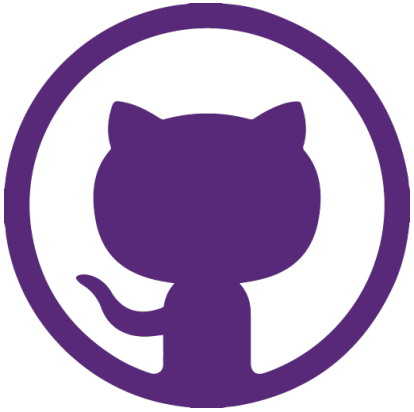


Version Control, git, and GitHub: An Introduction

The University of Oklahoma

Goal

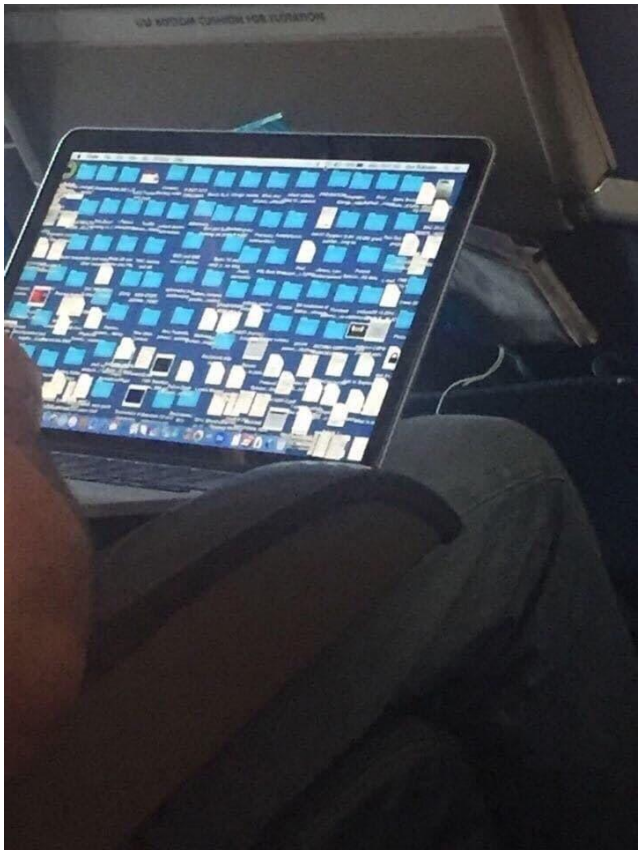
Introduce version control and git and walk through the mechanics of using it. Hopefully you walk away with a confidence to start using it and learn on your own.

Motivation

Imagine you are writing a paper and you delete a section during edits. The next day you decide you want to keep that section, but you saved over the old version! Maybe you decide to save a unique file for every iteration…

Maybe you add a line of code to your programming project and now it breaks at runtime. If you can't find that new line again you are left to sift through your code...

Both of these problems could have been served using version control!



What is Version Control?

Briefly stated, version control is the “management of changes to documents” (shamelessly stolen from Wikipedia).

Version control software allows you to:

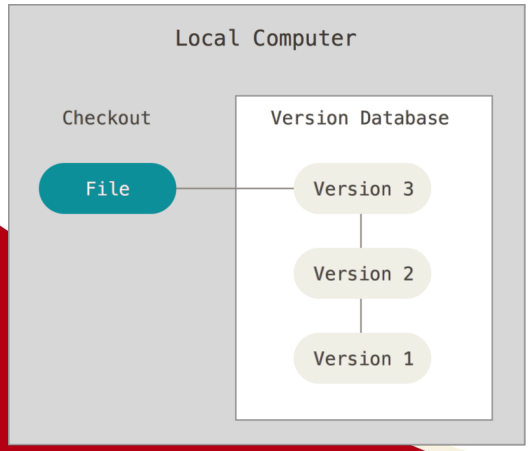
- record previous iterations of your projects without saving them each of them as separate files
- navigate these previous iterations and restore them if needed

In short: maintain a SINGLE set of files while tracking what changes have occurred

