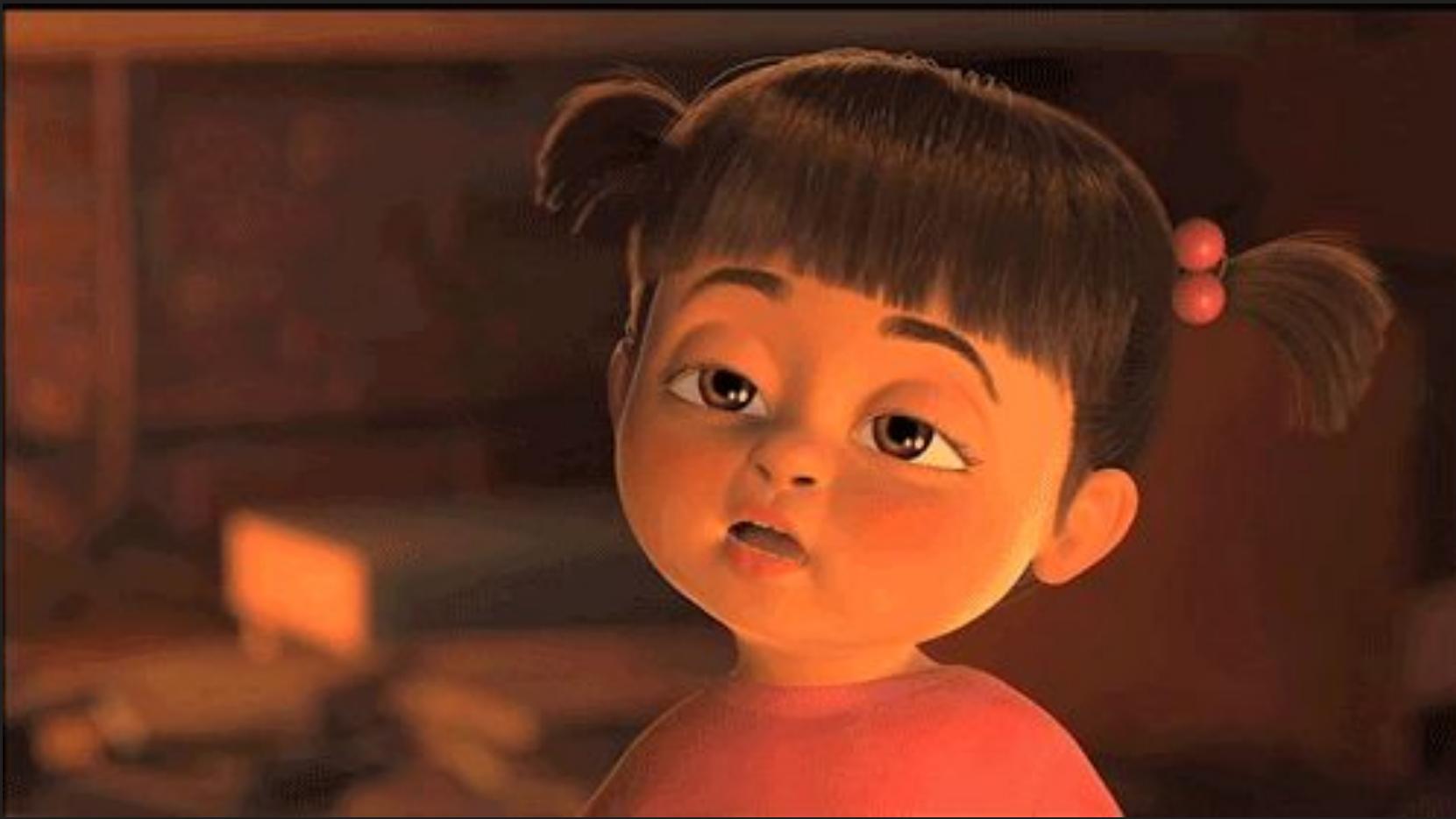
branch

merge

reset

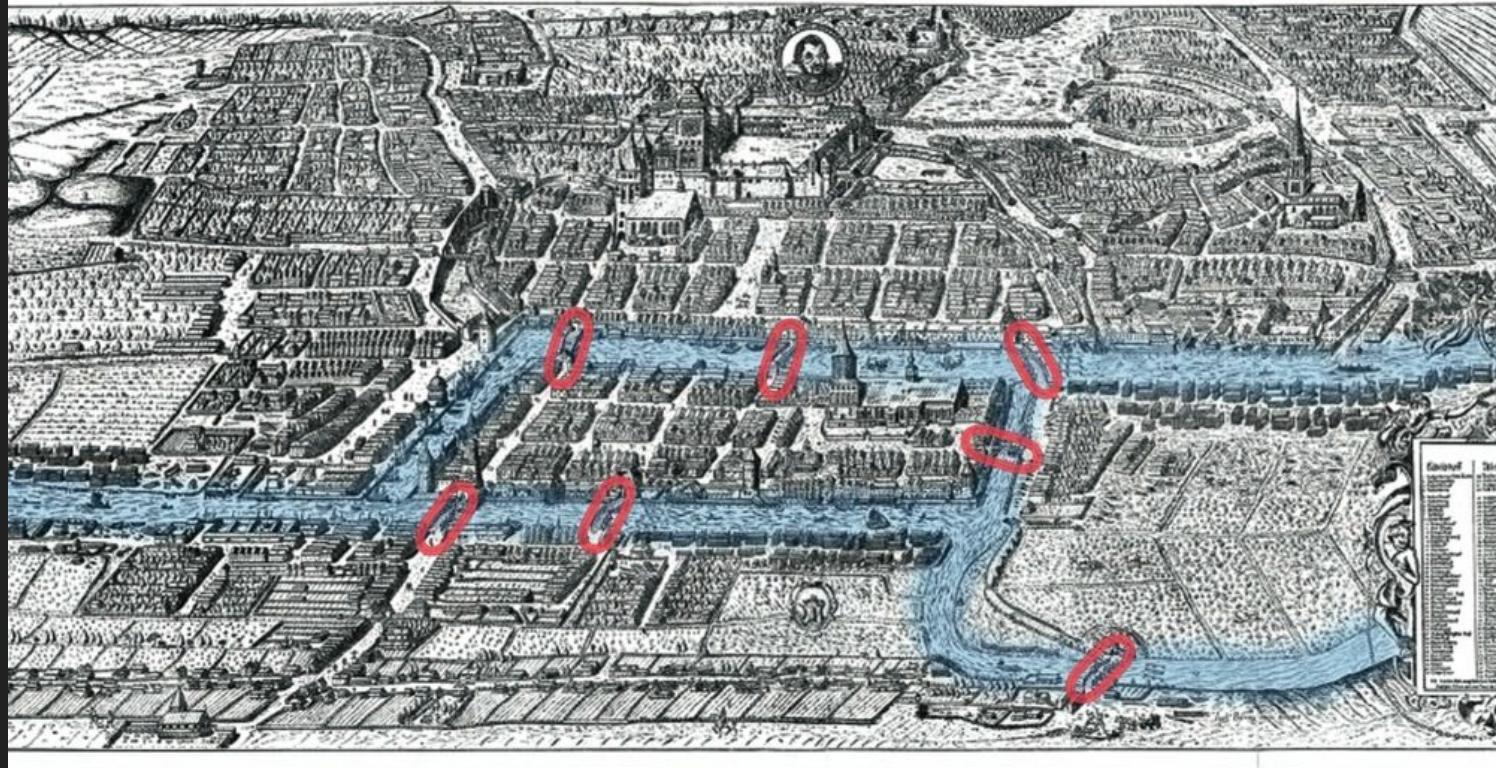
# rebase

Un peu de théorie ?

