Intro to version control

Version control with git

Gineke ten Holt

Target Holding

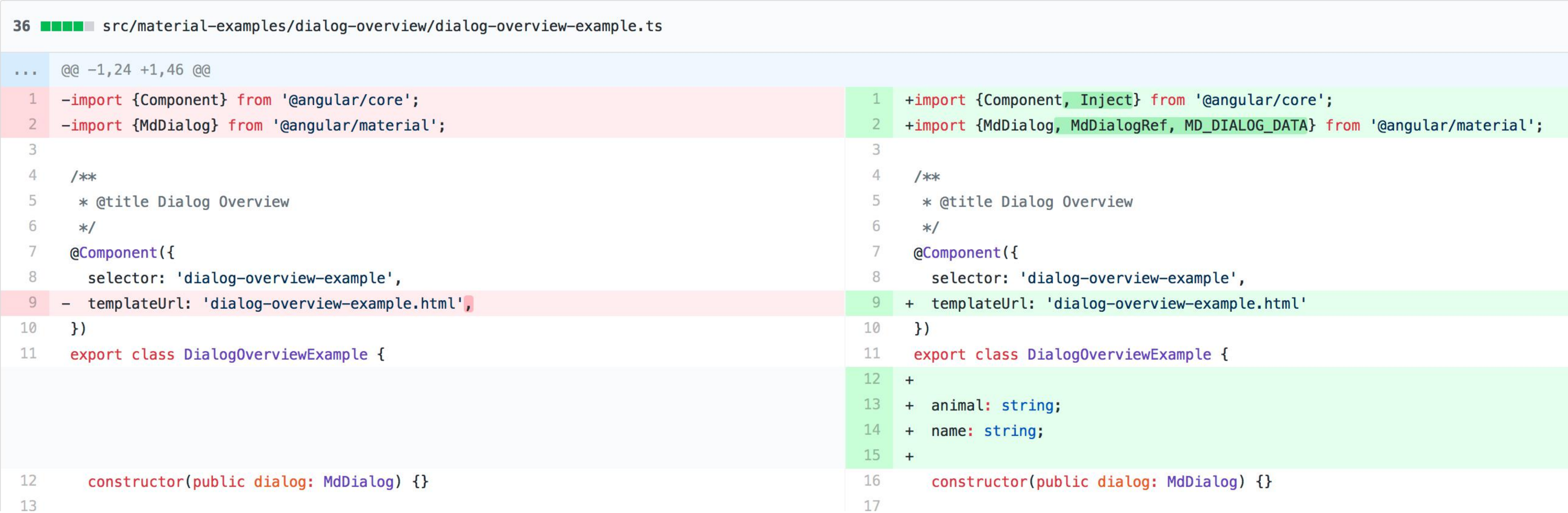
Overview

- Why version control?
- The diff (Unix, 1974)
- Snapshots
- Commit history
- Branches

Why version control?

You use source control?
No, I am an advocate of free software. Source control and free source are incompatible.
Umm...'free source' and 'source control' are not the same concept.
Yeah, of course not.
But it's easier to tell the boss than "I'm too damn lazy to learn to use Git".
Git indeed.
© 2017 The Code Zone
Visit angriestprogrammer.com for your dose of cynicism

The Diff



```
36 🟩🟩🟩🟩⬜ src/material-examples/dialog-overview/dialog-overview-example.ts
... @@ -1,24 +1,46 @@
```

```
-import {Component} from '@angular/core';
-import {MdDialog} from '@angular/material';

 /**
  * @title Dialog Overview
  */
 @Component({
   selector: 'dialog-overview-example',
-  templateUrl: 'dialog-overview-example.html',
 })
 export class DialogOverviewExample {
   constructor(public dialog: MdDialog) {}