Different Version Control Systems

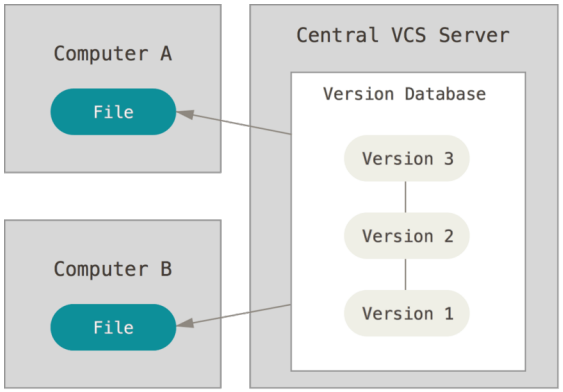
Local

Beyond copying files into another directory, local VCSs use a simple database to control changes



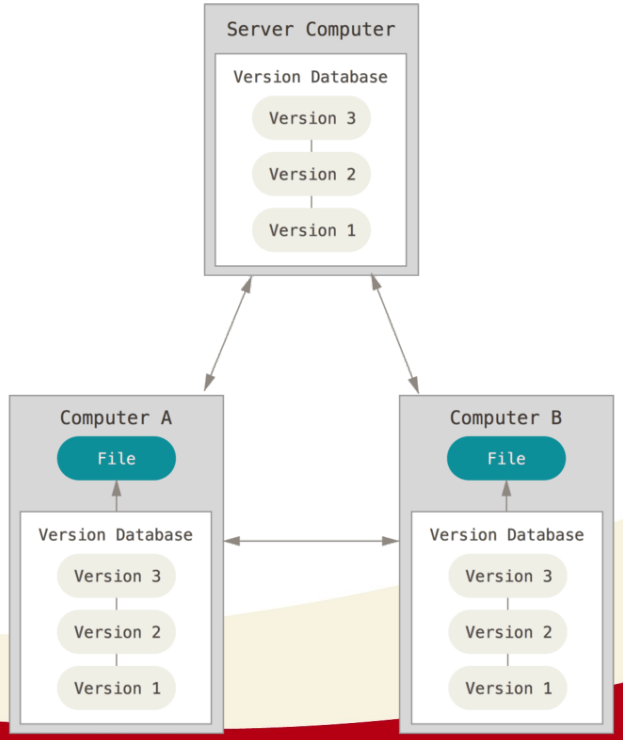
Central

Servers allow developers to collaborate. Administrators have control on who can do what but server failures stall progress.



Distributed

Clients fully mirror repositories; every clone is a full backup of the data!

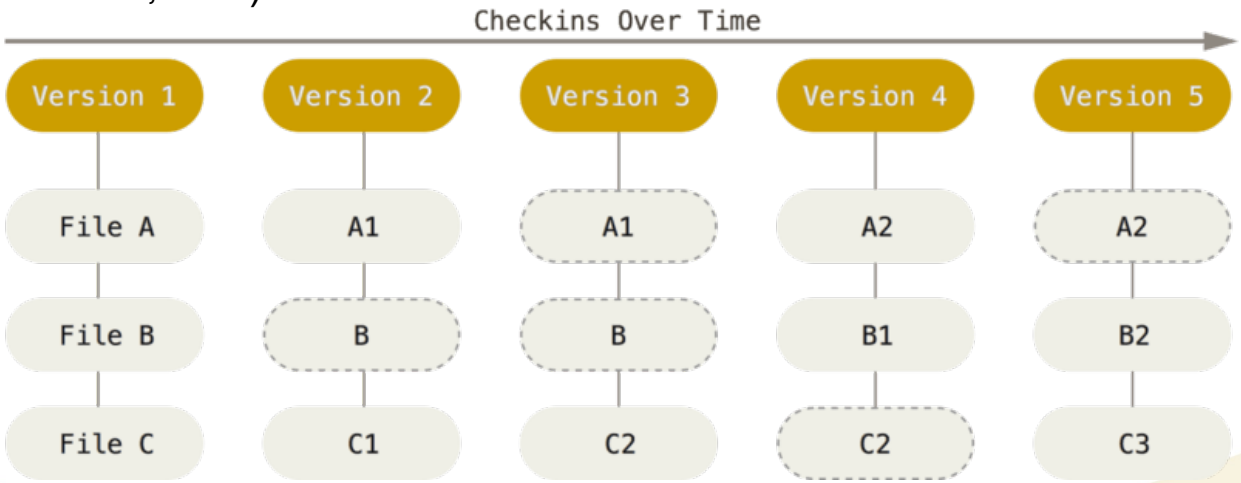


The University of Oklahoma

What is Git?

