

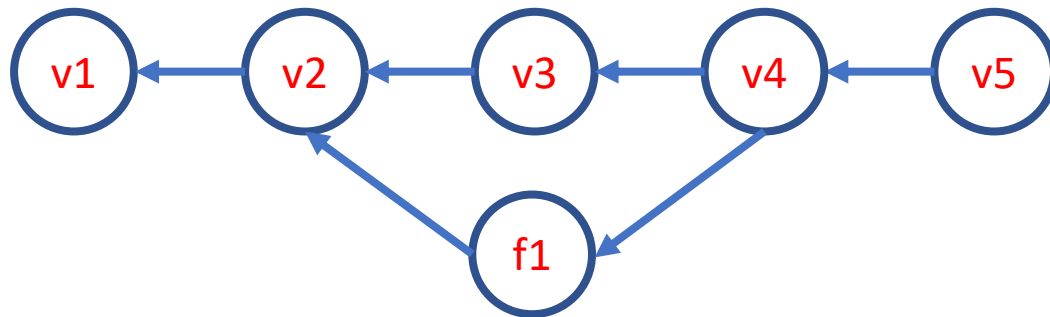
Intro to Git & GitHub

Jay / [TDMDAL](#)

Site for this workshop: <https://tdmdal.github.io/git-workshop>

What's Git git

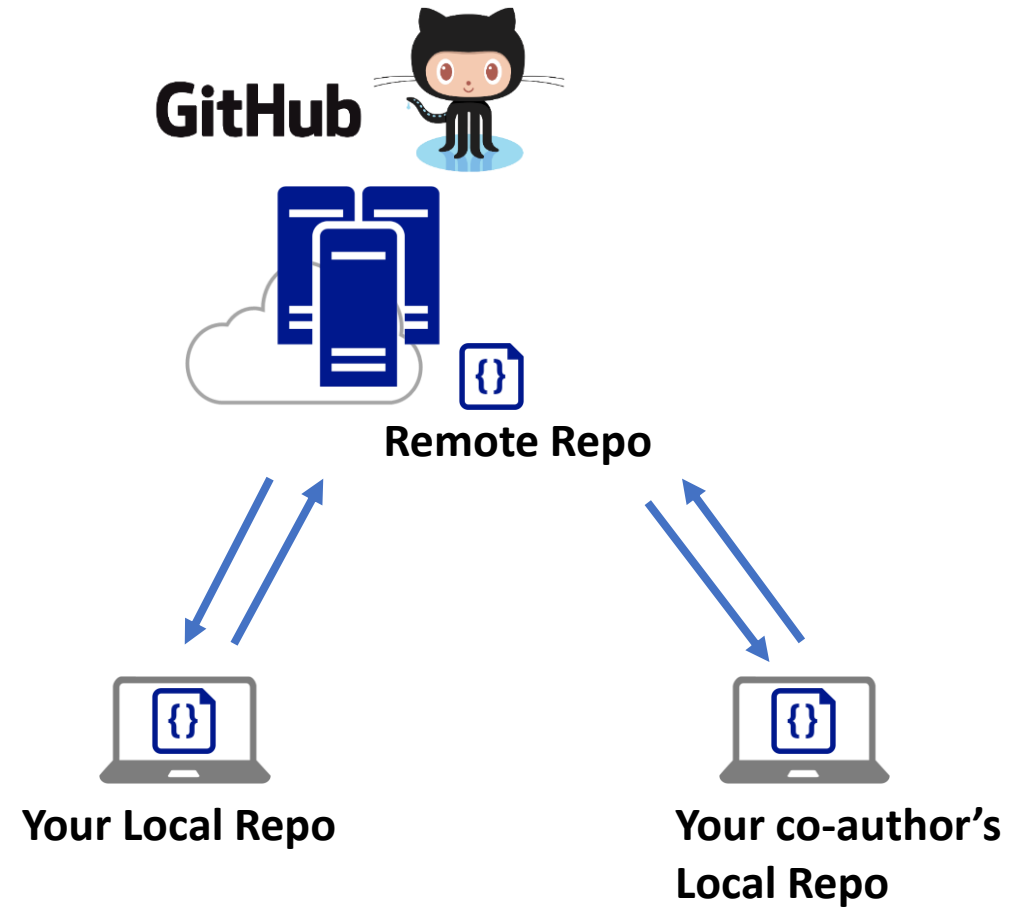
- A version control system
 - manage the evolution of a set of files (repository / repo)
 - usually for source code or text files
- Version control?
 - keep track of changes: version 1, version 2, etc.
 - like “Track Changes” or “undo” in MS Word, but much more powerful



What's GitHub

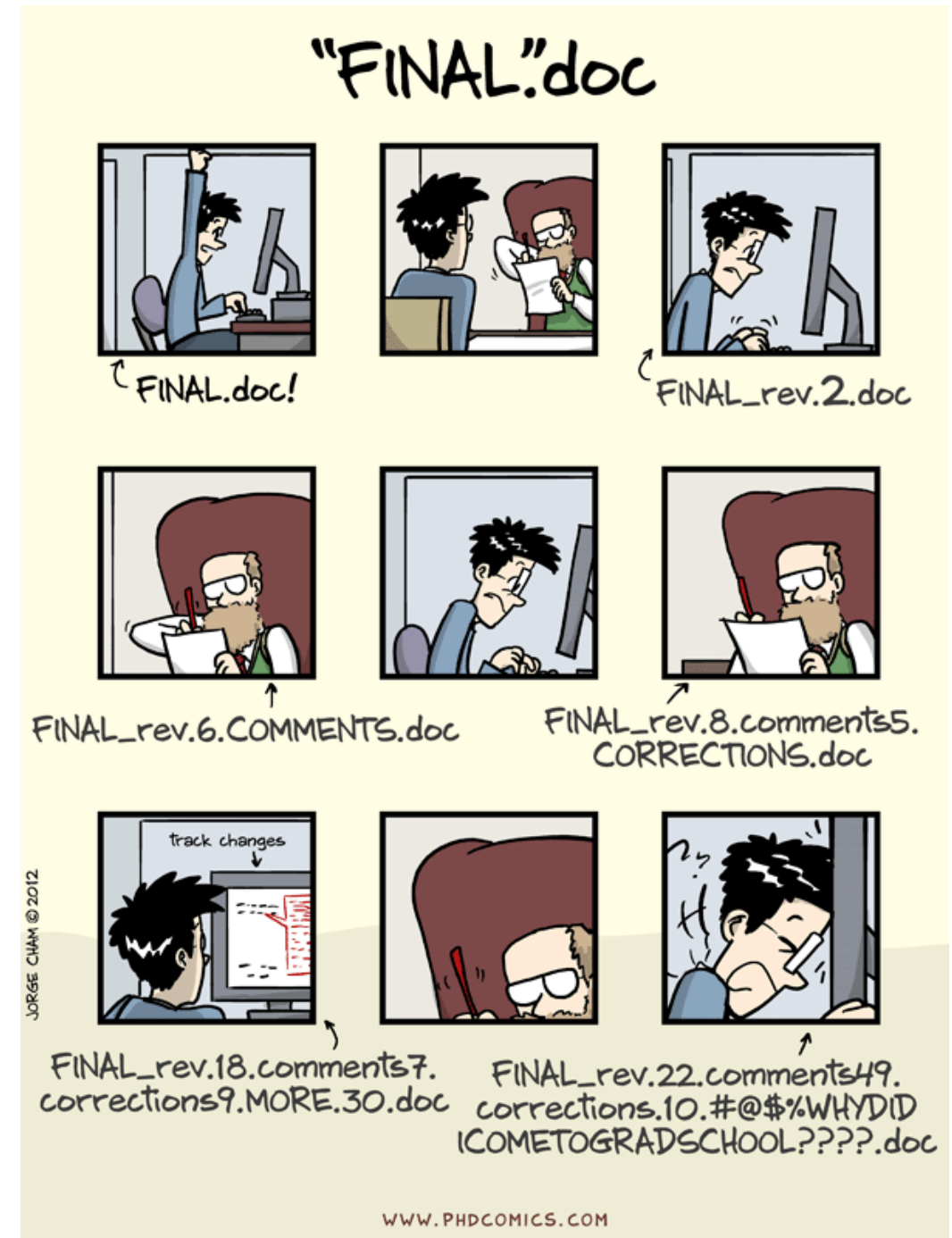
- A git-aware online repo host
- Enable repo sharing and collaboration
 - raise issues, pull request, etc.
- Free public and private repo (*)
- Other repo hosts exist
 - e.g. [gutbucket](https://github.com/pricing), [gitlab](https://github.com/pricing), etc.

