

# Git in action

tangdeping  
July 24, 2016

# About me

- Web, RESTful, Golang, etcd
- DevOps
- Leader of common services
- [tdp100.github.com](https://tdp100.github.com)

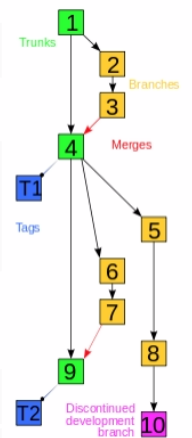


# Agenda

- History of Git
- Design of Git
- Commands of Git
- Git and Gitlab
- Best Practices in DevOps

# History of Git

V•T•E			Version control software	[hide]
Years, where available, indicate the date of first stable release. Systems with names <i>in italics</i> are no longer maintained or have planned end-of-life dates.				
Local only	Free/open-source	RCS (1982) • SCCS (1972)		
	Proprietary	PVCS (1985) • QVCS (1991)		
Client–server	Free/open-source	CVS (1986, 1990 in C) • CVSNT (1998) • QVCS Enterprise (1998) • Subversion (2000)		
	Proprietary	AccuRev SCM (2002) • ClearCase (1992) • CMVC (1994) • Dimensions CM (1980s) • DSEE (1984) • Endeavor (1980s) • Integrity (2001) • Panvalet (1970s) • Perforce (1995) • Software Change Manager (1970s) • StarTeam (1995) • Surround SCM (2002) • Synergy (1990) • Team Concert (2008) • Team Foundation Server (2005) • Visual Studio Team Services (2014) • Vault (2003) • Visual SourceSafe (1994)		
Distributed	Free/open-source	ArX (2003) • BitKeeper (1998) • Codeville (2005) • Darcs (2002) • DCVS (2002) • Fossil (2007) • Git (2005) • GNU arch (2001) • GNU Bazaar (2005) • Mercurial (2005) • Monotone (2003) • SVK (2003) • Veracity (2010)		
	Proprietary	TeamWare (1990s?) • Code Co-op (1997) • Plastic SCM (2006) • Team Foundation Server (2013) • Visual Studio Team Services (2014)		
Concepts	Branch • Fork • Changeset • Commit • Interleaved deltas • Delta compression • Data comparison • Merge • Repository • Tag • Trunk			
Category • Comparison • List				



I've never actually used CVS for the kernel

BitKeeper is commercial software

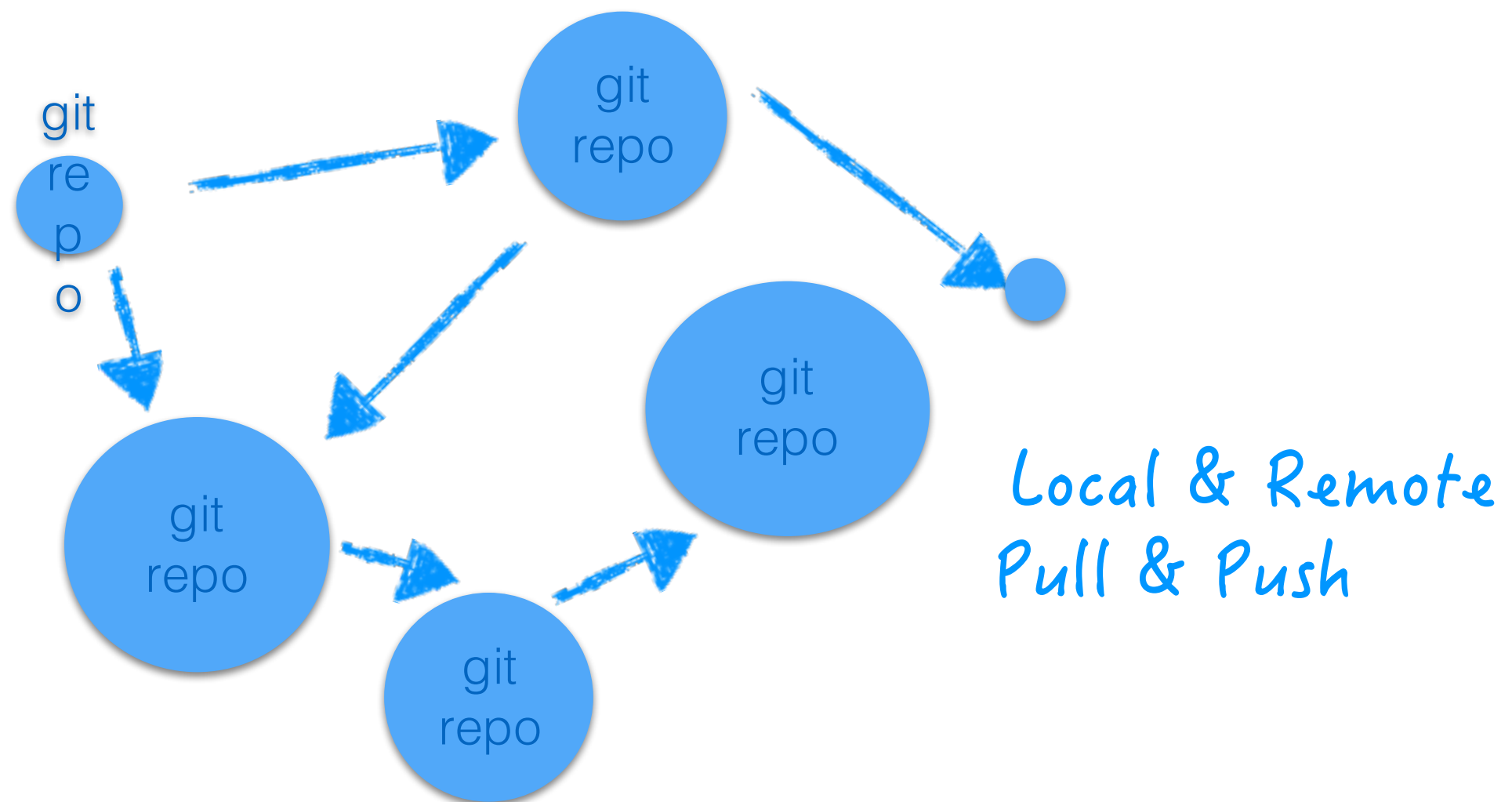
Git: a reliable, high performance, distributed content management thing, and the key word here is actually the "**distributed**" part

[https://en.wikipedia.org/wiki/Concurrent\\_Versions\\_System](https://en.wikipedia.org/wiki/Concurrent_Versions_System)