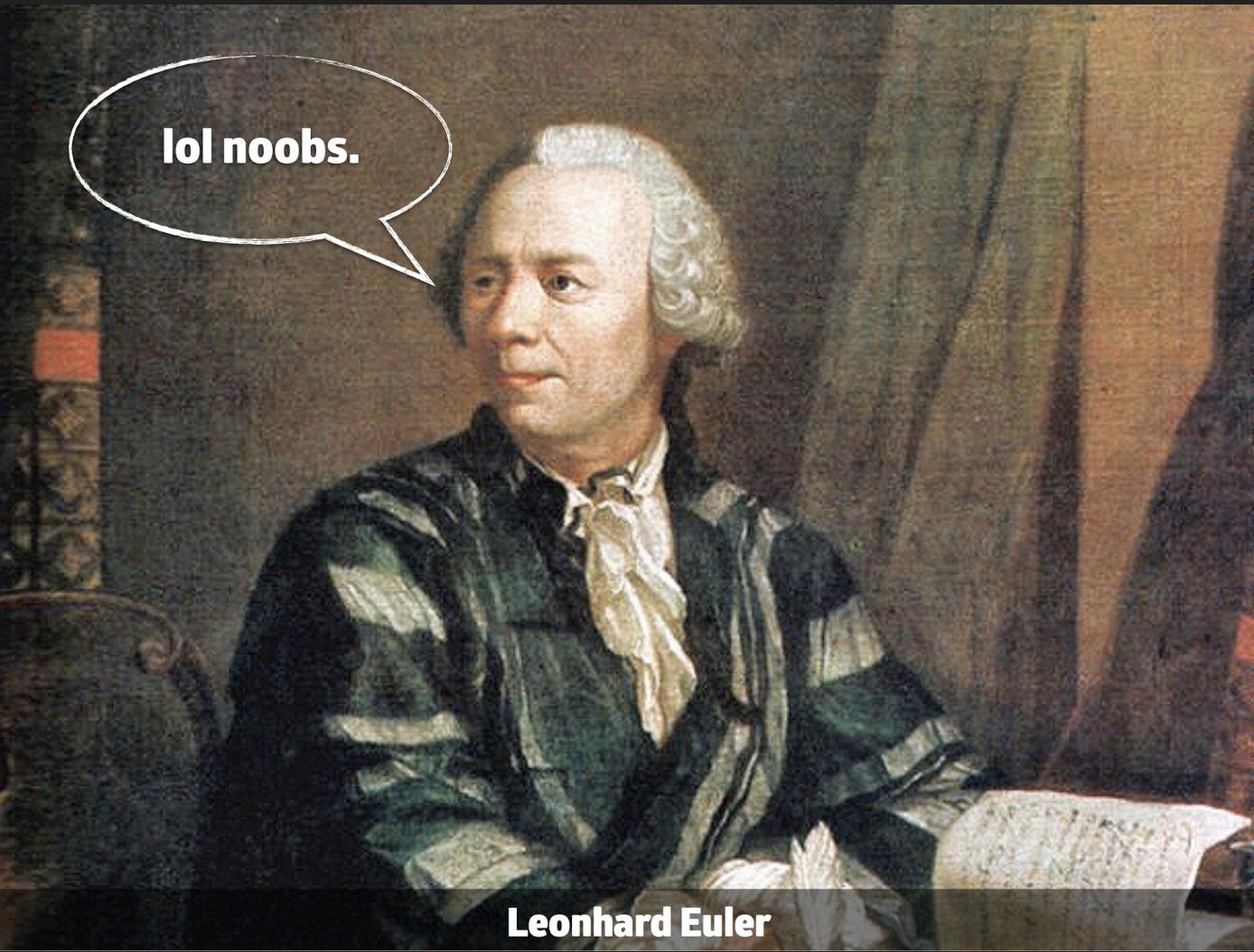


# La théorie des graphes

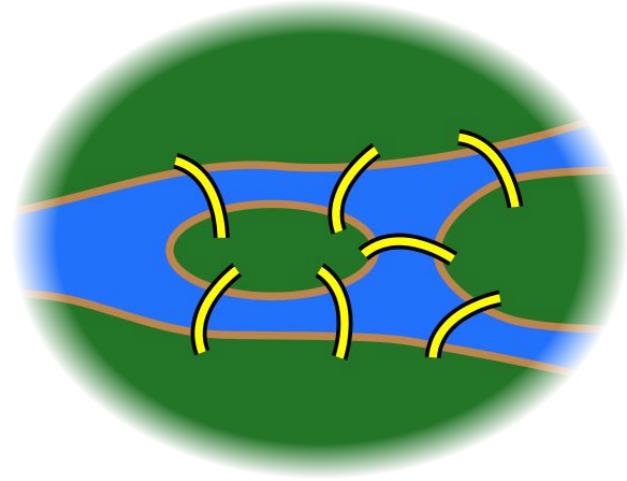
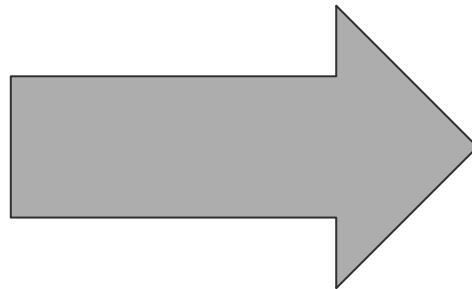
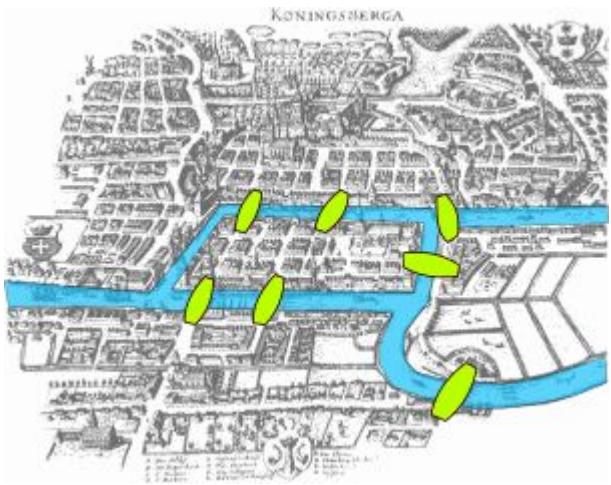
Gedenkblatt zur sechshundert jährigen Jubelfeier der Königlichen Haupt und Residenz-Stadt Königsberg in Preussen.

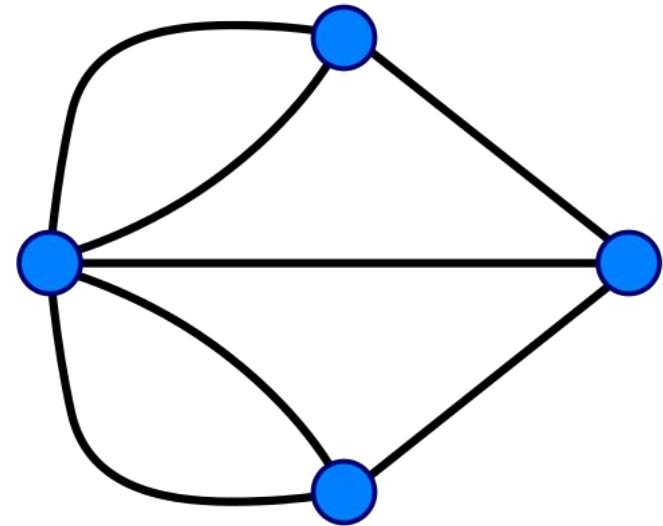
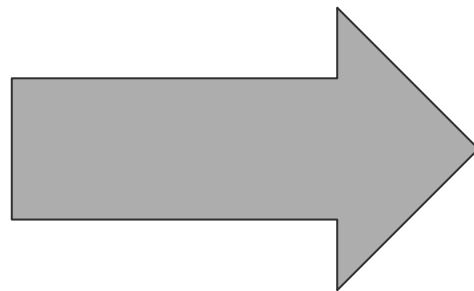
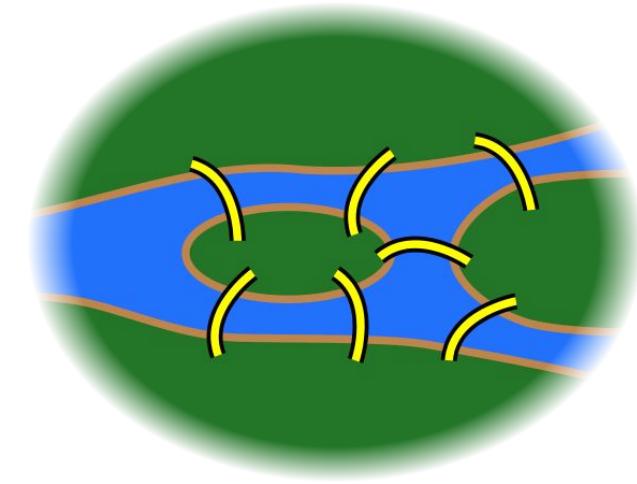


7 bridges of königsberg



**Leonhard Euler**





# GRAAAAPHES !!!



Un graphe est une collection d'éléments mis en relation entre eux.

Selon que l'on choisit d'orienter les arêtes ou de leur attribuer un poids (un coût de passage), on parle de graphes orientés ou de graphes pondérés.