Ref: <https://github.com/pricing>



Why Git & GitHub

- Organize (record keeping; traceability)
 - Track, compare and undo changes
 - Manage multiple versions/ideas at the same time efficiently
 - Backup your work
- Share
- Collaborate
 - co-authors
 - open source community

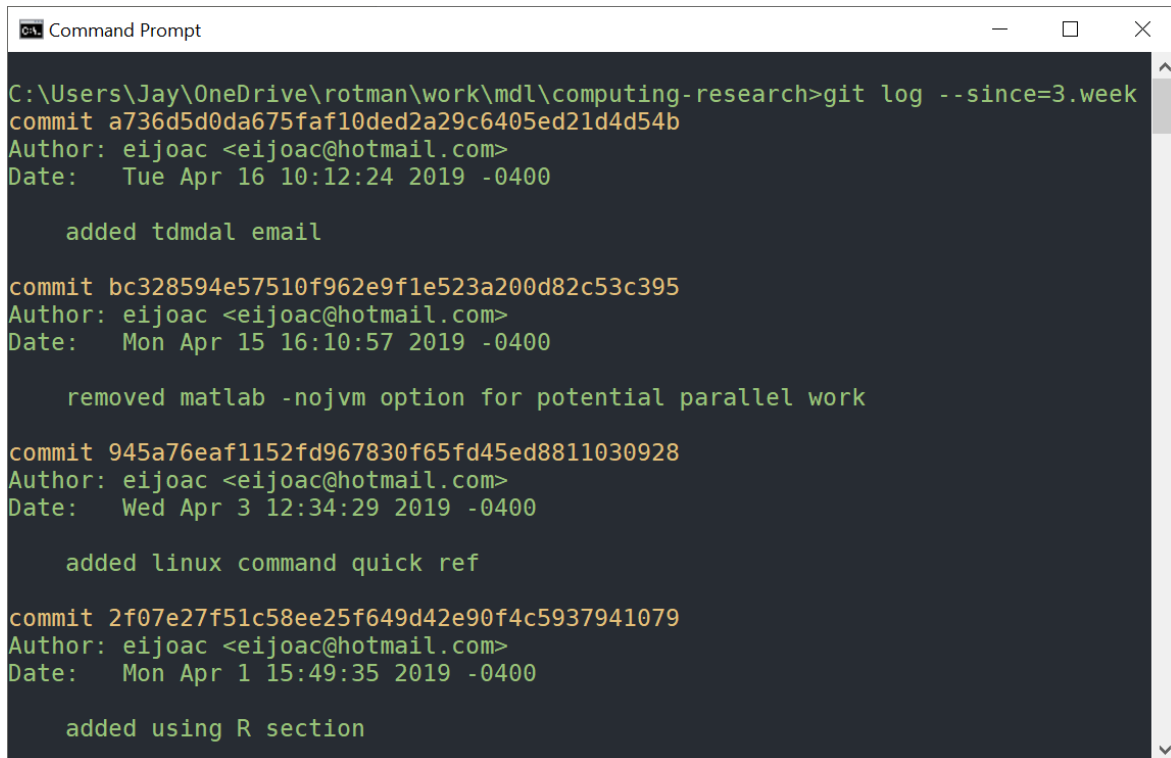


Other Benefits

- GitHub can host your (static) website, i.e., a free web host
 - GitHub Pages: <https://pages.github.com/>
 - ex. <https://tdmdal.github.io>
- Digital presence for your research or classroom work
 - ex. <https://github.com/jesusfv/>
 - ex. <https://github.com/Computational-Content-Analysis-2018>

Using Git: Command Line vs GUI Clients

- Installation: <https://git-scm.com/downloads>
- Command Line vs GUI clients (many of them)



```
Command Prompt

C:\Users\Jay\OneDrive\rotman\work\mdl\computing-research>git log --since=3.week
commit a736d5d0da675faf10ded2a29c6405ed21d4d54b
Author: eiyoac <eiyoac@hotmail.com>
Date: Tue Apr 16 10:12:24 2019 -0400

    added tdmdal email

commit bc328594e57510f962e9f1e523a200d82c53c395
Author: eiyoac <eiyoac@hotmail.com>
Date: Mon Apr 15 16:10:57 2019 -0400

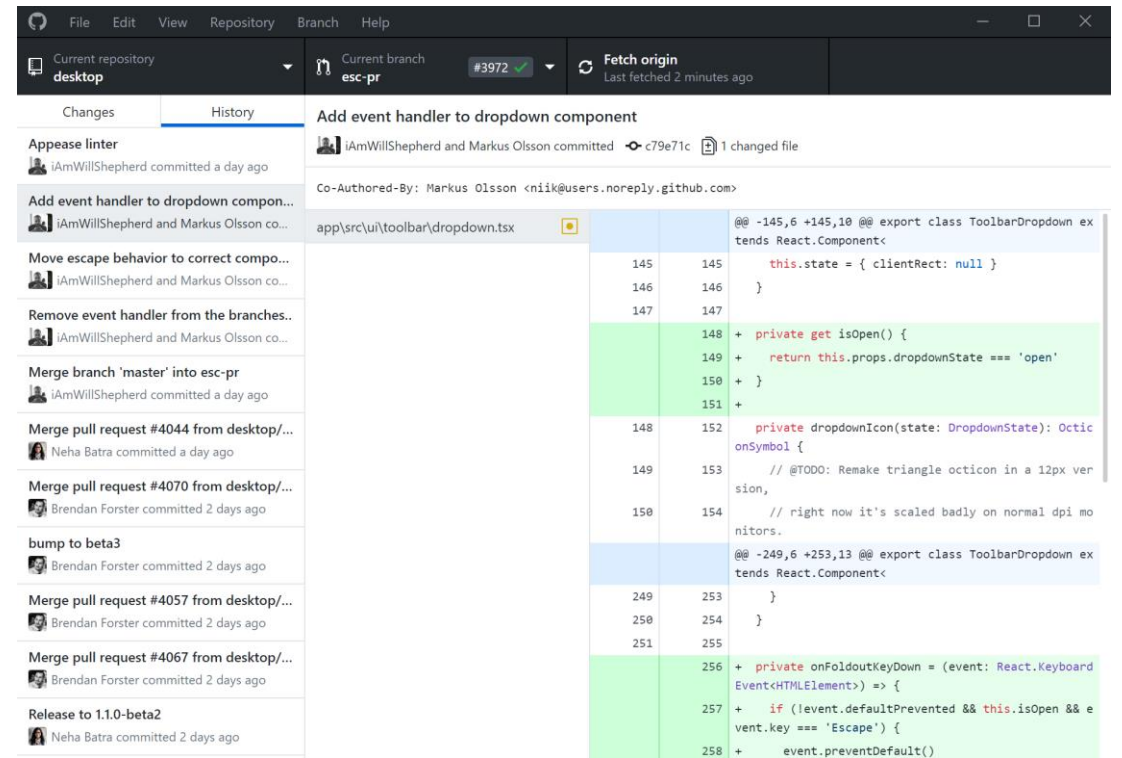
    removed matlab -nojvm option for potential parallel work

commit 945a76eaf1152fd967830f65fd45ed8811030928
Author: eiyoac <eiyoac@hotmail.com>
Date: Wed Apr 3 12:34:29 2019 -0400

    added linux command quick ref

commit 2f07e27f51c58ee25f649d42e90f4c5937941079
Author: eiyoac <eiyoac@hotmail.com>
Date: Mon Apr 1 15:49:35 2019 -0400

    added using R section
```



GitHub Desktop interface showing commit history and a diff view.

Current repository: desktop | Current branch: esc-pr | Fetch origin (Last fetched 2 minutes ago)

Changes | History

Appense linter
iAmWillShepherd committed a day ago

Add event handler to dropdown compon...
iAmWillShepherd and Markus Olsson co...

Move escape behavior to correct compo...
iAmWillShepherd and Markus Olsson co...

Remove event handler from the branches..
iAmWillShepherd and Markus Olsson co...

Merge branch 'master' into esc-pr
iAmWillShepherd committed a day ago

Merge pull request #4044 from desktop/...
Neha Batra committed a day ago

Merge pull request #4070 from desktop/...
Brendan Forster committed 2 days ago

bump to beta3
Brendan Forster committed 2 days ago

Merge pull request #4057 from desktop/...
Brendan Forster committed 2 days ago

Merge pull request #4067 from desktop/...
Brendan Forster committed 2 days ago

Release to 1.1.0-beta2
Neha Batra committed 2 days ago

Add event handler to dropdown component
iAmWillShepherd and Markus Olsson committed c79e71c 1 changed file

Co-Authored-By: Markus Olsson <niki@users.noreply.github.com>

app\src\ui\toolbar\dropdown.tsx

Line	Diff
145	@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
146	this.state = { clientRect: null }
147	}
148	+ private get isOpen() {
149	+ return this.props.dropdownState === 'open'
150	+ }
151	+ }
152	private dropdownIcon(state: DropdownState): OcticonSymbol {
153	// @TODO: Remake triangle octicon in a 12px version,
154	// right now it's scaled badly on normal dpi monitors.
155	@@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
249	}
250	}
251	}
252	}
253	}
254	}
255	}
256	+ private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
257	+ if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
258	+ event.preventDefault()

The simplest git workflow (demo)

1. make changes to your files
2. add/stage the changes (to the staging area): `git add`
3. commit the changes (i.e., record version history): `git commit`
4. repeat (back to 1)...

