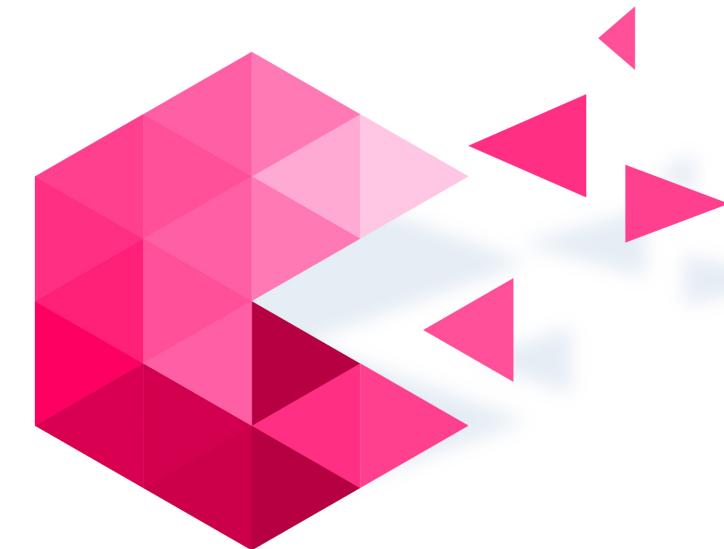


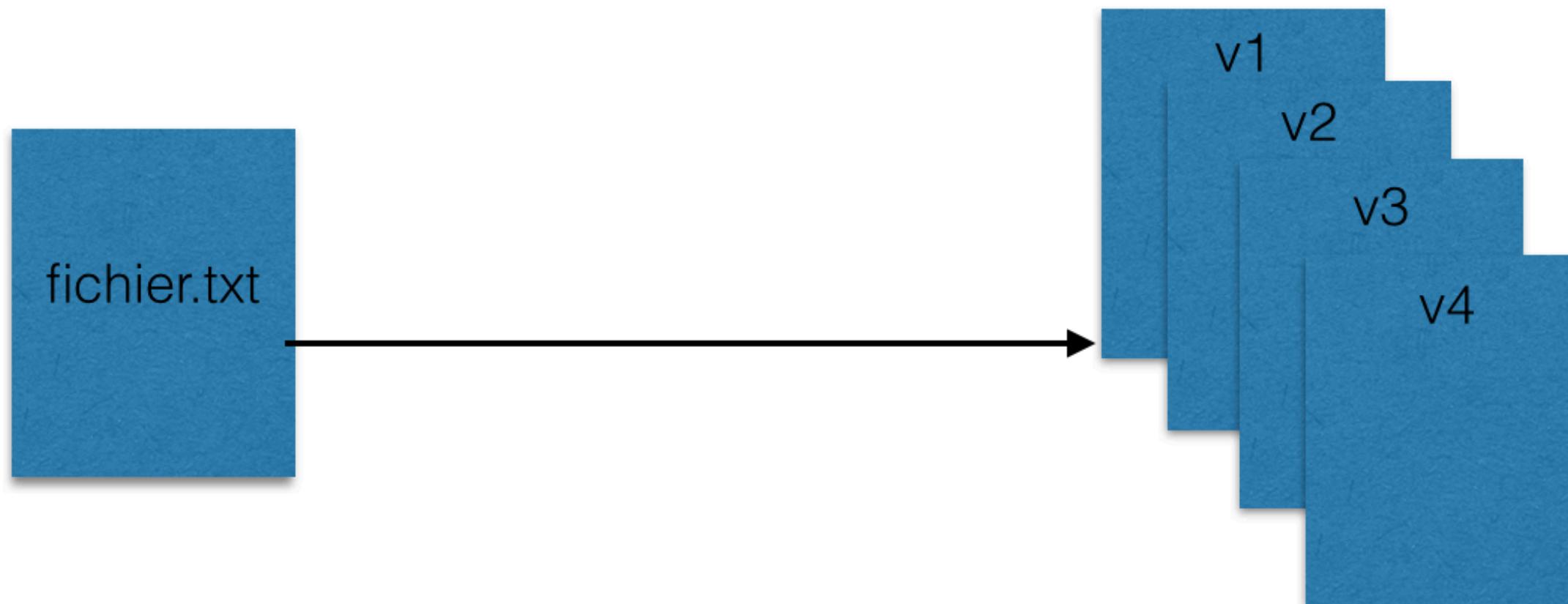
# Cours 1 : Introduction



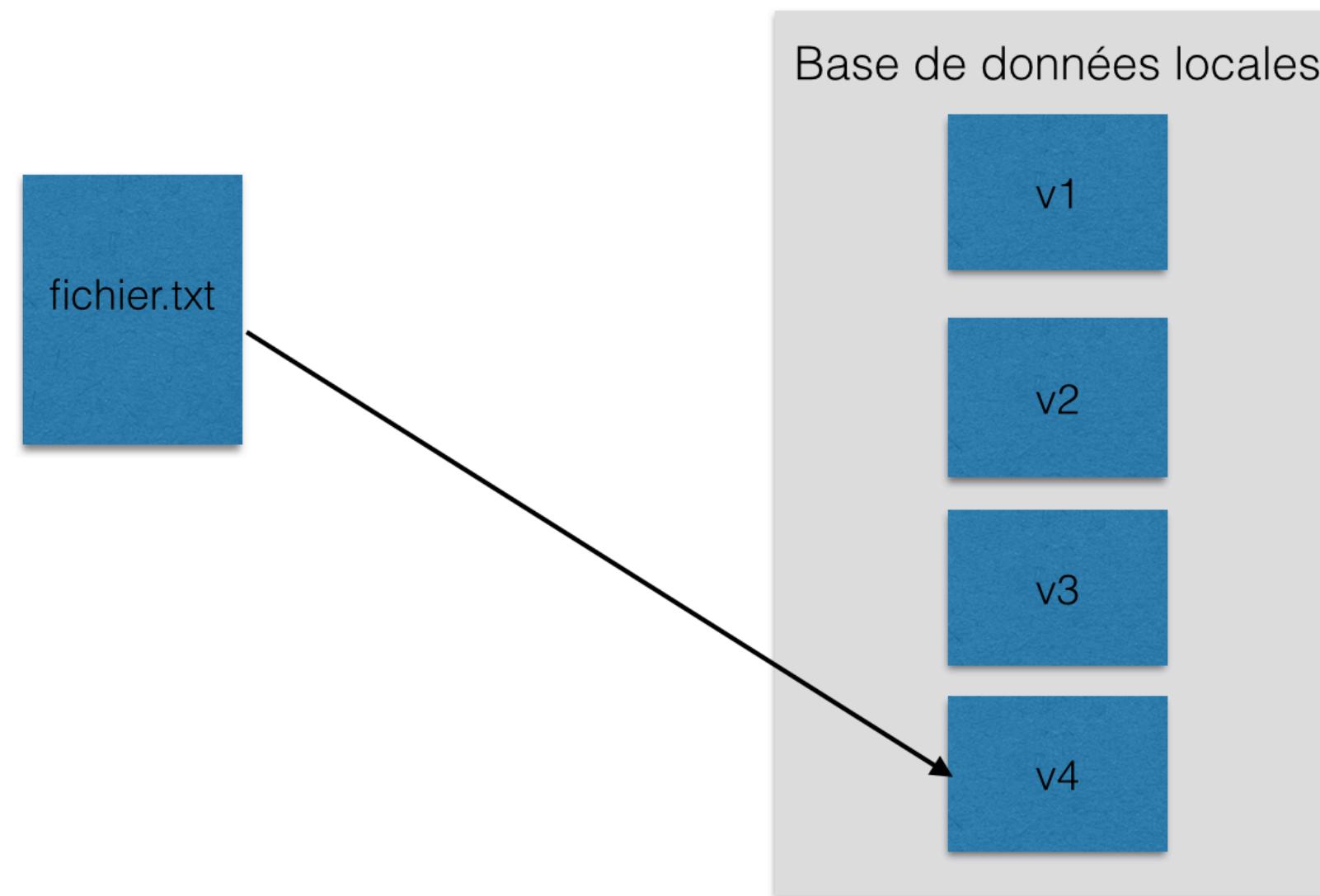
le  
**campus**  
**numérique**  
in the ALPS

# Une petite histoire des gestionnaires de version.

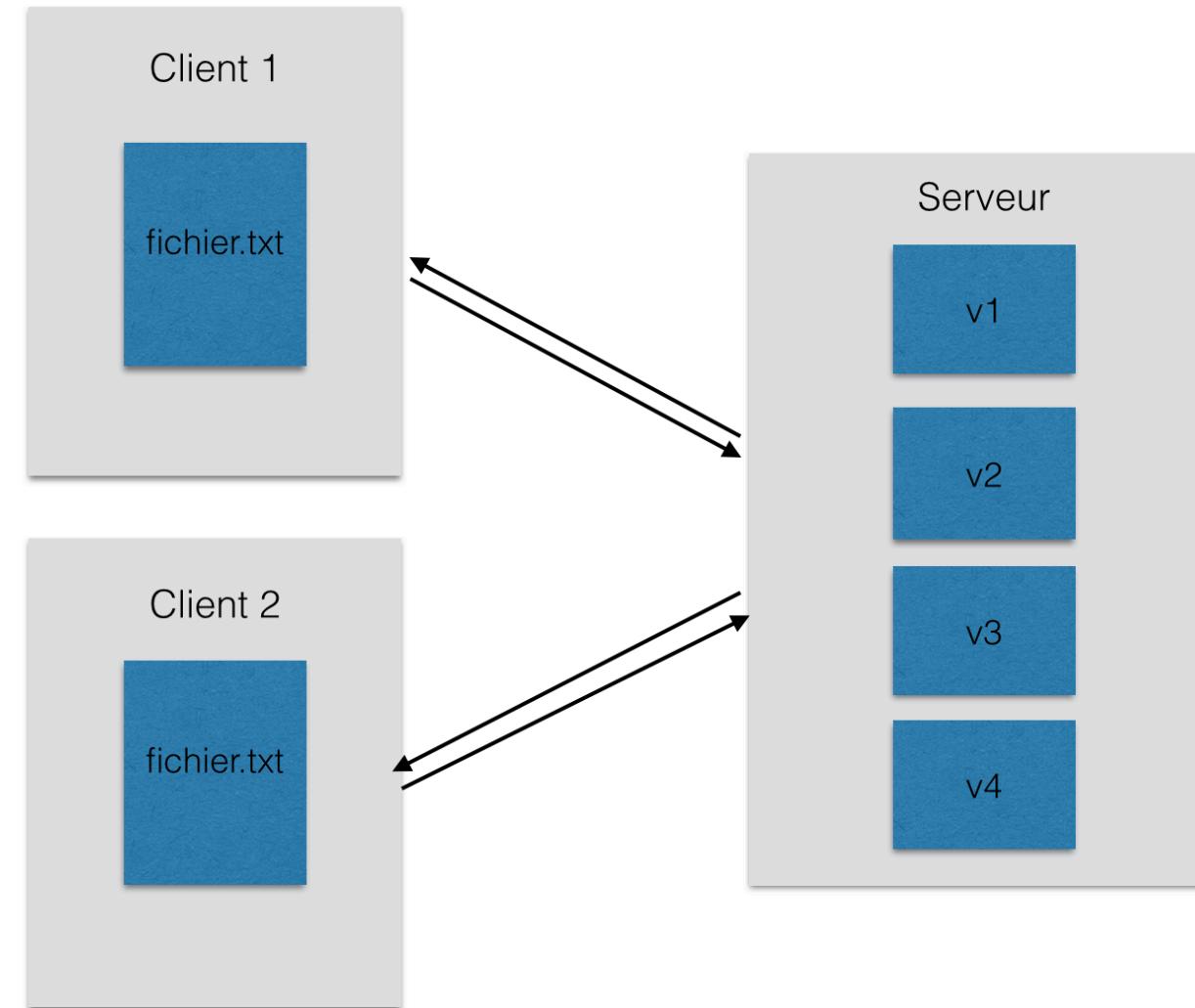
# Gestionnaire de version local par duplication.



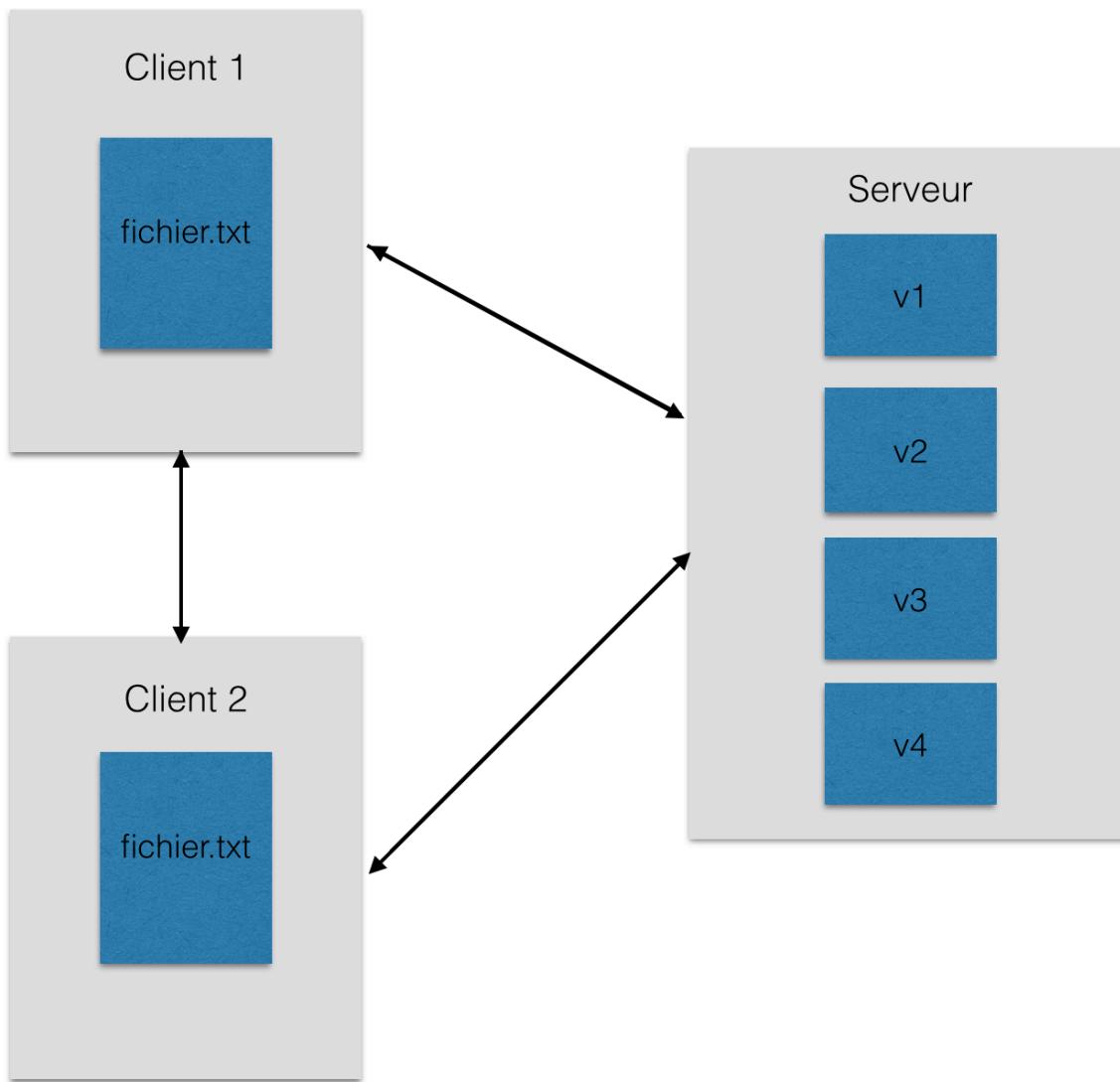
# Gestionnaire de version avec base de données locale.



# Gestionnaire de version centralisé.



# Gestionnaire de version décentralisé .



# Histoire de git.

- Jusqu'en 2002 : utilisation de patchs.
- De 2002 à 2005 : BitKeeper propriétaire mais gratuit.
- Développé en quelques jours par Linus pour le noyau Linux<sup>1</sup>.

---

<sup>1</sup> L'histoire de git est étroitement liée avec celle du noyau linux.

# Les avantages de git

- La plupart des opérations sont locales.
- Git gère l'intégrité.
- Chaque ordinateur est une sauvegarde complète du repository.
- Très peu d'opérations destructrices.

# Installation de Git



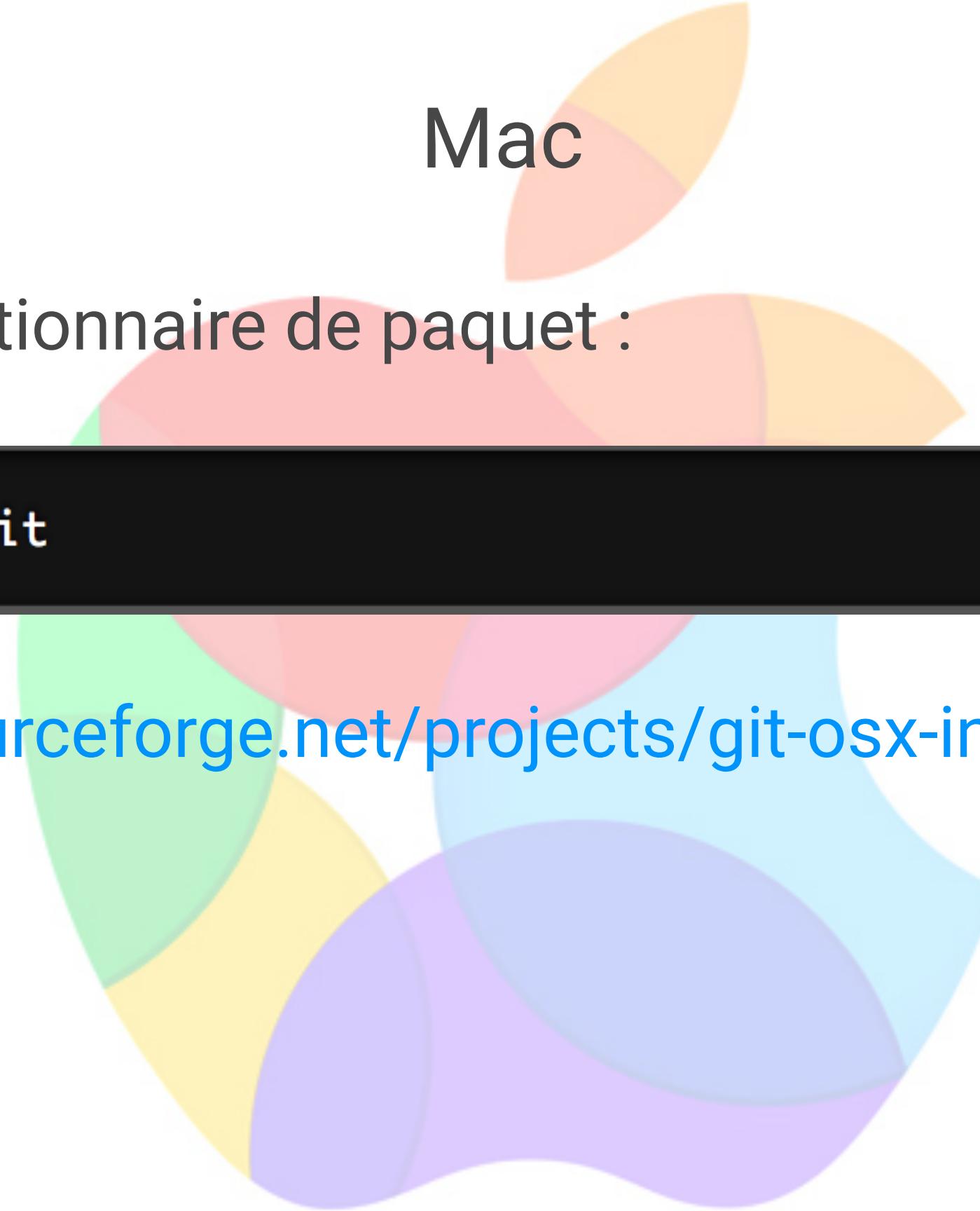
# Linux

Le plus simple est de l'installer<sup>2</sup> via votre gestionnaire de paquets.

```
1 | apt-get install git  
2 | yum install git
```

---

<sup>2</sup> Il est très probable que git soit par défaut sur votre distribution.



Mac

Soit via votre gestionnaire de paquet :

```
1 | brew install git
```

Soit via <http://sourceforge.net/projects/git-osx-installer/>

# Windows

Installer msysGit<sup>3</sup> : <http://msysgit.github.io>



---

<sup>3</sup> Vous devez utiliser Git avec le shell bash fourni par msysGit. Retrouvez le raccourci sur votre bureau.

# Ligne de commande

Pour vérifier que git est installé, ouvrez votre terminal<sup>4</sup>:

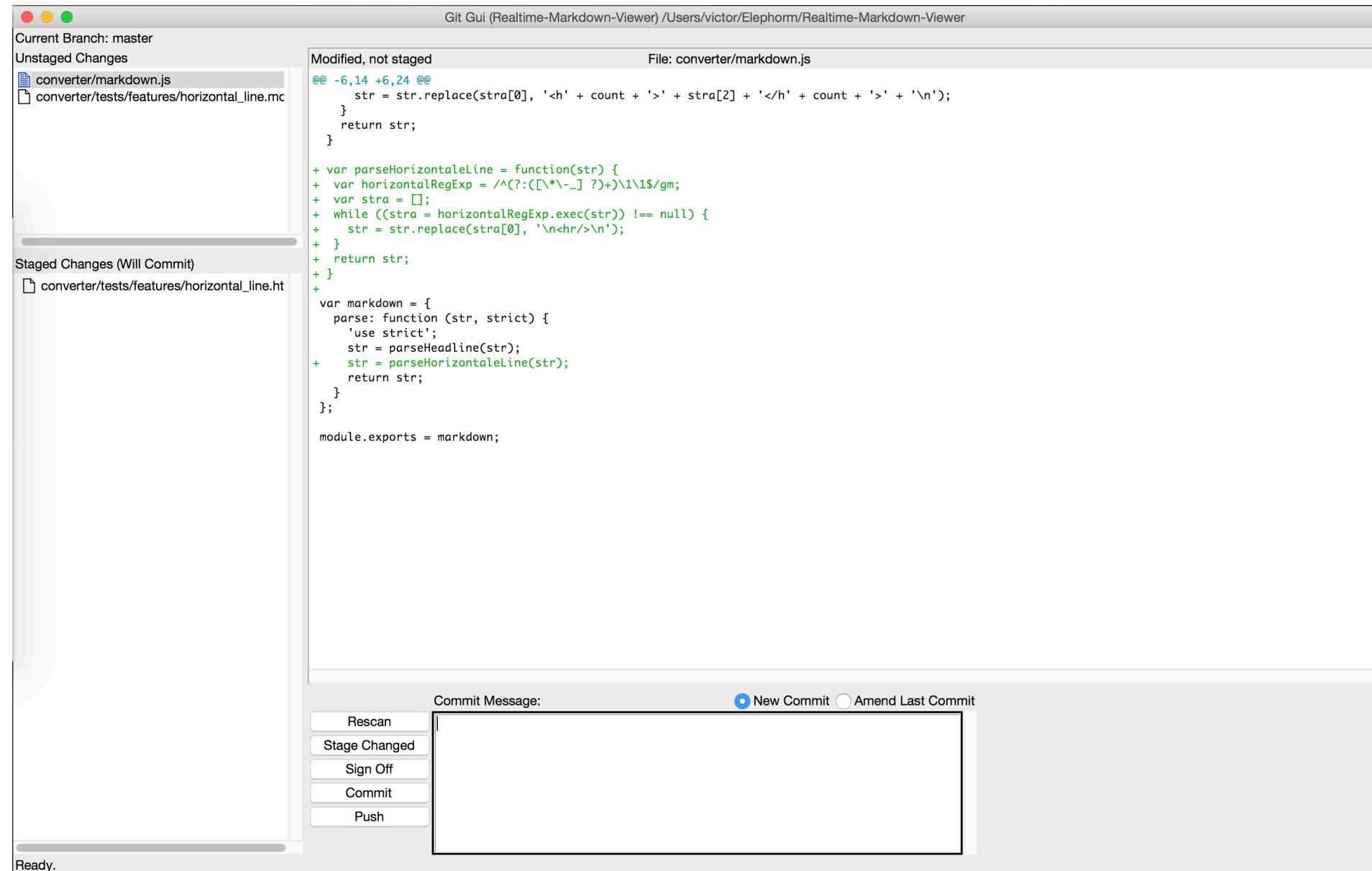


```
Realtime-Markdown-Viewer — victor@MacBook-Pro-de-victor: ~/Elephorm/Realtime-Markdow...
→ Realtime-Markdown-Viewer git:(master) git --version
git version 2.3.1
→ Realtime-Markdown-Viewer git:(master) which git
/usr/local/bin/git
```

---

<sup>4</sup> Si votre version est légèrement différente, ce n'est pas grave. Néanmoins assurez vous d'être au moins à la version 2.0.

# git gui



# gitk

The screenshot shows the gitk graphical interface for Git. The main window displays a timeline of commits on the left, a log of commits on the right, and a detailed view of a selected commit in the bottom half.

**Timeline (Top Left):** Shows a sequence of commits from `tp5-start` to `tp2-start`. A specific commit, `b496d8c48c7289ac8d4dc9d032412284e18bbe64`, is highlighted. The commit message is "Revert \"frf\"".

