

CSE 374 - Week 6 (Fri)

Git

Instructor: Andrew Hu

TAs: Jim Limprasert, Kaelin Laundry,
Keith Jiang, Nick Durand, Simon
McFarlane, Yitian Hao



Source: XKCD

Plan for the Week

- Software engineering week!
- Problems we will look at
 - ~~Monday: What does it mean for our program to be correct?~~
 - ~~Wednesday: How do we test our program's correctness?~~
 - Friday: How can we collaborate and track changes to our code?
- Solutions
 - ~~Monday: pre and post conditions~~
 - ~~Wednesday: testing~~
 - Friday: git

Version Control: Individual

- Does any of the following sound familiar?
 - Your code was working great! Then *you made a few changes* and now everything is broken and you **saved over the previous version?**
 - You **accidentally delete** a critical file and can't get it back.
 - Your computer was broken or stole and now **all of your files are gone!**
 - While writing a paper for one of your classes you save each version as **final_paper.doc, final_paper2.doc, final_paper_actually_this_time.doc, UGH.doc**
- There has to be a better way to manage versions...

Version Control: Teams

- Does any of the following sound familiar?
 - My partner and I are paired up for a project for one of our CSE classes. We usually pair program together in the labs but sometimes we have to work remotely. **Who keeps the most up-to-date version of the project?** How do we **share changes** with each other? What if I want to compare the changes my partner made?
 - How do we **keep backups** of important files? Who stores them on their computer?

Version Control

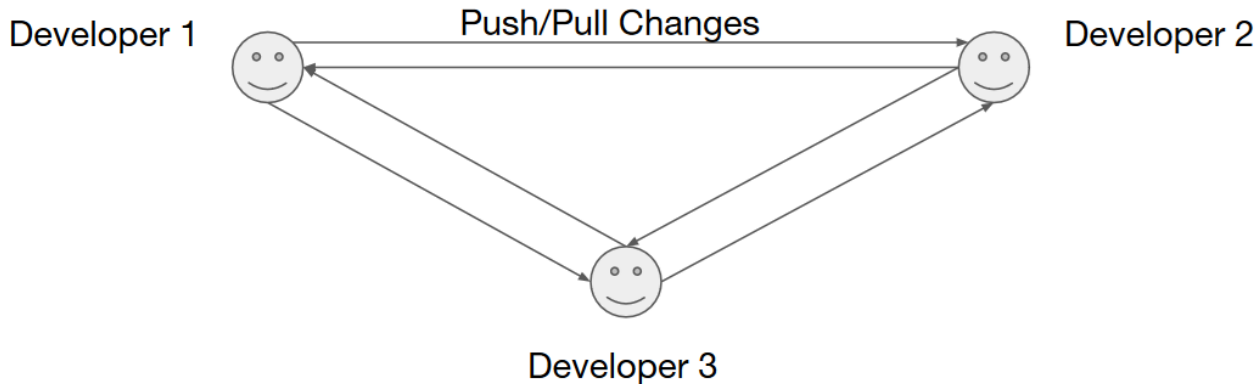
- Version Control: Software that keeps track of changes to a set of files.
- You likely use version control all the time:
 - In Microsoft Word, you might use Ctrl+Z to undo changes and go back to an earlier version of the document
 - In Google Docs you can see who made what changes to a file
- Lots of people have a use-case for version control
 - We often think of version control as related to managing code bases, but it's also used by other industries such as law firms when keeping track of document changes over time

Repository

- A repository, commonly referred to as a **repo** is a location that stores a copy of all files
 - The working directory(or working tree) is different from the repository (see next slide)
- What should be inside of a repository?
 - Source code files (i.e. .c files, .java files, etc)
 - Build files (Makefiles, build.xml)
 - Images, general resources files
- What should not be inside of a repository (generally)
 - Object files (i.e. .class files, .o files)
 - Executables

