# Jujutsu Git-compatible version control

## CONCEPTS

There is no staging area: the working copy is itself a commit (@), and file edits amend it automatically.

**Edit files** amends @
↓ `jj describe`
**Describe** set the message
↓ `jj new`
**New change** fresh empty @
↓ `jj bookmark set`
**Bookmark** branch pointer
↓ `jj git push`
**Remote** origin

Change IDs (letters) stay stable when a change is rewritten; commit IDs (hex) change. Use change IDs.

Bookmarks do not follow @ automatically: move them with jj bookmark set before pushing.

## SETUP

**jj git clone** *<url>*
Clone a Git repository

**jj git init --colocate**
Use jj in an existing Git repo

**jj git init**
Create a new repository here

**jj config set --user user.name "***<name>***"**
Set your name

**jj config set --user user.email "***<email>***"**
Set your email

## CHANGES

**jj st**
Show the working-copy change

**jj diff**
Diff of the current change

**jj diff -r** *<rev>*
Diff of another change

**jj show** *<rev>*
Message and diff of a change

**jj describe -m "***<msg>***"**
Set the current change's message

## CHANGES (cont.)

**jj describe** *<rev>*
Edit another change's message

**jj commit -m "***<msg>***"**
Describe @ and start a new one

**jj new**
Start a new empty change on @

**jj new** *<rev>*
Start a new change on <rev>

**jj new -m "***<msg>***"**
New change with a message

**jj edit** *<rev>*
Make <rev> the working copy

**jj next / jj prev**
Step to child or parent

**jj restore** *<path>*
Discard changes to a file

## HISTORY

**jj log**
Show the revision graph

**jj log -r** *<revset>*
Log selected revisions

**jj log -p**
Log with patches

**jj file annotate** *<path>*
Blame a file

**jj file show -r** *<rev>* *<path>*
Print a file at a revision

## REWRITING

**jj squash**
Move @ into its parent

**jj squash --into** *<rev>*
Move @ into any ancestor

**jj squash -i**
Pick which parts to move

**jj split**
Split @ into two changes

**jj absorb**
Auto-squash edits into ancestors

**jj rebase -s** *<rev>* **-o** *<dest>*
Move a change and its descendants

**jj rebase -r** *<rev>* **-o** *<dest>*
Move only one change

## REWRITING (cont.)

**jj rebase -b** *<rev>* **-o** *<dest>*
Move a whole branch

**jj abandon** *<rev>*
Drop a change, keep its children

## BOOKMARKS & REMOTES

**jj bookmark list**
List bookmarks

**jj bookmark create** *<name>*
Create a bookmark at @

**jj bookmark set** *<name>* **-r** *<rev>*
Point a bookmark at a revision

**jj bookmark delete** *<name>*
Delete a bookmark

**jj bookmark track** *<name>*@*<remote>*
Track a remote bookmark

**jj git fetch**
Fetch from the remote

**jj git push**
Push changed bookmarks

**jj git push -b** *<name>*
Push one bookmark

**jj git push -c** *<rev>*
Push with an auto-named bookmark

**jj git remote add** *<name>* *<url>*
Add a remote

## UNDO & CONFLICTS

**jj undo**
Undo the last operation

**jj redo**
Redo an undone operation

**jj op log**
List past operations

**jj op restore** *<id>*
Restore the repo to an operation

**jj resolve**
Resolve conflicts with a merge tool

**jj resolve --list**
List conflicted files