**GIT?**





# Noeuds



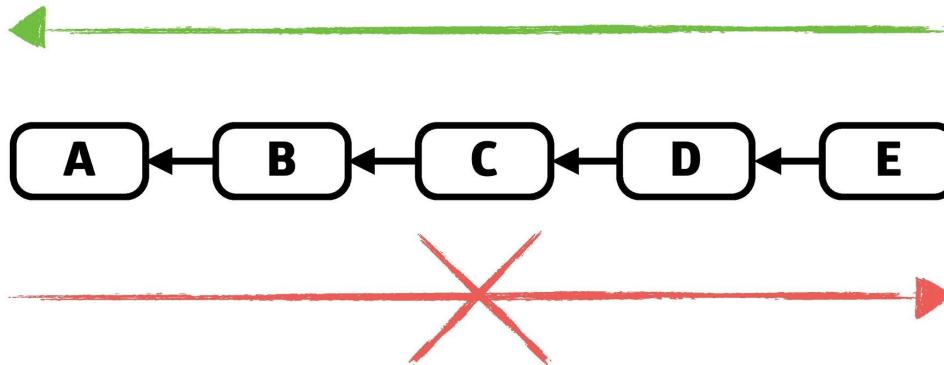
# Arêtes



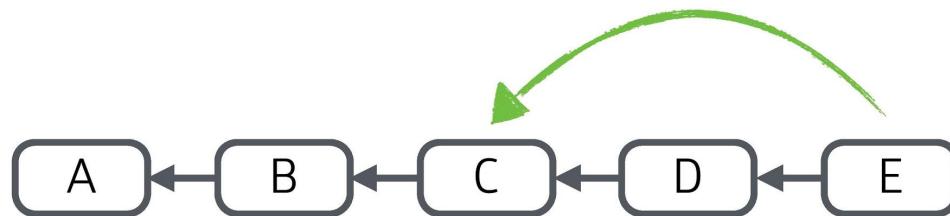
# Graph orienté



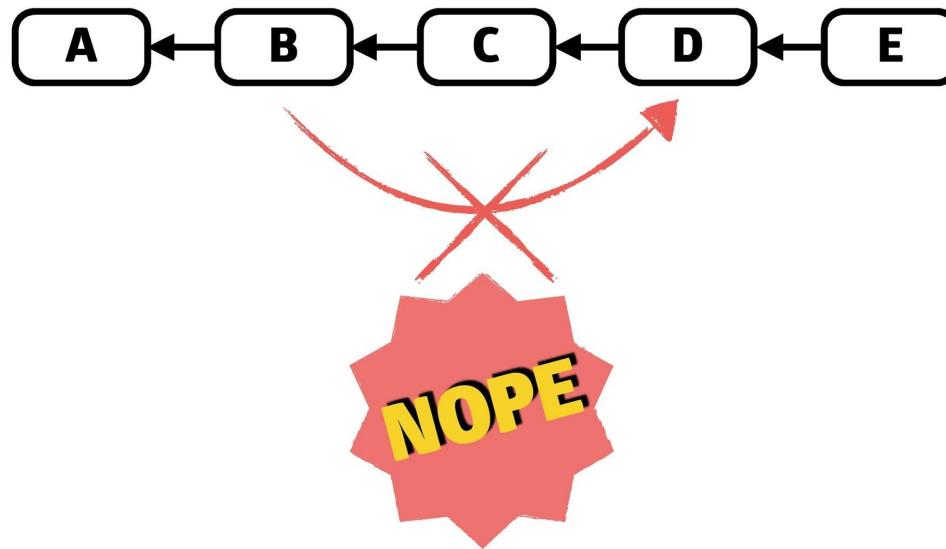
# Graph orienté



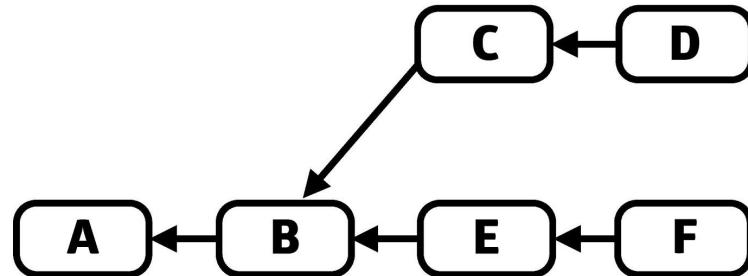
# Atteignabilité



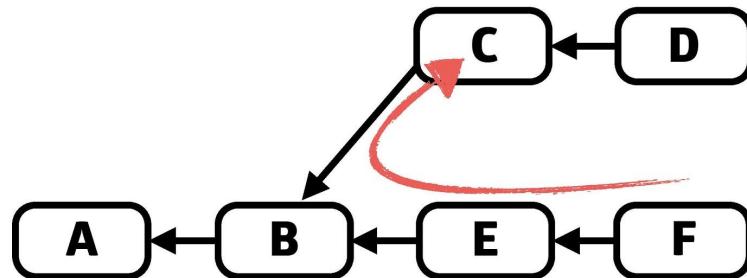
# Atteignabilité



# Atteignabilité



# Atteignabilité



D n'est pas atteignable depuis F

Qu'est-ce qu'une branche ?

```
$ ls .git/refs/heads/
```

```
master
```

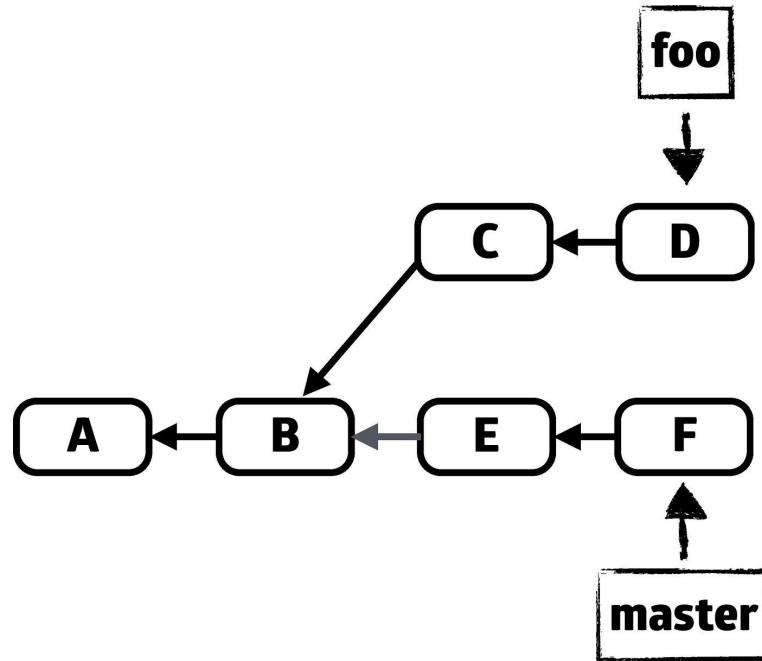
```
$ cat .git/refs/heads/master
```

```
58bd01cded7904f1596cb12aac1d3e0e8ae6b558
```

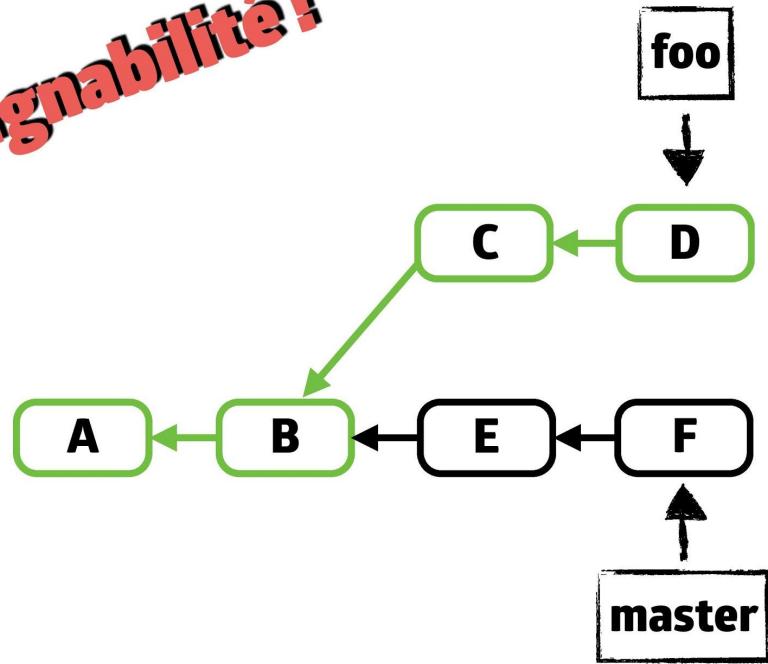
```
$ git branch
```

```
* master
```

```
foo
```

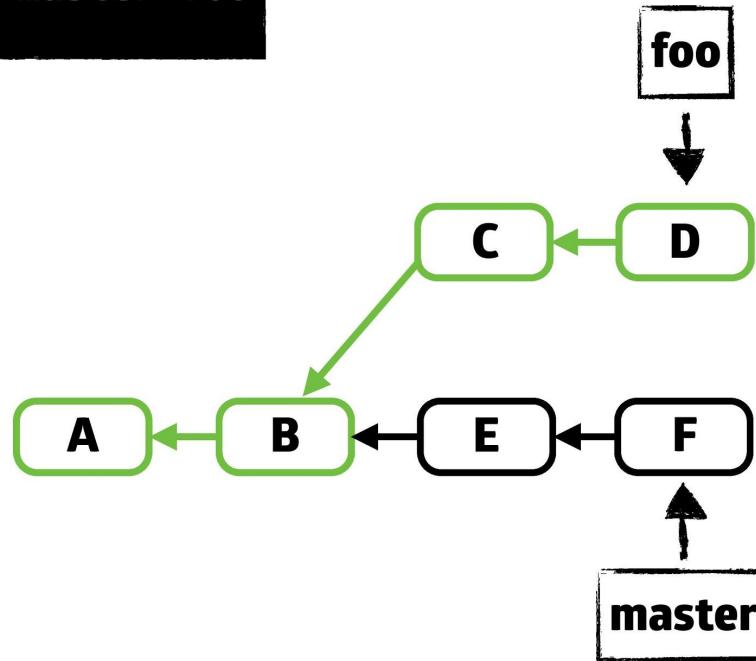


**Atteignabilité!**

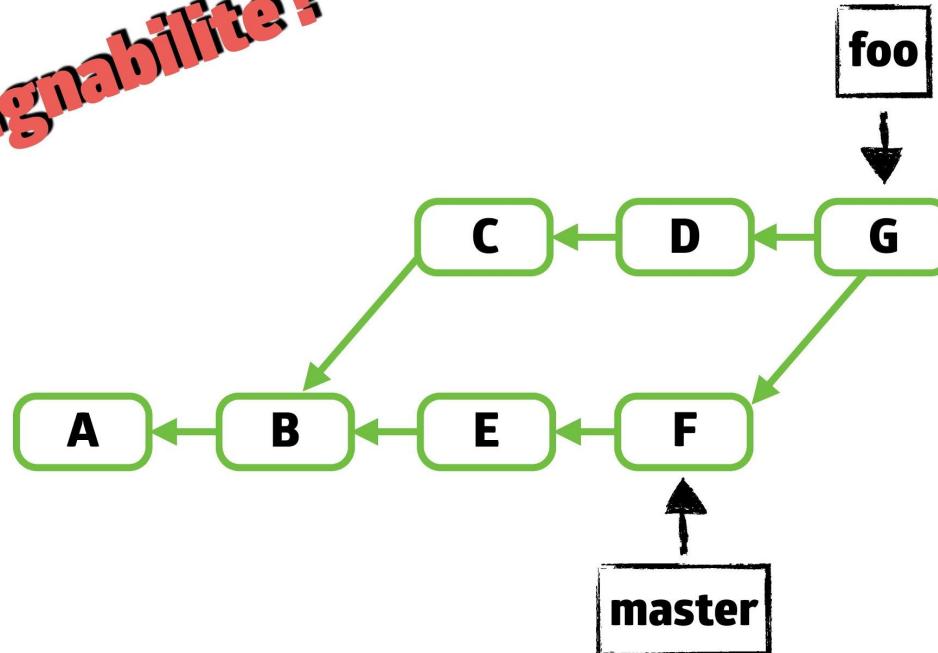


Qu'est-ce qu'un merge ?

