GIT 101

IN 5 ACTS

1. The Scene
2. The Cast
3. Call for Extras
4. Pilot Episode
5. Wrap

YO. I HEARD YOU LIKED GIT...
SO I CREATED A GIT TO HELP YOU MANAGE YOUR GIT WHEN YOU USE GIT.
memegenerator.net

SO META

Use git to get this presentation

```
git clone https://github.com/royragsdale/git-101.git
```

THE SCENE

git

BY THIS GUY

LETS RTFM

```
GIT(1)                  Git Manual                 GIT(1)

NAME
       git - the stupid content tracker

DESCRIPTION
       Git is a fast, scalable, distributed revision
       control system with an unusually rich command
       set that provides both high-level operations
       and full access to internals.
```

LETS TRY THE WEBSITE

> *Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.*

HOW ABOUT THE BOOK

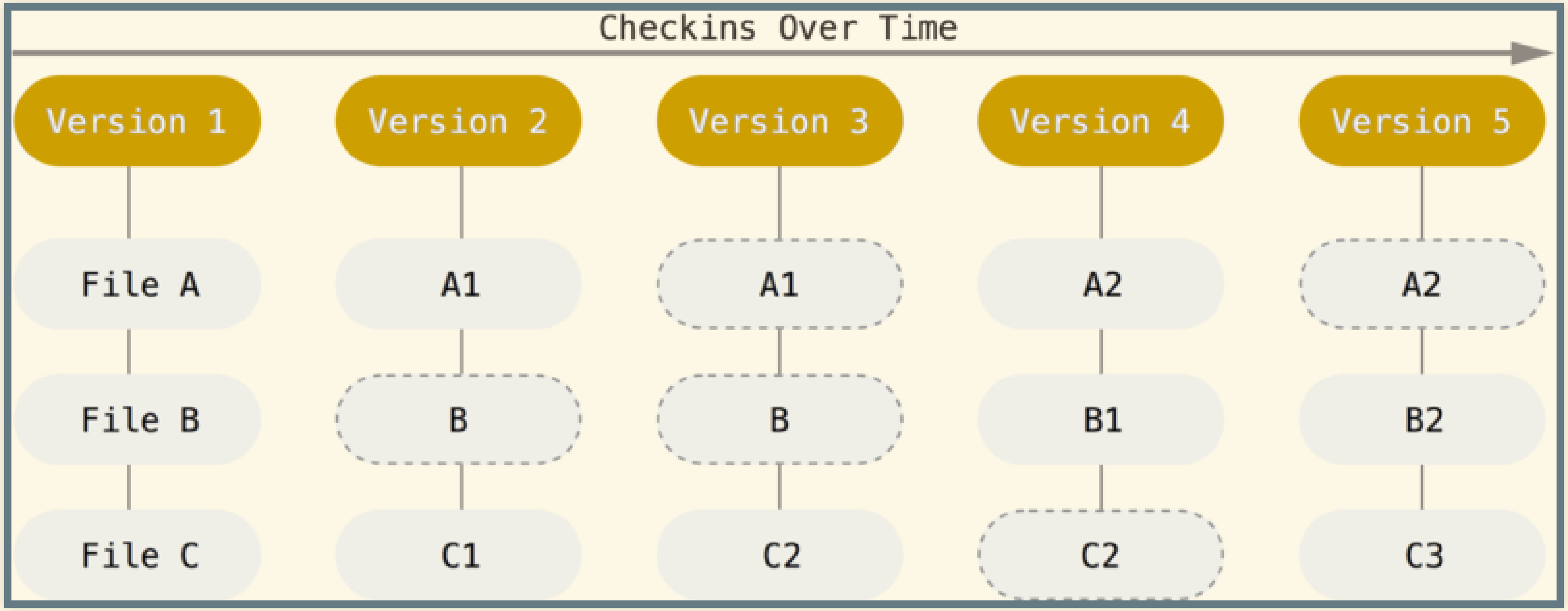
PRO GIT

https://git-scm.com/book/en/v2

DISTRIBUTED VERSION CONTROL SYSTEM

VERSION CONTROL

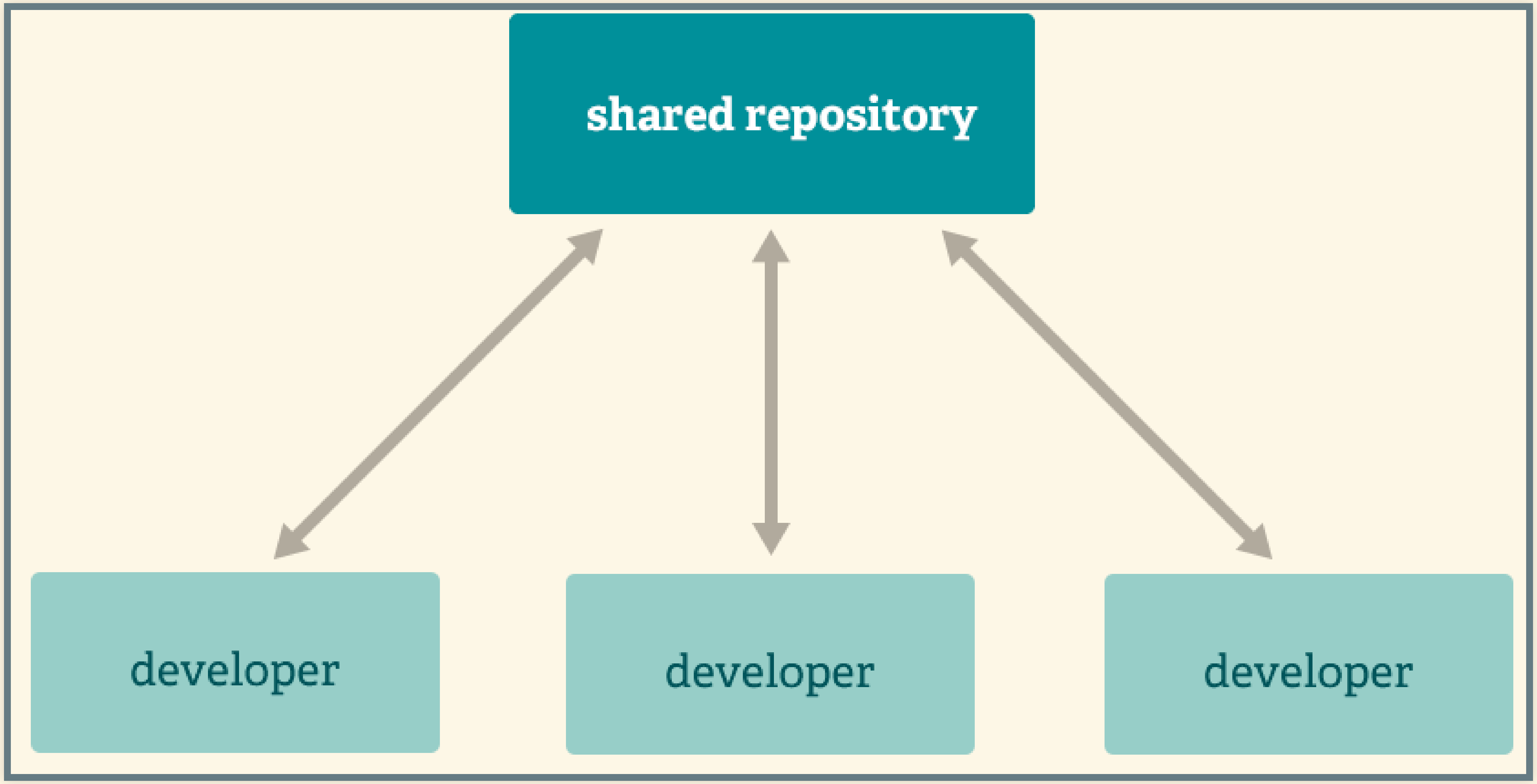
snapshots over time



DISTRIBUTED

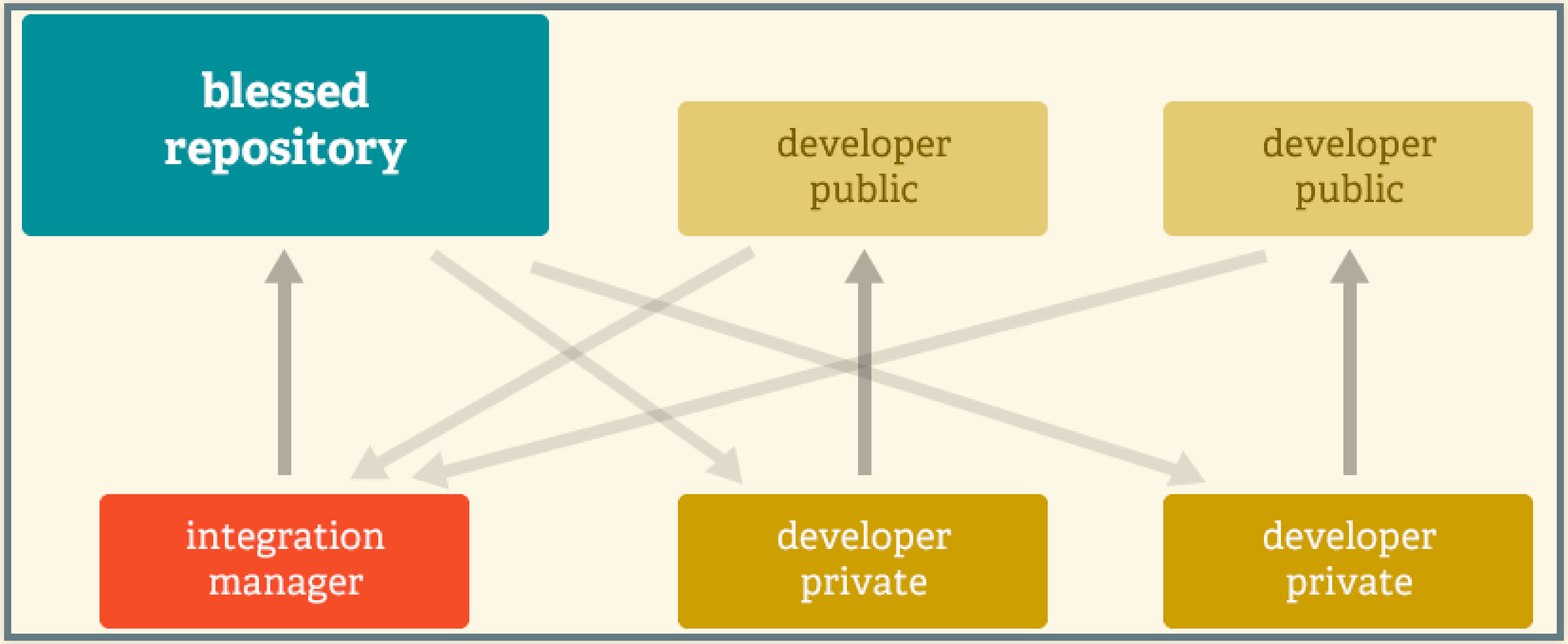
You get a copy! You get a copy! EVERYONE gets a copy!

CENTRALIZED



e.g. you and your buddies

INTEGRATION MANAGER / PROTECTED



e.g. many open source repositories

SO WHAT?

CODE FAST

CODE SAFE

CODE TOGETHER

CODE FEARLESSLY

read

THE NOUNS

REPOSITORY

BRANCH

COMMIT

REMOTE

REPOSITORY

Typically this will map to a project.

Stores files and metadata.

REPOSITORY

basically a .git directory with subdirectories for objects, refs/heads, refs/tags, and template files. An initial HEAD file that references the HEAD of the master branch is also created.

reference

```
$ ls .git/
branches  COMMIT_EDITMSG  config  description  HEAD
hooks  index  info  logs objects  ORIG_HEAD  refs
```

BRANCH

A pointer to a specific history of commits.

Also known as a head.

