



git

# Gestion de version (versionning)

Un logiciel de gestion de versions est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications.

Les logiciels de gestion de versions sont utilisés notamment en développement logiciel pour conserver le code source relatif aux différentes versions d'un logiciel.



# Quelques logiciels

Nécessitant un serveur

**CVS** 1990

**SVN** 2000

**Team Foundation Server** 2005

Fonctionnement en mode décentralisé (sans serveur)

**BitKeeper** 1998

**Mercurial** 2005

**Git** 2005



# Création de git

Après un désaccord sur l'utilisation de BitKeeper pour la gestion du projet Linux, Linus Torvald décide de créer son propre logiciel de gestion de version.

Git est à peu près l'équivalent de “connard” en argot britannique.



# Principes

Git permet de garder un historique des modifications des fichiers de votre projet.

A chaque modification importante, vous indiquez à Git qu'il faut prendre un instantané de vos fichiers (commit).

Ensuite vous pouvez naviguer dans la liste des modifications et revenir à une version antérieure.



# Création d'un dépôt

```
git init
```

Le dépôt est créé dans le répertoire .git

# Fonctionnement

```
git add .
```

Ajoute les nouveaux fichiers ou les fichiers modifiés dans l'index

```
git commit
```

Enregistre l'état du dossier dans le dépôt

# Fonctionnement

```
git commit -am 'message de commit'
```

Plus rapide ...



# Voyage dans le temps

**git log**

Affiche la liste des commit

**git checkout**

Permet de revenir à un état précédent

**git checkout master**

Retour au dernier commit

# Coller des étiquettes

```
git tag
```

Affiche la liste des étiquettes

```
git tag -a v1 74c35
```

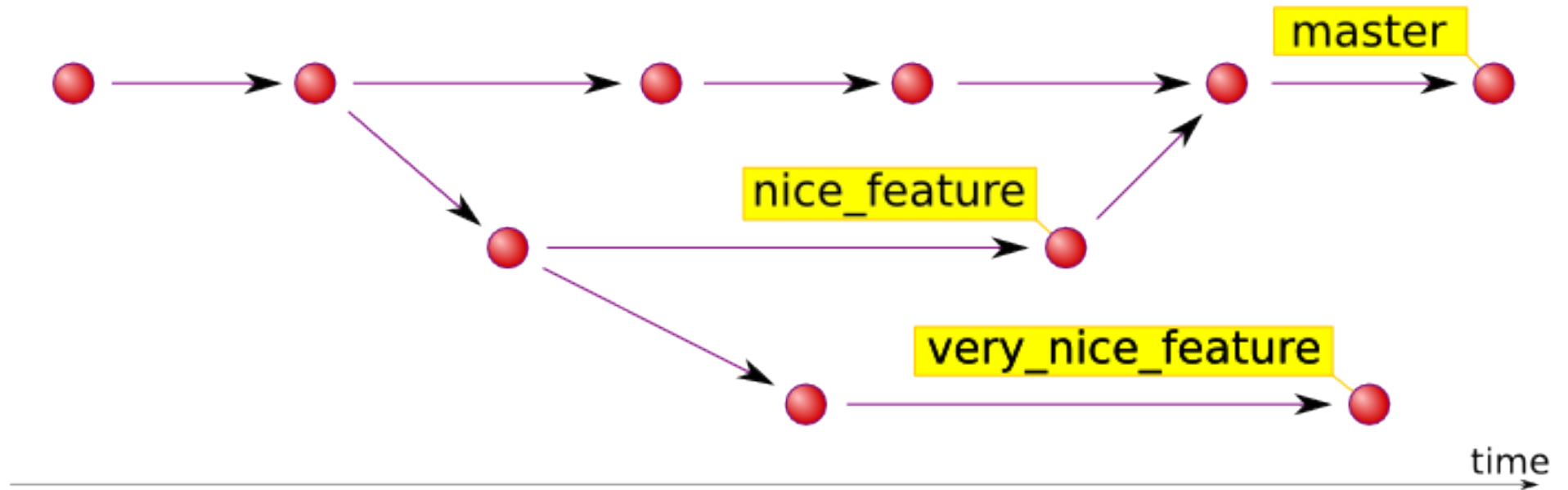
Pose l'étiquette  
v1 sur le commit  
74C35

```
git checkout v1
```

Et hop ...



# Les branches



Les branches permettent de manipuler les différents états d'un projet.