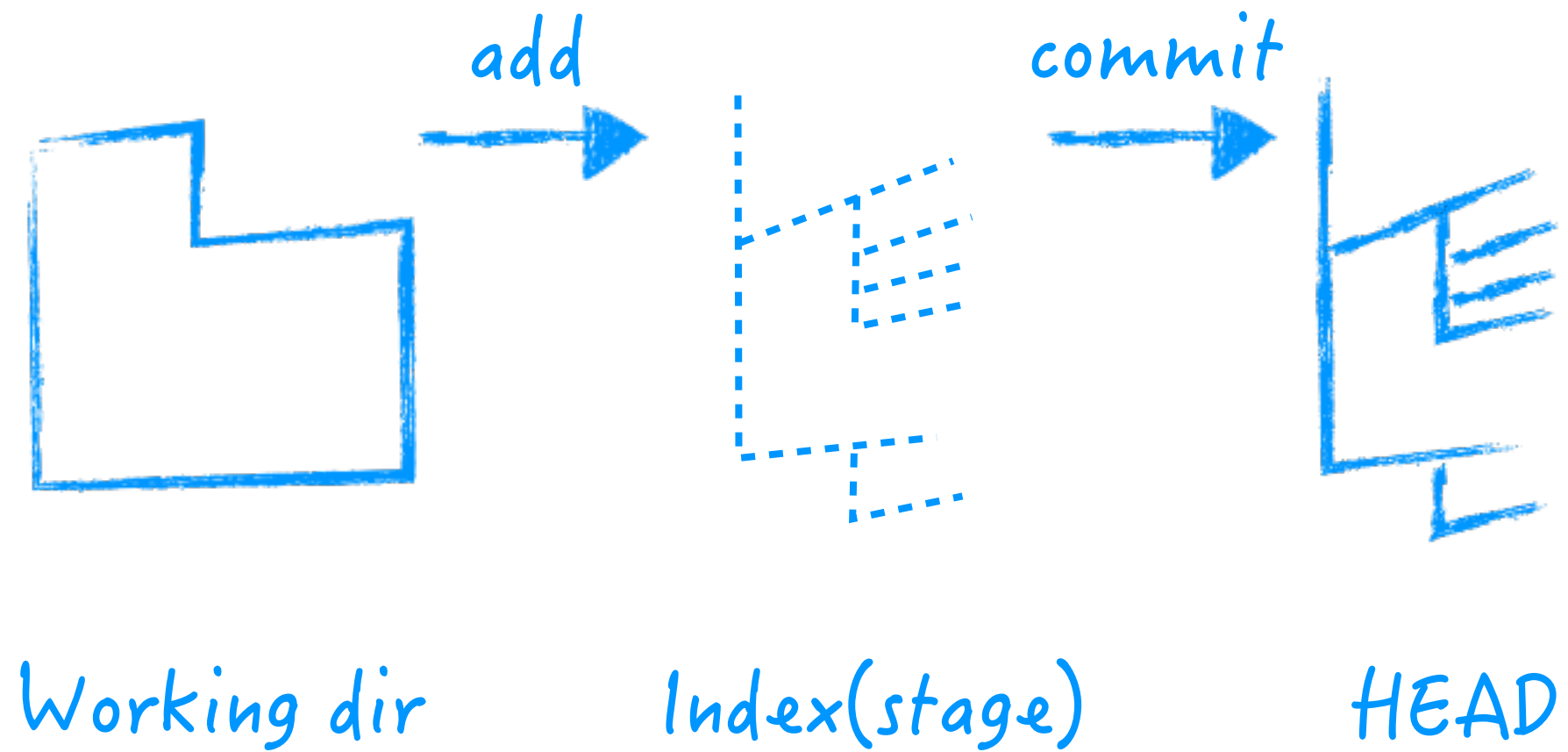
See: Tech Talk: Linus Torvalds on Git

# Design of Git

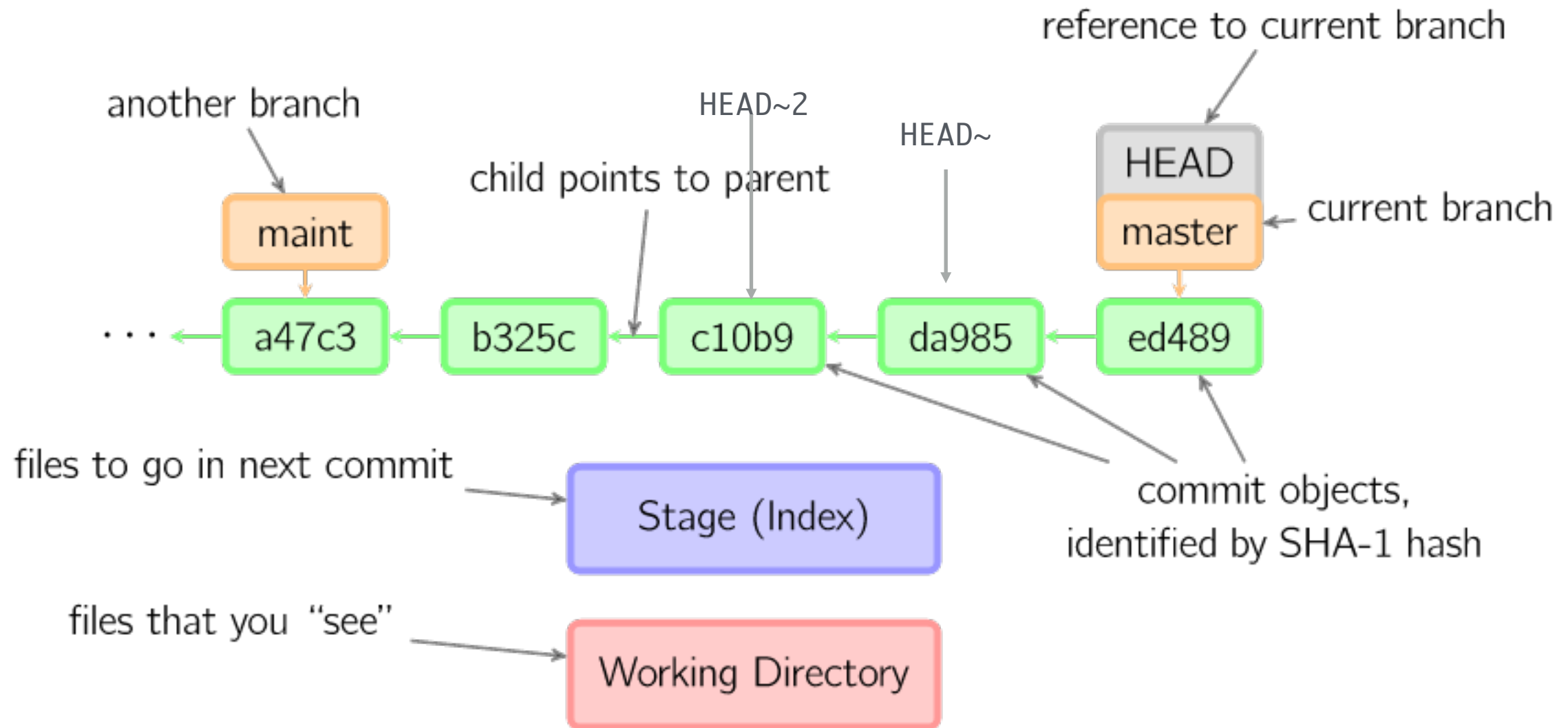
- Distributed



- Workflow



- Pointer



# Commands of Git

## **1. For local Git repo**

- add
- commit
- checkout
- reset
- merge