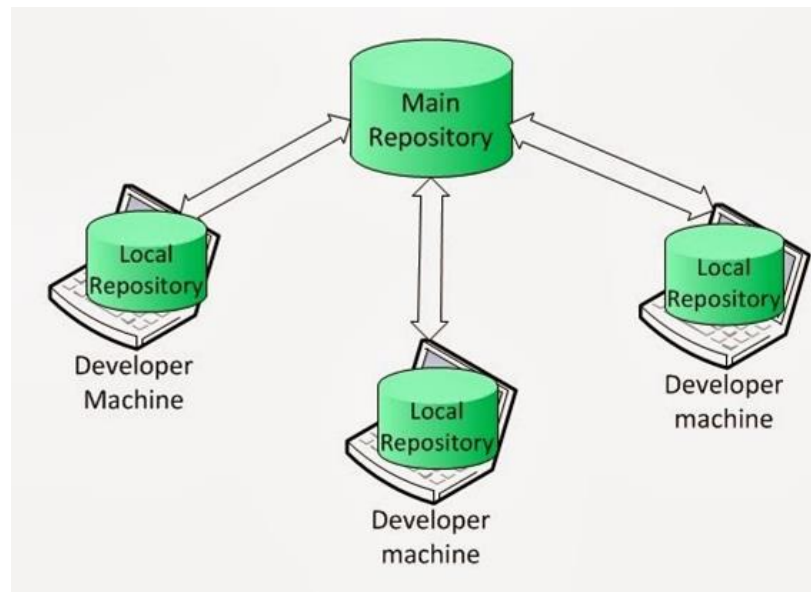
Git

- With git, everyone working on the project has a complete copy of the repository and its history
 - Everyone has a local copy of the repository, which is what we use to make our own changes.
 - We share changes by *pushing* and *pulling*



Central git repos

- Keep an "origin" copy of the repo on a Git server
 - The remote repository is the *defacto* central repository
 - Remote repositories are hosted on services like GitHub or Gitlab
- Everyone can push their changes and pull others' changes



A repository's history is a series of "commits"

- Each commit makes changes to the files in the repo
- *Commit history* serves as a log of the changes everyone made
- Commits are easy, and free! Commit early and often.
 - Ever accidentally deleted something and forgot what it was?
 - If you make a mistake, you can look at the changes since the last commit

What is a "commit"?

- A "commit" is a **single set of changes** made to your repository
- Also records:
 - The name of the author
 - The date and time
 - A "commit message": short sentence describing what that commit did
- Identified by an ID, or "SHA"

```
commit 6b2186c8105d15774591d8cc949de6ea7b1922fd
Author: Kaelin Laundry <kaelinl@cs.washington.edu>
Date:   Wed Jul 29 23:43:04 2020 -0700


hw4: begin introducing optional bugs
```