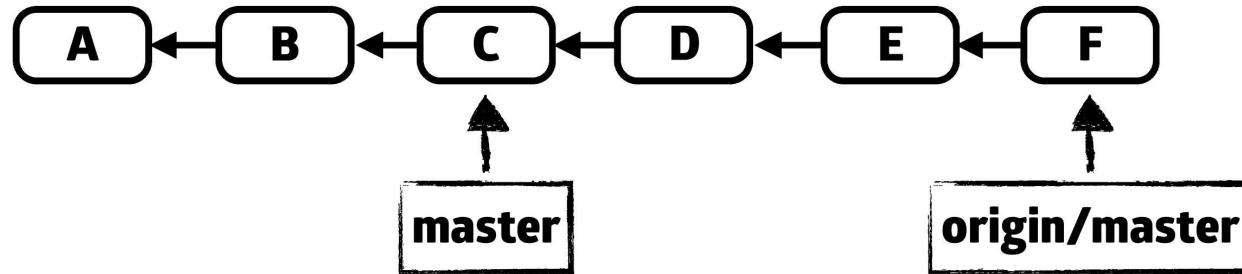
```
$ git merge master foo
```



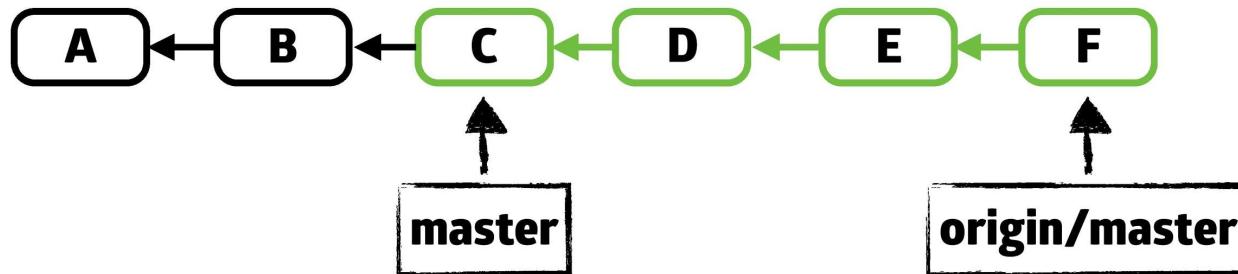
**Atteignabilité!**



Qu'est-ce qu'un  
fast-forward ?

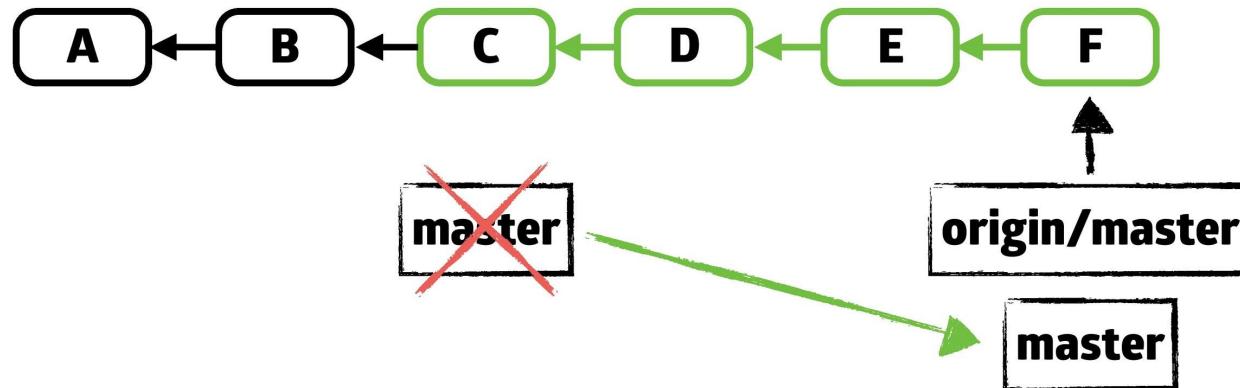


```
$ git merge origin/master
```



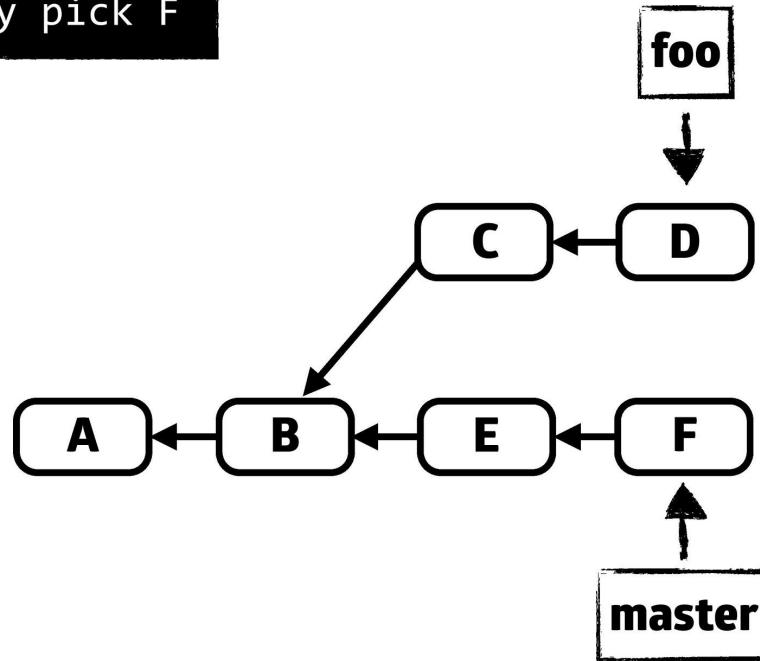
**Atteignable !**

```
$ git merge origin/master
```

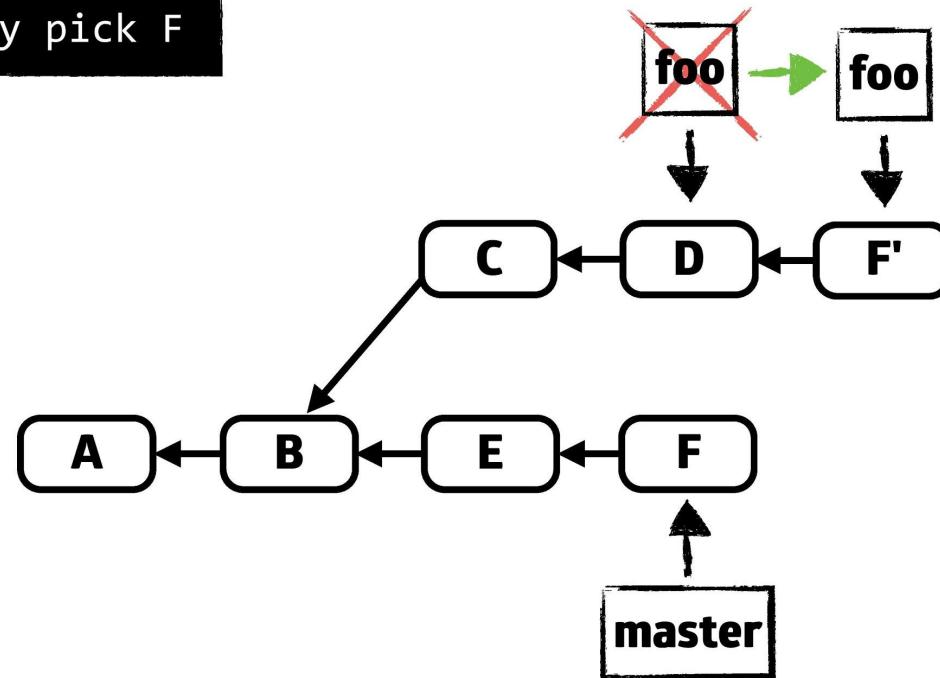


Qu'est-ce qu'un  
cherry-pick ?

```
$ git checkout foo  
$ git cherry pick F
```

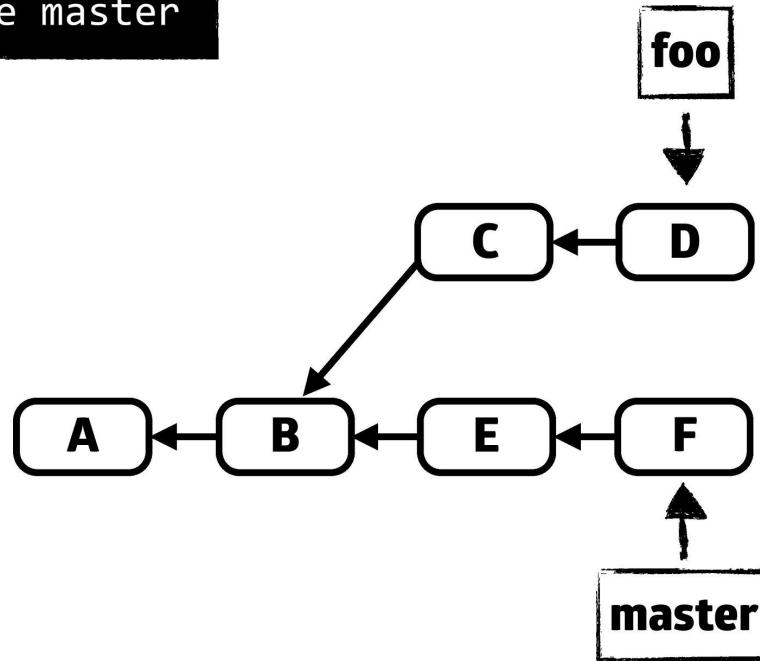


```
$ git checkout foo  
$ git cherry pick F
```

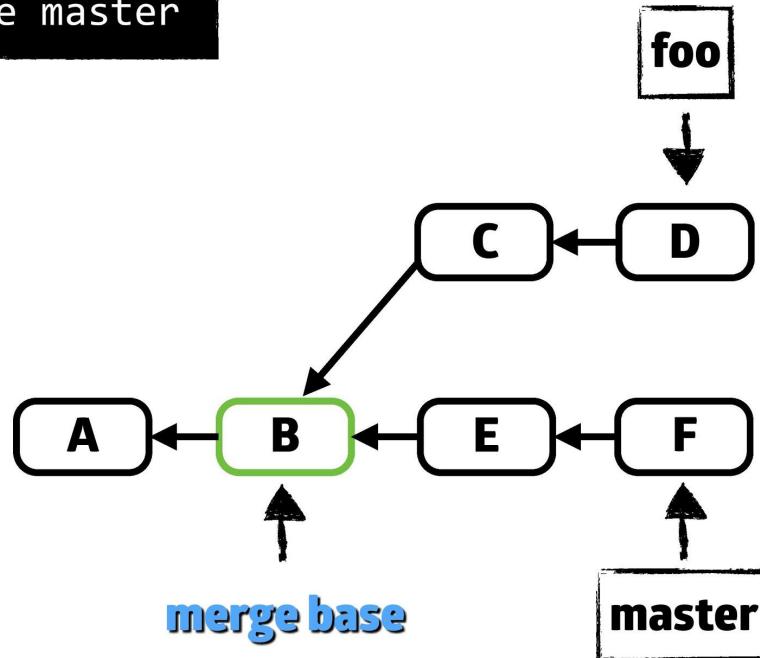


Qu'est-ce qu'un  
rebase ?