COMMIT

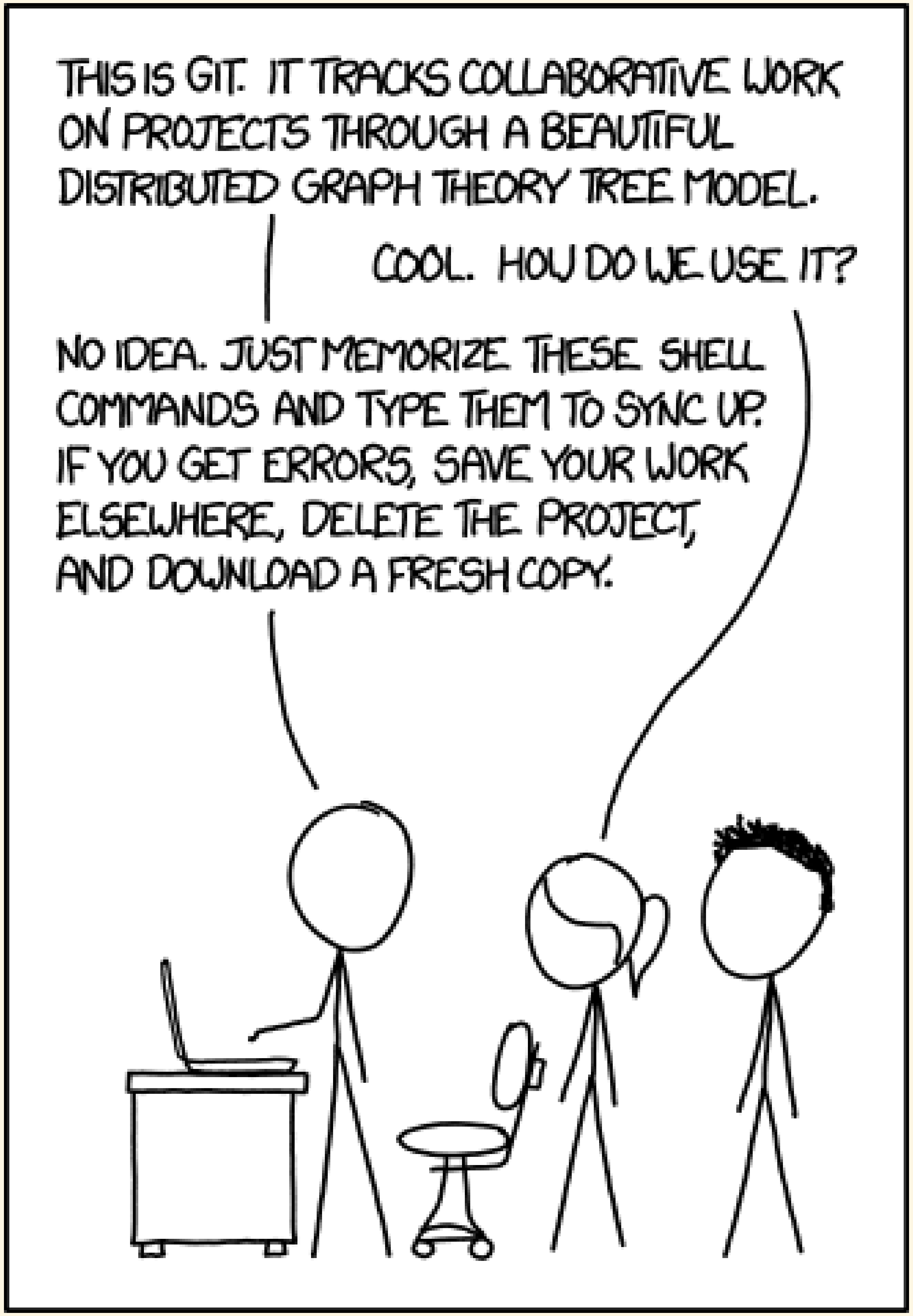
A specific snapshot (revision) of your repository.

REMOTE

Another copy of the repository. For example:

- GitHub
- GitLab

THE VERBS


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.

CORE COMMANDS

- setup
- snapshot
- branch and merge
- share and update

SETUP

init

```
$ git init
Initialized empty Git repository in /home/roy/code/git-101/.git/
```

clone

```
$ git clone https://github.com/royragsdale/git-101.git
Cloning into 'git-101'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

$ ls
git-101
```

WHAT IS GOING ON?