```

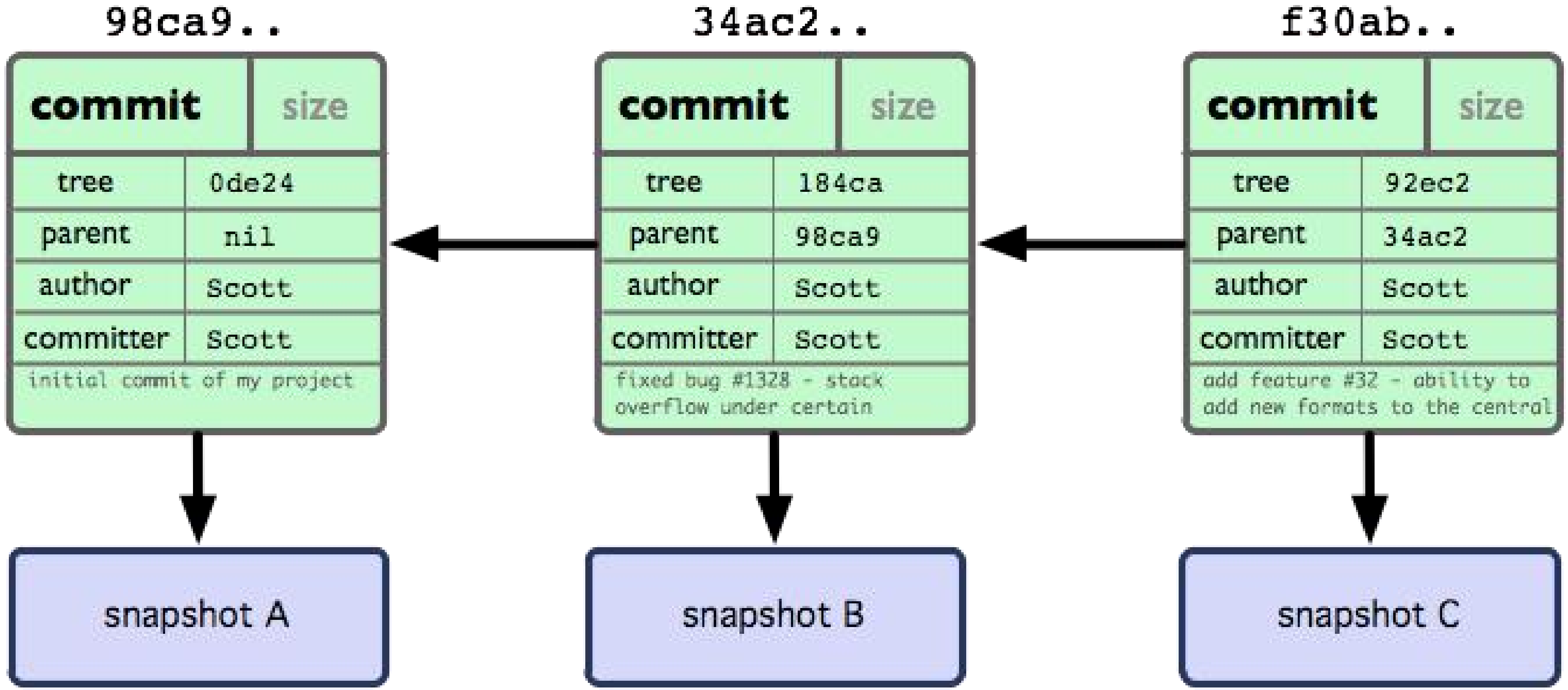
```
+import {Component, Inject} from '@angular/core';
+import {MdDialog, MdDialogRef, MD_DIALOG_DATA} from '@angular/material';

 /**
  * @title Dialog Overview
  */
 @Component({
   selector: 'dialog-overview-example',
+  templateUrl: 'dialog-overview-example.html'
 })
 export class DialogOverviewExample {
+
+  animal: string;
+  name: string;
+
   constructor(public dialog: MdDialog) {}