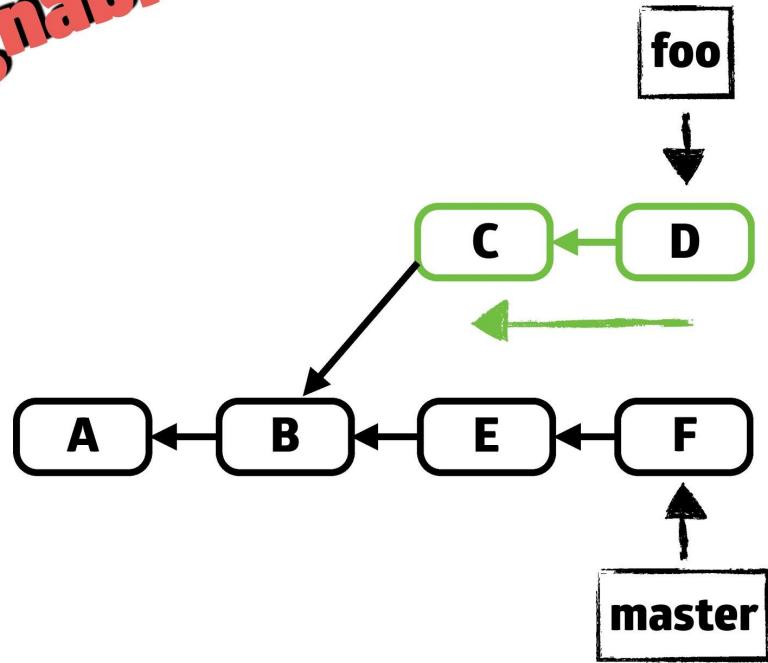
```
$ git checkout foo  
$ git rebase master
```



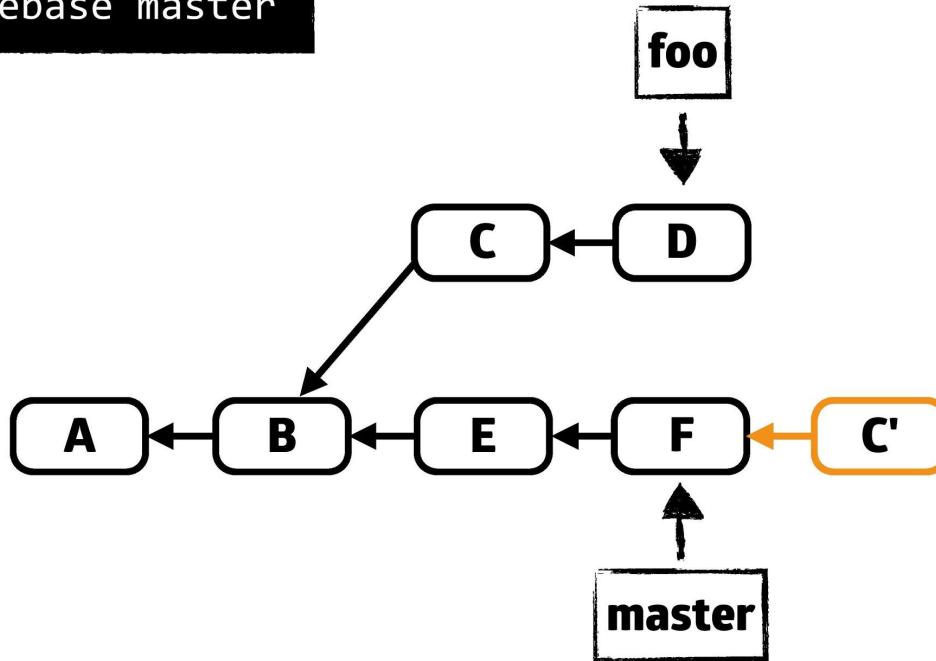
```
$ git checkout foo  
$ git rebase master
```



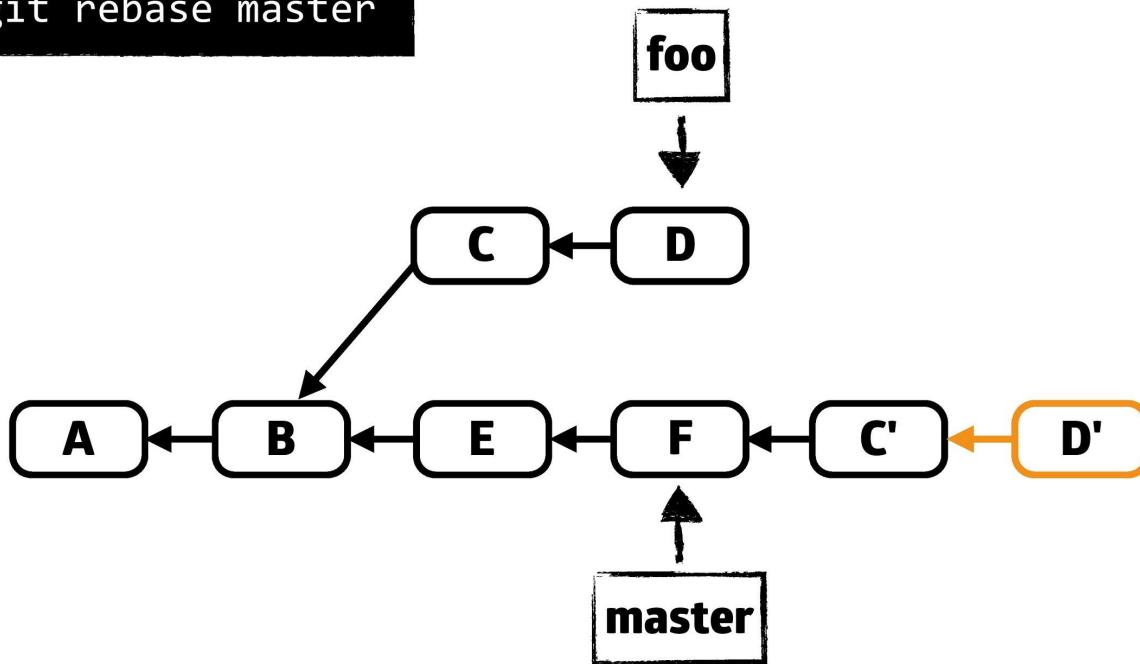
**Atteignable!**



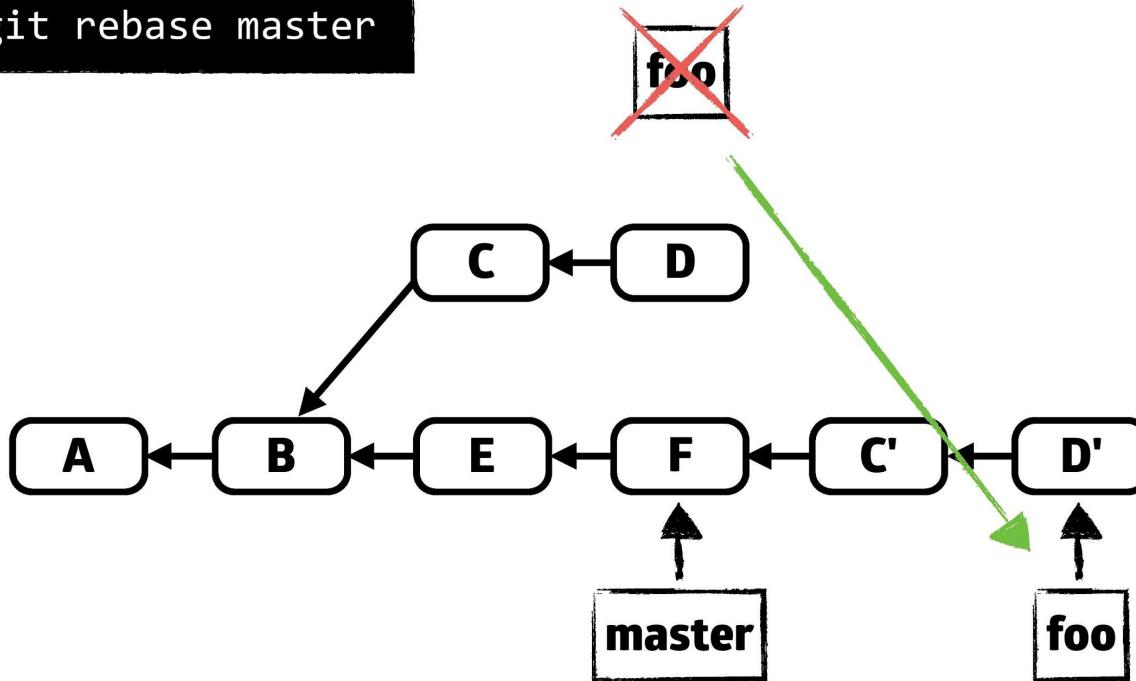
```
$ git checkout foo  
$ git rebase master
```



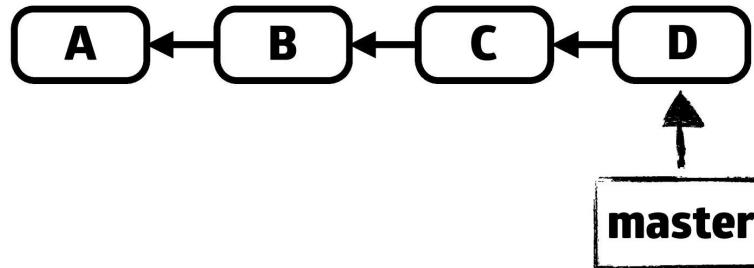
```
$ git checkout foo  
$ git rebase master
```