`git status`

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   example.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

GIT STATUS

SNAPSHOT

A two step process, so you can review your changes.

add

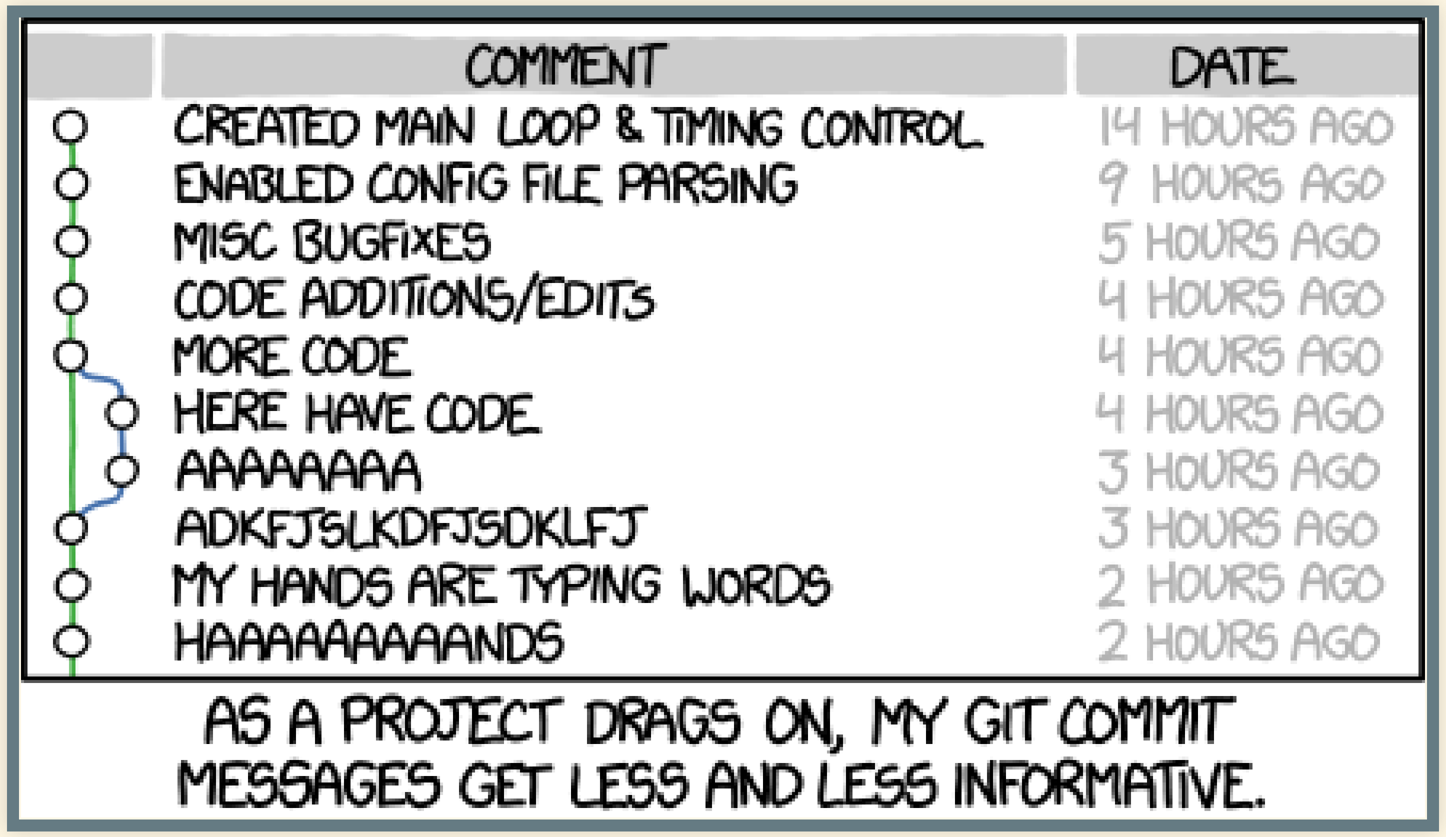
```
$ git add example.txt

$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   example.txt
```

commit

COMMIT MESSAGES



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

MODEL COMIT

```
Here’s a model Git commit message:

Capitalized, short (50 chars or less) summary

More detailed explanatory text, if necessary.  Wrap it to about 72
characters or so.  In some contexts, the first line is treated as the
subject of an email and the rest of the text as the body.  The blank
line separating the summary from the body is critical (unless you omit
the body entirely); tools like rebase can get confused if you run the
two together.

Write your commit message in the imperative: "Fix bug" and not "Fixed bug"
or "Fixes bug."  This convention matches up with commit messages generated
by commands like git merge and git revert.

Further paragraphs come after blank lines.
```

From Tim Pope.

COMMIT MESSAGE

Other useful resources for more explanation.