Configure git for first-time use: `git config`

Create a new local repo: `git init`

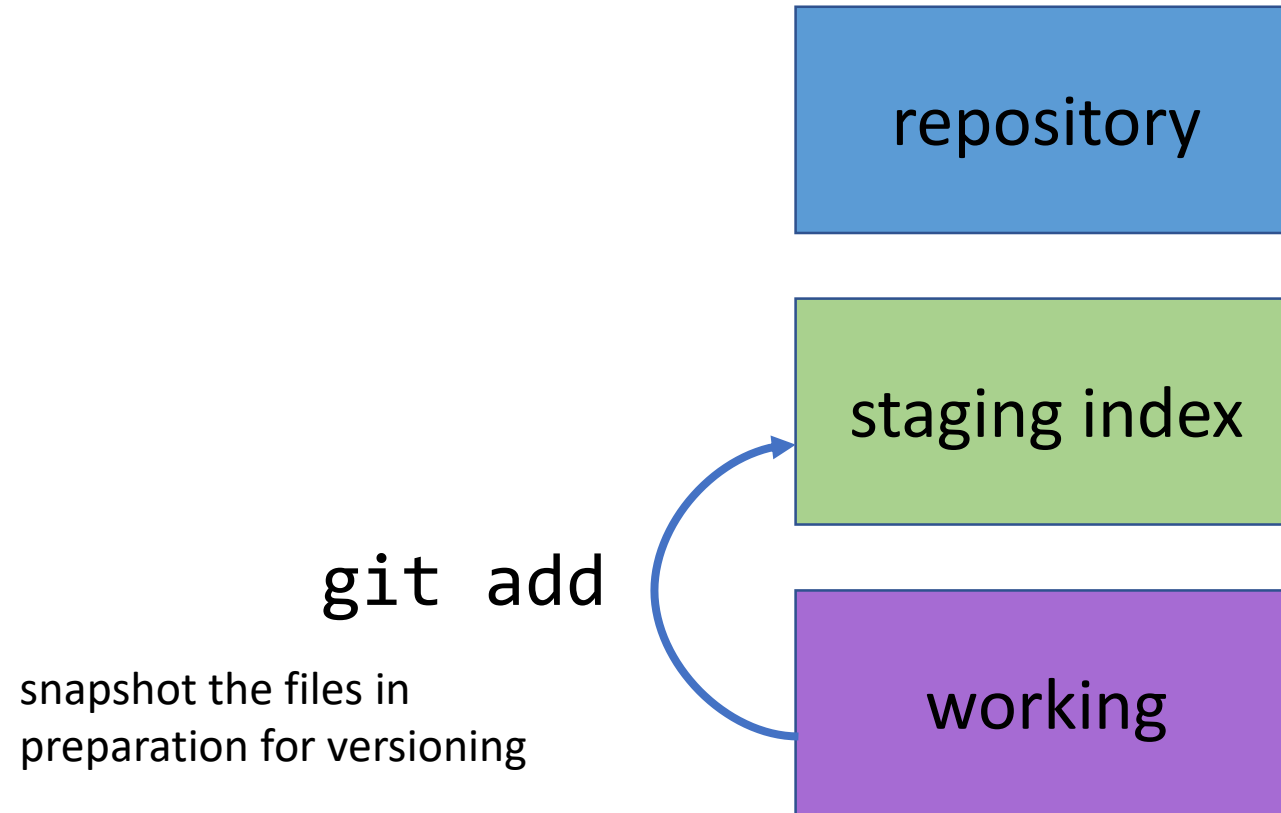
check commit history: `git log`; `git show`

compare difference between changes: `git diff`

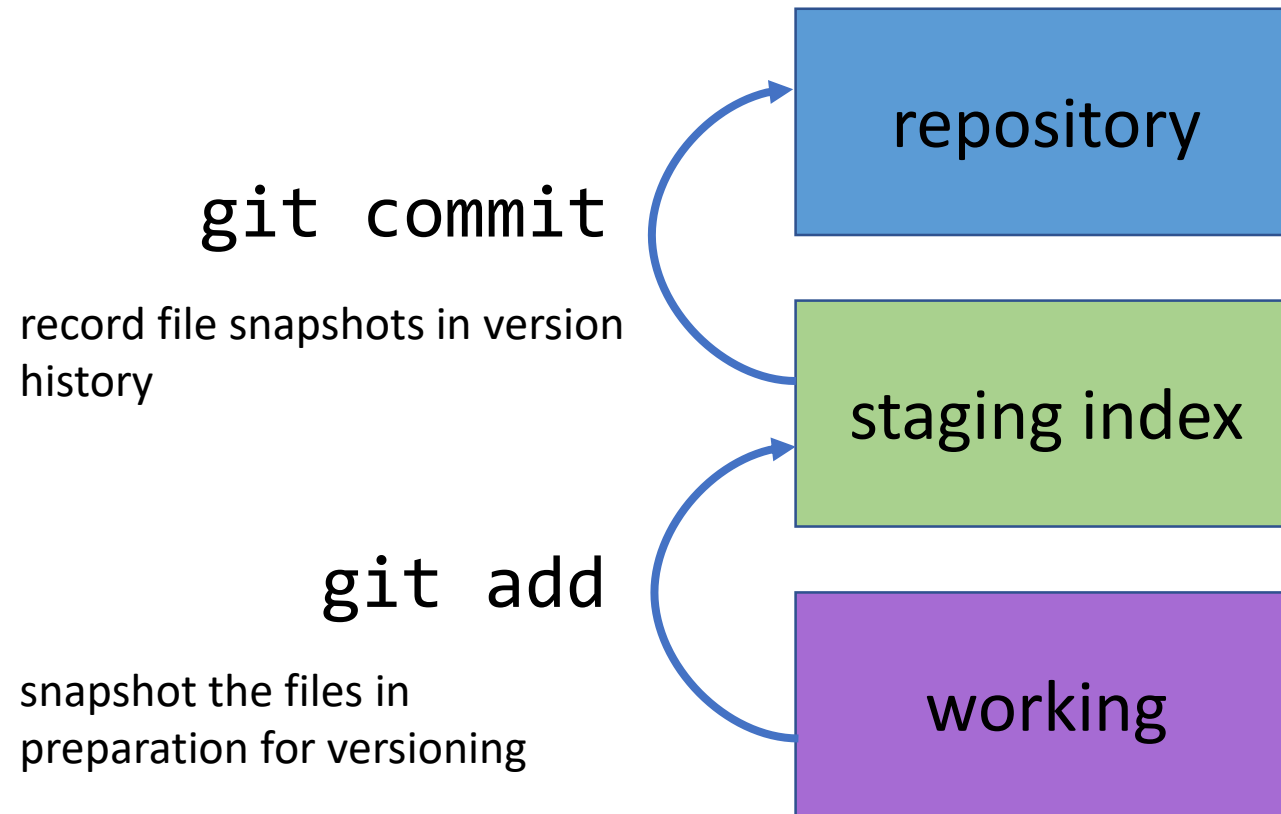
Git Concepts – three trees/areas



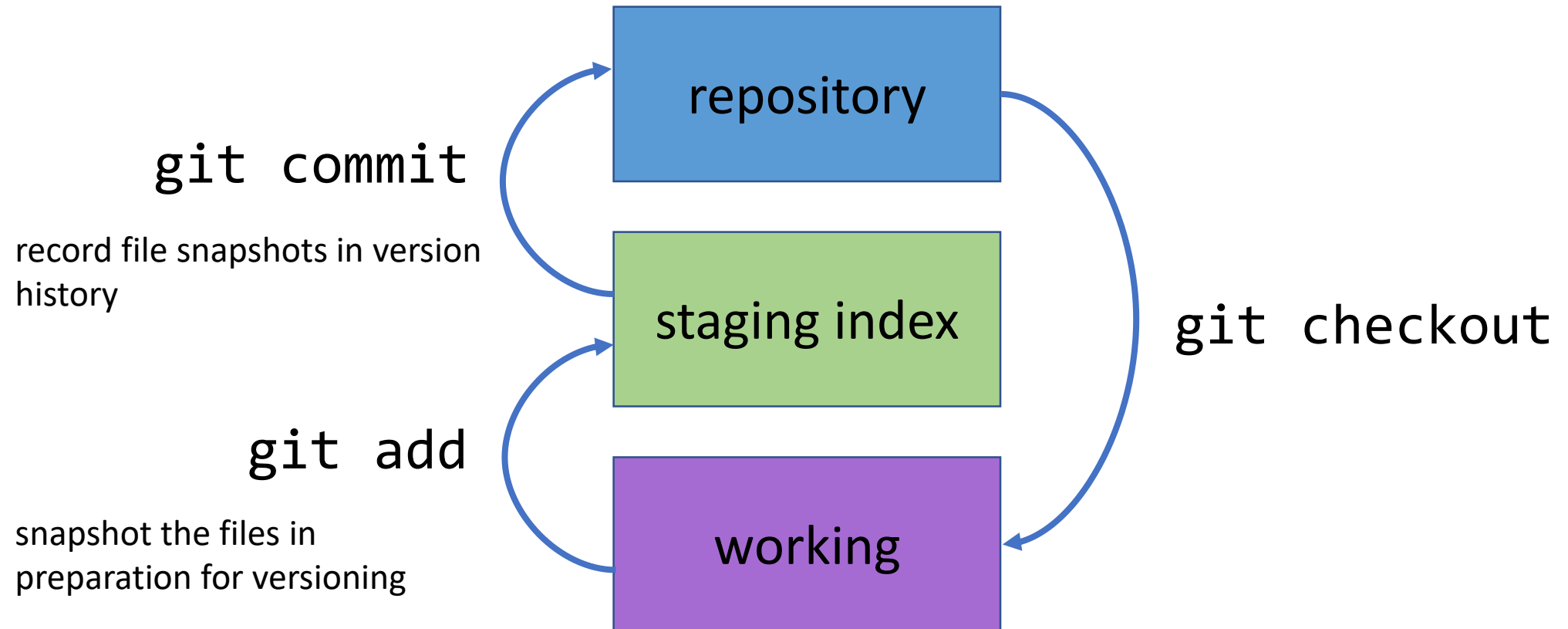
Git Concepts – three trees/areas



Git Concepts – three trees/areas



Git Concepts – three trees/areas



Hands-on?

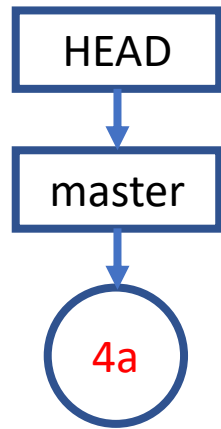
Option 1: Install Git: <https://git-scm.com/downloads>

Option2: Use this in-browser [Linux emulator](#) for Git practice.