Commit messages

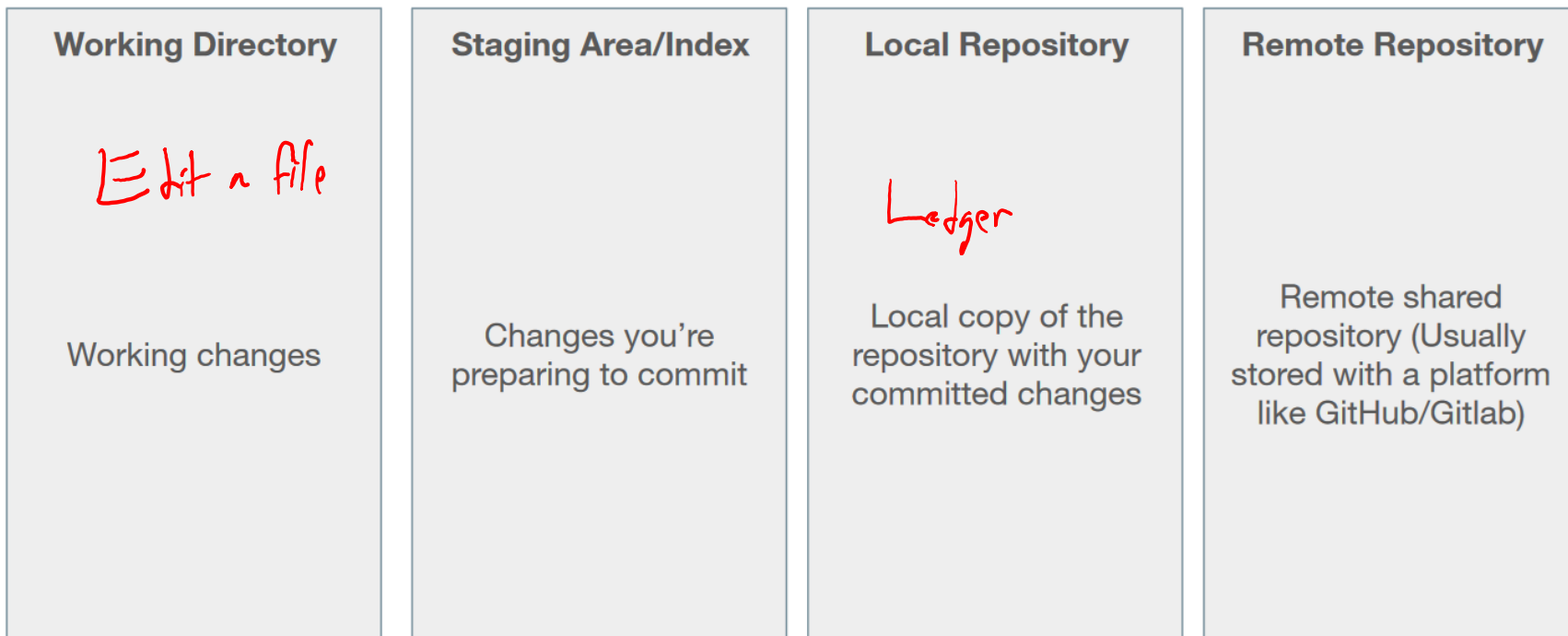
- Commit messages are the way you remind yourself and tell others what you did
- Commit messages should be **descriptive**
 - E.g. "Added test for predicting null string"
 - *not* "changed test"
- Commit messages should be **short/medium length**
 - If you want to know *exactly* what code was changed, you can check the full changes.

The "git" command

- The "git" command is the primary way of interacting with git
- You must "cd" into the folder where your repo is stored, or any subfolder within it
- Used like any other shell command!