Git is a popular distributed version control system created in 2005 by Linus Torvalds in response to the DVCS used for the development of the Linux kernel becoming a paid service.

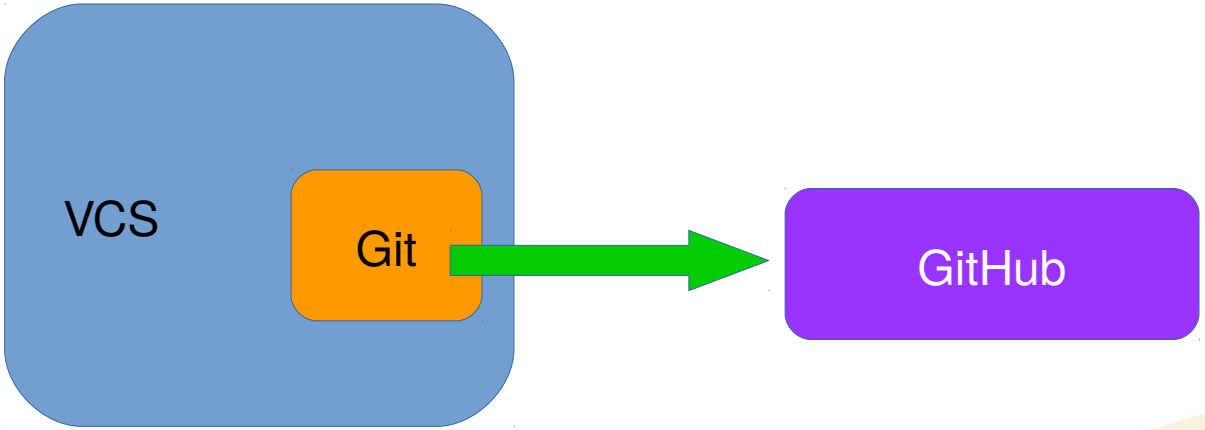
Git has many nice features in addition to the ones that come from being DVCS (offline work, it is ‘small’, etc.)



What is GitHub?

GitHub is a web service that hosts Git repositories. It is best known for its website and is in many ways a social platform.

GitHub allows the world to see what contributions you make to open source projects. They also offer private repositories (students get them for free!)



Starting with Git

Installation:

Chances are you already have it installed!

Otherwise, consult the official website:

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Configuring your user info:

```
$ git config --global user.name "Jane Doe"
$ git config --global user.email jdoe@ou.edu
```

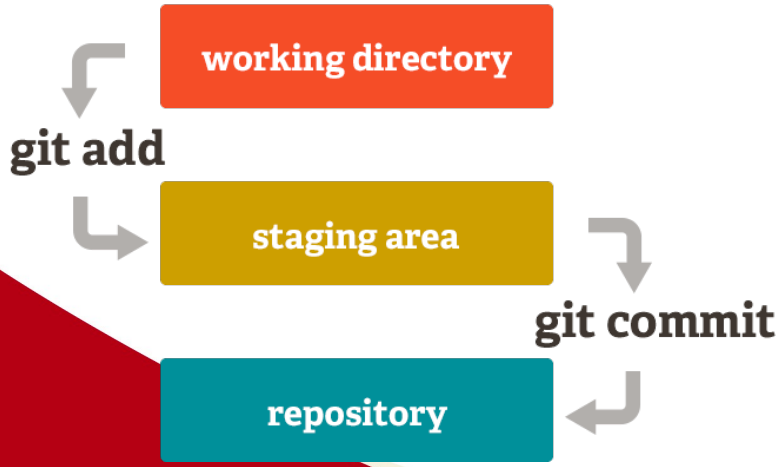
This is so contributions are tagged by their creator

Creating a repository:

```
$ git init
  OR
$ git clone https://www.URL.com/project.git
```

Functional Git

An understanding of how Git operates is needed. The history of changes in a repository come in the form of **commits**. Changes must be **staged** before they can be committed.




```
$ git status
```

Inspect status of directory...

```
$ git add <files-to-be-staged>
```

...STAGE your changes…

```
$ git commit -m "commit msg"
```

...then COMMIT them

These 3 commands comprise half of your Git usage!