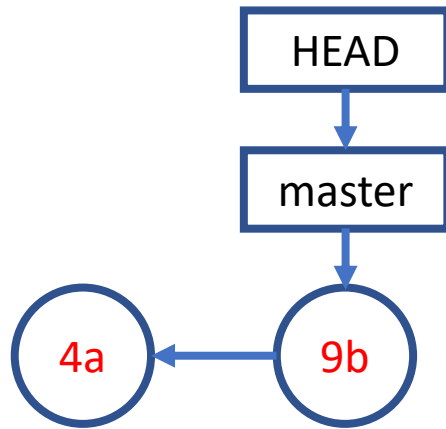
- may have problem accessing internet (i.e. when you use github)

<https://tdmdal.github.io/git-workshop/basic-git-workflow.html>

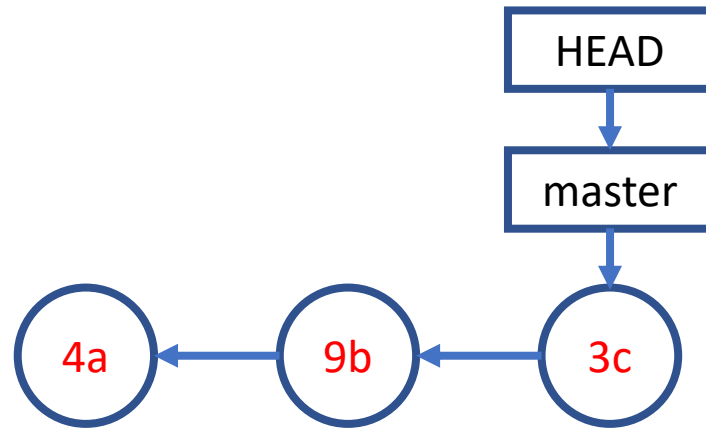
Git Concepts – First commit



Git Concepts – Second commit



Git Concepts – Third commit and so on...



Remove and Rename Files (FYI)

- Remove files (**demo**)

```
git rm <file>
```

- Rename files

```
git mv <file_old> <file_new>
```

- After removing or rename files

```
git commit -m "<remove or rename msg>"
```


Undo (1 / FYI)

- Undo working directory changes

```
git checkout -- <file>
```

- Retrieve old version of a file (to staging index & working dir) (demo)

```
git checkout <commit-id> -- <file>
```

- Unstaging files

```
git reset HEAD <file>
```

Undo (2 / FYI)

- Amending last commit

```
git commit -amend -m "commit message"
```

- Reverting a commit (by adding a new commit to undo last commit)

```
git revert <commit-id>
```

- Undo multiple commits

```
git reset [--soft|--mixed|--hard] <commit-id>
```

Suppress Tracking: .gitignore file

- Don't track certain files for a project
 - e.g. my_project/.gitignore

```
my_project/.gitignore
```

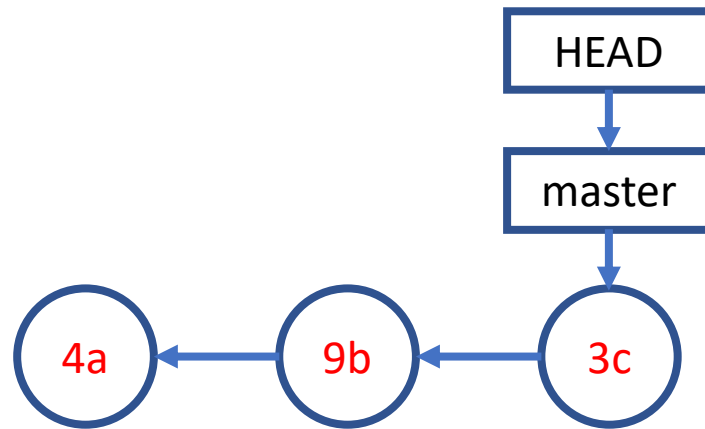
```
*.log
```

```
log/
```

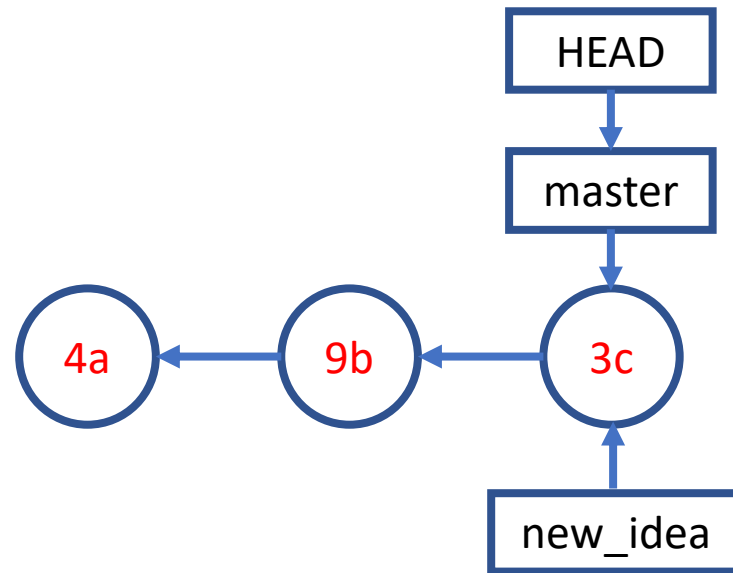
```
data/
```

```
!data/sample.csv
```

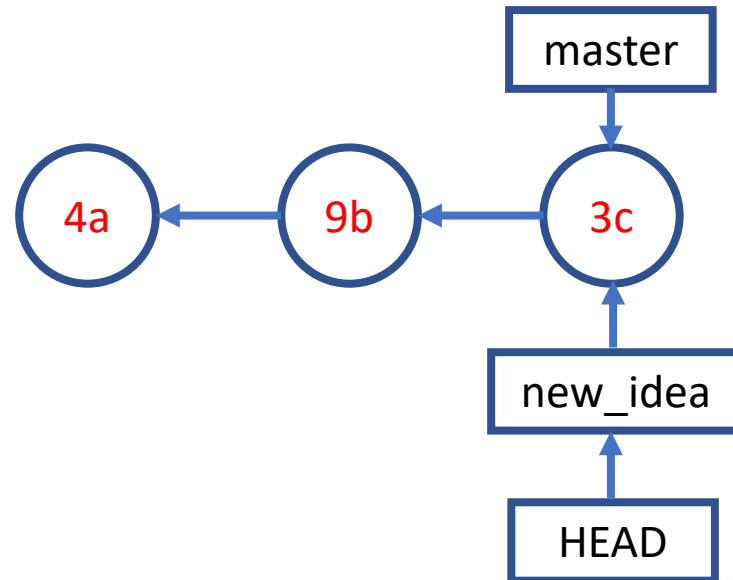
Branching (another common git workflow)