- rebase
- cherry-pick
- stash

## **2. For remote Git repo**

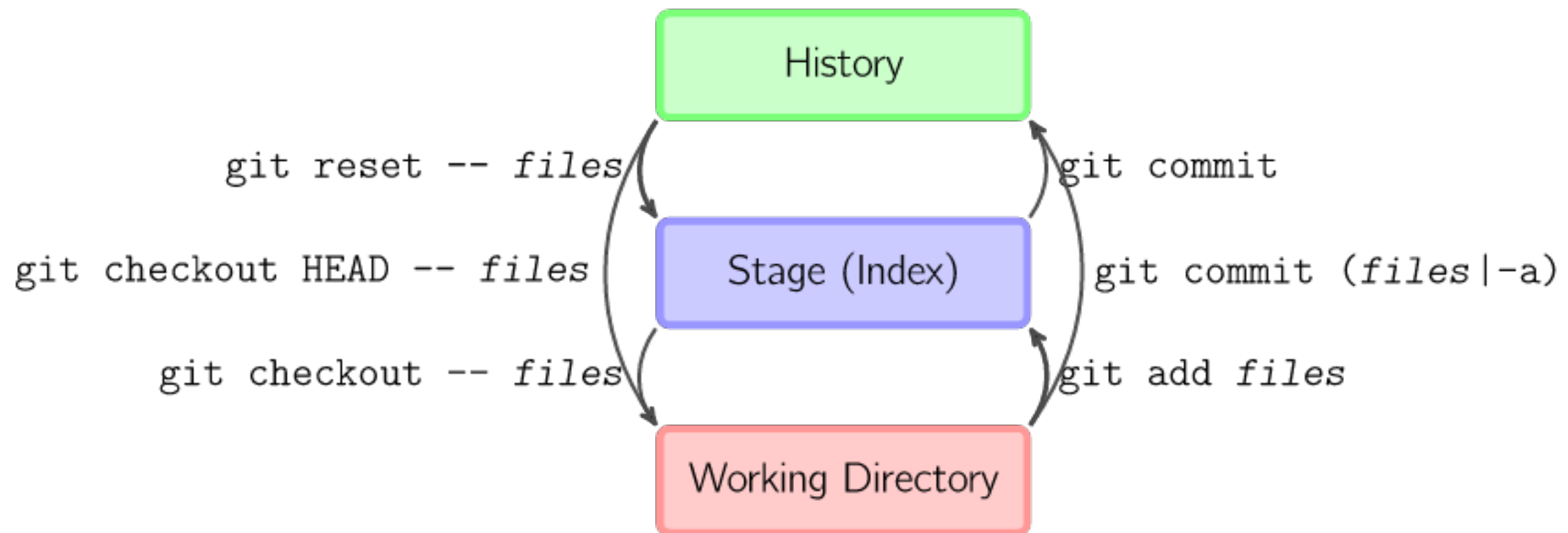
- clone
- pull
- push
- remote

## **3. For Git tool**

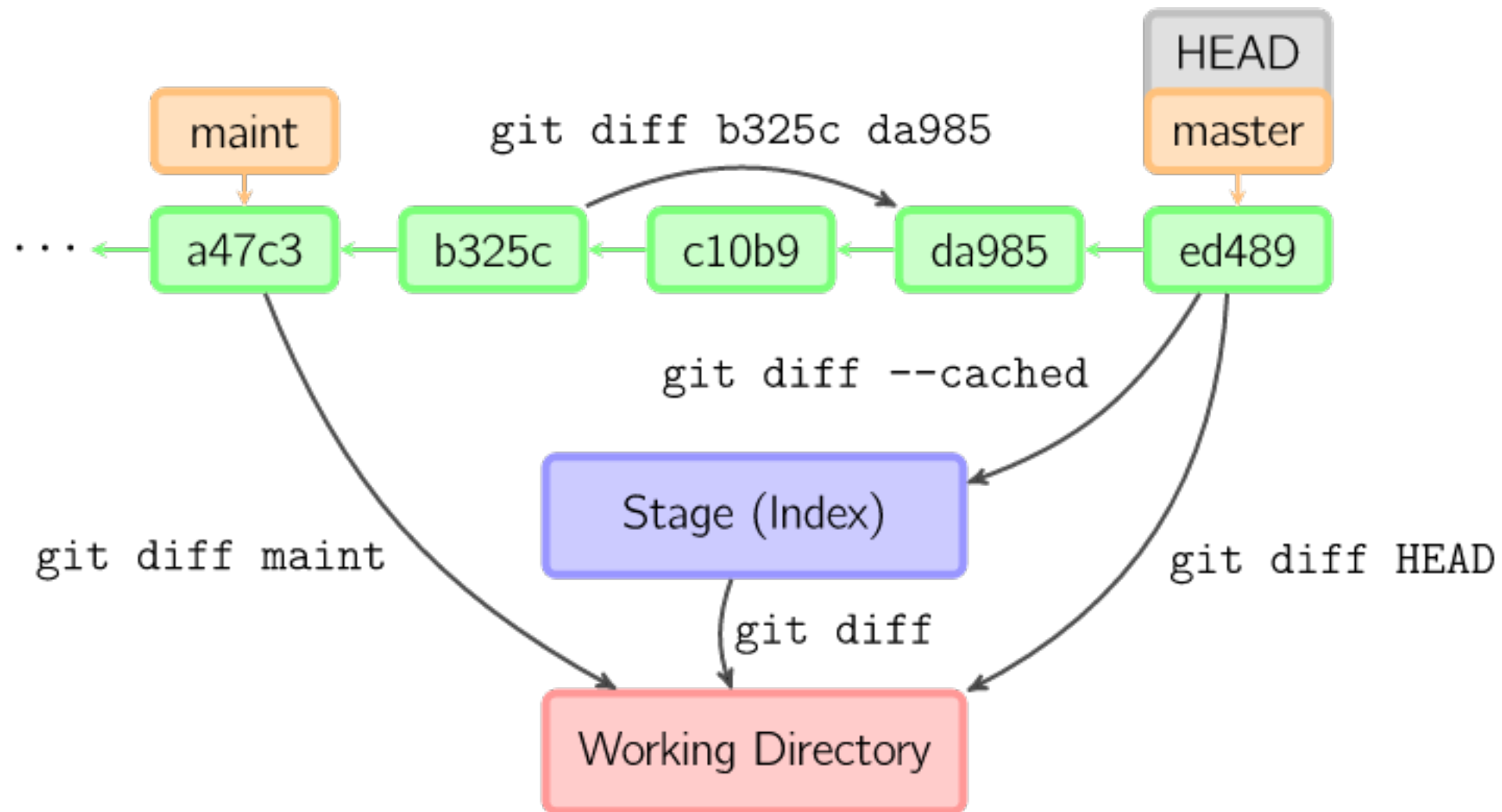
- log
- status
- config



# add/commit/checkout/reset

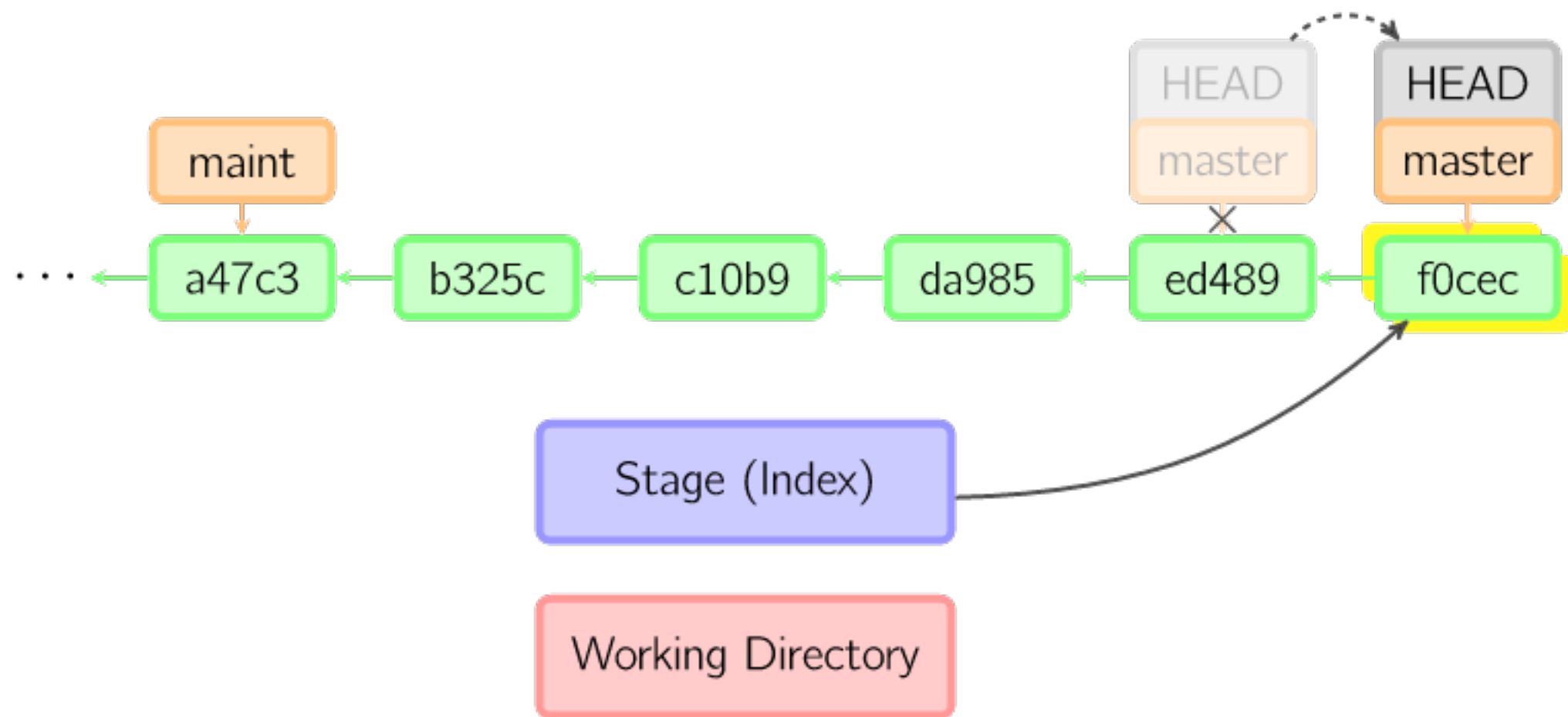