The University of Oklahoma

Remote Repositories

These are commands related to communicating with remote repositories. These are repos that do not live on your local machine.

```
$ git remote add <shortname> <url>
```

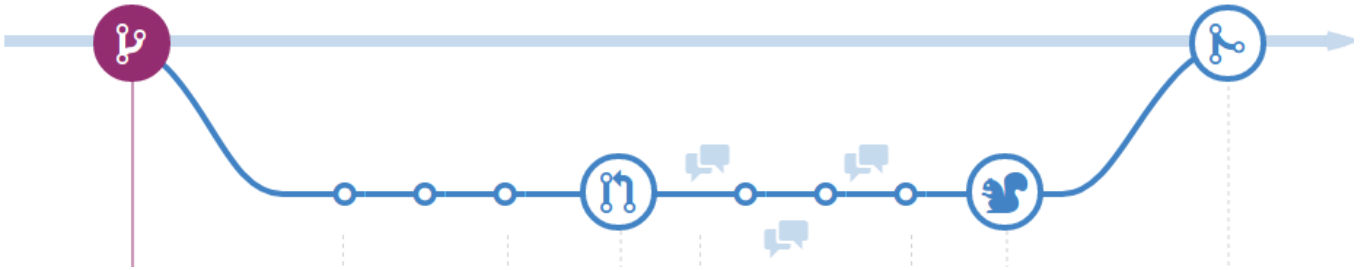
```
$ git push <remote> <branch>
```

```
$ git pull <remote> <branch>
```

=

```
$ git fetch
$ git merge
```

Branches



Branches allow versions of files to exist in parallel! Good for maintaining a stable release version, experimental version, and smaller branches for day to day work. Important when adding big changes, working with multiple people, etc.

List all branches

```
$ git branch -av
```

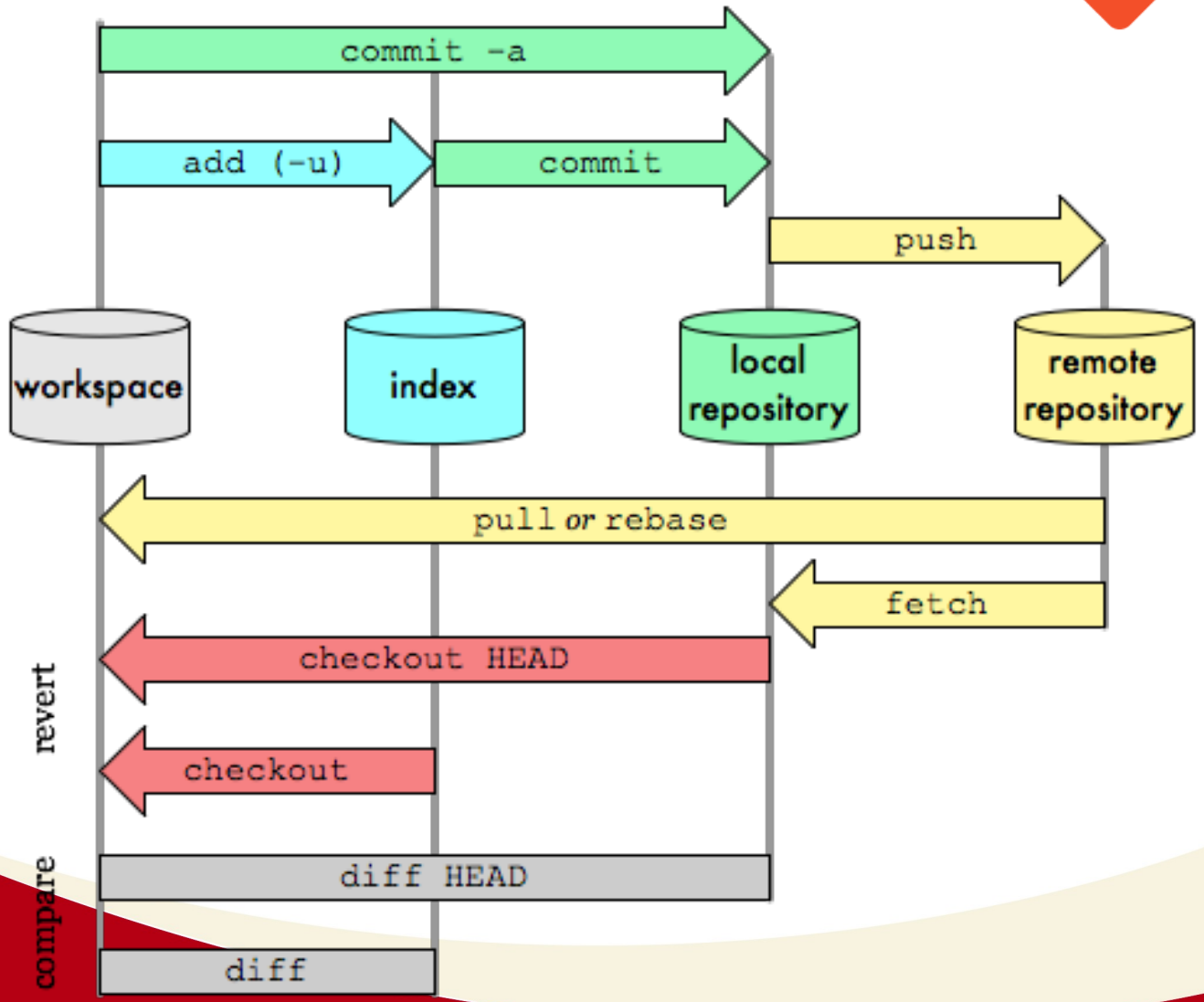
Switch HEAD branch

```
$ git checkout <branch>
```