# Gestion des branches

**git branch**

Affiche la liste des branches

**git branch develop**

Création de la  
branche develop, à  
partir de l'état actuel

**git checkout develop**

Changement de  
branche

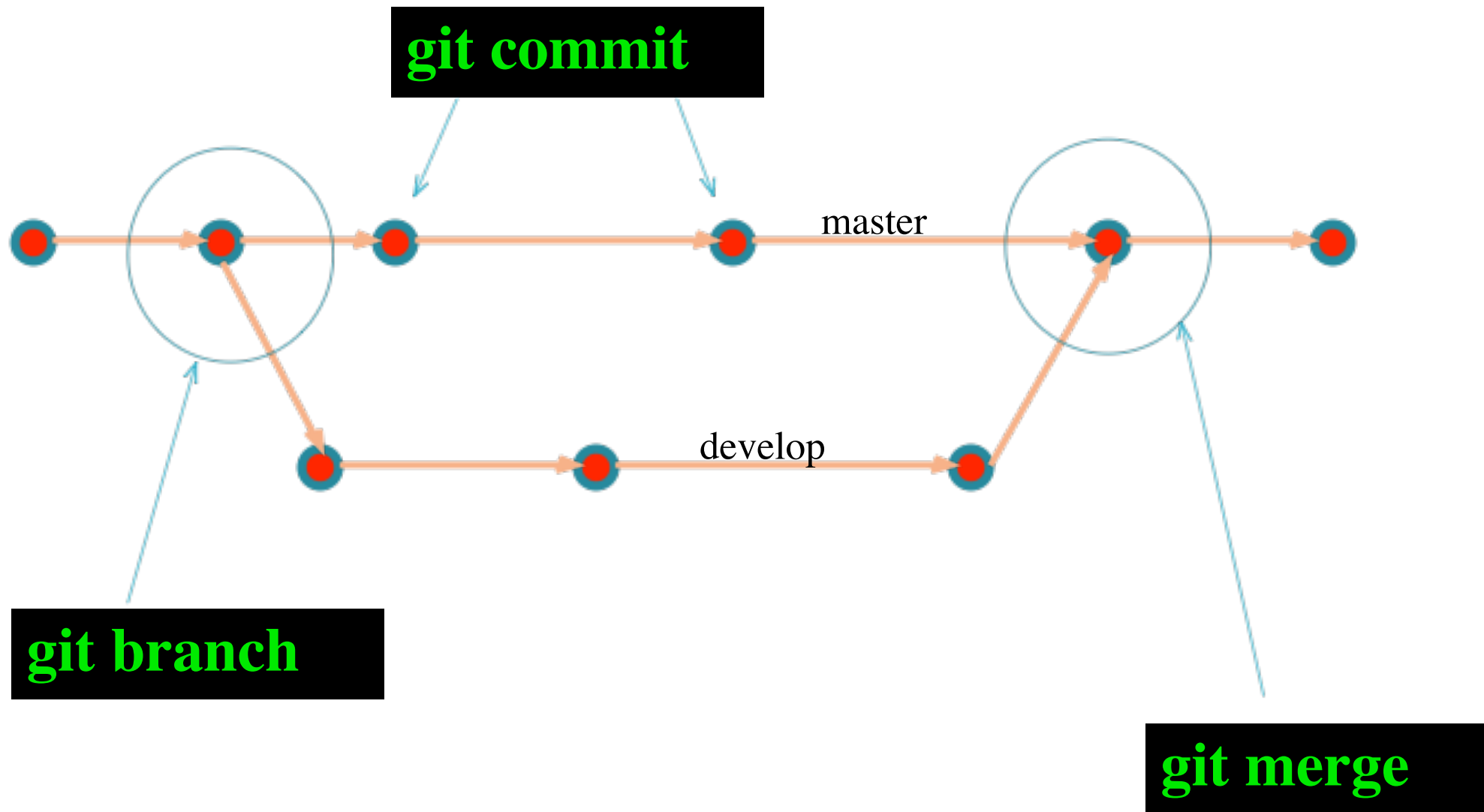
# Fusion de branches

## git merge develop

Fusionne la branche develop dans la branche active

Si un conflit apparait, vous devez:

- Résoudre le conflit (manuellement)
- Effectuer un commit



**git checkout**

Pour se positionner dans l'arbre


# Dépôts distants

```
git clone url
```

Initialise un dépôt local à partir  
du distant

```
git remote add origin url
```