# Diff

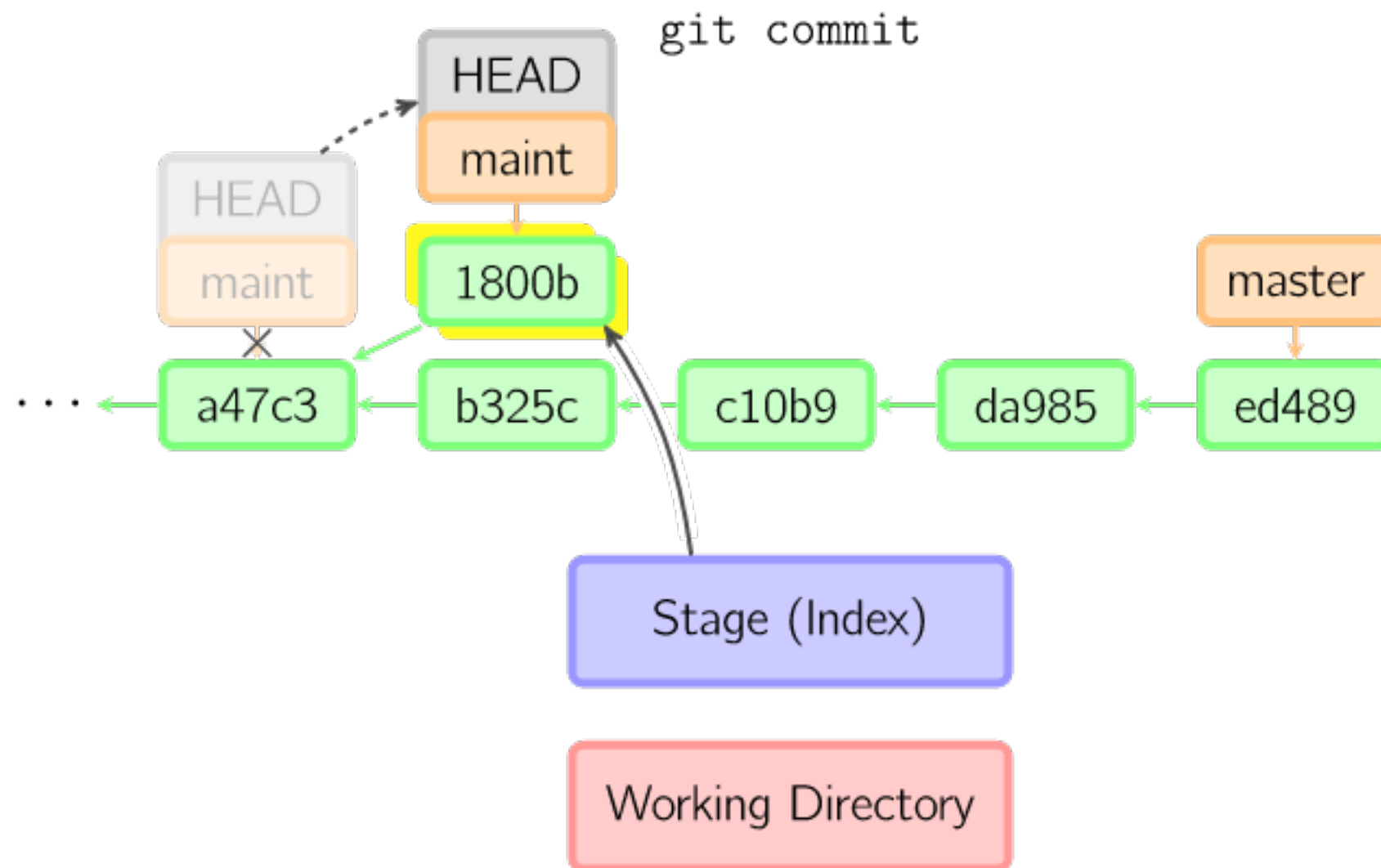


# commit

`git commit`

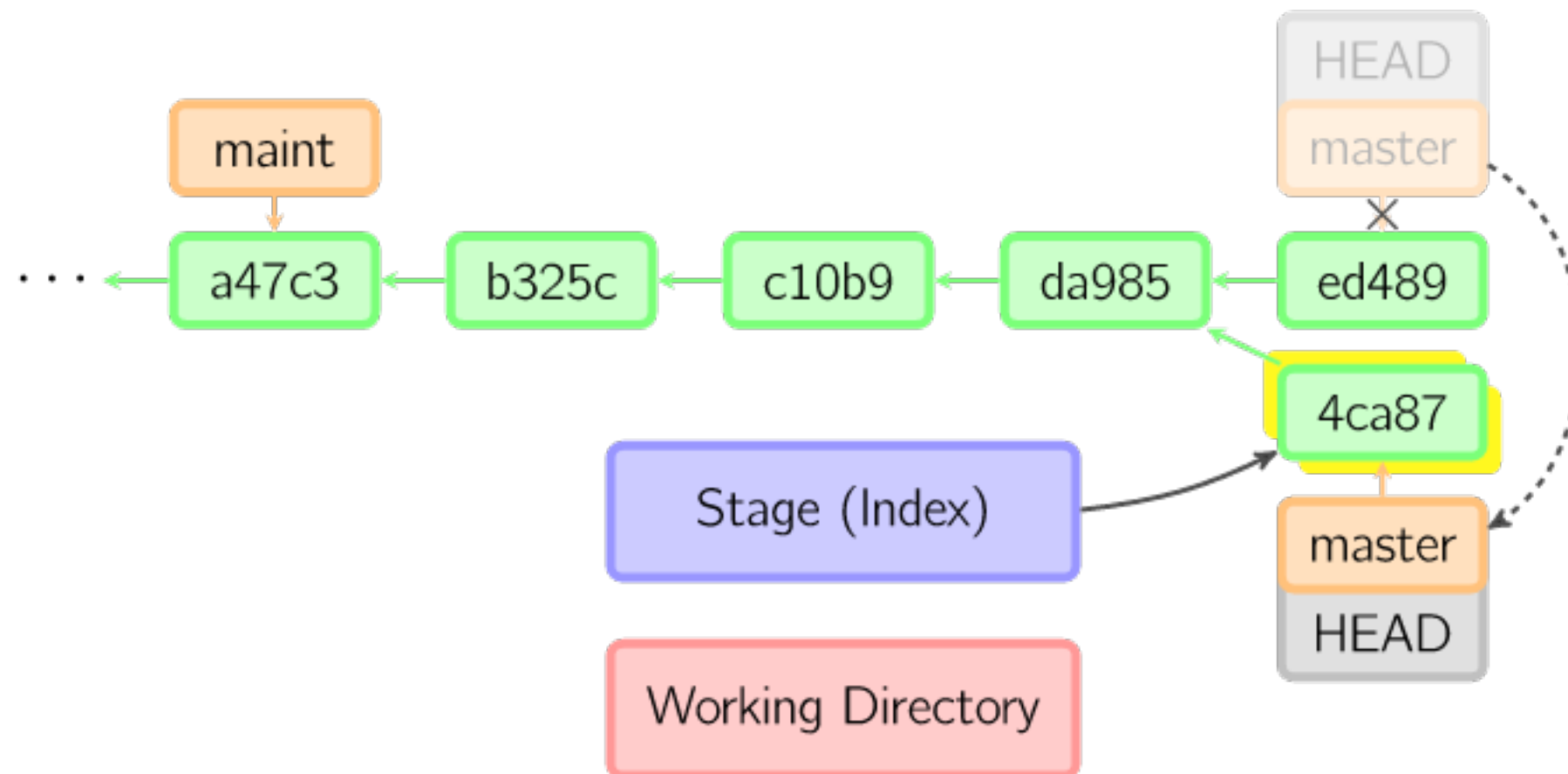


- need merge



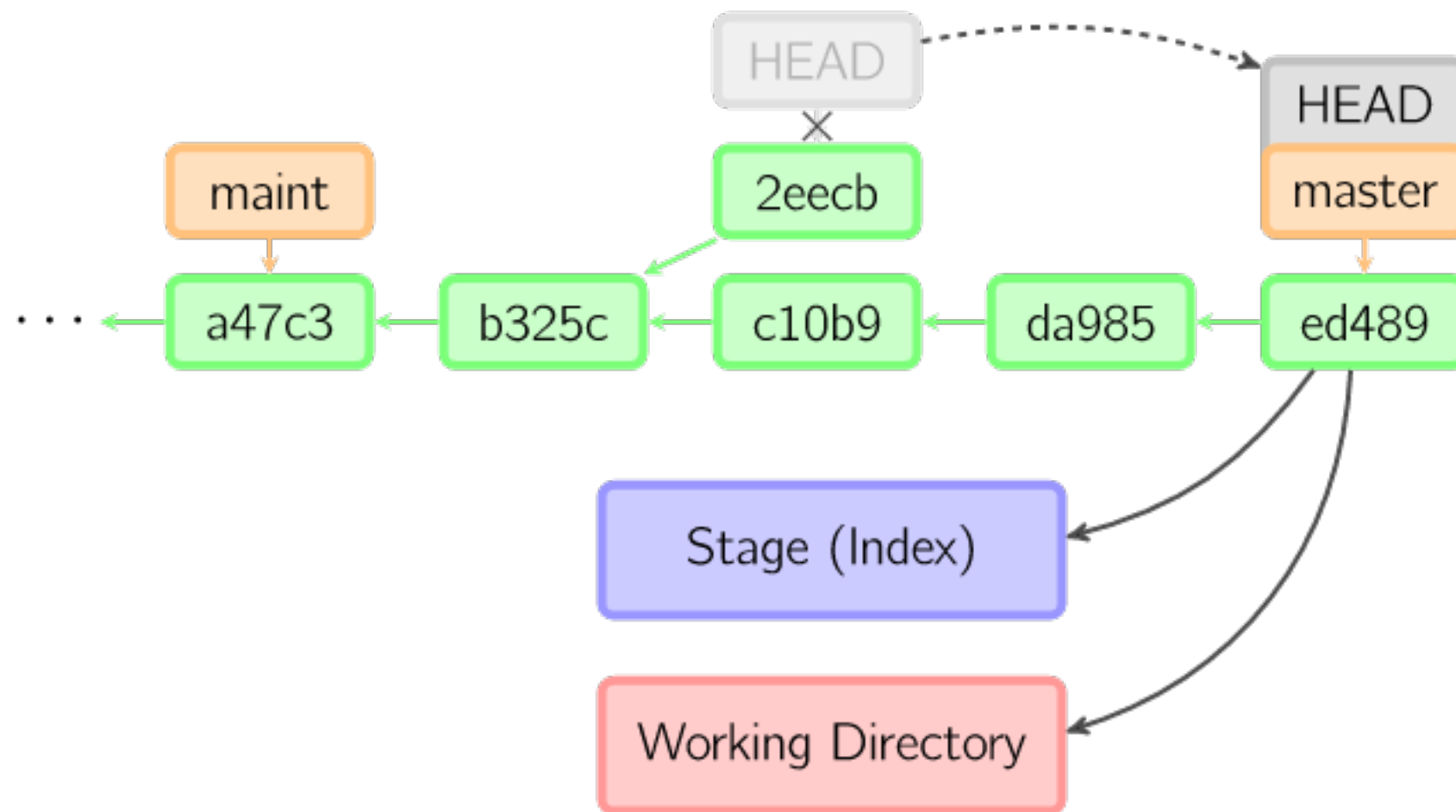
- `commit --amend`

`git commit --amend`

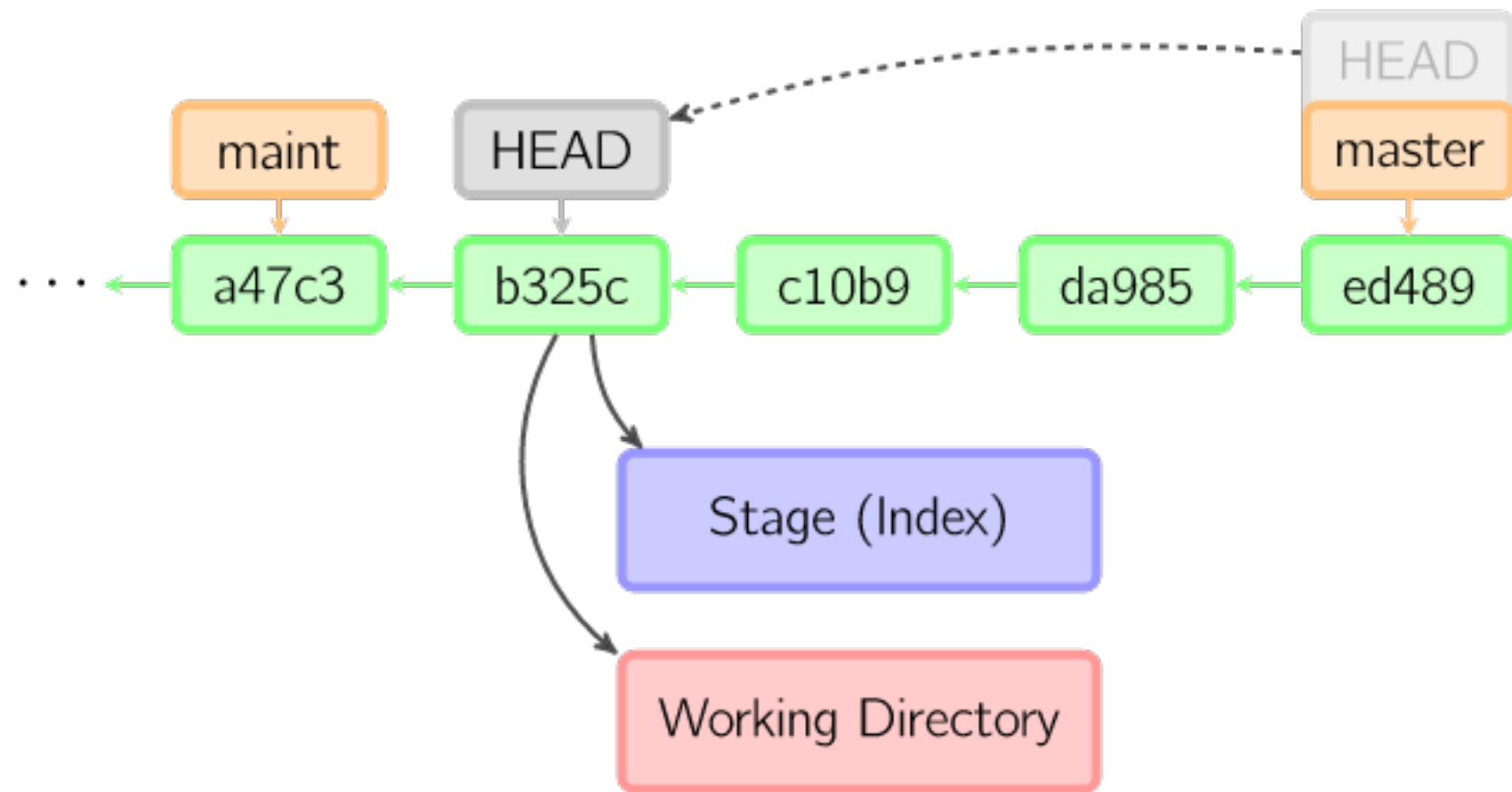


