

# Solution de versionning

ESGI - 3ème année IW 2020

# Introduction

# Généralités sur le cours

Introduction

# Objectifs

- Connaître certaines solutions existantes
- Utiliser efficacement git pour:
  - Être capable de revenir à une version précise d'une base de code
  - Travailler en équipe
  - Repérer un commit défectueux
- Comprendre le fonctionnement de l'outil

# Pourquoi versionner

Introduction

# Pourquoi versionner

- Sauvegarder les changements du projet (historique)
- Revenir à un état plus stable
- Travailler en parallèle sur plusieurs fonctionnalités
- Modifier les mêmes fichiers sur plusieurs postes

# Les outils existants




Introduction

# Les principaux outils existants

- Subversion (SVN)
- Mercurial (hg)
- Git



# Les principaux outils existants : comparaison

	Subversion 	Mercurial 	Git 
Année de création	2000	2005	2005
Centralisé	✓	✗	✗
Gestion des utilisateurs	✓	✗	✗
Synchronisation de branche individuelles	✓	✗	✓
Lightweight branches	✗	✓	✓
Licence	Apache License 2.0	GPL version 2 or any later version ("GPLv2+")	GNU General Public License version 2.0

# Historique de création de git

Introduction

# Historique de création de git

- Utilisation de Bitkeeper pour versionner Linux
  - accompagnée de contraintes limitant les développeurs
- Trois ans plus tard: fin de la licence gratuite
- Réponse de Linus Torvalds: git
  - Branching facile
  - Décentralisé
  - Licence open source

Vidéo explicative: <https://www.youtube.com/watch?v=LDy6Rv0kU1Q>

# Conclusion

Git devient l'outil de gestion de version du projet open-source le plus conséquent de l'époque, poussant son adoption massive par la communauté.