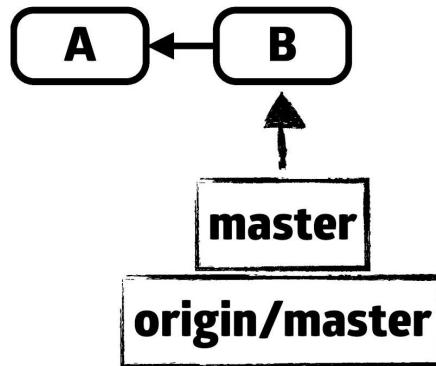
```
$ git checkout foo  
$ git rebase master
```



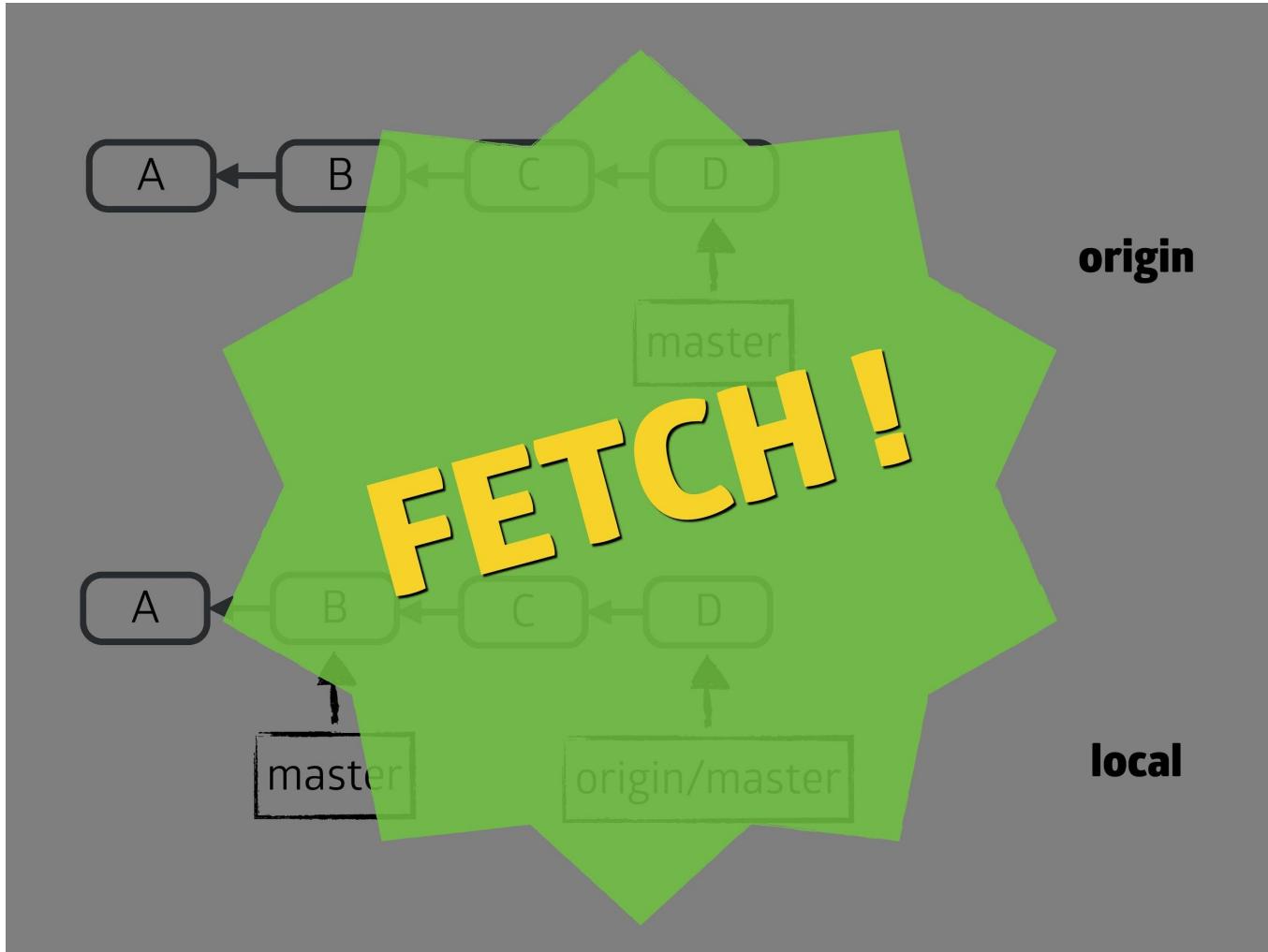
Qu'est-ce qu'un  
fetch ?

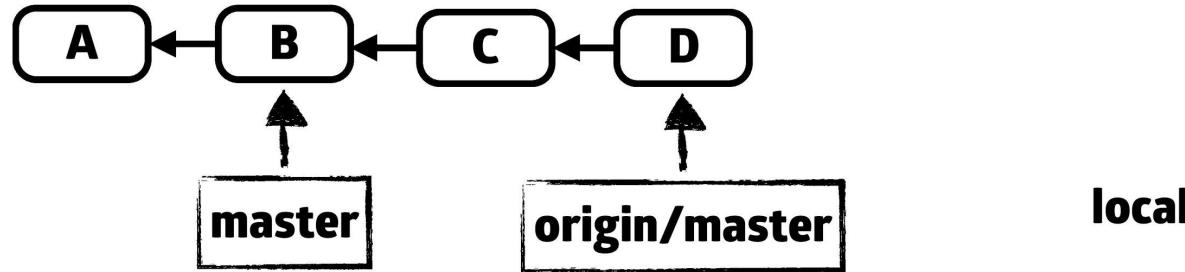
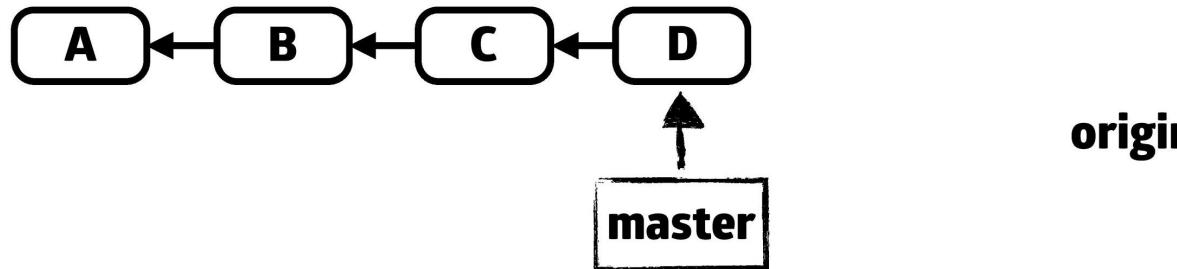


**origin**

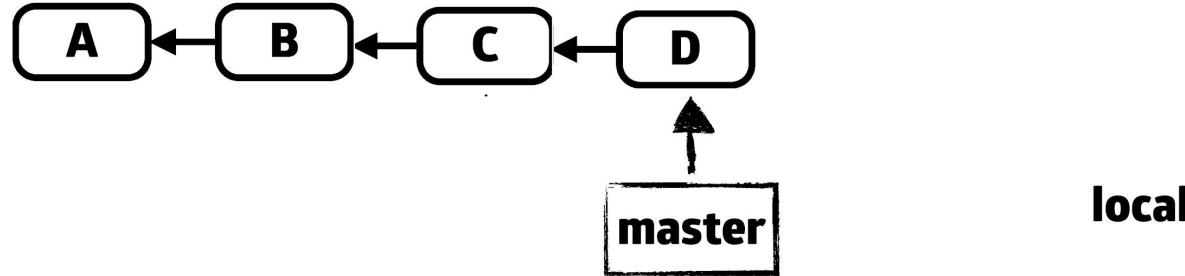
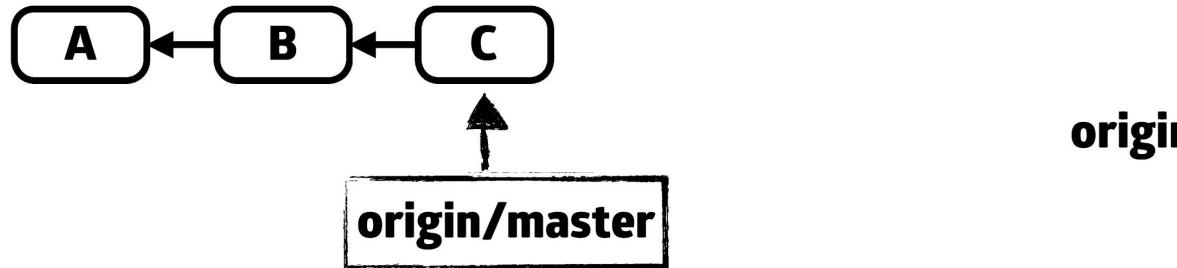