```

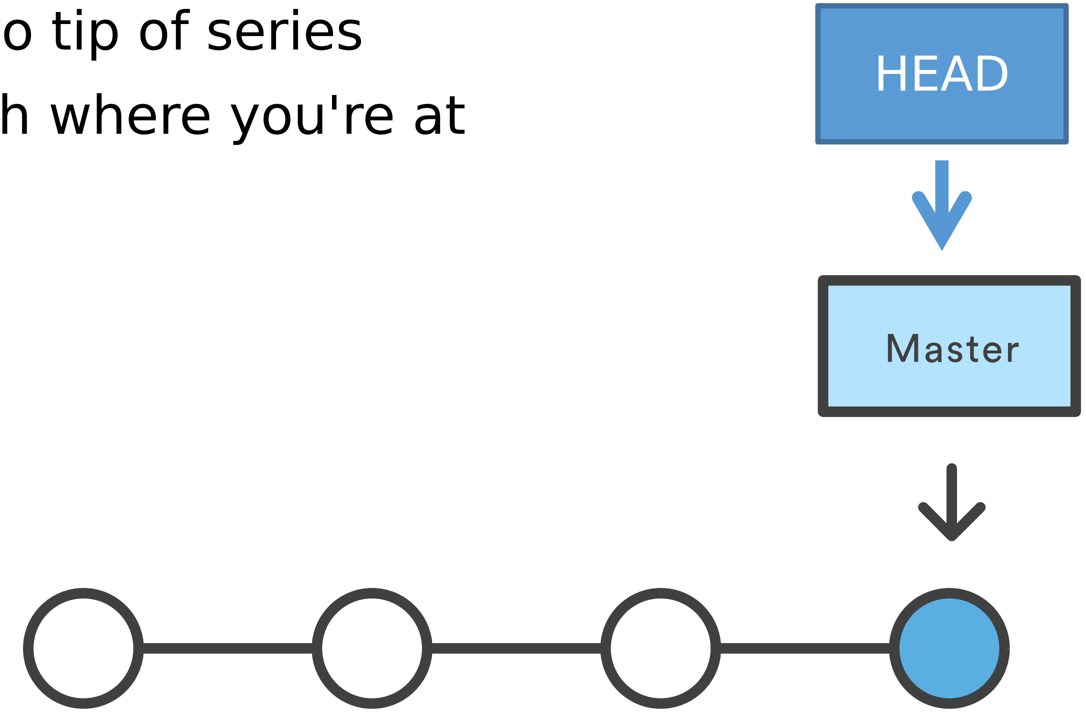
Snapshots

- The state your code was in at the moment of commit
- Repacking saves space



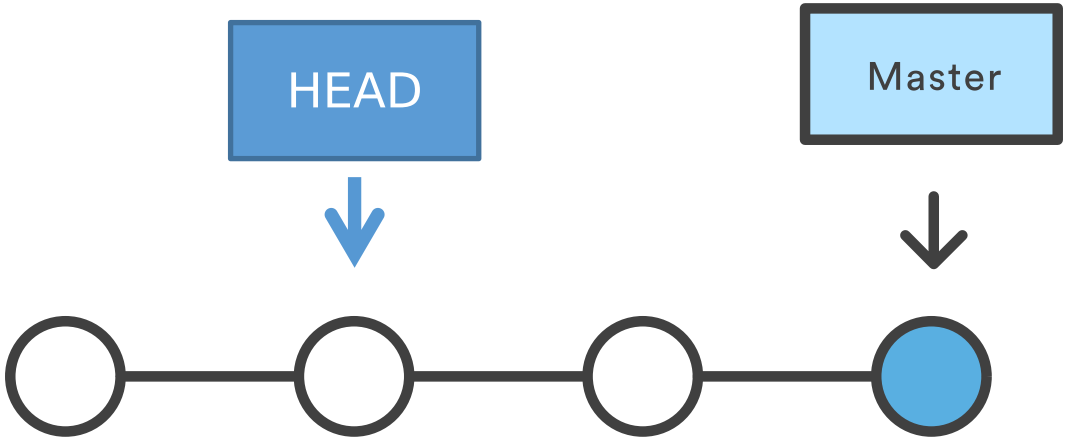
Commit history

- Series of commits / snapshots
- Branch name points to tip of series
- HEAD points at branch where you're at