# checkout

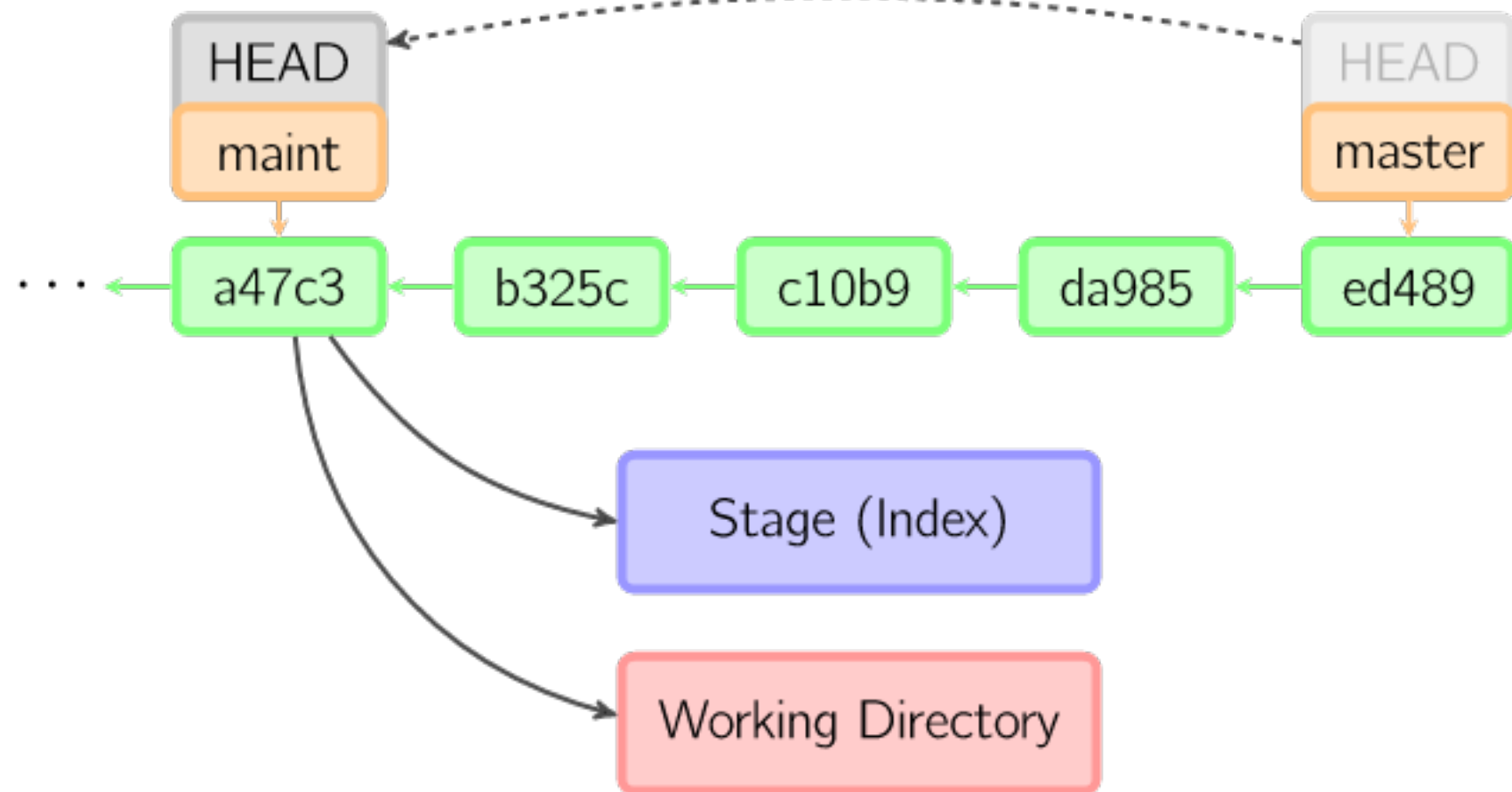
`git checkout master`



git checkout master~3



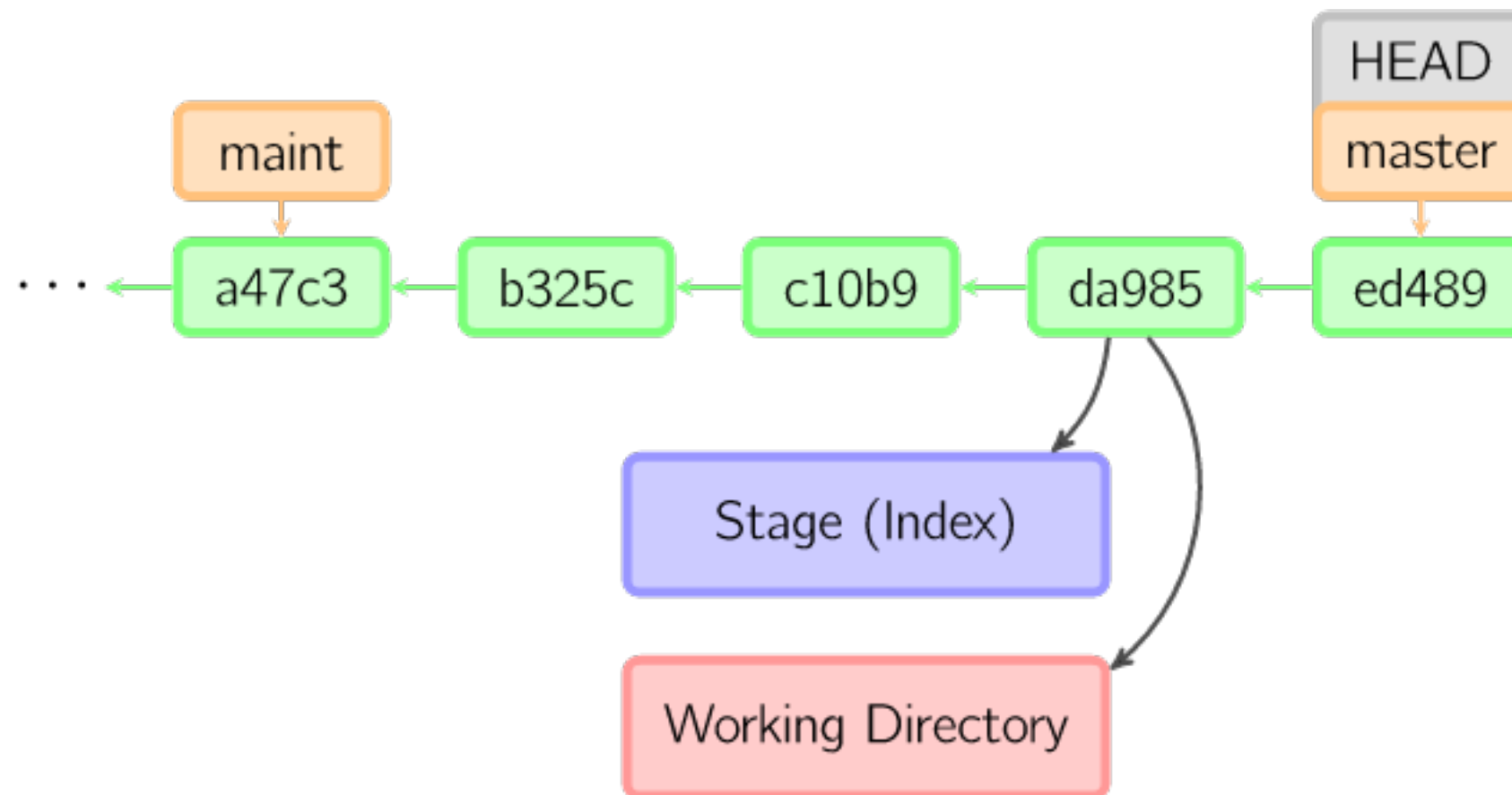
git checkout maint



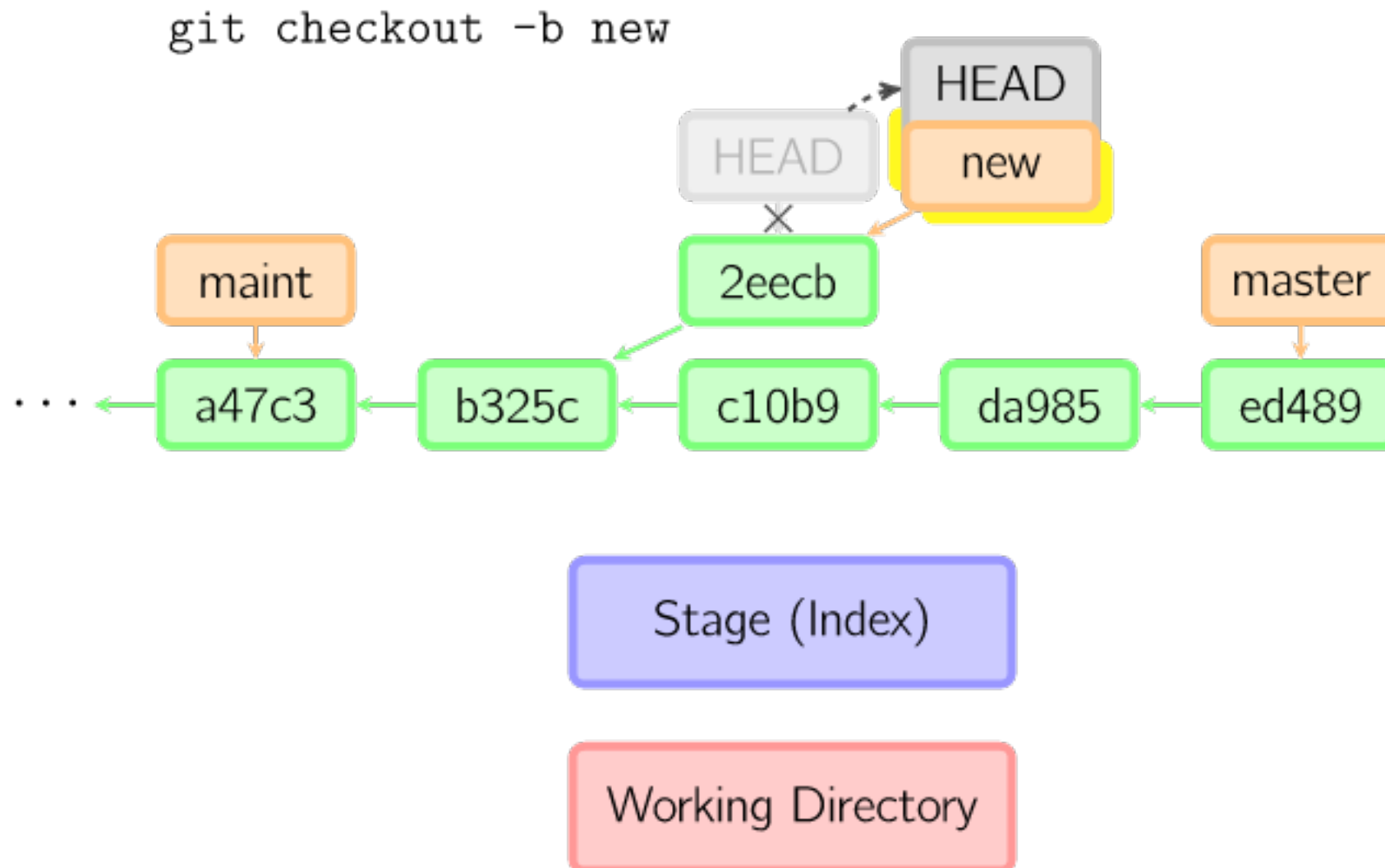


- Do not change HEAD pointer

`git checkout HEAD~ files`