- Git Source
- Chris Beams
- Caleb Thompson
- Code Like a Girl

reference

BRANCH AND MERGE

walkthrough

- `checkout`
- `merge`

SHARE AND UPDATE

push : Share your code

```
$ git push --set-upstream origin add-inital-slides
Counting objects: 285, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (256/256), done.
Writing objects: 100% (285/285), 6.49 MiB | 8.79 MiB/s, done.
Total 285 (delta 50), reused 0 (delta 0)
remote: Resolving deltas: 100% (50/50), done.
remote:
remote: Create a pull request for 'add-inital-slides' on GitHub by visiting:
remote:      https://github.com/royragsdale/git-101/pull/new/add-inital-slides
remote:
To github.com:royragsdale/git-101.git
 * [new branch]      add-inital-slides -> add-inital-slides
Branch add-inital-slides set up to track remote branch add-inital-slides from
origin.
```

PUSH

```
$ git status
On branch add-inital-slides
Your branch is ahead of 'origin/add-inital-slides' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

$ git push
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 56.12 KiB | 0 bytes/s, done.
Total 13 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:royragsdale/git-101.git
   b76e17b..88b80c1  add-inital-slides -> add-inital-slides
```

SHARE AND UPDATE

`pull` : Update your local copy

```
$ git pull
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 3), reused 13 (delta 3), pack-reused 0
Unpacking objects: 100% (13/13), done.
From github.com:royragsdale/git-101
   b76e17b..88b80c1  add-inital-slides -> origin/add-inital-slides
Updating b76e17b..88b80c1
Fast-forward
 slides/content/home/background.md     |   2 +-
 slides/content/home/practice.md       |  17 ++++++++++++++
 slides/static/img/xkcd_git_commit.png | Bin 0 -> 53731 bytes
 6 files changed, 146 insertions(+), 3 deletions(-)
 create mode 100644 slides/content/home/practice.md
 create mode 100644 slides/static/img/xkcd_git_commit.png
```

INFO

log : see the history

```
commit 88b80c1d1085677497ab1e74db69c4d5031529d3
Author: Roy Ragsdale <roy@ragsdale.xyz>
Date:   Mon Oct 1 19:03:33 2018 -0400

    Work through commands

commit b76e17b74f7a6a4a0d69456dfd5485b2a2910fe1
Author: Roy Ragsdale <roy@ragsdale.xyz>
Date:   Mon Oct 1 18:21:30 2018 -0400

    Pixelize Linus
```

INFO

```
git log --oneline
* 88b80c1 Work through commands
* b76e17b Pixelize Linus
* 6689733 Add so what
* c212f4f Add new machine sections
* 805c3e3 Add most of background
* 6c28b6a Add initial command list to cover
* 48f6f38 Split out sections into individual files
* b9ffda8 Add initial theming and yo setup
* 364e9cb Vendor reveal-hugo theme to build slide deck
* ea513f0 Initial Commit - add Readme
```

INFO

diff: see what has changed

```
$ git status
On branch add-inital-slides
Your branch is up-to-date with 'origin/add-inital-slides'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   content/home/commands.md

no changes added to commit (use "git add" and/or "git commit -a")

$ git diff content/home/commands.md
diff --git a/slides/content/home/commands.md b/slides/content/home/commands.md
index f32820a..a86c8bf 100644
--- a/slides/content/home/commands.md
+++ b/slides/content/home/commands.md
```

INFO

See what has changed on your branch

```
git diff master..
```

HELP

```
$ git help <verb>
$ man git-<verb>
```

reference

FROM SCRATCH

- put it all together
- new machine
- keys!