git 

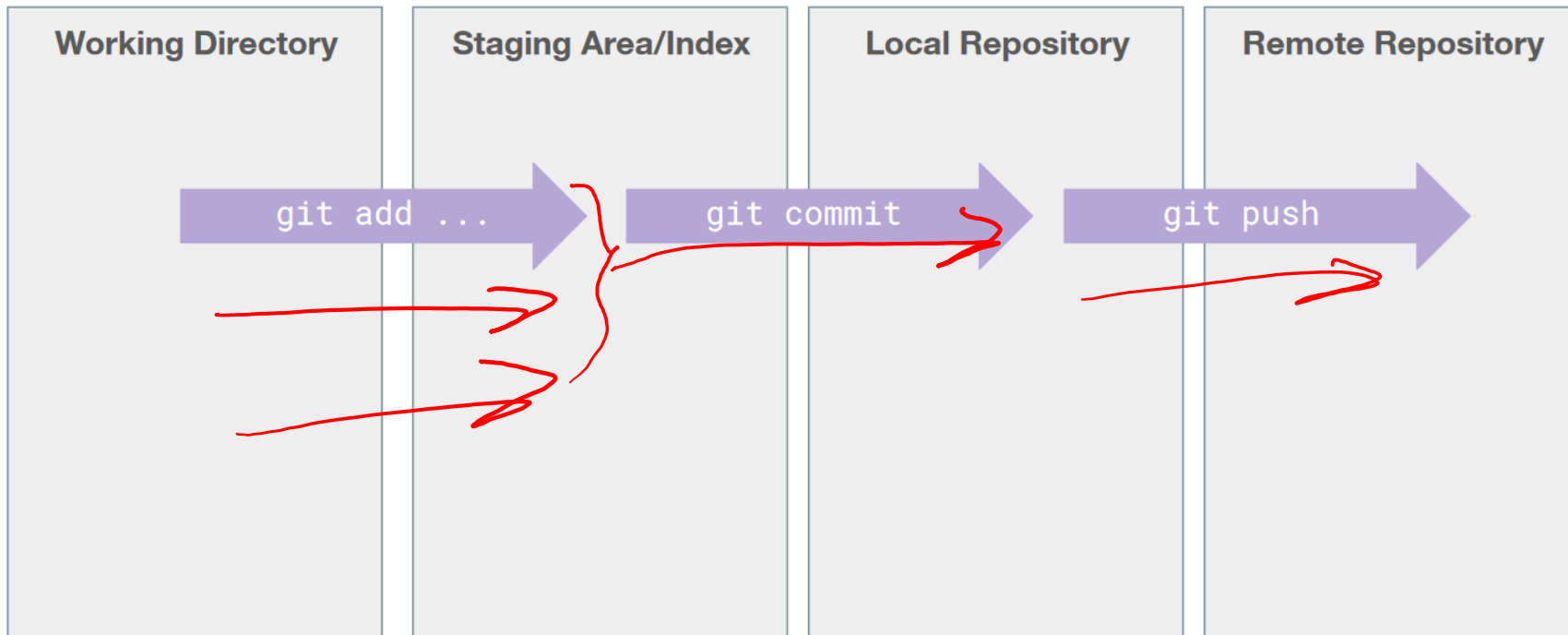
Git: Four Phases



Git: Four Phases



Git: Four Phases



NOTE: There are way more git commands than what is listed here - this is a simplified model to get us started.

Inspecting a Git Repository

- `git status`
 - Lists the files which you have changed but not yet committed
 - Working directory (1)
 - Staging area (2)
 - Indicates how many commits have made but not yet pushed
- `git log`
 - Shows the commit history
 - Press "q" to exit



Poll Question: [Pollev.com/andrewhu](https://pollev.com/andrewhu)



Poll Question ([PollEv.com/andrewhu](https://pollev.com/andrewhu))

If git status shows the following, which phase is file1.txt in?

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file1.txt

10 A. Working directory

2 B. Staging area/index

1 C. ~~Local repo~~

0 D. ~~Remote repo~~

Creating a Commit

- Make a change to any file
- "add" the change
 - `git add path/to/file.ext`
- "commit" the change
 - `git commit --message "made some changes"`