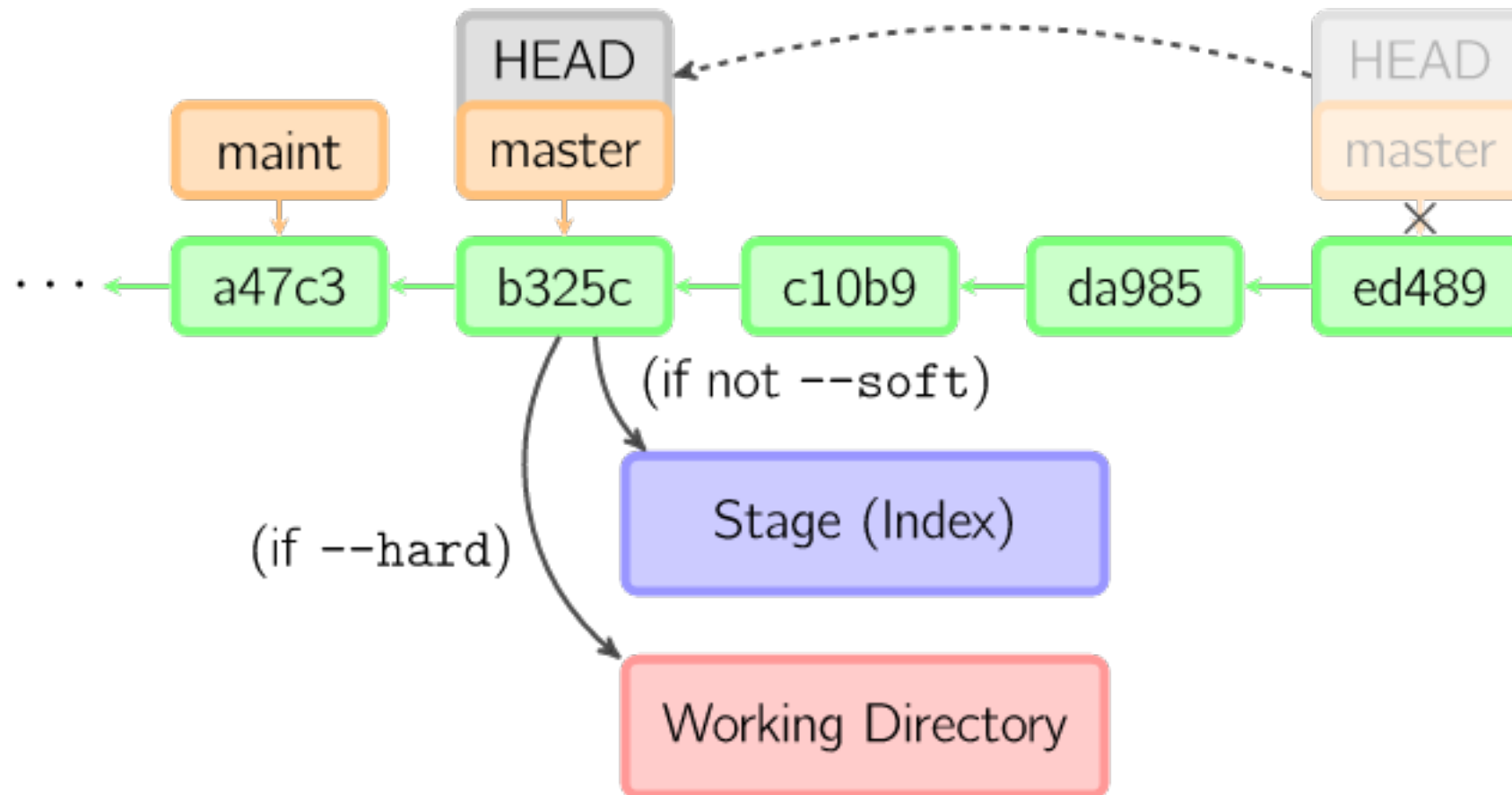


- checkout -b new

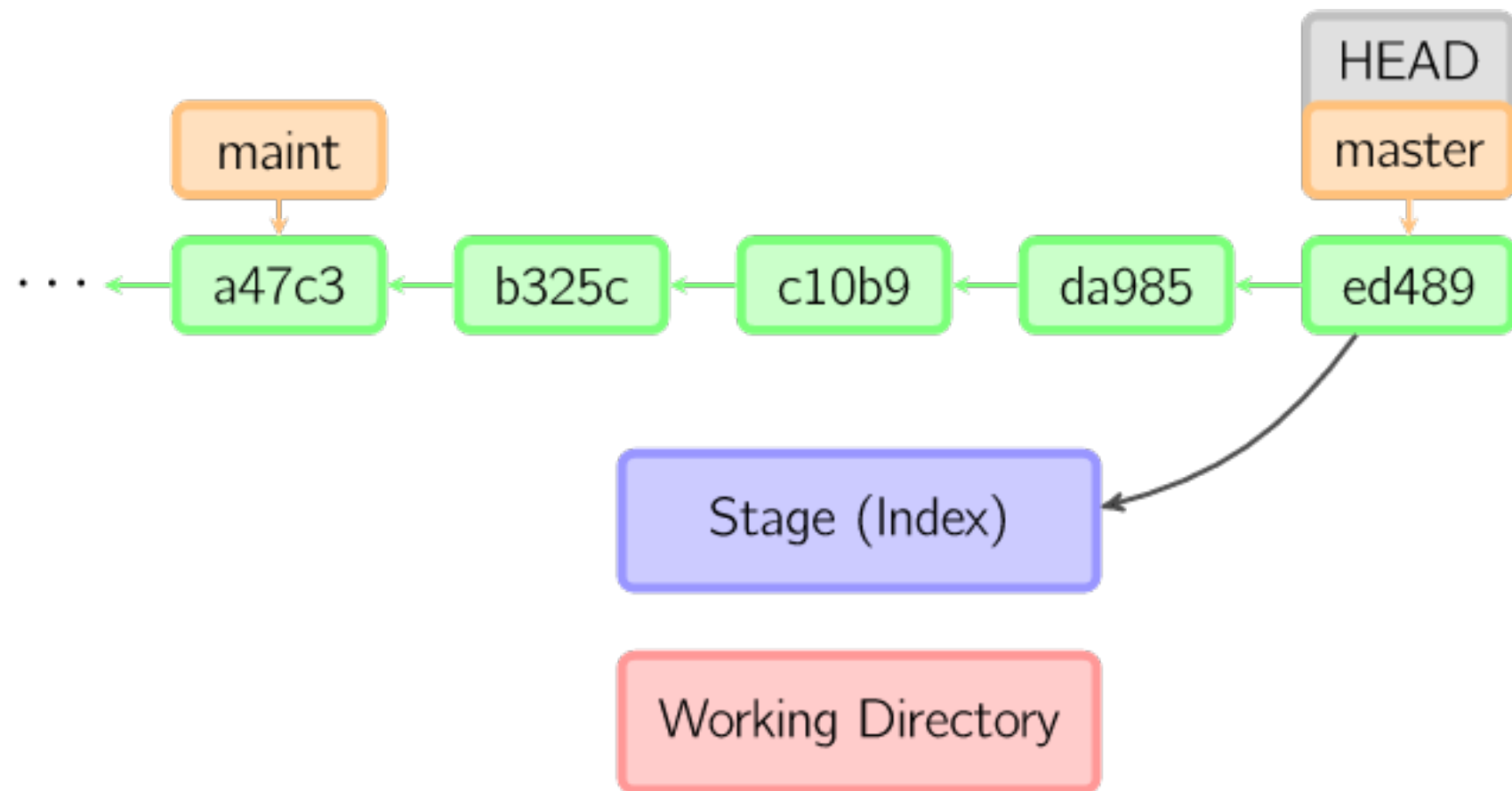


# reset

```
git reset HEAD~3
```



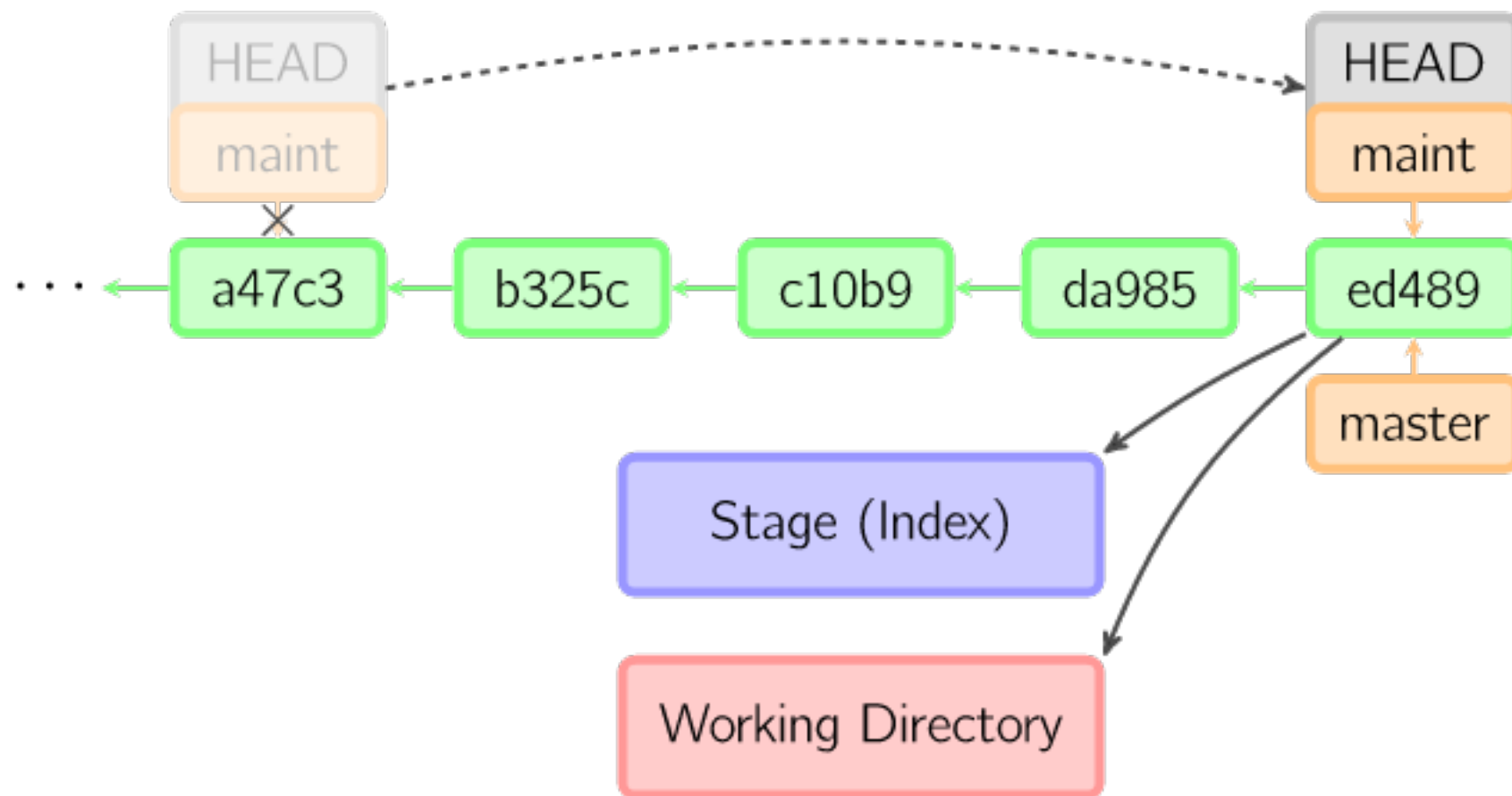
```
git reset -- files
```



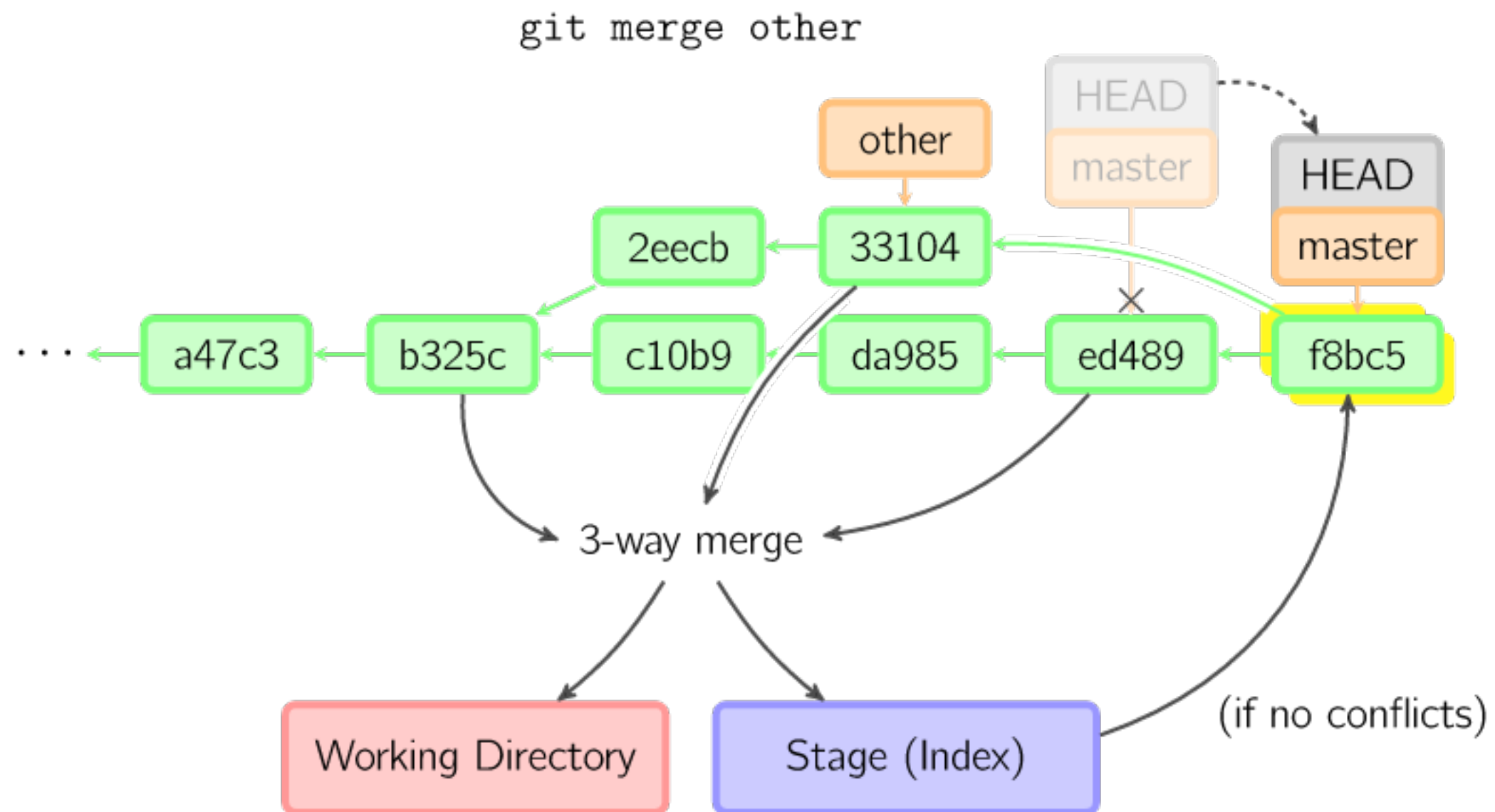
# merge

- fast-forward

