



Git

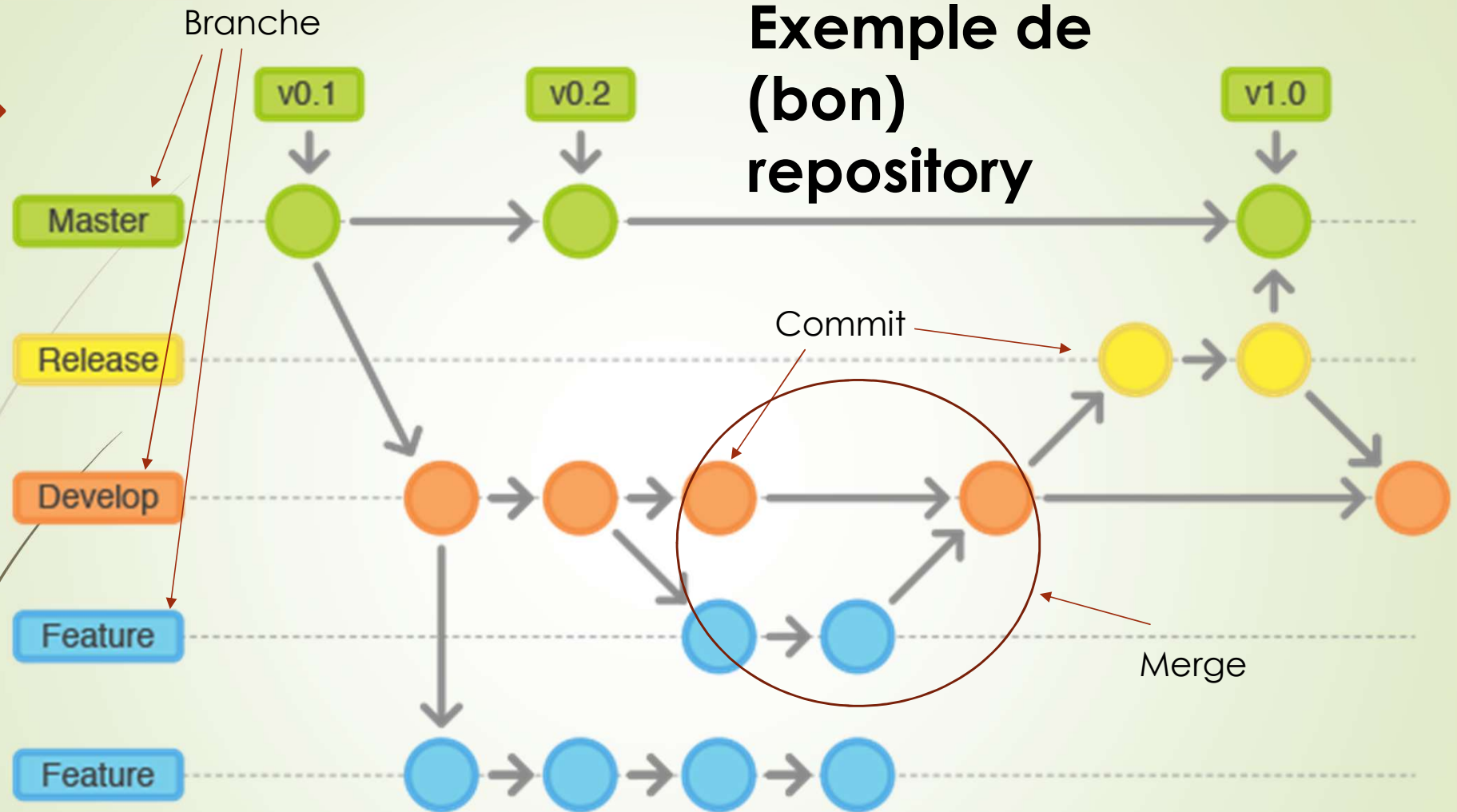
## Qu'est ce que c'est ?

- Outil de versionnage
- Suivre un projet et ses modifications
- Savoir ce qui a été modifié?, quand?, par qui? ( ultime flicage !)
- Travailler en parallèle sans difficultés

# Pourquoi?

- Facilite la relecture d'une année sur l'autre
- INDISPENSABLE en industrie
- Pas de peur de modifier un fichier
- Rigorisme
- La fin des noms du type :
  - Version\_finale
  - Version\_finaleDefinitive
  - BonCetteFoisCestladerniereVersion
  - ...
  - Koala+++eucalyptus\_vert\_feuille2015bis

## Exemple de (bon) repository



# Comment s'en servir?

- Vocabulaire important:
  - Repository : là où se situe l'arbre
  - Create branch: créer une branche
  - Pull : mettre à jour l'arbre
  - Commit : soumettre une modification de branche
  - Push : mettre un commit sur internet
  - Merge : fusion de branches WARNING

# Code de bonne conduite

- Chaque projet différent: Nouvelle Branche
- TOUJOURS vérifier qu'on travaille sur la bonne branche
- JAMAIS de merge vers le Master
- TOUJOURS tester son résultat de merge ( marche jamais du premier coup)
- TOUJOURS mettre un commentaire de commit explicite
- TOUJOURS pull avant de push, de modifier ou de travailler
- En cas de doute: demander

In case of fire



1. git commit



2. git push



3. leave building

# Défauts

- Facile de faire des bêtises
- Prise en main peut être difficile
- Changer ses habitudes

# Travaux pratiques

- Installer **git extension** en **ANGLAIS** (traductions bancales...)
- Se créer une **clef SSH** privée et publique
- **Clone repository**
- Enjoy!



Questions ?



# Demonstration!