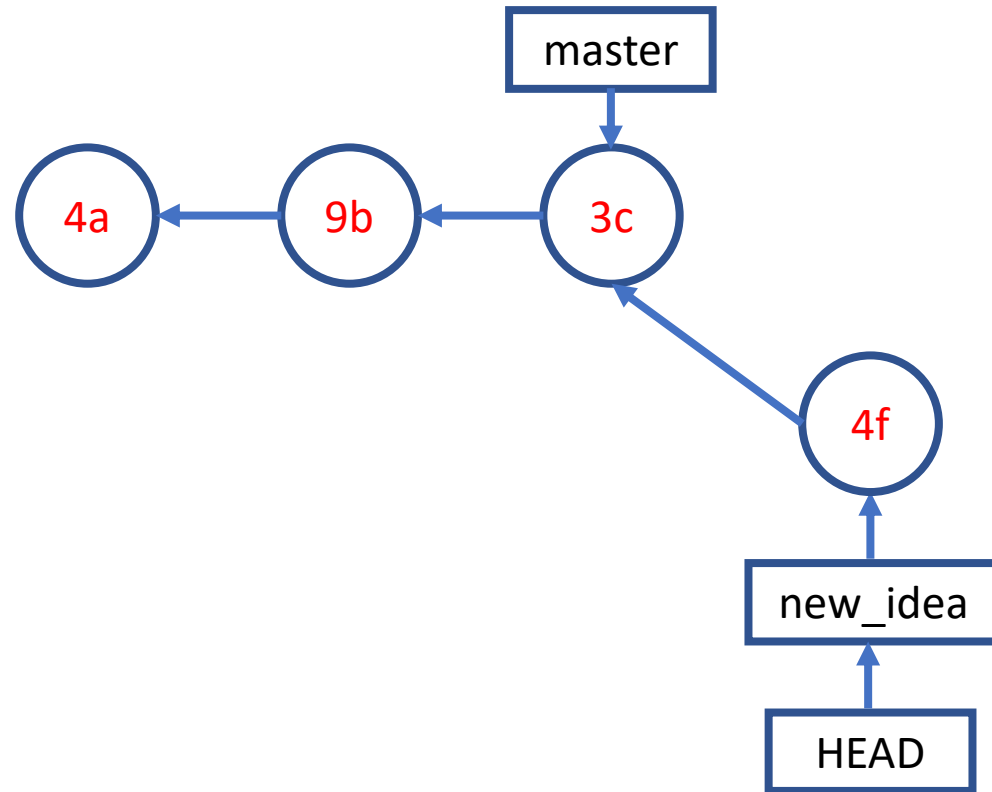
Branch `git branch new_idea`



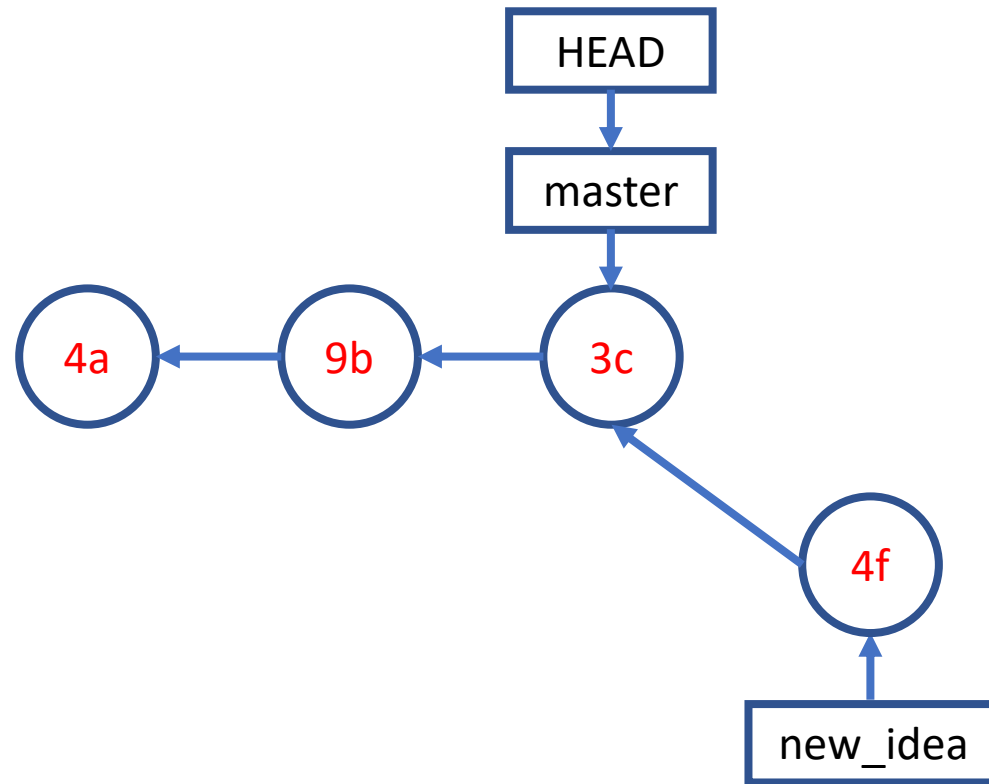
Branch `git checkout new_idea`



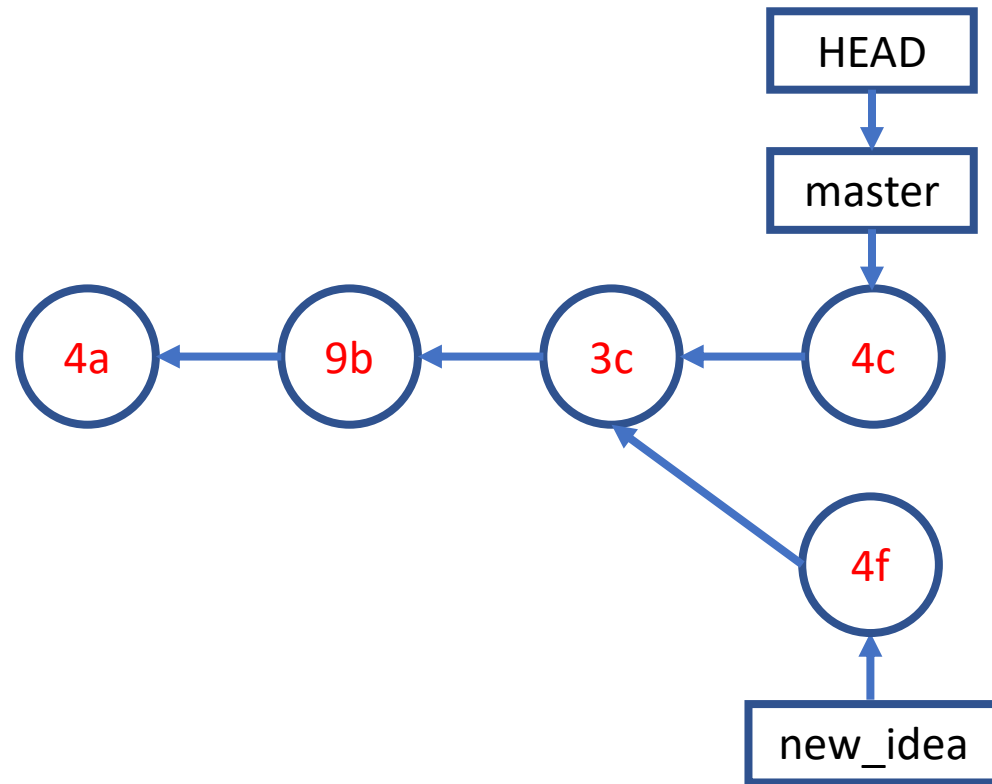
Branch `git add; git commit;`



Branch `git checkout master`

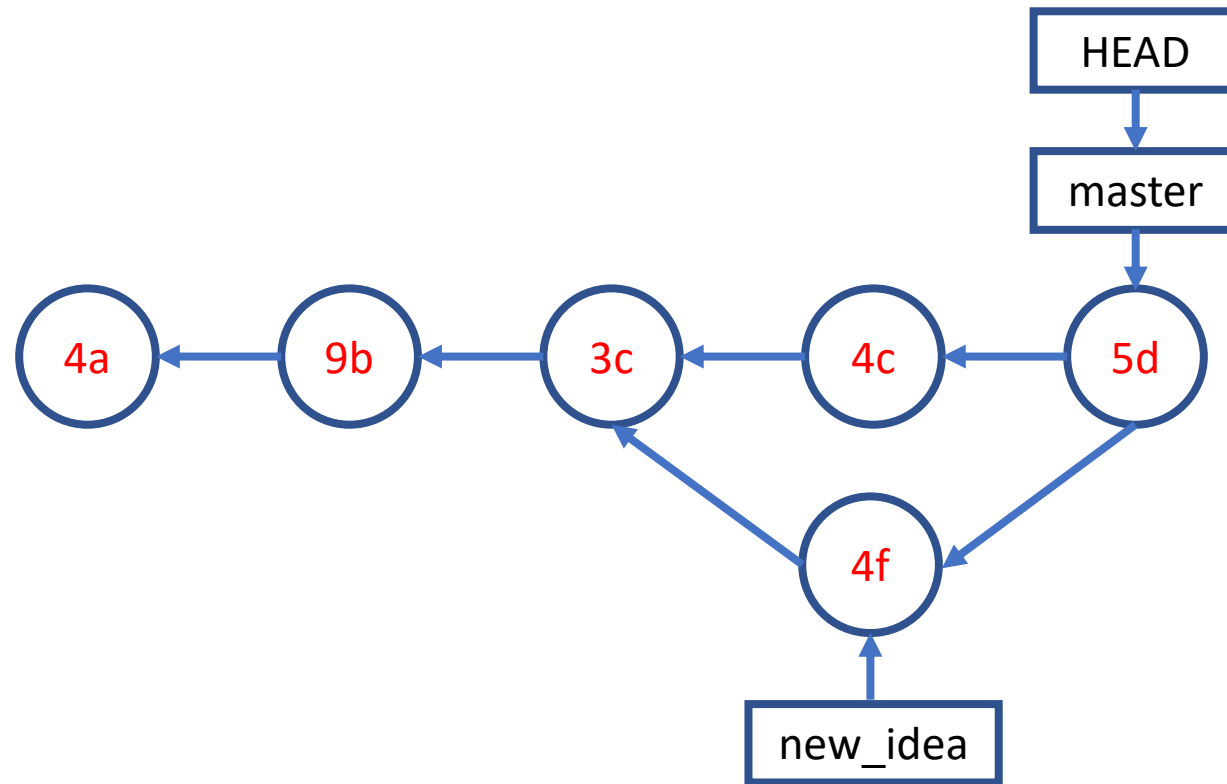


Branch `git add; git commit;`



Merge

```
git merge new_idea
```



Work with GitHub (demo)

- GitHub Account

- Create a GitHub project repo & push your code there
 - backup
 - collaborate with your co-authors
 - collaborate with open source community

```
git remote add  
git push
```

- Use a public repo as your project starting point

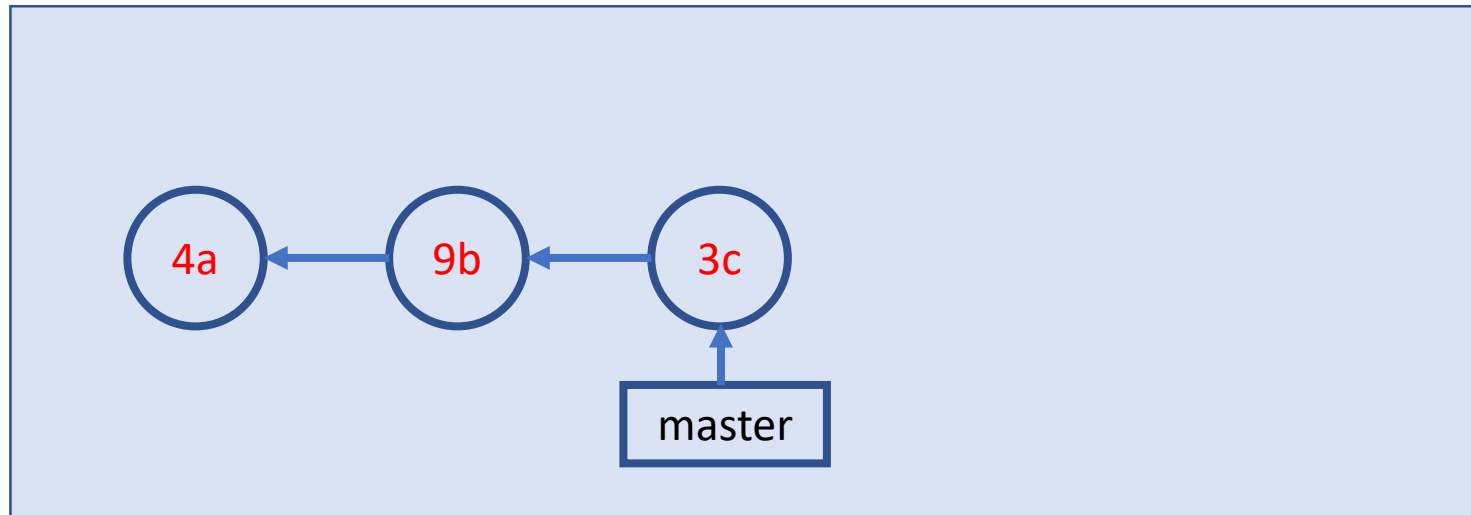
```
fork & git clone
```

A Simple Remote Repo Workflow

Remote Repo

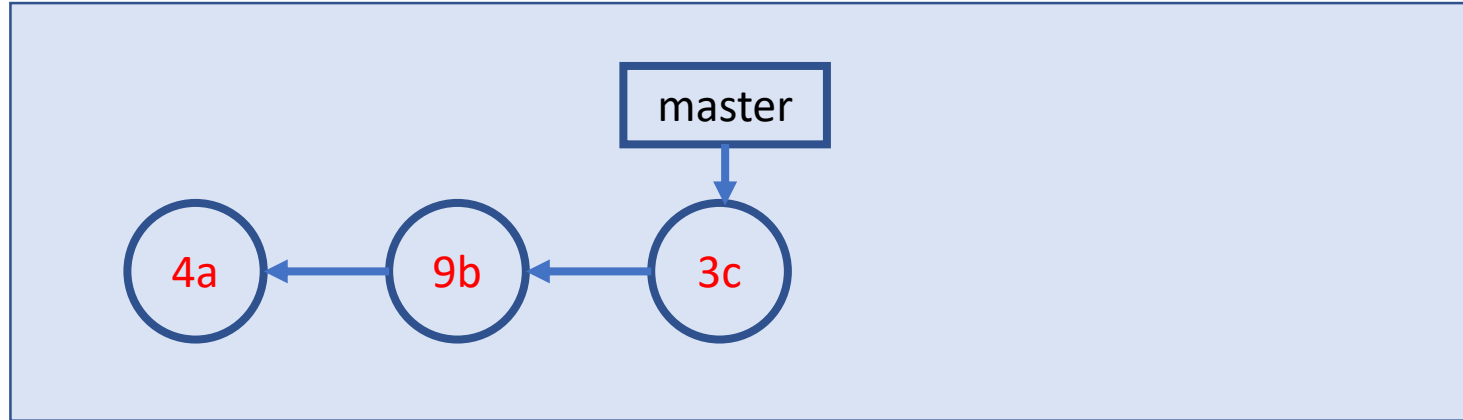


Local Repo

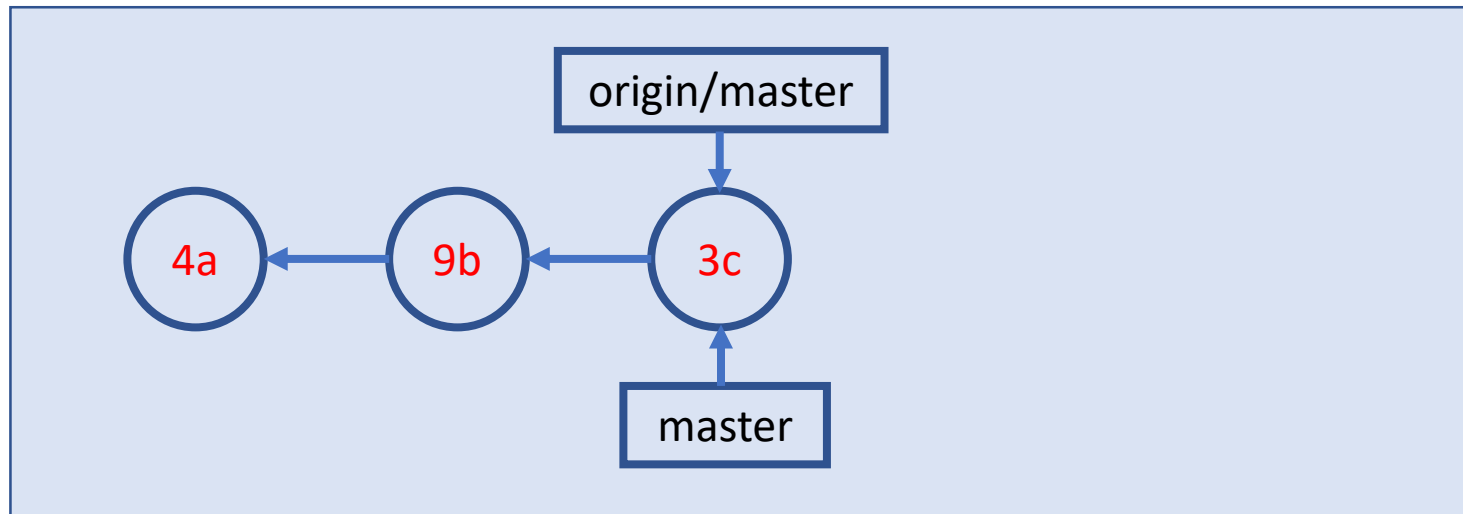


A Simple Remote Repo Workflow `git push`

Remote Repo

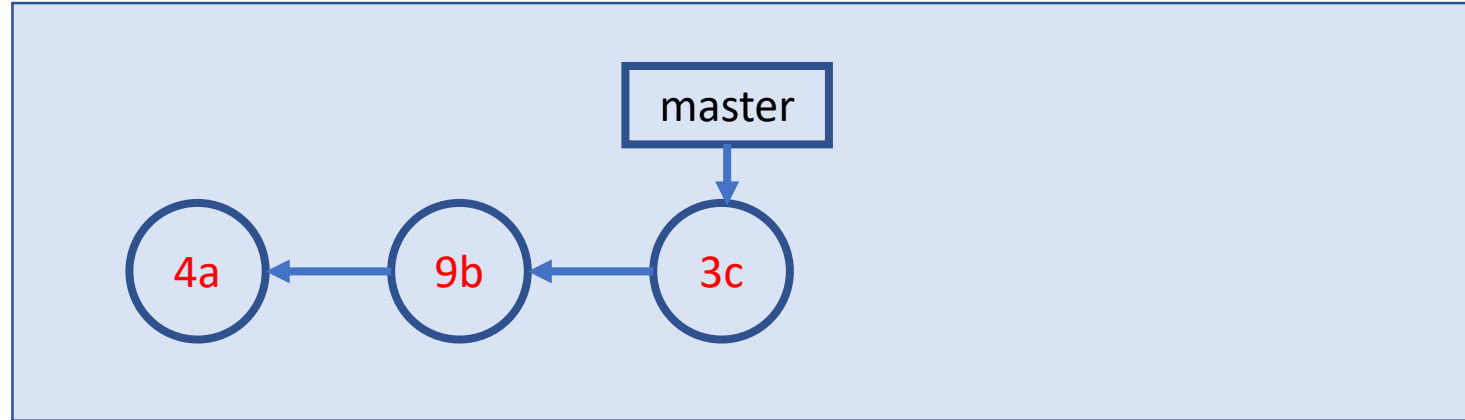


Local Repo

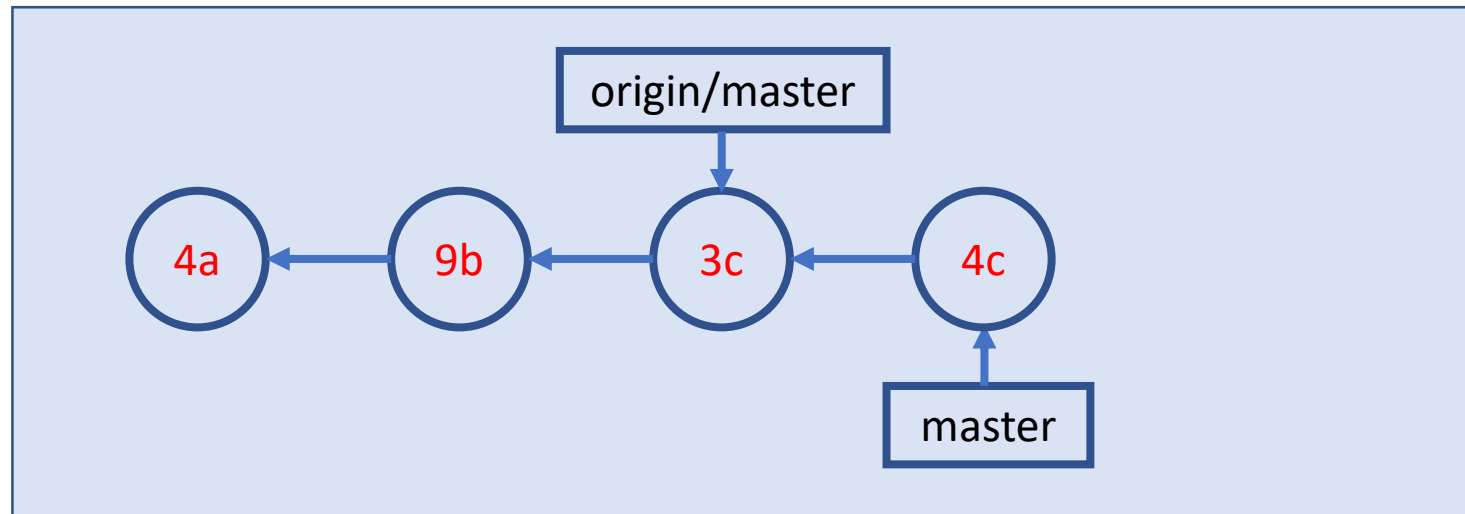


A Simple Remote Repo Workflow

Remote Repo

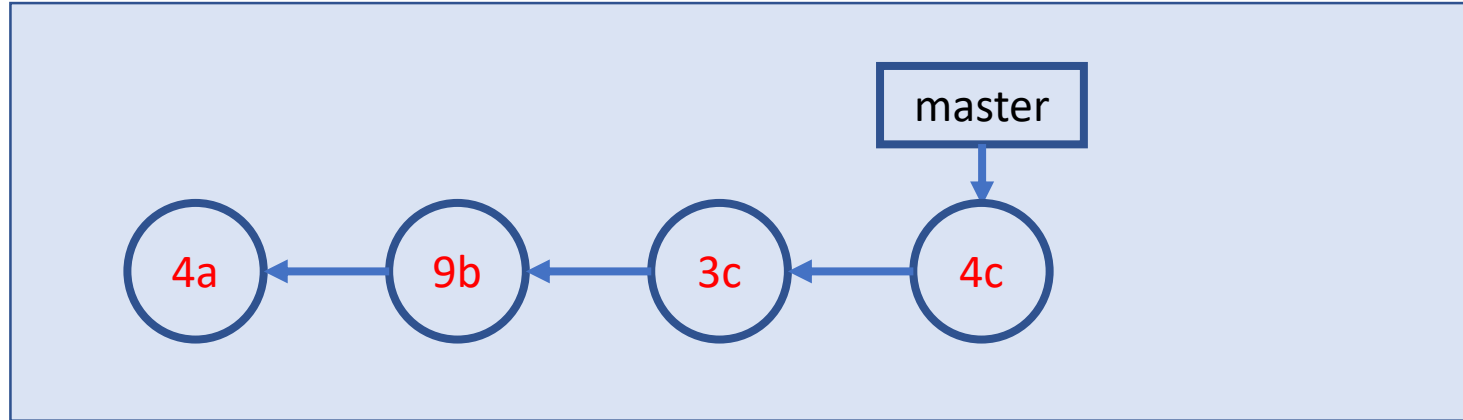


Local Repo

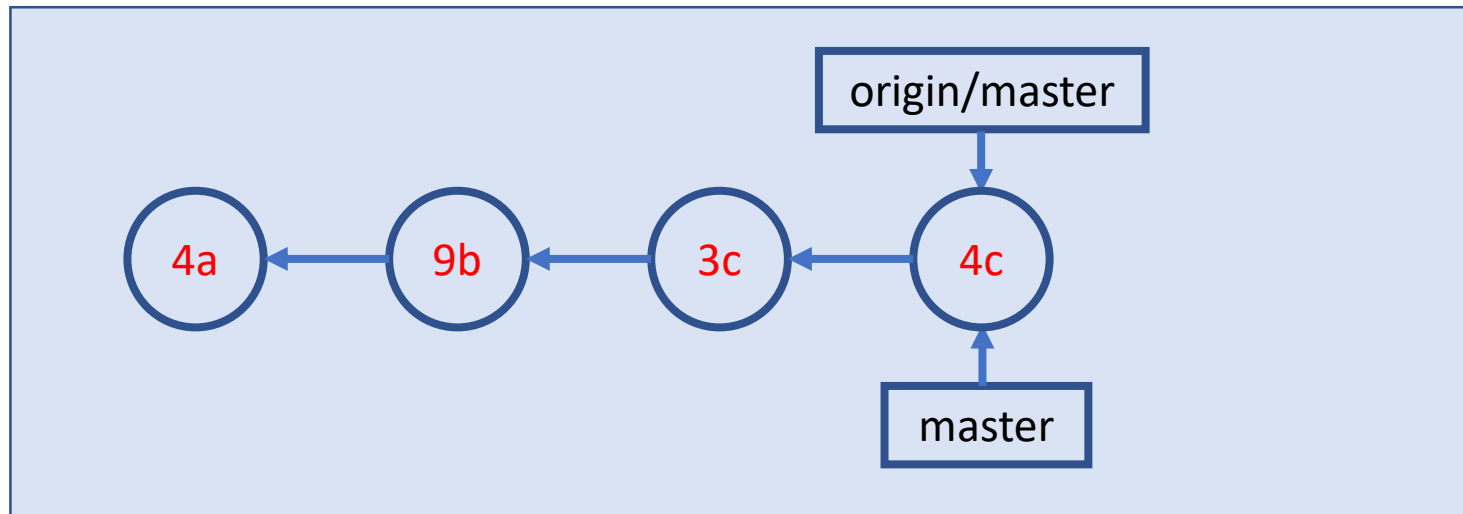


A Simple Remote Repo Workflow `git push`