`git merge master`

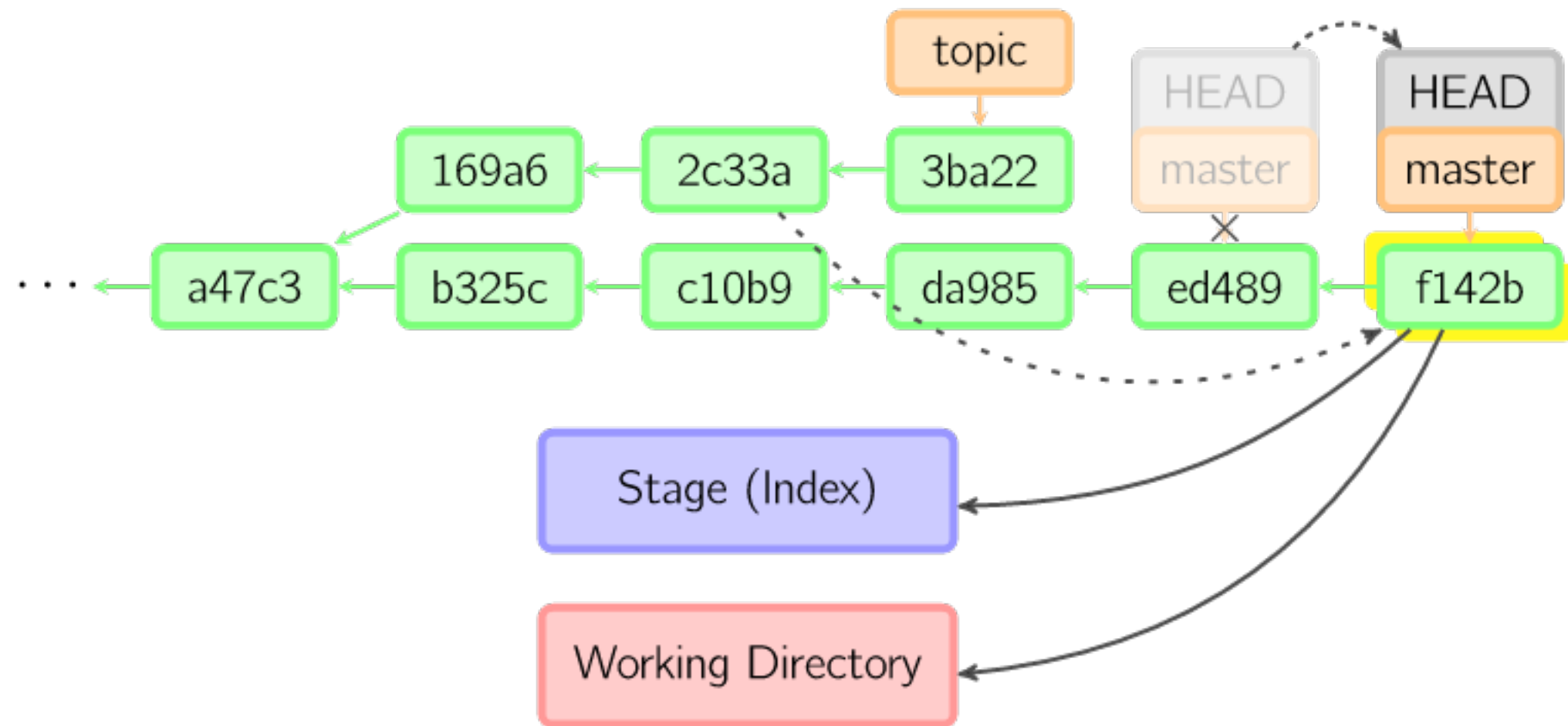


- three-way merge

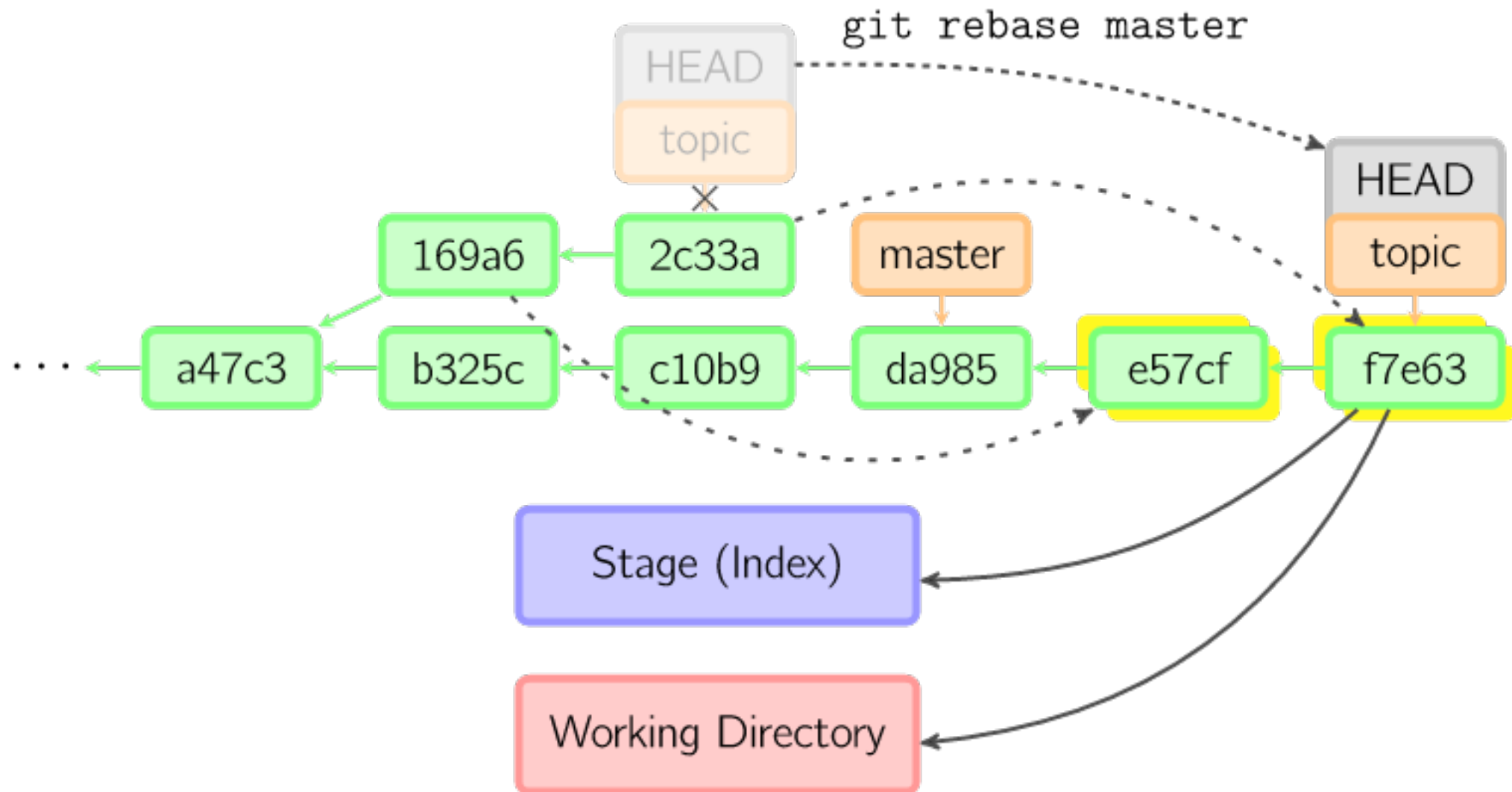


# cherry-pick(copy)

git cherry-pick 2c33a

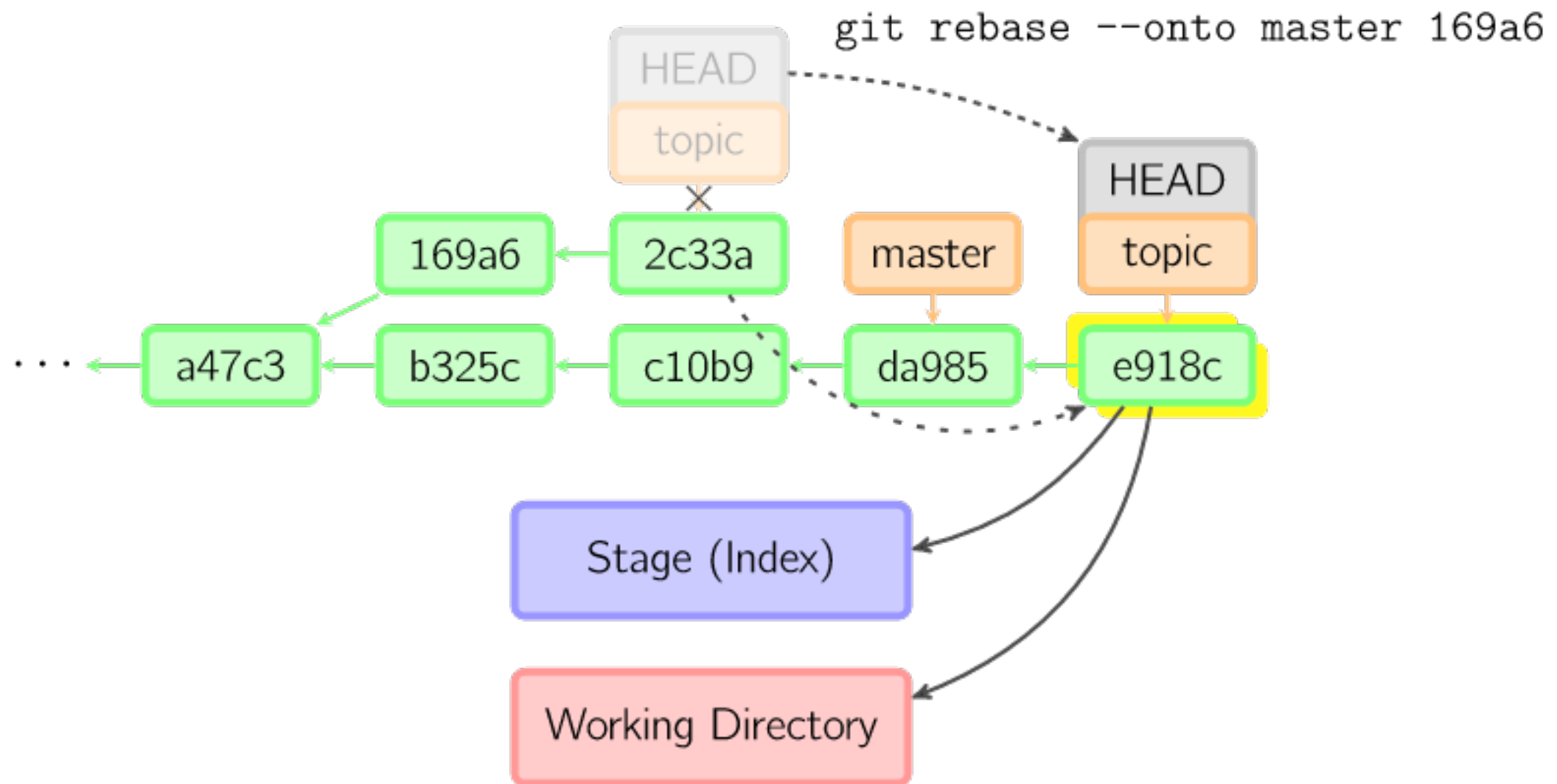


# rebase(copy)





- rebase - -onto



The command replays onto master the most recent commits on the current branch

# Git and GitLab

- Create organisation
- Create repo
- Setting: members, privileges
- SSH token

# Best Practices in DevOps

- Team Culture, +1
- Branch, tdp/feature
- Codereview
- Style of code and comments
- Wiki of markdown

End