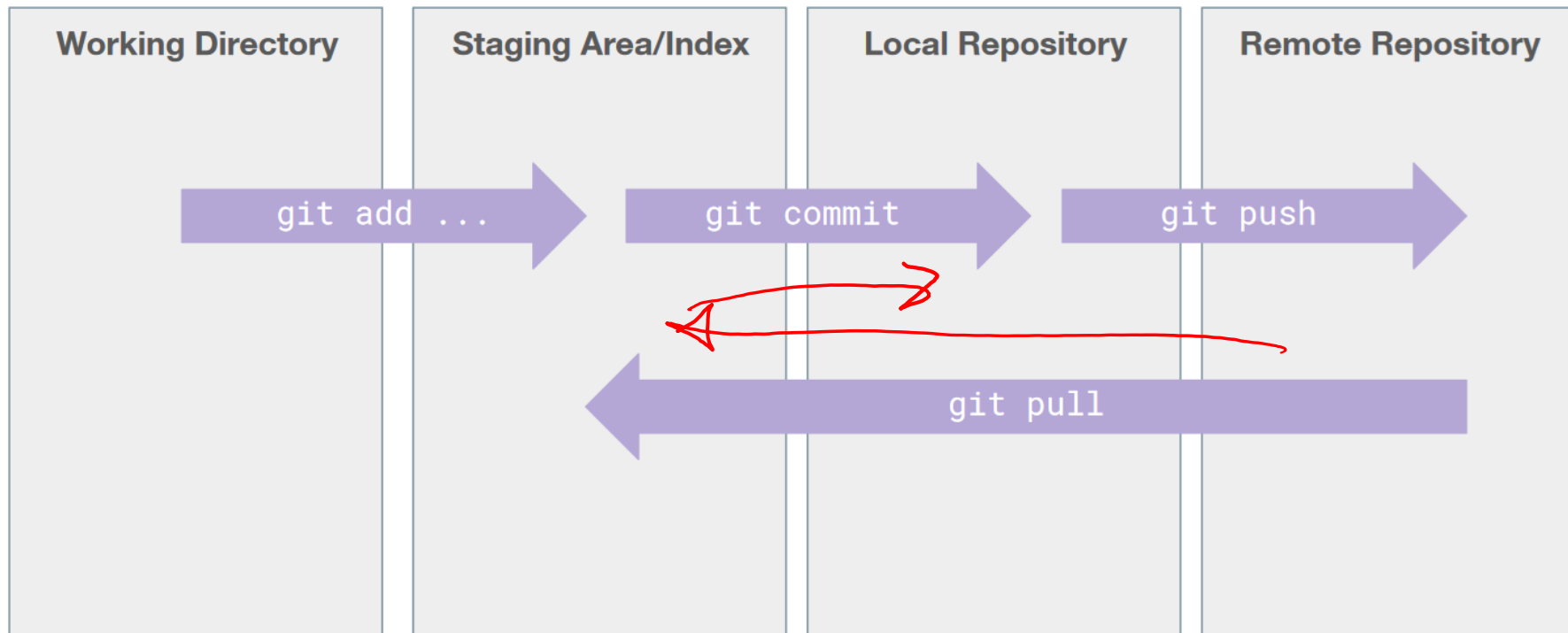
-m

How do I fix git?!

- Even for experienced users, sometimes you may accidentally get your git repo into an undesirable situation
- There is always a way to fix this, although it's not always obvious how
- Online resources are helpful
 - <https://ohshitgit.com/>

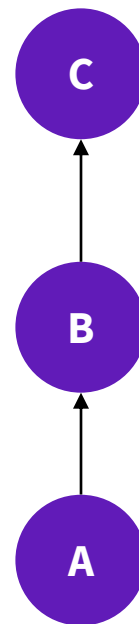
Demo: Working with `git` Locally

Git: Four Phases with Remote



Collaboration in a git repo: simple case

- A "linear" history
 - **Alice** makes a commit and **pushes**
 - **Bob pulls**, makes a change, commits the change, and **pushes**
 - **Alice pulls**, makes a change, commits, and **pushes**
 - ...etc.



CSE Gitlab

- Github and Gitlab are just websites that store git repos
- You can create a repo on the website and `git clone` to edit it on your computer (e.g. laptop, klaatu, etc.)
- CSE has its own version of Gitlab where you will be given a repository
 - <https://gitlab.cs.washington.edu/>

Cloning From Remote

The screenshot shows the GitLab interface for a repository named 'cse374-20su' (Project ID: 50790). The repository is a 'Staff repo' with 315 Commits, 4 Branches, 0 Tags, 8 MB Files, and 8 MB Storage. The 'Clone' button is circled in red. A dropdown menu is open, showing two options: 'Clone with SSH' and 'Clone with HTTPS'. The SSH URL is 'git@gitlab.cs.washington.edu:cs' and the HTTPS URL is 'https://gitlab.cs.washington.edu'. The HTTPS URL and its corresponding icon are also circled in red. A red arrow points from the text 'git clone' to the HTTPS URL.