Remote Repo

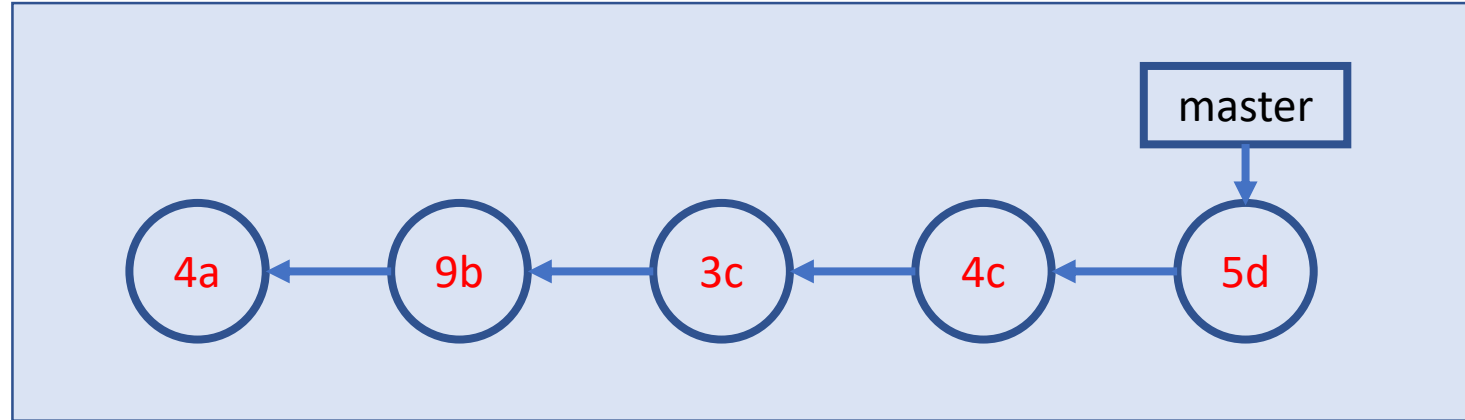


Local Repo

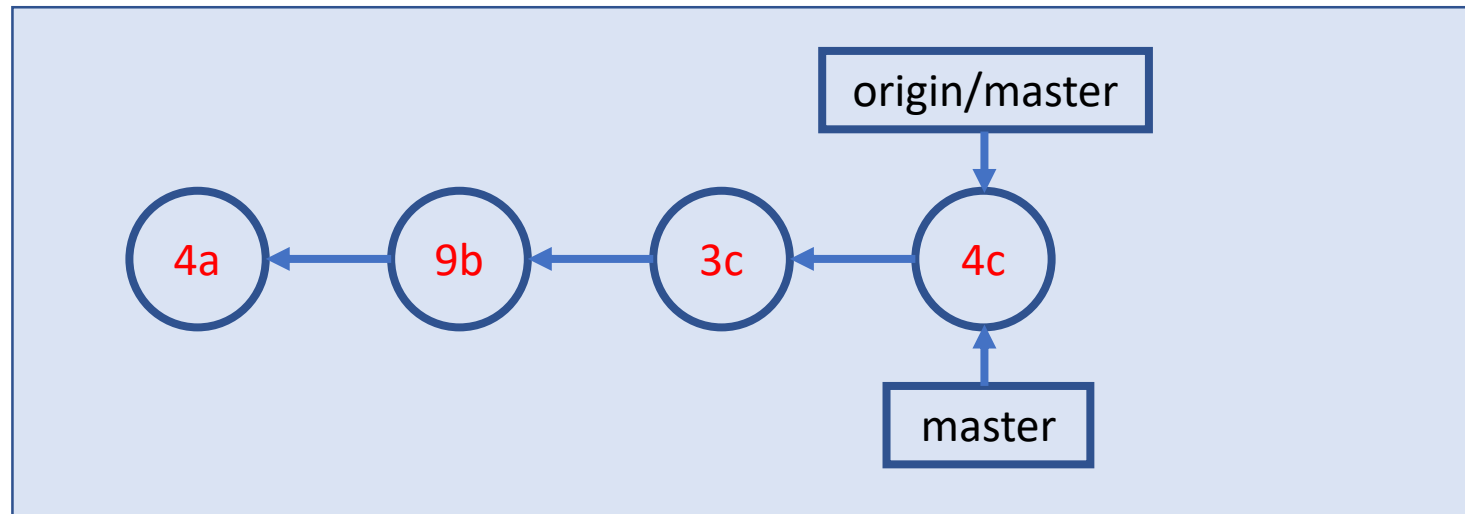


A Simple Remote Repo Workflow

Remote Repo

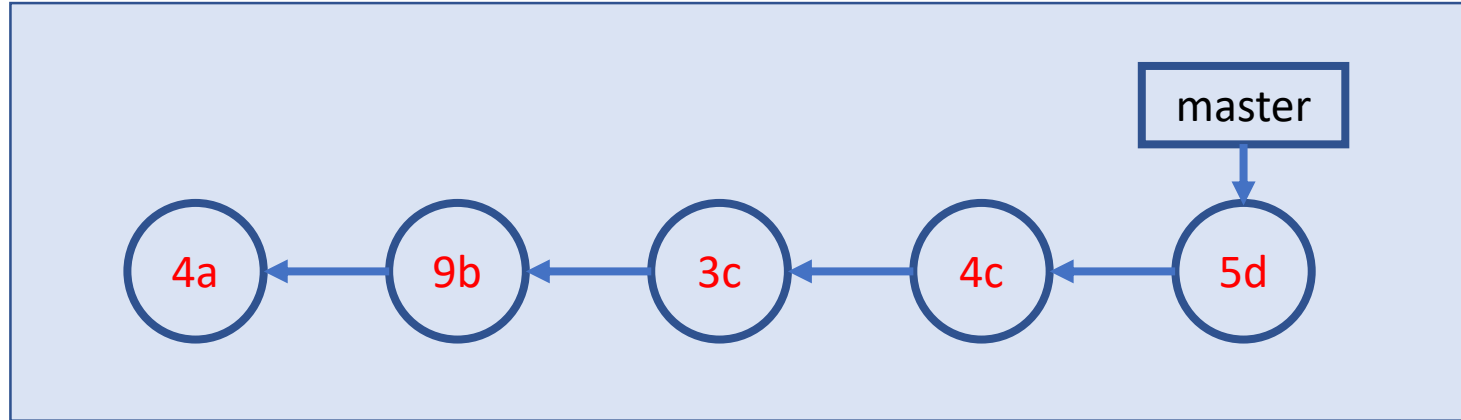


Local Repo

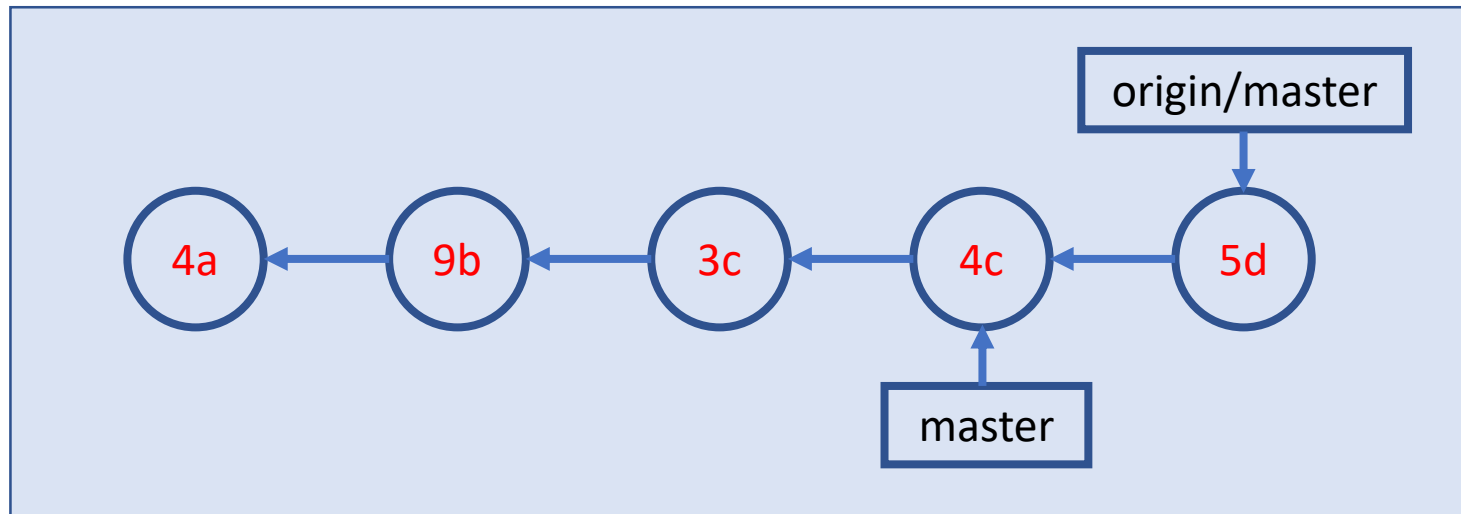


A Simple Remote Repo Workflow `git fetch`

Remote Repo

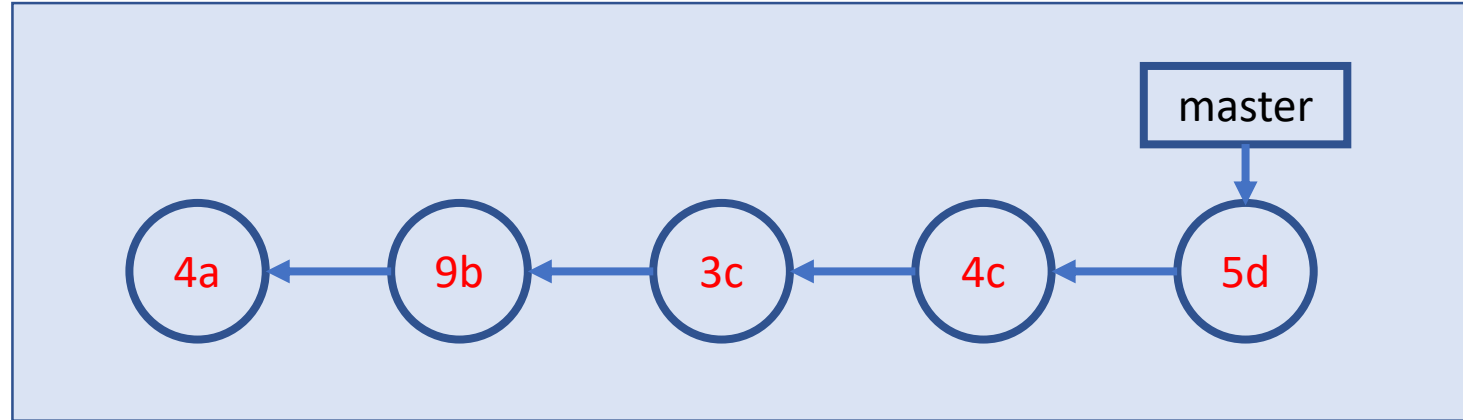


Local Repo

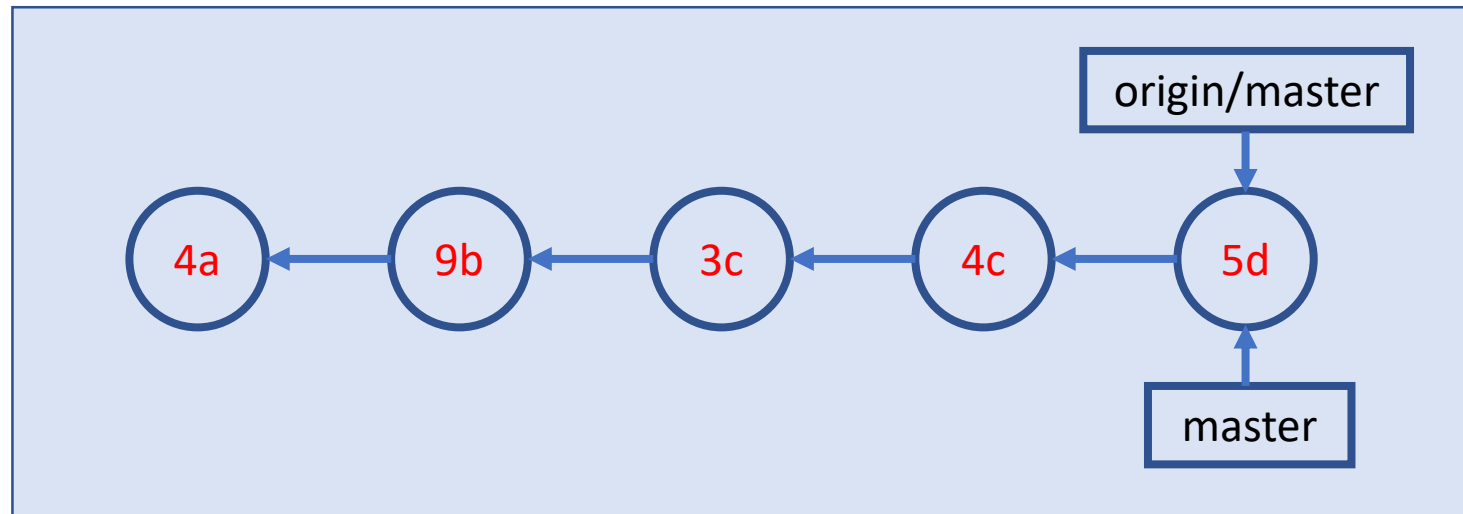


A Simple Remote Repo Workflow `git merge`

Remote Repo



Local Repo



Many more to explore on your own

- Git concept / command
 - merge conflict
 - remote branch
 - git reset
 - git stash, rebase, bisect
 - ...
- Git best practice
 - workflows
 - commit size / message
 - ...

Resources

- Git Ref Book: <https://git-scm.com/book/en/v2>
- Git Tutorials
 - [Version Control with Git](#) by Software Carpentry
 - [Git Essential Training](#) by Kevin Skoglund at lynda.com
 - login from [here](#) for UofT free access
 - [Get Started Tutorials](#) from Bitbucket Atlassian
 - [GitHub Guides](#)
- Git GUI (I recommend starting with command line)
 - dedicated GUI client: <https://git-scm.com/downloads/guis>
 - GUI integrated with IDE or code editor (e.g. RStudio, vscode, etc.)