**local**

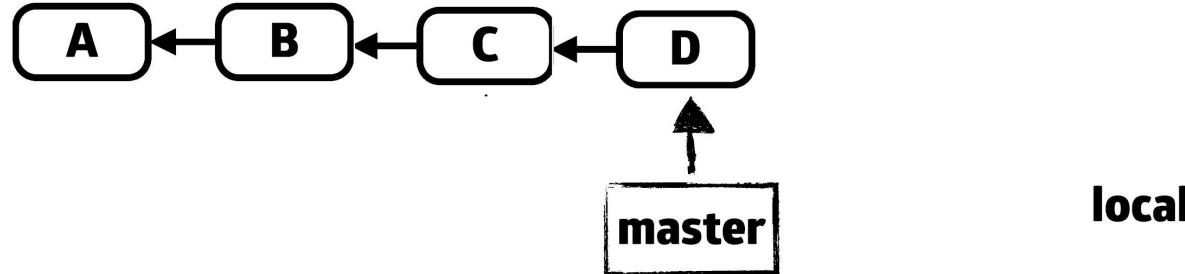
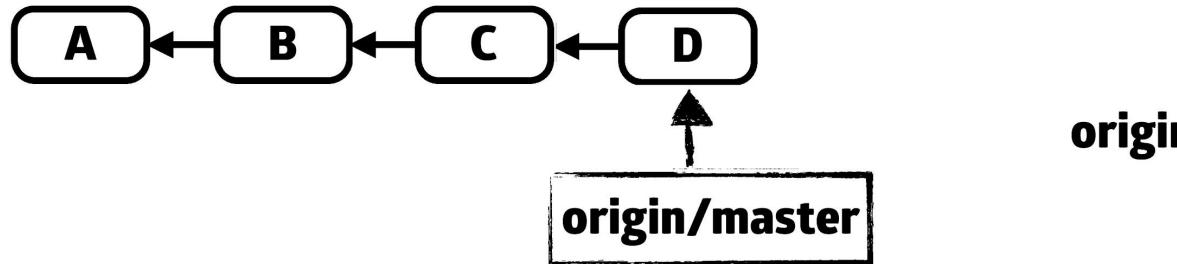


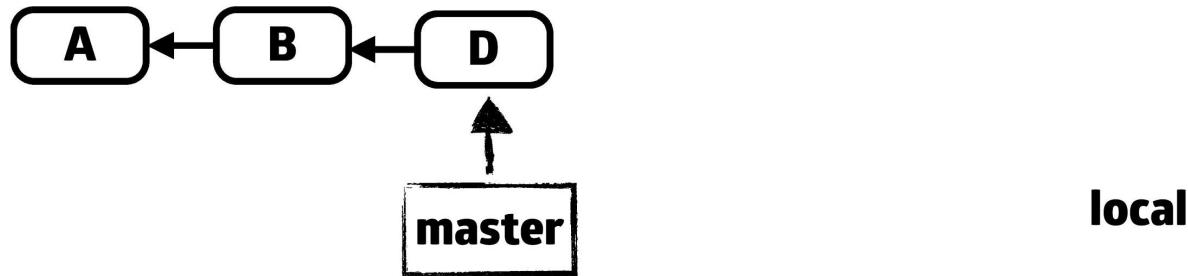
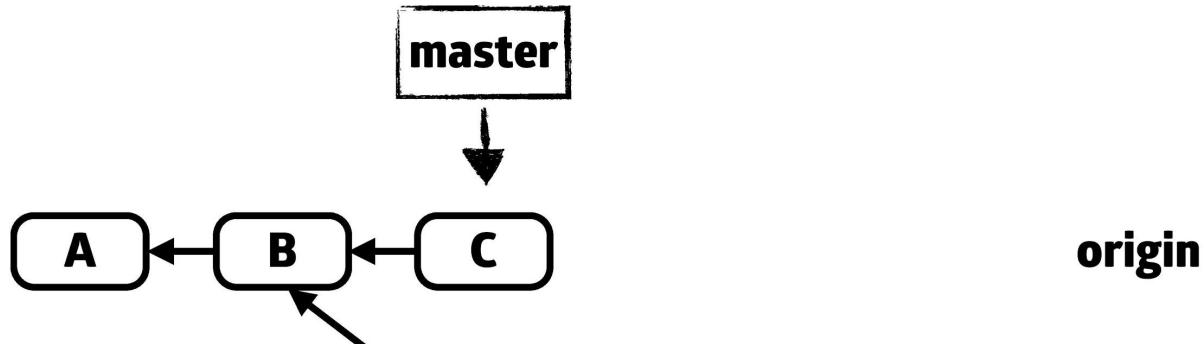


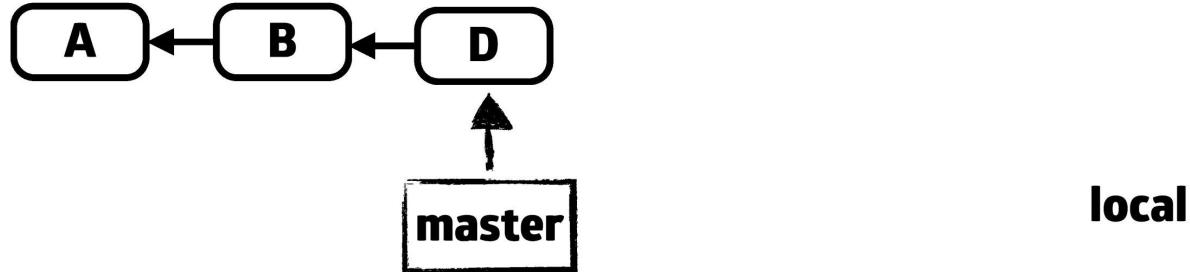
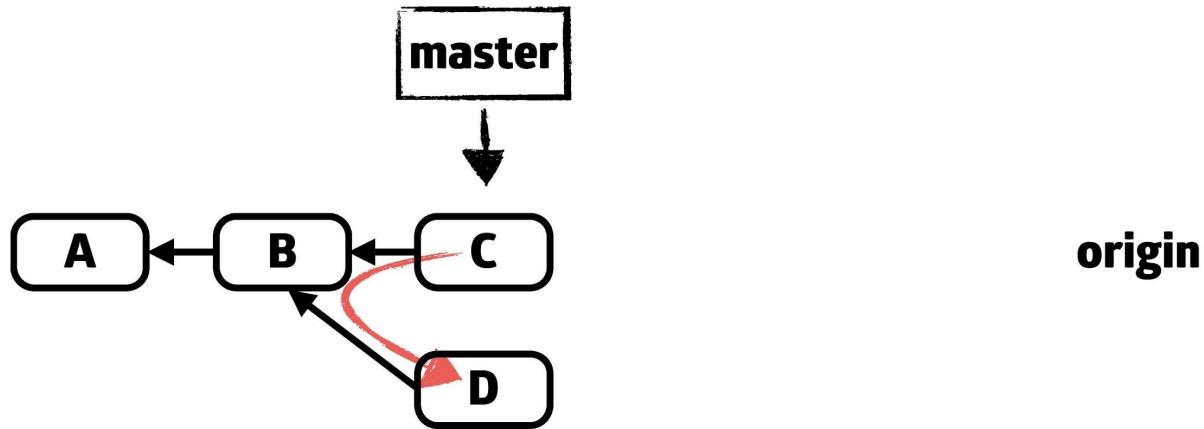
Qu'est-ce qu'un  
push ?

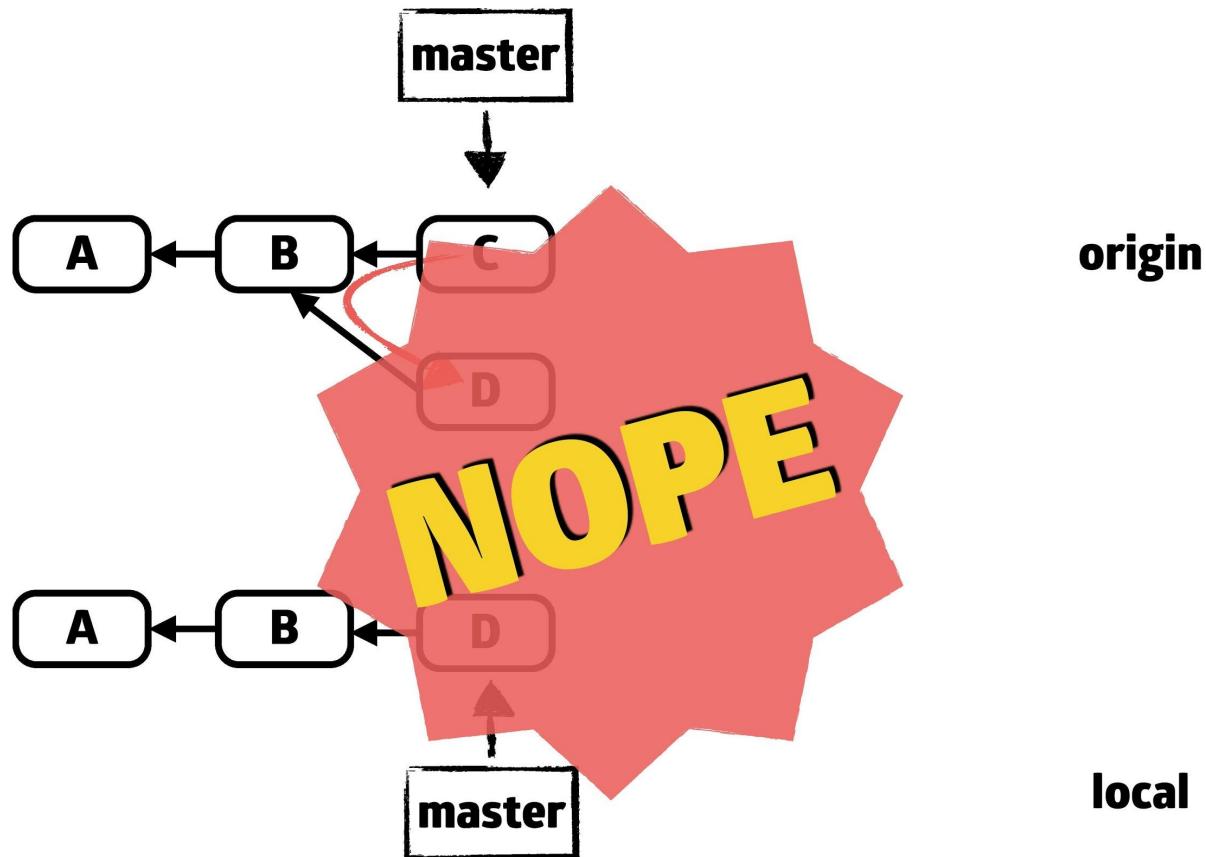


```
git push origin master
```









# Qu'est-ce qu'on fait ?



git pull origin master

git rebase origin/master

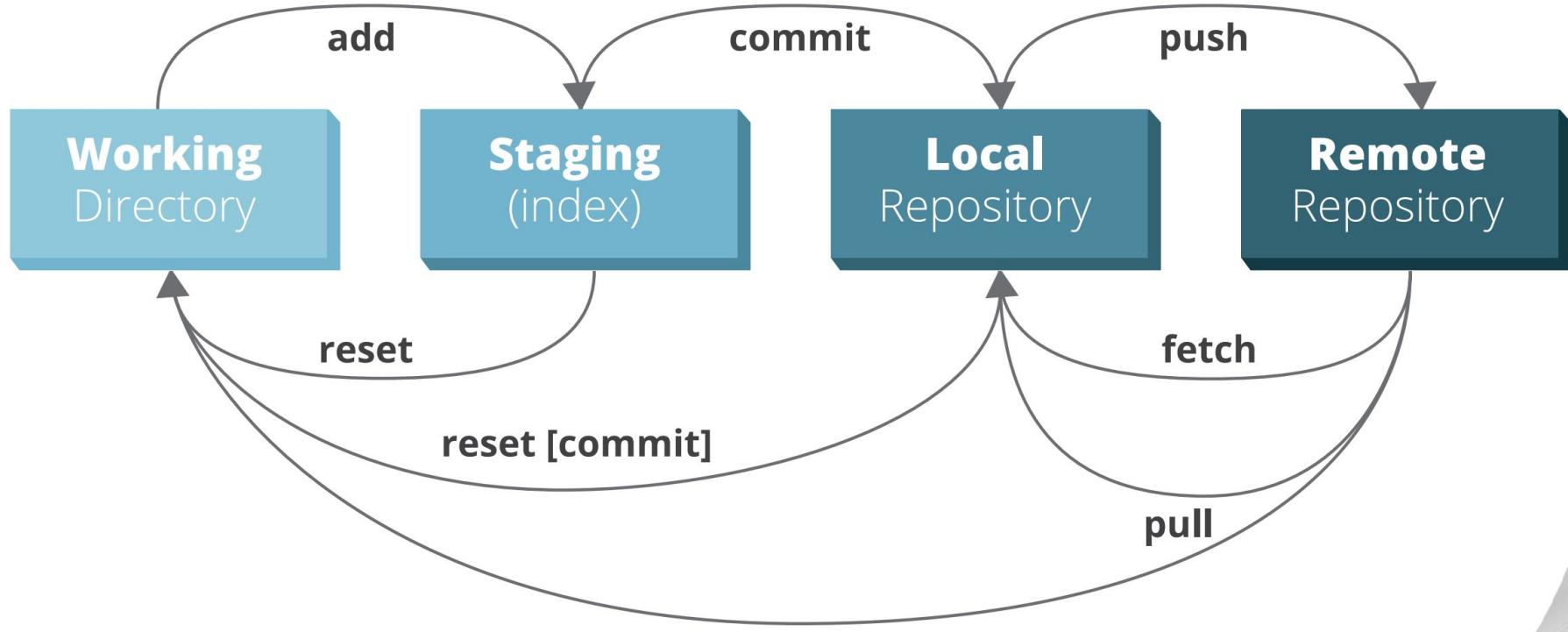
git push --force



```
git push  
--force-with-lease
```



DRAPPH



**HEAD ?**

# Trucs et astuces !



Annuler ses modifications  
en local

git checkout .

Annuler ses modifications  
en local d'un seul fichier

```
git checkout -- filename
```

# Annuler des modifications dans le staging

git reset *filename*

# Créer une branche

git branch ma\_branch

```
git checkout -b  
ma_branche
```

```
git push  
--set-upstream  
origin ma_branche
```

Voir l'historique

git log



```
git log  
  --oneline  
  --decorate  
  --graph
```



# Résoudre les conflits

git mergetool

# Stash

git stash save

git stash pop

git stash list

.gitignore  
Ne me git pas !

```
*.log  
/tmp/*  
*.pyc  
node_module  
vendor  
.DS_Store  
/tmp  
!/tmp/.gitkeep
```

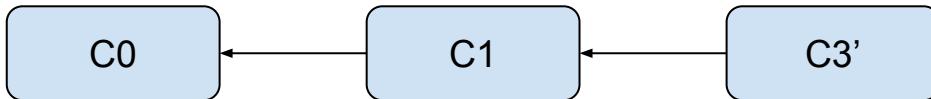
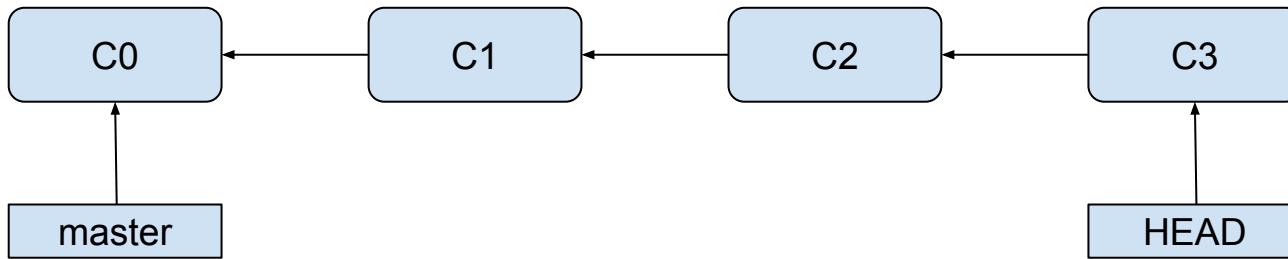
# Nettoyer le graphe

```
git rebase -i HEAD~5
```

```
pick 31f6585 mocky
pick d8d7834 logger middleware
pick 7a00bee settings
pick 788f871 settings page
pick b22831b Settings routing

# Rebase 287f87f..b22831b onto 287f87f (5 commands)
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
```

# Squash de C2



# Workflow

Don't touch master !

# Utilisez des branches

# Pull request

# Pull request

**POULE**

Badum tss



**REQUEST**

[Code](#)[Issues 675](#)[Pull requests 130](#)[Projects 3](#)[Insights](#)

## Fixed mistake in exception expectation #24557

[Open](#) uuf6429 wants to merge 1 commit into `symfony:3.3` from `uuf6429:Fixed-mistake-in-exception-expectation`[Conversation 0](#)[Commits 1](#)[Files changed 4](#)

Changes from all commits ▾ Jump to... ▾ +29 -15

[Unified](#)[Split](#)[Review changes ▾](#) This repository has not been [set-up on Scrutinizer](#).9  src/Symfony/Bundle/FrameworkBundle/Tests/DependencyInjection/ConfigurationTest.php[No coverage](#)[View](#)

```
@@ -165,12 +165,11 @@ public function testAssetsCanBeEnabled()
165     */
166     public function testInvalidAssetsConfiguration(array $assetConfig, $expectedMessage)
167     {
168 -     $this->{method_exists($this, $_ = 'expectException') ? $_ : 'setExpectedException'}(
169 -         InvalidConfigurationException::class,
170 -         $expectedMessage
171 -     );
172 -     if (method_exists($this, 'expectExceptionMessage')) {
173         $this->expectExceptionMessage($expectedMessage);
174     }
175
176     $processor = new Processor();

```

```
165     */
166     public function testInvalidAssetsConfiguration(array $assetConfig, $expectedMessage)
167     {
168 +     if (method_exists($this, 'expectException')) {
169 +         $this->expectException(InvalidConfigurationException::class);
170
171         $this->expectExceptionMessage($expectedMessage);
172     } else {
173         $this->setExpectedException(InvalidConfigurationException::class,
174         $expectedMessage);
175     }
176     $processor = new Processor();

```

11  src/Symfony/Component/Form/Tests/ButtonBuilderTest.php[No coverage](#)[View](#)

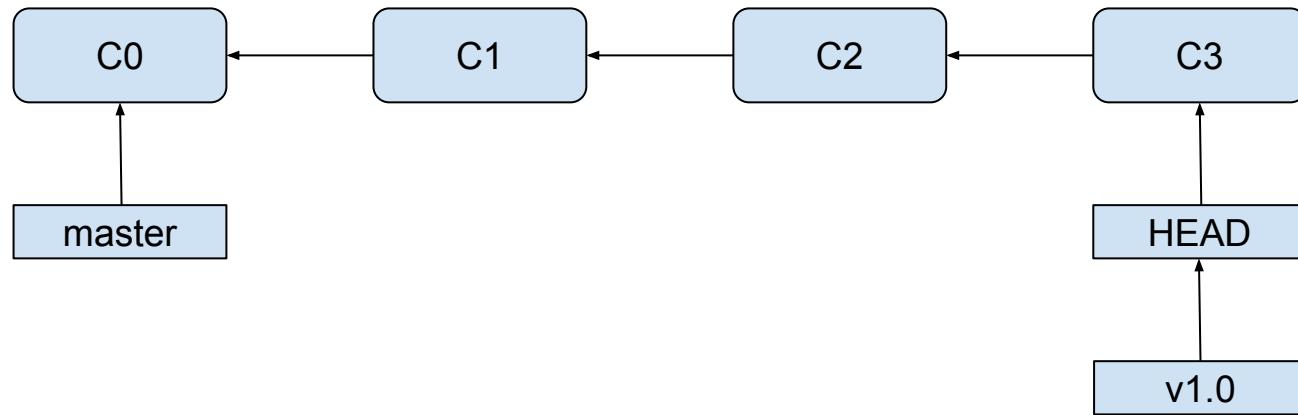
```
@@ -13,6 +13,7 @@
13
14     use PHPUnit\Framework\TestCase;
15     use Symfony\Component\Form\ButtonBuilder;
```

```
13
14     use PHPUnit\Framework\TestCase;
15     use Symfony\Component\Form\ButtonBuilder;
16     +use Symfony\Component\Form\Exception\InvalidArgumentException;
```

# Faire des commit atomique

# Faire des tag

git tag v1.0



Faites des conneries !

<http://bit.ly/2xZNsme>

Crédits : @ubermuda