cse374-20su Project ID: 50790

🔔 Unstar 1 🍴 Fork 0

📁 315 Commits 🌿 4 Branches 🏷️ 0 Tags 📁 8 MB Files 📁 8 MB Storage

Staff repo

master cse374-20su / +

History 🔍 Find file Web IDE 📄 Clone

Merge branch 'hw4-autograder' into 'master' ...
Kaelin Laundry authored 1 day ago

📖 README 📄 Add LICENSE 📄 Add CHANGELOG 📄 Add CONTRIBUTING 📄 Env

📄 Set up CI/CD

Clone with SSH
git@gitlab.cs.washington.edu:cs

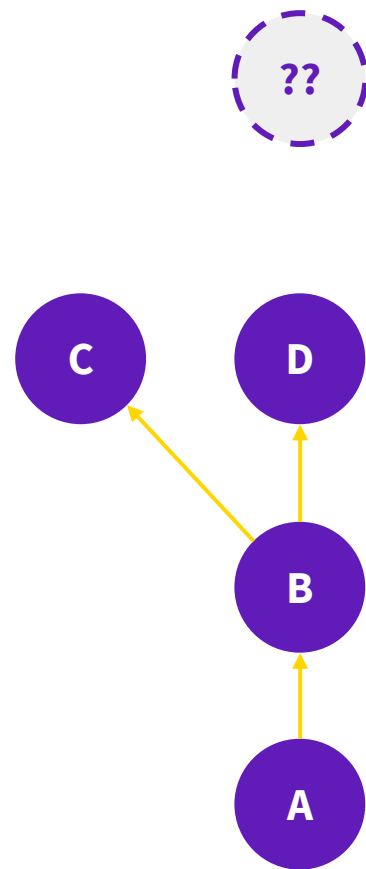
Clone with HTTPS
https://gitlab.cs.washington.edu

git clone →

Demo: `git` with Gitlab

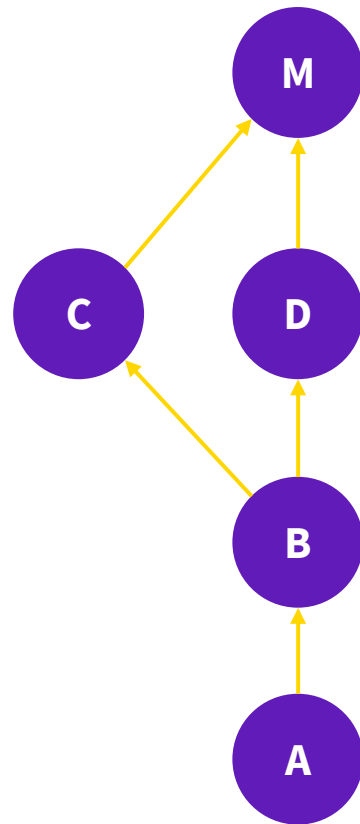
Collaboration: Reality

- We said the "commit history" is a list of commits. What happens here?
 - **Charlie** makes a change and creates commit C, but **doesn't push**
 - **Diane** also makes a change and commit D, and **pushes**
 - **Charlie pulls** from the remote repo
 - It's no longer a list! The history has **diverged**
- Does Charlie just have to delete, pull and start over?



Merging

- A **merge commit** is a commit which has two "parents"
 - Combines the changes in each
 - Commit "M" includes all of Diane's changes, *plus* all of Charlie's



How do we merge?

- `git pull`
 - Automatically fetches the changes and merges them into yours
 - Then, `git push`
 - Others can now work off of your combined changes
- Sometimes, the changes you make will "conflict" with the changes others make
 - e.g., you both edit the same line
 - Resolving merge conflicts is more complicated; come to OH or post on Ed!