A WORKFLOW

1. Make Sure you are up to date

```
git checkout master
git pull
```

2. Create a branch

```
git checkout -b add-workflow
```

3. Add commits (repeat)

```
git add file
git commit
```

CONTINUED

Push

```
git push -u origin add-workflow
```

or Merge

```
git checkout master
git merge --no-ff add-workflow
```

NEW MACHINE

You only need to do this on a brand new machine

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com

$ git config --global core.editor "gedit -s"
```

reference

SSH BASED AUTH

You only need to do this on a brand new machine

WE'LL DO IT LIVE

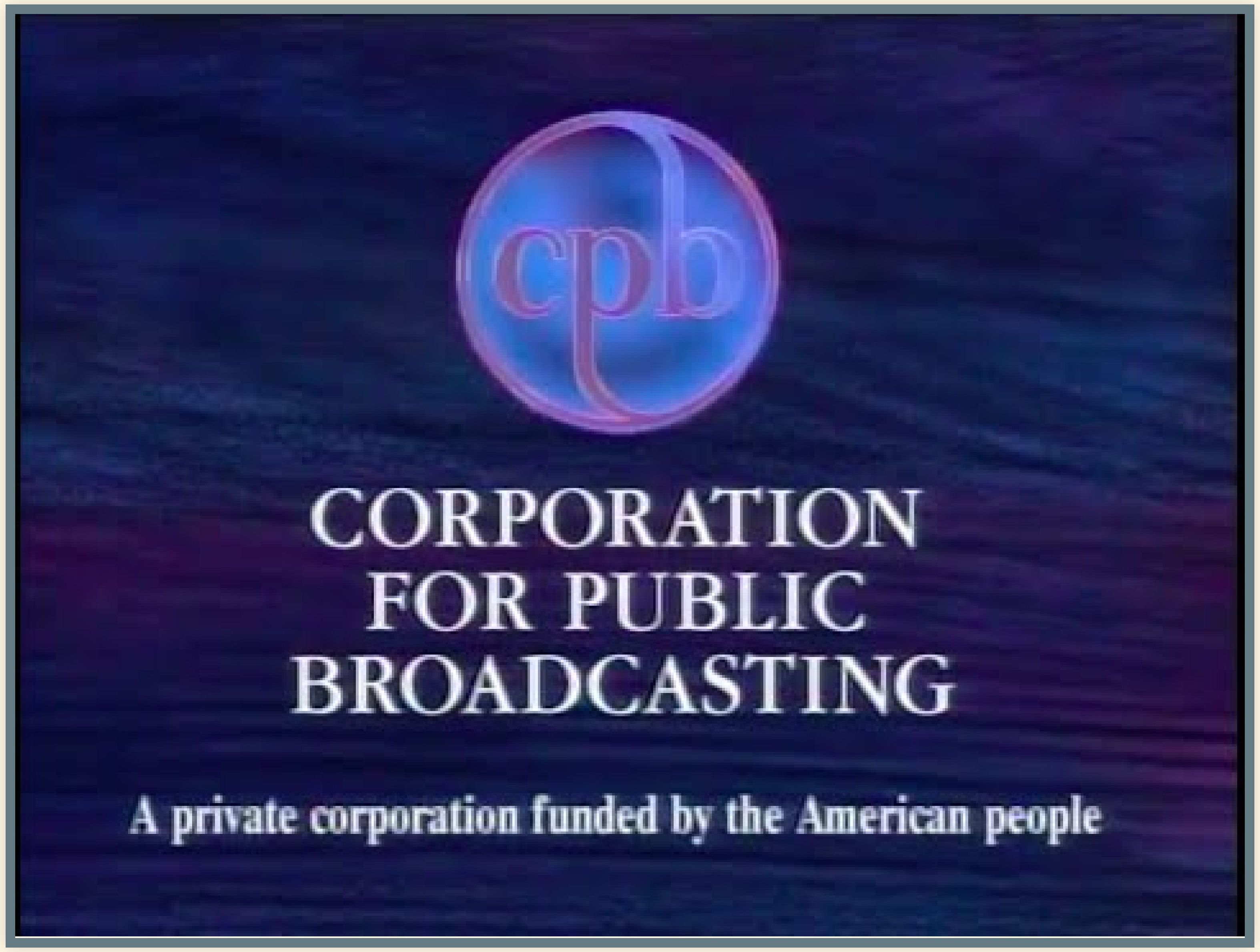
HTTPS://LEARNGITBRANCHING.JS.ORG/

FLASK DEMO

RESOURCES

- Documentation
- Cheat Sheet
- Exercises
  - Learn Git Branching
  - Codecademy
- Learn Git with Bitbucket Cloud
- More links

THIS PROGRAM IS BROUGHT TO YOU BY...


cpb
CORPORATION FOR PUBLIC BROADCASTING
A private corporation funded by the American people

- Yo Dawg Meme
- Git Logo
- Linus
- Snapshots
- Oprah
- Distributed
- XKCD 1, 2
- Corporation for Public Broadcasting

THANK YOU