**Log (Top Right):** A list of commits by user `sabativi <victor.sabatier@gmail.com>` from July 2015, with the most recent at the top.

**Commit Details (Bottom Left):** Shows the commit `b496d8c48c7289ac8d4dc9d032412284e18bbe64` in detail. It includes the author (`sabativi <victor.sabatier@gmail.com> 2015-07-21 18:04:42`), committer (`sabativi <victor.sabatier@gmail.com> 2015-07-21 18:04:42`), parent (`0baab1092449e02bee078472bba4a4e175d0dbce (frf)`), branch (`tp5`), follows (`tp5-start`), and precedes (`tp2-start`). The commit message is "Revert \"frf\"".

**Diff View (Bottom):** Displays the diff between the current commit and its parent. The diff shows changes in `converter/markdown.js`. The context lines are set to 3. The diff highlights additions in green and deletions in red.

```
diff --git a/converter/markdown.js b/converter/markdown.js
@@ -49,12 +49,7 @@ var parseStrong = function(str) {
     var linkRegExp = /^[<[^>]+>]\(([^>]+)\)\)/;
     var stra = [];
     while ((stra = linkRegExp.exec(str)) !== null) {
-       console.log(stra);
-       if (stra[0].substr(0, 1) === '!') {
-           str = str.replace(stra[0], '\n');
-       } else {
-           str = str.replace(stra[0], '<a ' + 'href="' + stra[2] + '">' + stra[1] + '</a>');
-       }
+       str = str.replace(stra[0], '<a ' + 'href="' + stra[2] + '">' + stra[1] + '</a>');
     }
     return str;
   }
@@ -95,7 +90,6 @@ var parseStrong = function(str) {
```

# Où héberger son code ?



Github : gratuit pour les projets open source.

Gitlab : open source.

Bitbucket : gratuit aussi pour les projets privés.

# Configurer Git.

# Quelle est la configuration minimale de git ?

# Trois niveaux de configuration :

Système : /etc/gitconfig

Global : ~/.gitconfig

Local : .git/config

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
  ignorecase = true
  precomposeunicode = true
[remote "origin"]
  url = https://github.com/Elephorm-tp-git/Realtime-Markdown-Viewer.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
```

# Pour afficher toute votre config :

```
git config -l
```

Par exemple :

```
➔ Realtime-Markdown-Viewer git:(code) git config -l
user.name=sabativi
user.email=victor.sabatier@gmail.com
alias.amend=commit --amend --no-edit
alias.st=status
alias.who=shortlog -sne
alias.l=log --pretty=oneline --abbrev-commit --graph --decorate
alias/lg=log --pretty=oneline --abbrev-commit --graph --decorate --all
alias.changes=diff --name-status
alias.dic=diff --cached
alias.diffstat=diff --stat
```

# Ajouter une valeur :

```
git config [--global|--system] KEY VALUE
```

# Ou éditer directement le fichier :

```
git config [--global] -e
```

# Configuration minimale

Vous devez avoir une adresse mail et un nom pour enregister votre première modification sur git :

```
git config --global user.name NAME  
git config --global user.email EMAIL
```

# Quelques options<sup>5</sup>:

```
git config --global core.editor vim/emacs/notepad/...
git config --global color.ui true
git config --global help.autocorrect 1
git config --global core.autocrlf true // à faire sous windows
```

```
git config --global alias.st status
git config --global alias.co checkout
```

---

<sup>5</sup> Vous pouvez télécharger un exemple de config plus complet dans le TP.

# Consulter l'aide<sup>6</sup>

```
git help command
```

GIT-HELP(1)	Git Manual	GIT-HELP(1)
<b>NAME</b> git-help - Display help information about Git		
<b>SYNOPSIS</b> <code>git help [-a --all] [-g --guide] [-i --info -m --man -w --web] [COMMAND GUIDE]</code>		
<b>DESCRIPTION</b> With no options and no COMMAND or GUIDE given, the synopsis of the <code>git</code> command and a list of the most commonly used Git commands are printed on the standard output.  If the option <code>--all</code> or <code>-a</code> is given, all available commands are printed on the standard output.  If the option <code>--guide</code> or <code>-g</code> is given, a list of the useful Git guides is also printed on the standard output.		

<sup>6</sup> L'aide de git est complète.