Reference: Git Commands

<code>git clone <i>url</i> [<i>dir</i>]</code>	Downloads a copy of the git repository
<code>git add <i>files</i></code>	Adds file contents to staging area
<code>git commit</code>	Takes a snapshot of staging area and creates a commit
<code>git status</code>	View status of files in working directory and staging area
<code>git diff</code>	Show difference between staging area and working directory
<code>git pull</code>	Download ("fetch") history from the remote repository and merge it with local
<code>git push</code>	Upload your local commits to the remote repository/server

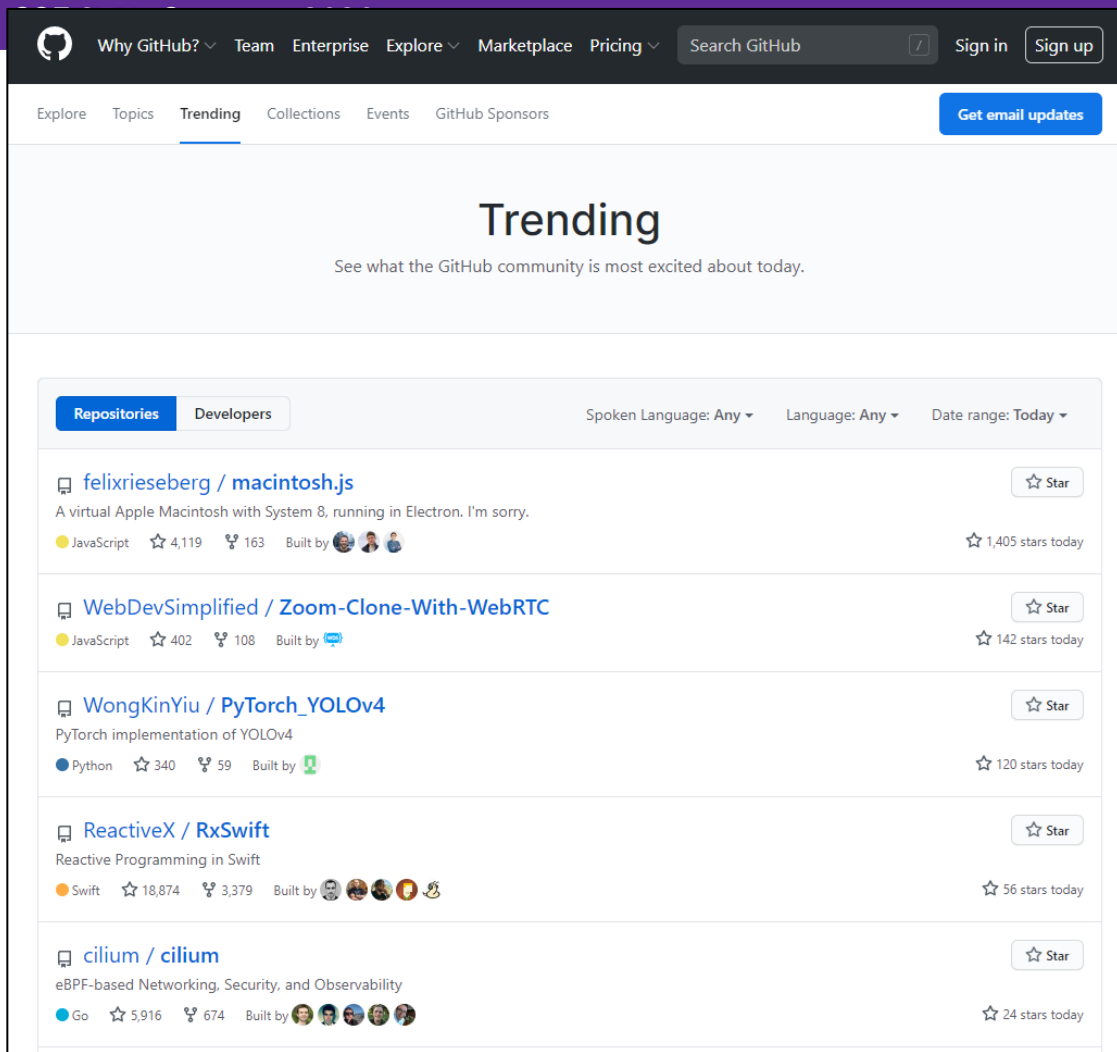
Git: learning more

- Lots of interesting and useful topics, including:
 - Branching, checkout
 - Resolving merge conflicts, mergetools
 - "Merge requests" (a.k.a. "Pull requests")
 - Rebase
- The web is your friend!
 - Official documentation
 - "Git Book": <https://git-scm.com/book/en/v2>

GitHub

Primarily public, open-source projects.

Check it out!



The screenshot shows the GitHub Trending page. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', 'Pricing', 'Sign in', and 'Sign up'. Below the navigation bar, there are tabs for 'Explore', 'Topics', 'Trending' (which is selected), 'Collections', 'Events', and 'GitHub Sponsors'. A 'Get email updates' button is located in the top right corner.

The main heading is 'Trending' with the subtitle 'See what the GitHub community is most excited about today.' Below this, there are filters for 'Repositories' (selected) and 'Developers', and dropdown menus for 'Spoken Language: Any', 'Language: Any', and 'Date range: Today'.

The list of trending repositories includes:

- felixrieseberg / macintosh.js**: A virtual Apple Macintosh with System 8, running in Electron. I'm sorry. JavaScript, 4,119 stars, 163 forks, built by 3 users. 1,405 stars today.
- WebDevSimplified / Zoom-Clone-With-WebRTC**: JavaScript, 402 stars, 108 forks, built by 1 user. 142 stars today.
- WongKinYiu / PyTorch_YOLOv4**: PyTorch implementation of YOLOv4. Python, 340 stars, 59 forks, built by 1 user. 120 stars today.