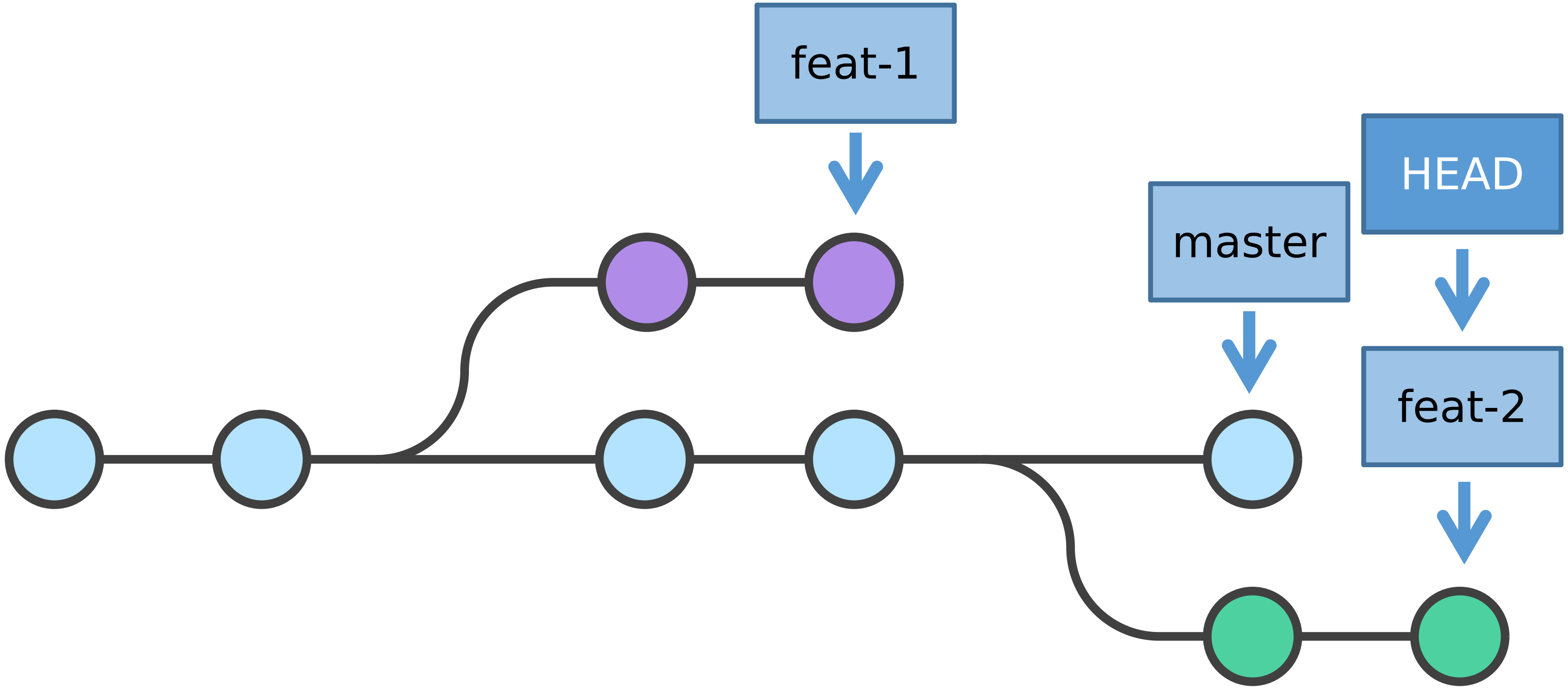
Commit history

- Easily go back to a previous state



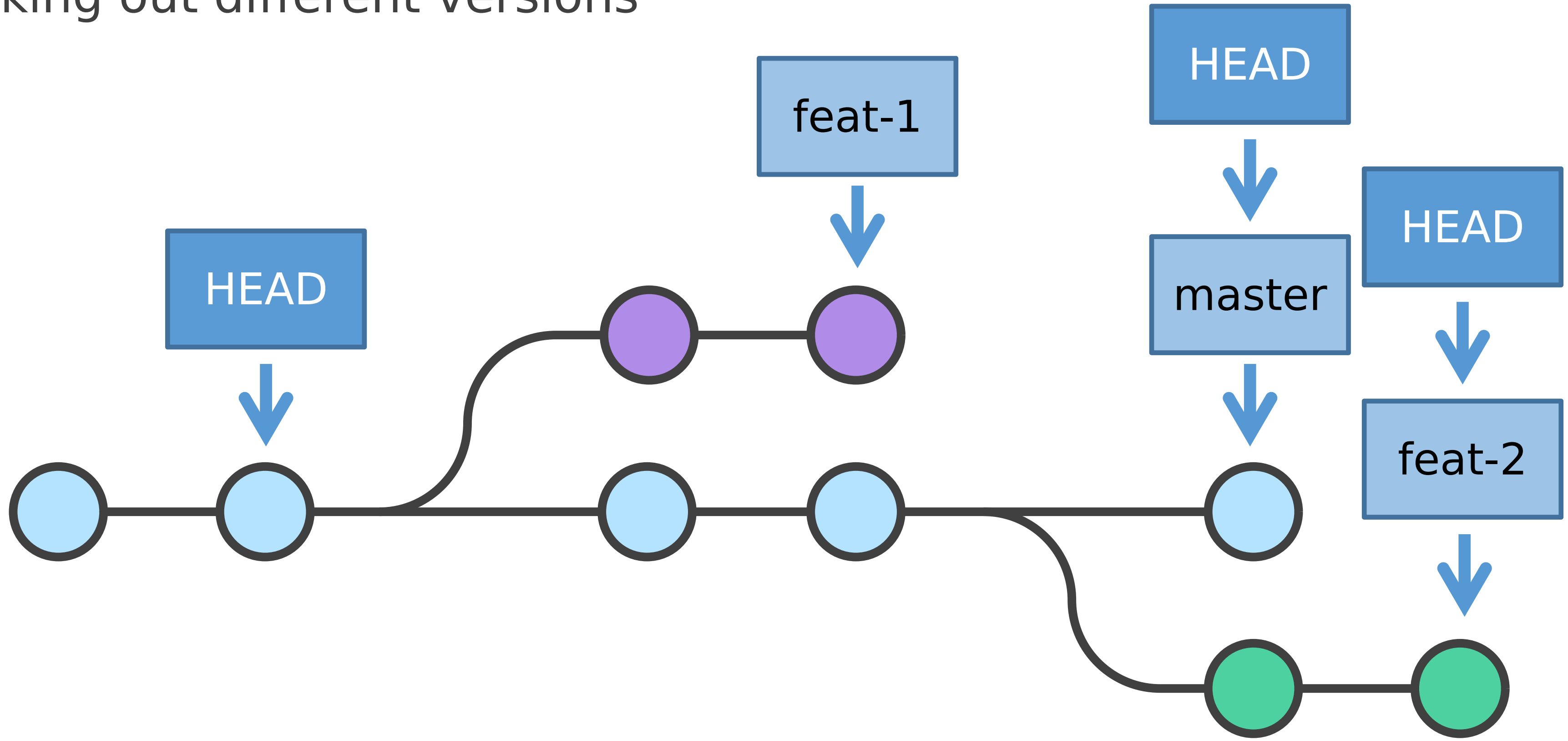
Branching

- A powerful way of keeping things organized



Commit history

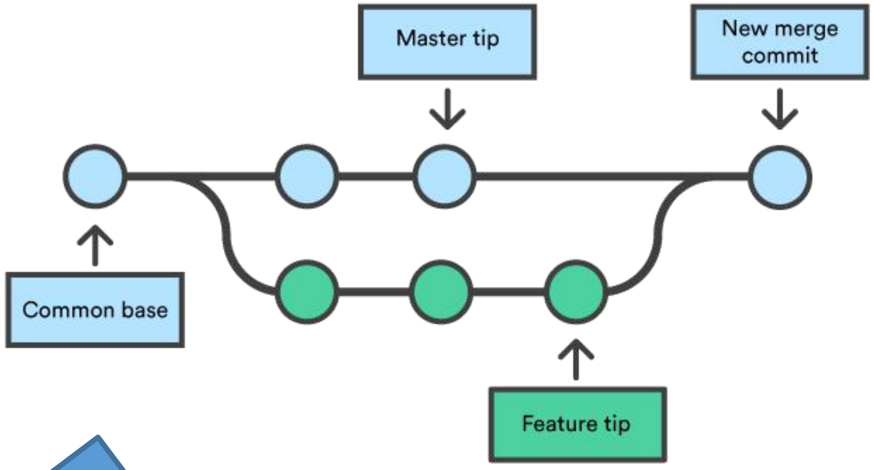
- keeping track of coherent states of your code
- checking out different versions