Create new branch based on HEAD

```
$ git branch <new-branch>
```

Big Picture



Oliver Steele
osteele.com

Other Commands

Changes to tracked files

```
$ git diff
```

Show all commits, starting with newest

```
$ git log
```

Revert to previous commit

```
$ git reset
```

Store mods and staged changes

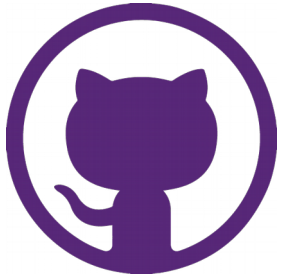
```
$ git stash
```

Who changed what and when in file

```
$ git blame <file>
```

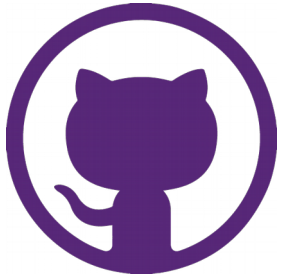
It is difficult to change/erase history in git. Almost every action adds information to your repository rather than remove it. Changing the history of your project (seldom performed) is a very deliberate process and practically impossible to do on accident.

GitHub Features



- Forks - duplicating someone else's repository onto your own account
- Pull Requests - formal way letting others know what changes you have made to your forks
  - if no conflicts exist, the changes can be accepted into the requested
  - they are, if nothing else, a platform for discussing the changes
  - any one can submit these which is how non-owners can contribute to projects
- Other features like Issues, etc.

These are features separate from vanilla Git



Offroad Tour of GitHub

The University of Oklahoma

Last Thoughts

- This was a lot of information at once but hopefully you have the confidence to try it yourself
- Many GUIs do exist but there's still a learning curve! I find the command line simpler.
- Git is useful in of itself but it is also a resume addition
  - Job candidates with Git listed appear considerably more competent
  - List or link your GitHub profile! This is crucial for jobs involving programming
  - Employers would probably expect you to be familiar with Git on the command line rather than on GUIs

Resources

- Git official site: https://git-scm.com/
  - much of this slideshow is stolen from there
- Pro Git - free pdf also featured on official site
  - paperback available for purchase
- Git cheat sheets
  - https://www.git-tower.com/blog/git-cheat-sheet/
  - https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
- Interactive git tutorial:
  - https://try.github.io/
- The best tutorial is using it on your own projects!