- ReactiveX / RxSwift**: Reactive Programming in Swift. Swift, 18,874 stars, 3,379 forks, built by 6 users. 56 stars today.
- cilium / cilium**: eBPF-based Networking, Security, and Observability. Go, 5,916 stars, 674 forks, built by 6 users. 24 stars today.

P.S.: git is complex!

Three of the top four most-upvoted questions on StackOverflow.

Everyone is learning!

stackoverflow Products Customers Use cases Search...

Home

PUBLIC

Stack Overflow

Tags Users

FIND A JOB Jobs Companies

TEAMS What's this? Free 30 Day Trial

All Questions

19,920,755 questions

Newest Active Bountied 439 Unanswered More Filter

24659 votes
26 answers
1.5m views

[Why is processing a sorted array faster than processing an unsorted array?](#)

Here is a piece of C++ code that shows some very peculiar behavior. For some strange reason, sorting the data miraculously makes the code almost six times faster: #include <algorithm> #include &...

java c++ performance optimization branch-prediction

asked Jun 27 '12 at 13:51

GManNickG 440k 47 450 530

21322 votes
84 answers
8.7m views

[How do I undo the most recent local commits in Git?](#)

I accidentally committed the wrong files to Git, but I haven't pushed the commit to the server yet. How can I undo those commits from the local repository?

git version-control git-commit undo pre-commit

community wiki 85 revs, 56 users 14% Peter Mortensen

17123 votes
40 answers
8.3m views

[How do I delete a Git branch locally and remotely?](#)

I want to delete a branch both locally and remotely. Failed Attempts to Delete a Remote Branch \$ git branch -d remotes/origin/bugfix error: branch 'remotes/origin/bugfix' not found. \$ git branch -d ...

git version-control git-branch git-push git-remote

asked Jan 5 '10 at 1:12

Matthew Rankin 380k 36 111 151

12034 votes
36 answers
2.8m views

[What is the difference between 'git pull' and 'git fetch'?](#)

What are the differences between git pull and git fetch?

git version-control git-pull git-fetch

asked Nov 15 '08 at 9:51

pupeno 244k 109 305 494