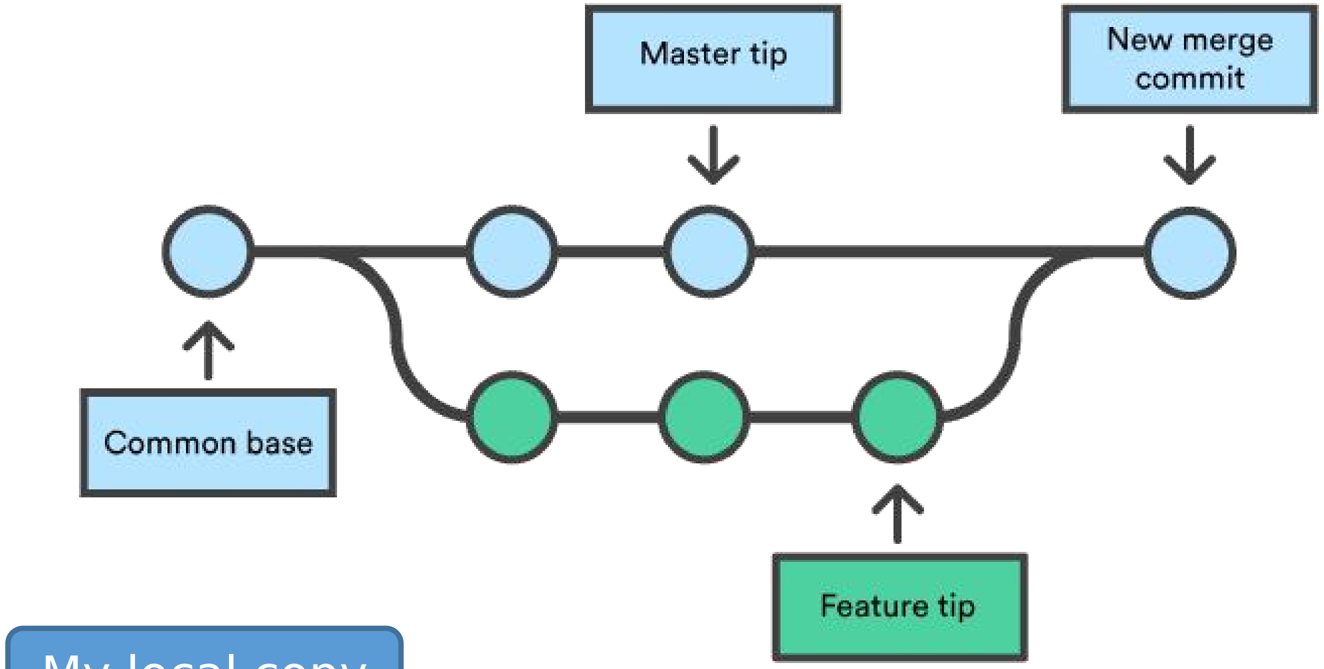
Distributed Version Control

- A Local and a Remote version of your repo
- *Clone* to get a local copy

Remote



git clone

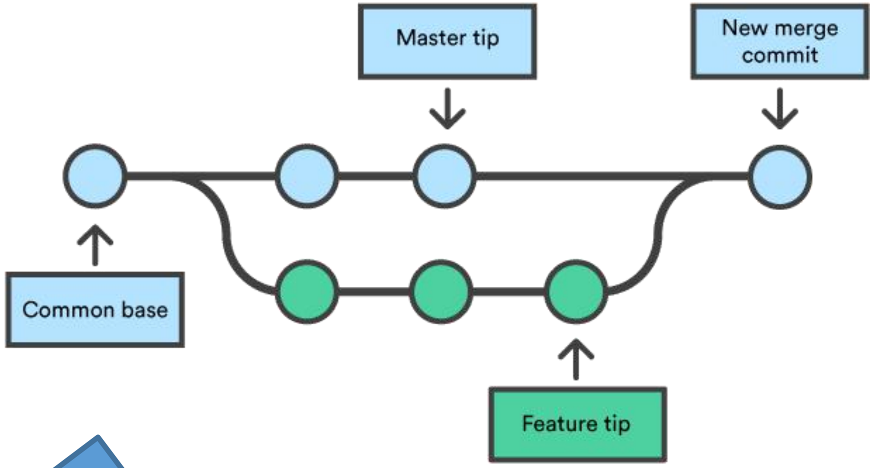


My local copy

Remote

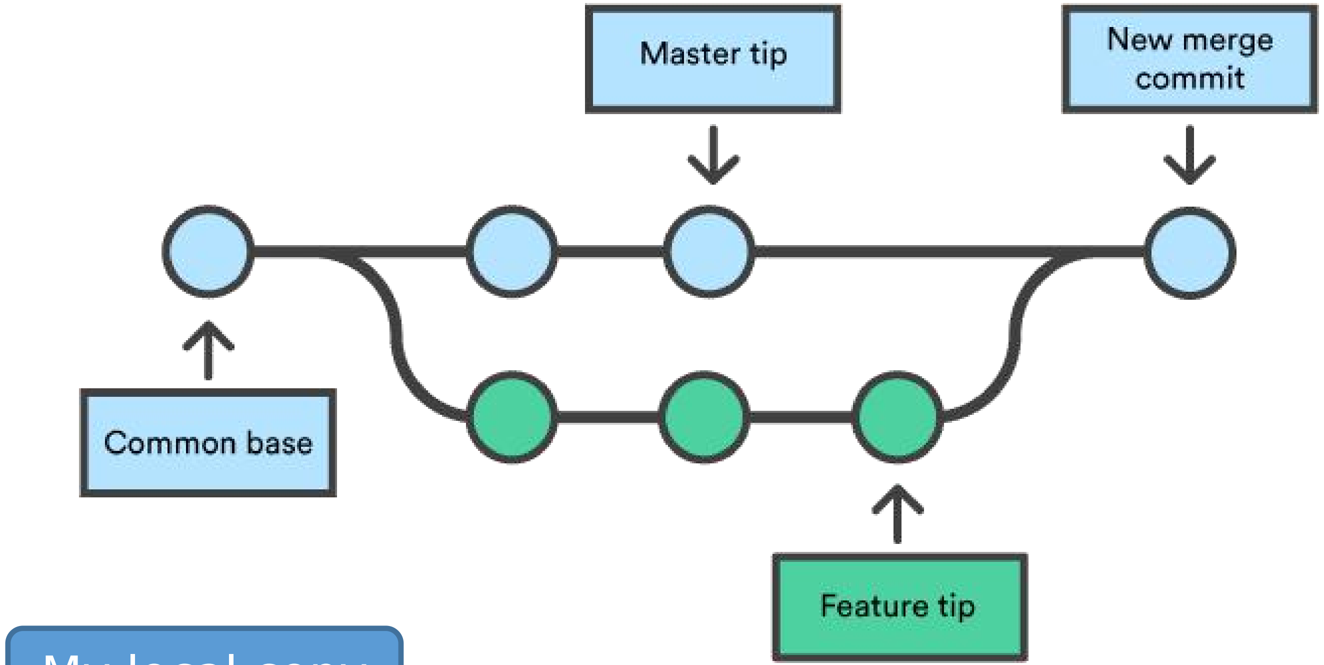
Distributed Version Control

- *Push* and *pull* to share and receive commits



