

Introduction to **Jujutsu**

| "Jū" can be translated as "gentle, soft, supple, flexible, pliable, or yielding"

| "jutsu" can be translated as "art or technique"





Introduction to Jujutsu

 Thomas Fisker Christensen

■ WHAT IS Jujutsu?

- it is a powerful version control system for software projects.
- initially developed by Martin von Zweigbergk (@ Google)
 - now part of the jj-vcs org on github
- used at Google, being rolled out for General Availability in the beginning of 2026 (<https://www.youtube.com/watch?v=v90b5yPpC0A>) for their monorepo
 - Google use a different backend (not git)



- automatic snapshotting
- atomic rebasing *
- anonymous first
- first class conflicts △
- pluggable backend

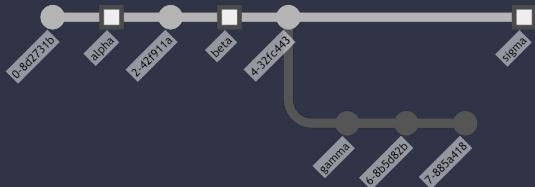


■ AUTOMATIC SNAPSHOTTING

- everything is always "checked" in (no dirty copy)
 - means you are always free to move around
 - everything is always committed
 - way of storing your changes
 - not necessarily ready to share (free to return to it and refine later)
 - every (non-ignored) file is always added
 - undeniably simpler™
 - contrast with git:
 - branch (naming is hard)
 - stage before commit
 - commit (naming is hard)



I constantly in the middle of an interactive rebase



- branches:
 - name when you are ready to share with others
- changes/commits:
 - describe when you are sure should how you want to "tell your story"



FIRST CLASS CONFLICTS



- conflicts are stored in your tree and can be solved later
- in one go or in many smaller attempts
- stored in commits
 - the inputs to the merge
 - cheap to recalculate what a conflict is
 - re-evaluate if it has been resolved
- jump around and do the work and resolve conflicts when it suits you
 - contrast: in git conflicts are "stop the world" events ☹



mergiraf (<https://mergiraf.org/>)

resolving conflicts at 60 km/h



■ mergiraf (<https://mergiraf.org/>)

■ Language semantics guided merge tool

- Rust (*.rs)
- Javascript (*.js, *.jsx, *.mjs), TypeScript (*.ts, *.tsx)
- C/C++ (*.c, *.h, *.cc, *.hh, *.cpp, *.hpp, *.cxx, *.hxx, *.c++, *.h++, *.mpp, *.ppm, *.ixx, *.tcc)
- C# (*.cs)
- Python (*.py)

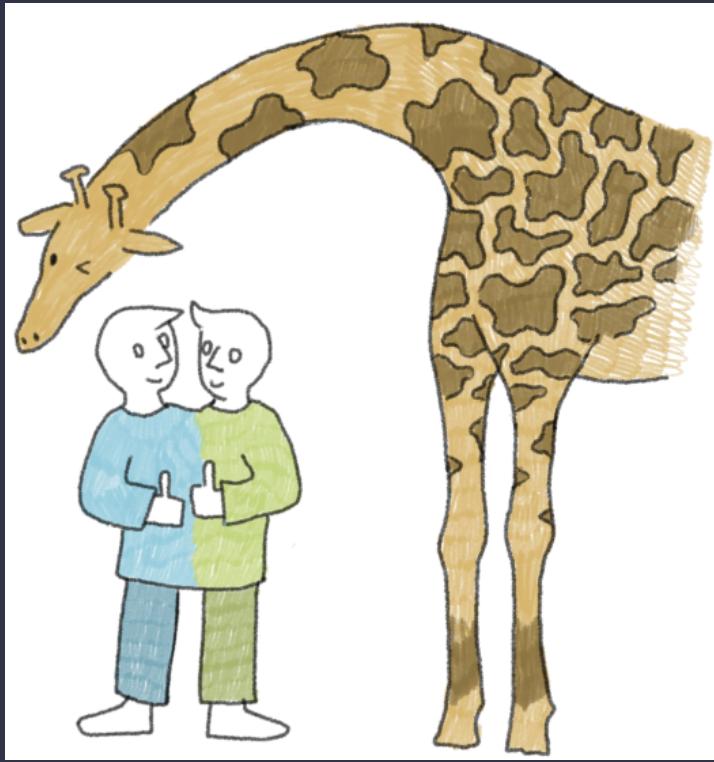
Supported out of the box in Jujutsu:

```
jj resolve --tool mergiraf
```









[back to Jujutsu](#)



■ PLUGGABLE BACKEND

- all the backends
 - "google"
 - git
- git is everywhere
- **Jujutsu** is transparent to others (git users)
- git has "ok"-ish UX



■ Stockholm syndrome?

| a psychological coping mechanism where hostages or abuse victims develop positive feelings, empathy, or even loyalty toward their captors or abusers

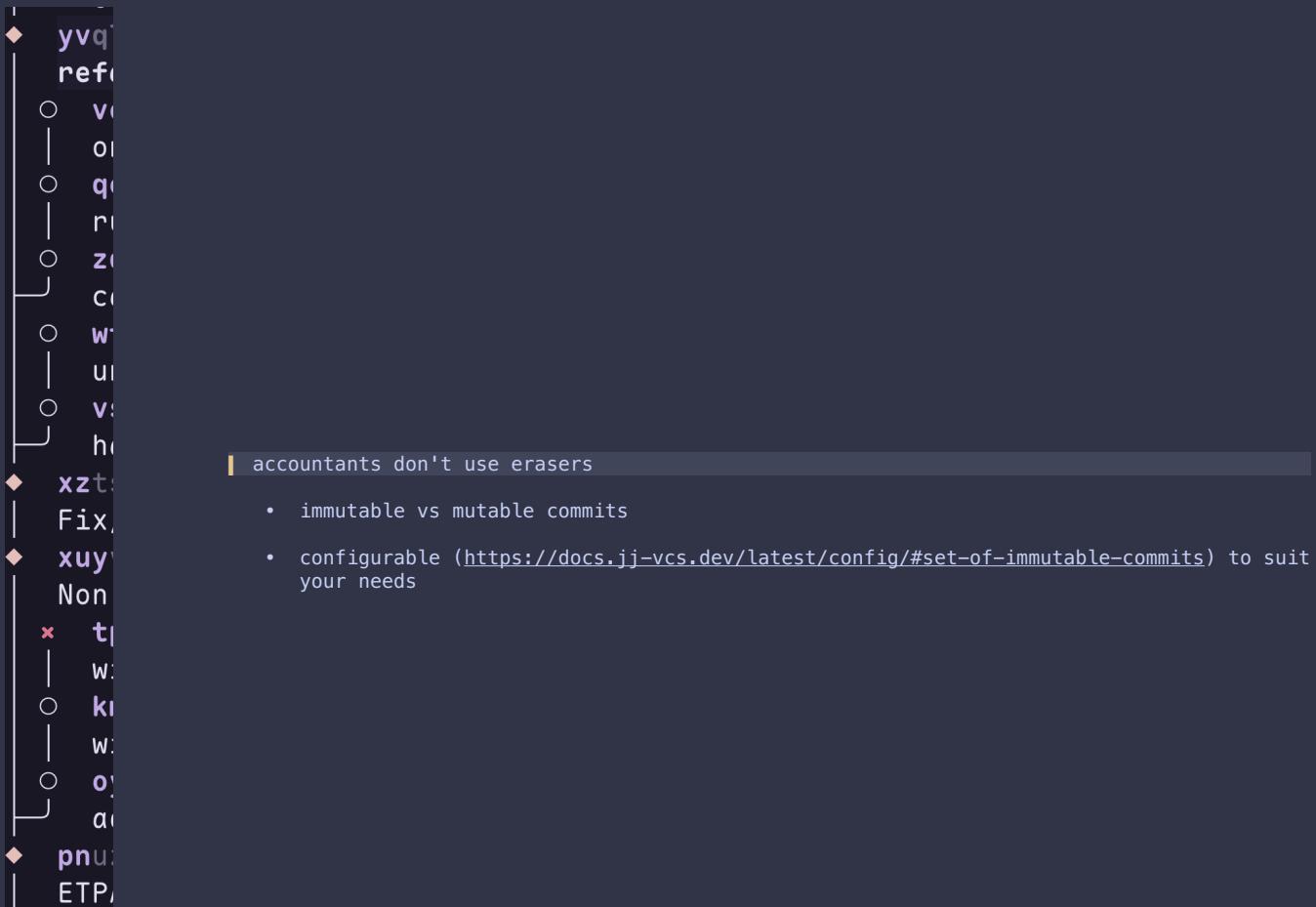
- who here uses vanilla commandline git ?
 - (without any aliases or helper scripts)?
- blink twice if you are being held against your will ☺



■ WHAT I LOVE ♥ FLUIDITY OF MOVING AROUND

- try different approaches, capture them all in VCS (not just in editor undo history)
- branch
- squash
- split
- abort
- rearrange
- rebase
- edit
- duplicate
- leads to fearless iteration/experimentation ↗
 - without ever getting in the way ♥





- git stashes are extremely linear
- Jujutsu's fluidity and constant snapshotting replace the whole concept



- `jj undo`
- `jj op log`: inspect operation log
- contrast with git:
 - `git reset`
 - hard, soft, mixed
 - `git reflog`





with help from jj and jjui (<https://github.com/idursun/jjui>)

LINK DUMP

- Homepage (<https://www.jj-vcs.dev/latest/>)
- GitHub (<https://github.com/jj-vcs/jj>)
- deadcode podcast with Steve Klabnik (<https://shows.acast.com/dead-code/episodes/deferred-conflict-with-steve-klabnik>)
- Solving Git's Pain Points with Jujutsu (with Martin von Zweigbergk) (https://www.youtube.com/watch?v=uIJ_Pw8qqsE)
- Jujutsu in practice (<https://arne.me/blog/jj-in-practice>)
- Jujutsu (jj), a git compatible VCS – Tony Finn (<https://tonyfinn.com/blog/jj/>)
- <https://schpet.com/note/why-i-think-jj-vcs-is-worth-your-time>
- <https://steveklabnik.github.io/jujutsu-tutorial/>
- <https://flames-of-code.netlify.app/blog/my-jj-workflow/>
- <https://etodd.io/2025/10/02/should-i-switch-from-git-to-jujutsu/>