Ajoute une référence au dépôt distant dans le  
dépôt local



# Dépôts distants

```
git push origin master
```

Envoie les modifications sur le dépôt distant

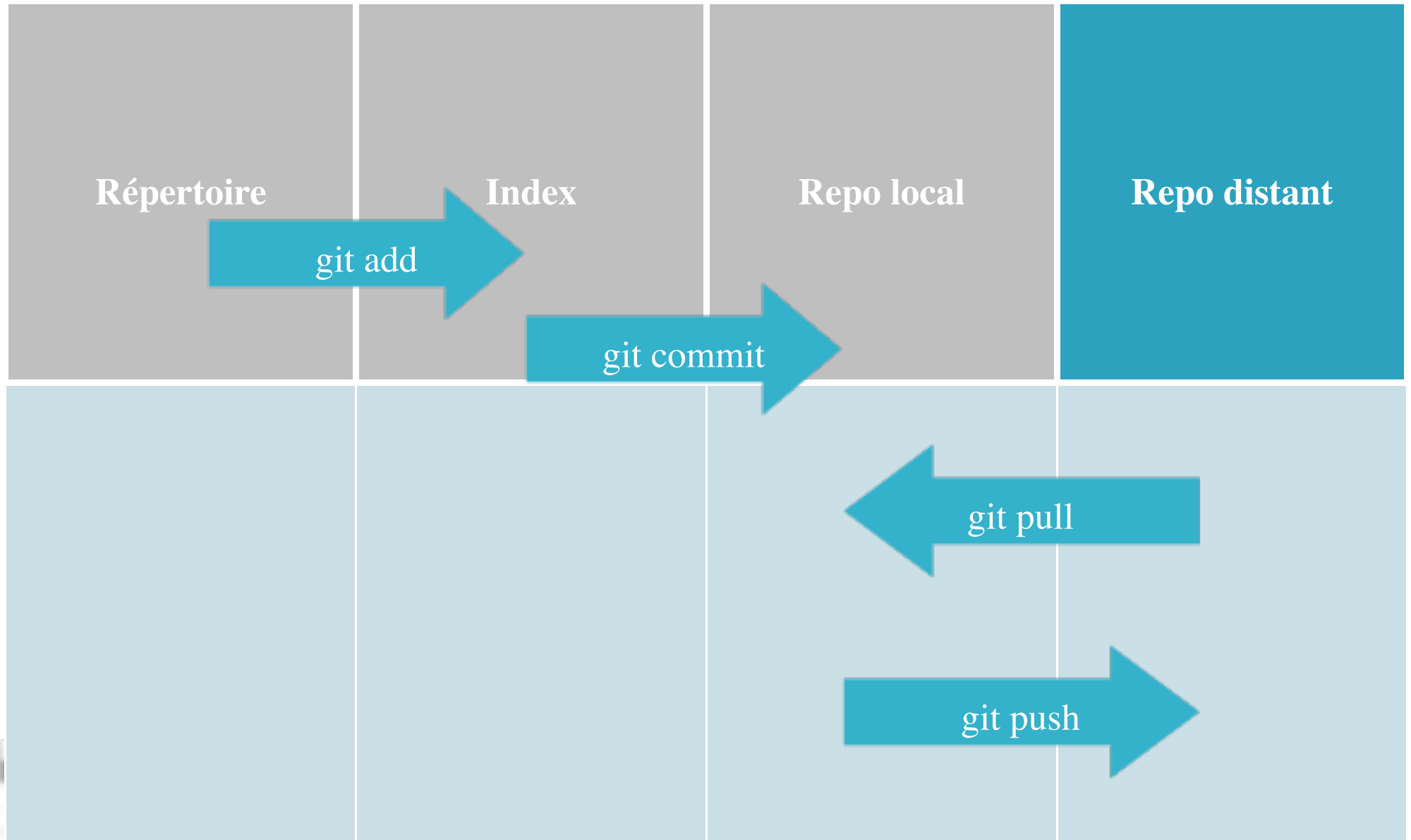
```
git pull origin master
```

Récupère les modifications depuis le dépôt distant.





# Cycle de travail



**Git gère les  
versions, à vous  
de gérer le reste**



**Git n'est pas un  
système de  
sauvegarde**



**La règle:**  
**On ne commit**  
**que du code qui**  
**marche**