git pull

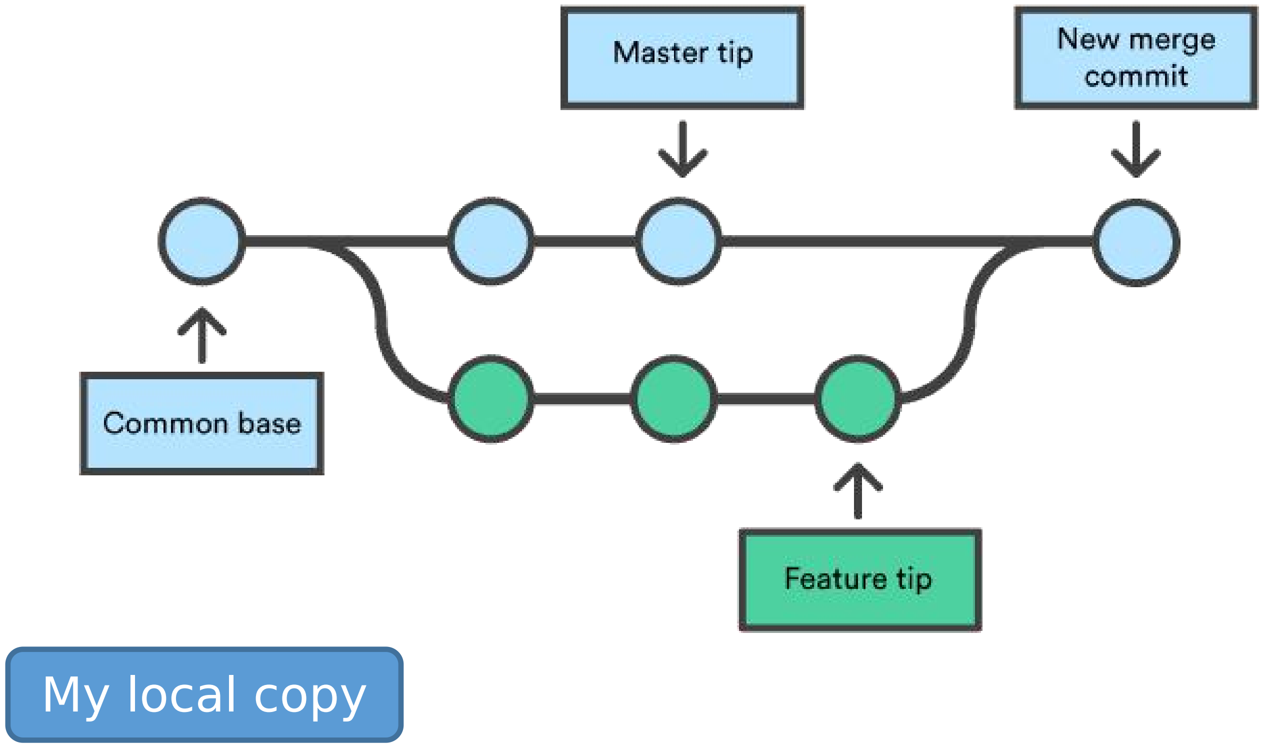
git push

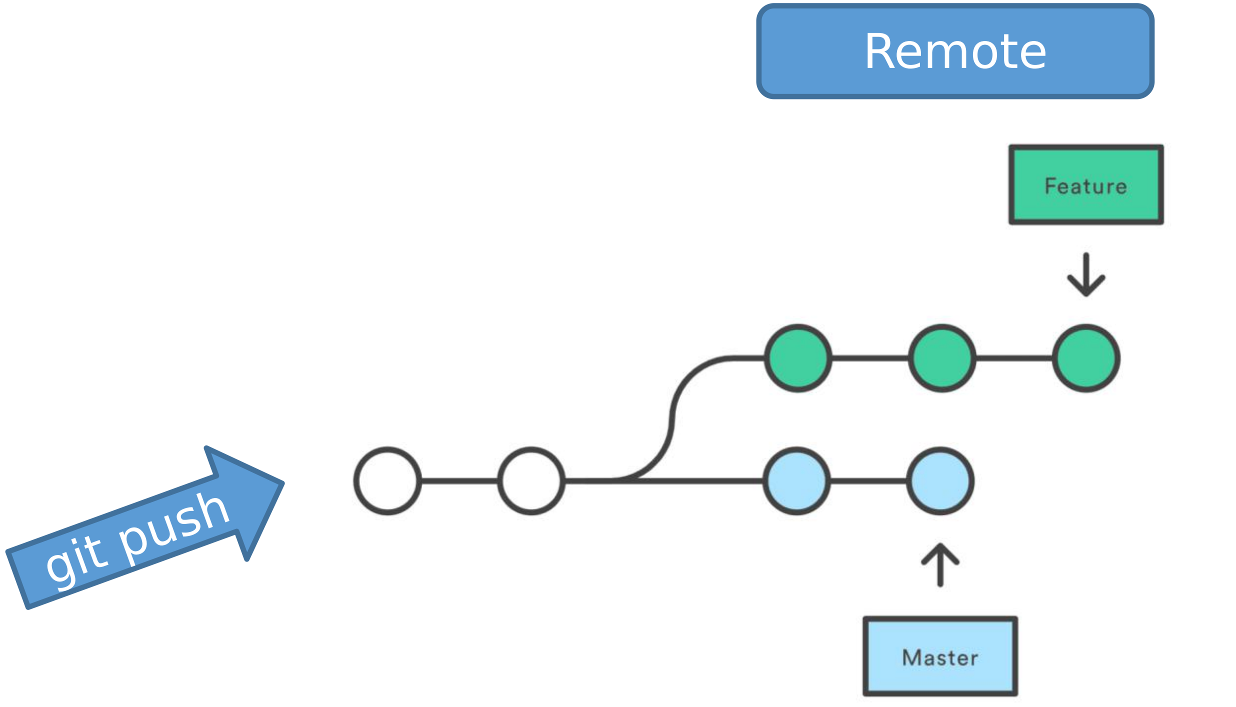


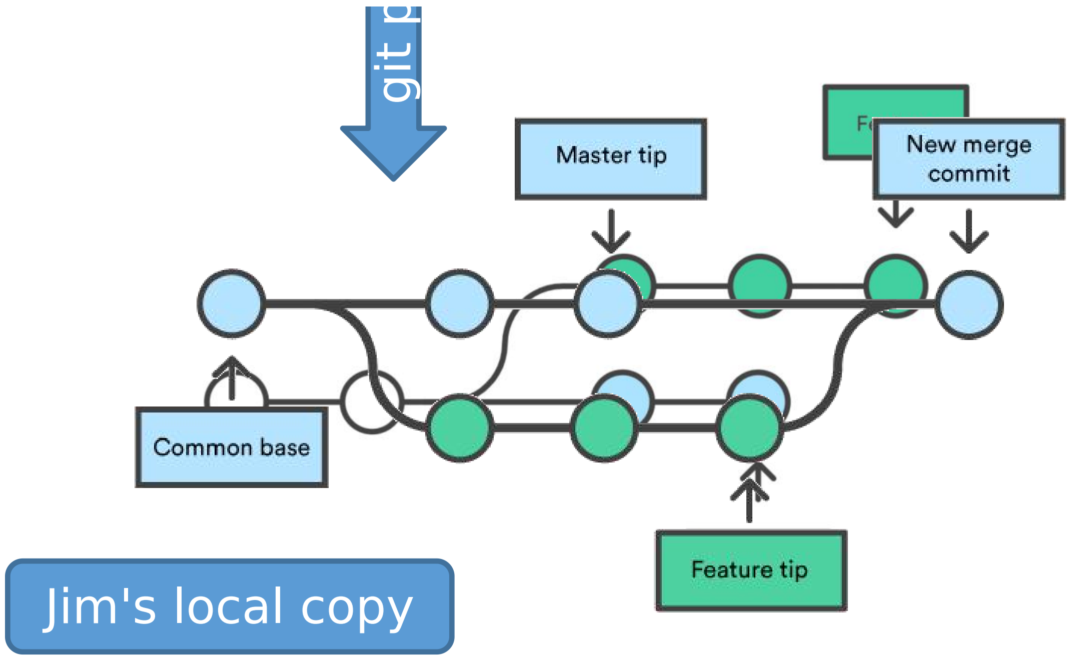
My local copy

Distributed Version Control

- Collaborate easily





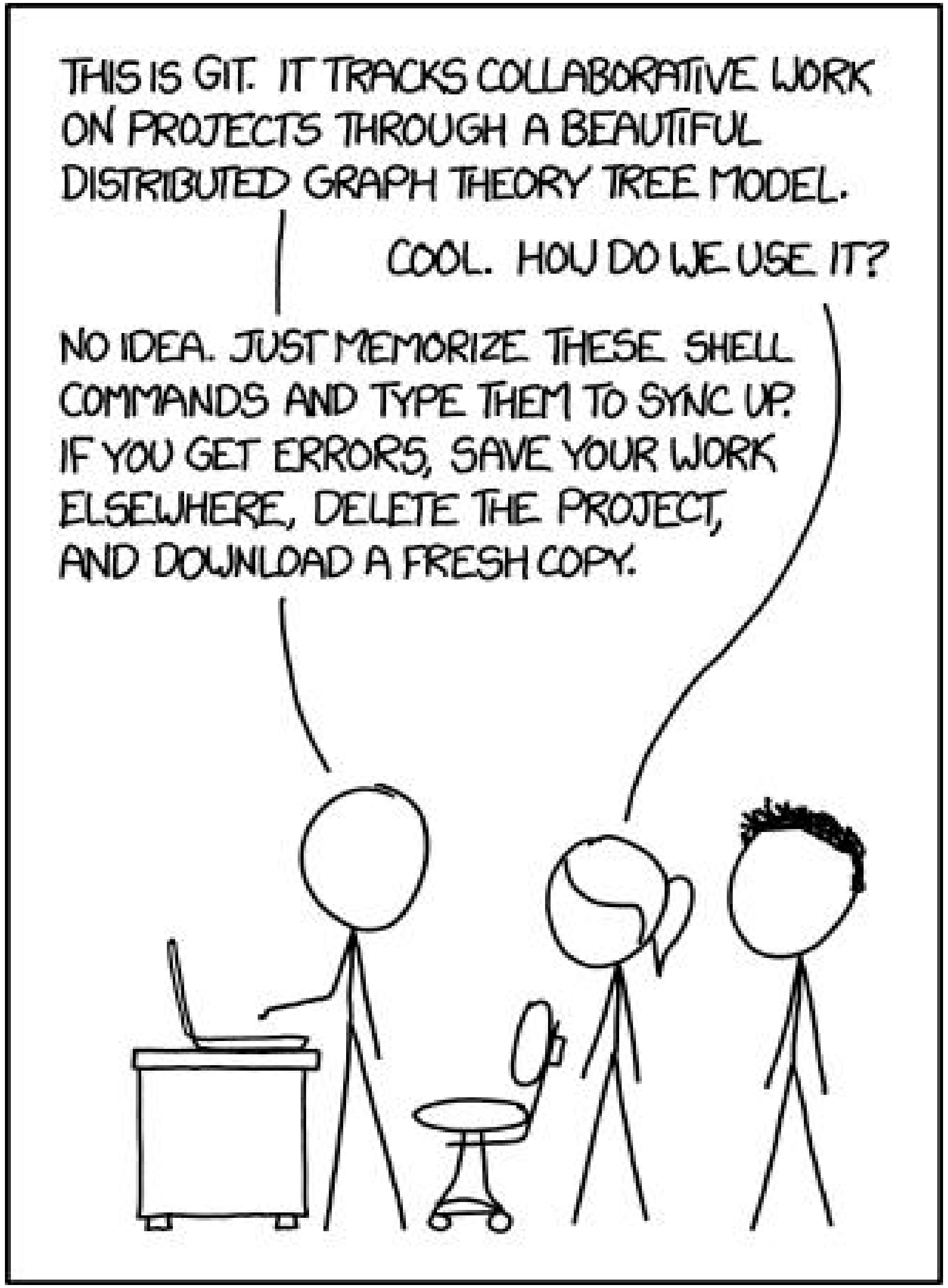


Resources & Examples

- Git tutorial : *https://git-scm.com/docs/gittutorial*
- Pro Git book : *https://git-scm.com/book/en/v2*
- The Git Parable : *http://tom.preston-werner.com/2009/05/19/the-git-parable.html*

- Try it in game form with Githug! : *https://www.tobscore.com/githug-what-is-it/ (ruby program)*

- Back to the future in 10 git actions : *http://www.mattluedke.com/back-git-history/*


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Thank you!

Want to try?

Clone the gwlc repo on github:
https://github.com/Lautian/gwlc
https://github.com/Lautian/gwlc.git

Or try the Atlassian tutorial
https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

git cheat sheet:
https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet