
Introduction to Git and GitHub



Fondren Library
Digital Scholarship Services



WHAT IS GIT?

- The most widely used version-control system!
 - Remembers entire project history
 - Easily undo changes
 - Track contributions
- Free and Open-Source Software
- Fast & Simple
- Optimized for collaboration
 - Non-linear development





WHAT IS GIT NOT?

- Affiliated with GitHub (which is owned by Microsoft)
- A file editor
- A graphical user interface
 - Many user interfaces exist
- Proactive
 - Must tell git about everything we do!





WHAT IS GITHUB?

- The largest host for Git repositories
- Free (for public repositories)
- A great place to collaborate
 - Support for unique features
 - Comment sections
 - Organizations
 - Issue tracking
- Extensible!



GitHub



WHAT IS GITHUB NOT?

- The only option
 - GitLab, Bitbucket, ...
- A version-control system
 - Everything is on top of git
- Only for source code and programming
- Difficult!



GITHUB & GITHUB DESKTOP

A walkthrough of the user interfaces





GETTING OUR BEARINGS: GITHUB.COM



GETTING OUR BEARINGS: USER PAGE



GETTING OUR BEARINGS: REPOSITORY



GETTING OUR BEARINGS: DESKTOP

STRUCTURE OF A GIT REPOSITORY

Snapshots, Locality, File States





SNAPSHOTS

- Sample Snapshot: Caption and Set of Files

devtools: move topbar out of scrollable container (#9077)
Hoten committed 4 days ago ✓

clients(devtools): ignore dark mode media query (#9082)
Hoten authored and brendankenny committed 4 days ago ✓

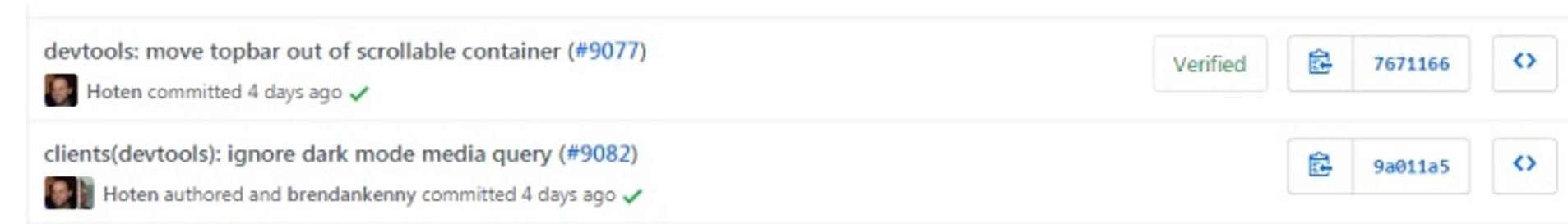
- Can see all files present at any of these times!

```
let turnOffTheLights = false;
if (window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
let turnOffTheLights = false;
// Do not query the system preferences for DevTools - DevTools should only apply dark theme
// if dark is selected in the settings panel.
if (!this._dom.isDevTools() && window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
```



SNAPSHOTS

- Sample Snapshot: Caption and Set of Files



devtools: move topbar out of scrollable container (#9077)
Hoten committed 4 days ago ✓ Verified 7671166

clients(devtools): ignore dark mode media query (#9082)
Hoten authored and brendankenny committed 4 days ago ✓ 9a011a5

- Can see all files present at any of these times!

```
let turnOffTheLights = false;
if (window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
let turnOffTheLights = false;
// Do not query the system preferences for DevTools - DevTools should only apply dark theme
// if dark is selected in the settings panel.
if (!this._dom.isDevTools() && window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
```



SNAPSHOTS

- Sample Snapshot: Caption and Set of Files

devtools: move topbar out of scrollable container (#9077)
Hoten committed 4 days ago ✓ Verified 7671166

clients(devtools): ignore dark mode media query (#9082)
Hoten authored and brendankenny committed 4 days ago ✓ 9a011a5

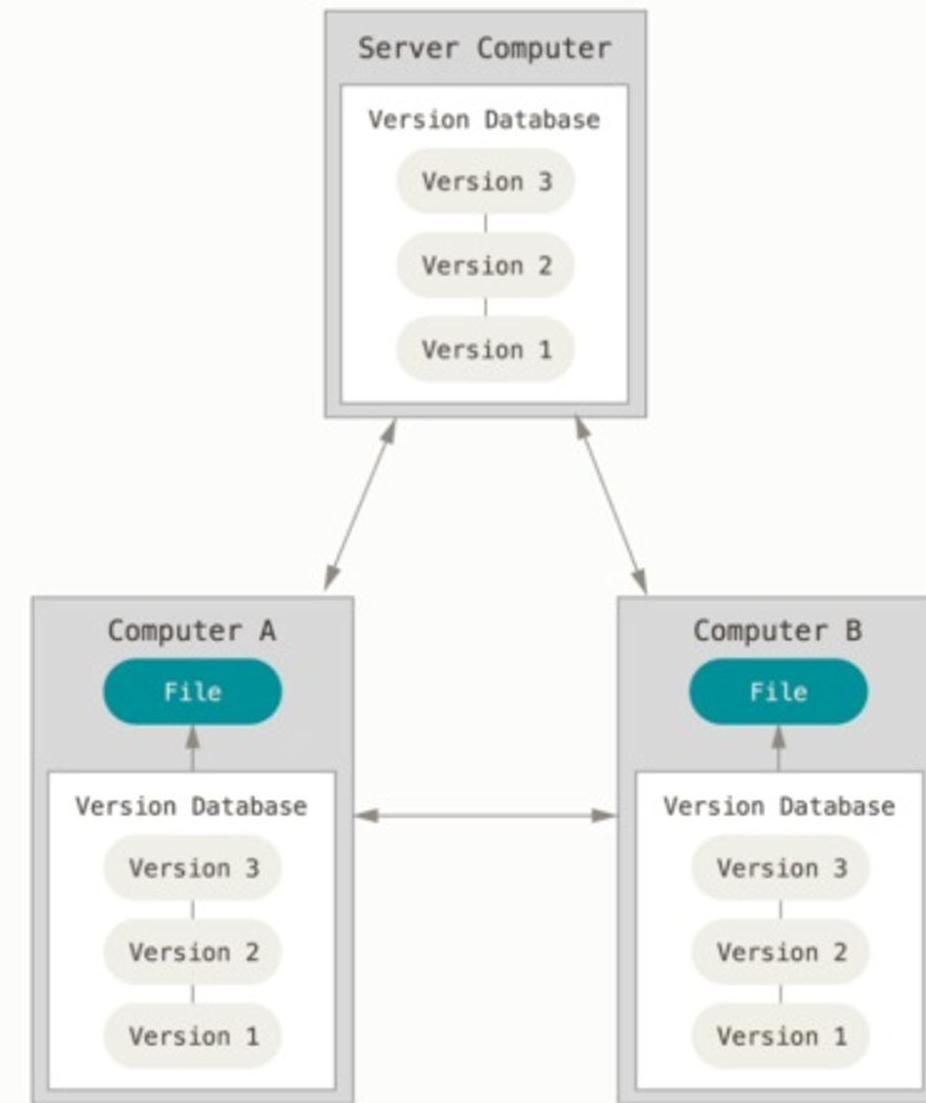
- Can see all files present at any of these times!

```
let turnOffTheLights = false;
if (window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
let turnOffTheLights = false;
// Do not query the system preferences for DevTools - DevTools should only apply dark theme
// if dark is selected in the settings panel.
if (!this._dom.isDevTools() && window.matchMedia('(prefers-color-scheme: dark)').matches) {
  turnOffTheLights = true;
}
```



LOCALITY

- Snapshots stored on your computer, not only on a server!
 - Hidden .git directory
 - Stores entire history of project (compressed!)





SPECIAL FILE: .GITIGNORE

- .gitignore (note the period!)
 - A set of filenames.
 - No file in here will ever be tracked by git
 - Common Uses: backup and temporary files, stored passwords for APIs

```
.gitignore
1
2 github/~/Introduction to Git & Github.pptx
3 github/Introduction to Git & Github.pptx
4 scripts/key.txt
5 github/.~.lock.Introduction to Git & GitHub.pptx#
6
```



SPECIAL FILE: README.md

- README.md (note the case!)
 - A text-based description of a directory or project
 - Uses Markdown language.
- Coming up next!

A screenshot of a GitHub README.md page for the Windows Calculator app. The page title is "Calculator". The content describes the app as a modern Windows app written in C++ that ships pre-installed with Windows, providing standard, scientific, and programmer calculator functionality, as well as converters between units of measurement and currencies. It notes that the app regularly receives updates and bug fixes, and links to the Microsoft Store for the latest version. At the bottom, there is a green "Azure Pipelines succeeded" status bar.

README.md

Calculator

The Windows Calculator app is a modern Windows app written in C++ that ships pre-installed with Windows. The app provides standard, scientific, and programmer calculator functionality, as well as a set of converters between various units of measurement and currencies.

Calculator ships regularly with new features and bug fixes. You can get the latest version of Calculator in the [Microsoft Store](#).

Azure Pipelines succeeded



MARKDOWN

- Simple formatting and text only
- Many dialects
- Used in a lot of places (Reddit)

Italic

Italic

Bold

Bold

Heading 1

=====

Heading 1

Heading 2

Heading 2

[Link][1]

:

[1]: <http://b.org>

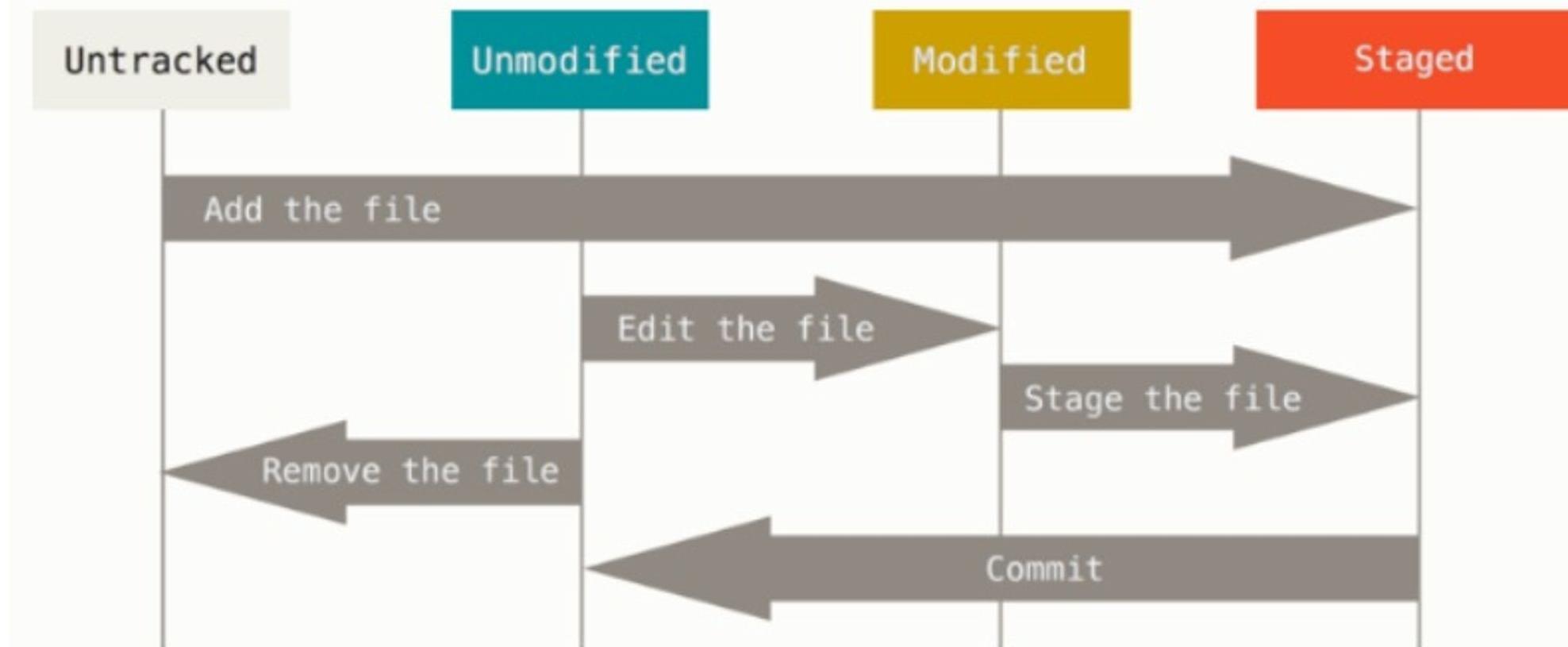
[Link](#)

```
# Calculator
The Windows Calculator app is a modern Windows app written in
C++ that ships pre-installed with Windows.
The app provides standard, scientific, and programmer
calculator functionality, as well as a set of converters
between various units of measurement and currencies.

Calculator ships regularly with new features and bug fixes.
You can get the latest version of Calculator in the \[Microsoft
Store\] (https://www.microsoft.com/store/apps/9WZDNCRFHVN5).
```

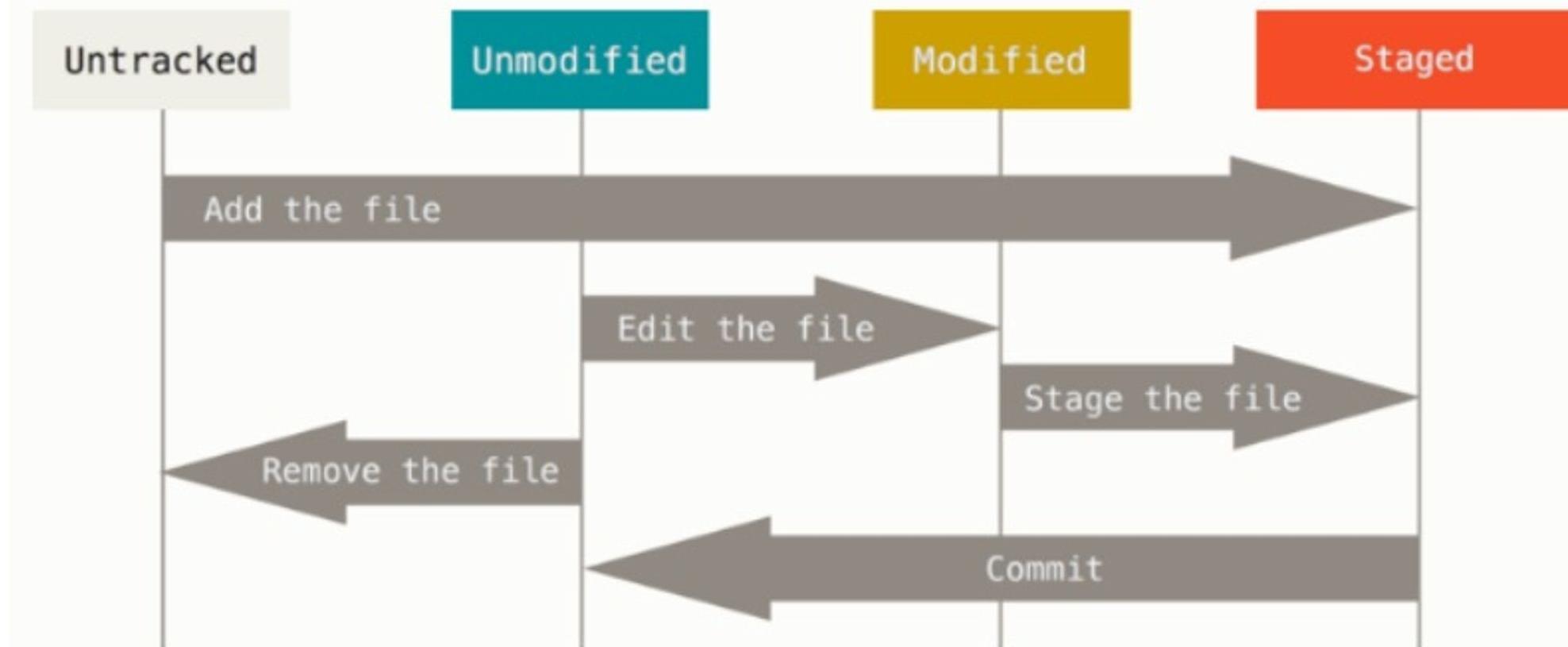


STATES OF FILES IN A REPOSITORY





STATES OF FILES IN A REPOSITORY



CREATING & CLONING REPOSITORIES

Initialization & Copying from a server





INITIALIZATION

- Create a repository, with a file index and database
- Stored in the hidden directory `.git`
- Note: No files are added to the repository by default!
 - We'll get to adding files a bit later

<input type="checkbox"/> Name	Date modified	Type
<input checked="" type="checkbox"/> .git	6/3/2019 11:11 AM	File folder
<input type="checkbox"/> .gitattributes	6/3/2019 11:11 AM	GITATTRIBUTES File



INITIALIZATION

- Create a repository, with a file index and database
- Stored in the hidden directory `.git`
- Note: No files are added to the repository by default!
 - We'll get to adding files a bit later

<input type="checkbox"/> Name	Date modified	Type
<input checked="" type="checkbox"/> .git	6/3/2019 11:11 AM	File folder
<input type="checkbox"/> .gitattributes	6/3/2019 11:11 AM	GITATTRIBUTES File



CLONING

Clone this repository from GitHub now!

<https://github.com/bakitybacon/rice-data-and-donuts-github>



CLONING

Clone this repository from GitHub now!

<https://github.com/bakitybacon/rice-data-and-donuts-github>



CLONING

Clone this repository from GitHub now!

<https://github.com/bakitybacon/rice-data-and-donuts-github>



CLONING

Clone this repository from GitHub now!

<https://github.com/bakitybacon/rice-data-and-donuts-github>

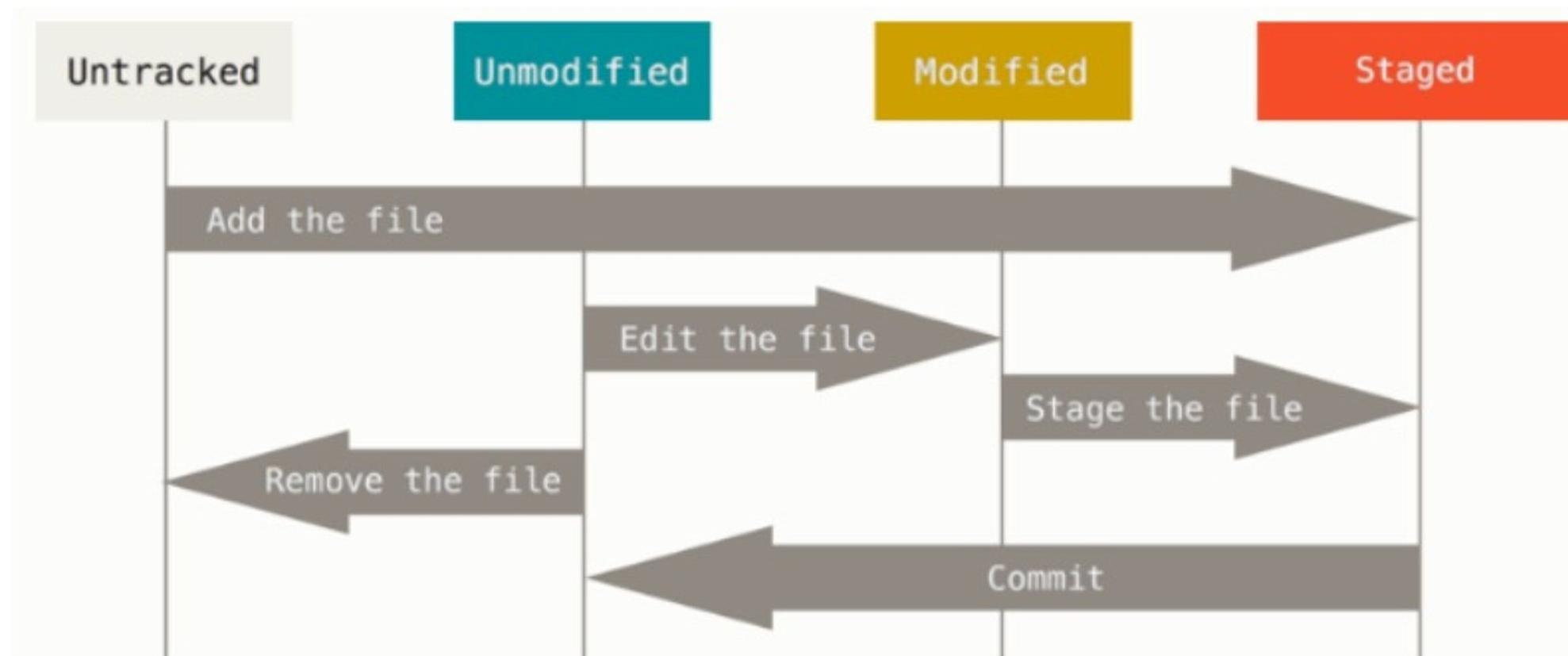
EDITING WORKFLOW

Editing, adding, staging, and committing





REVIEW OF FILE STATES





WORKFLOW

- **Working Directory**

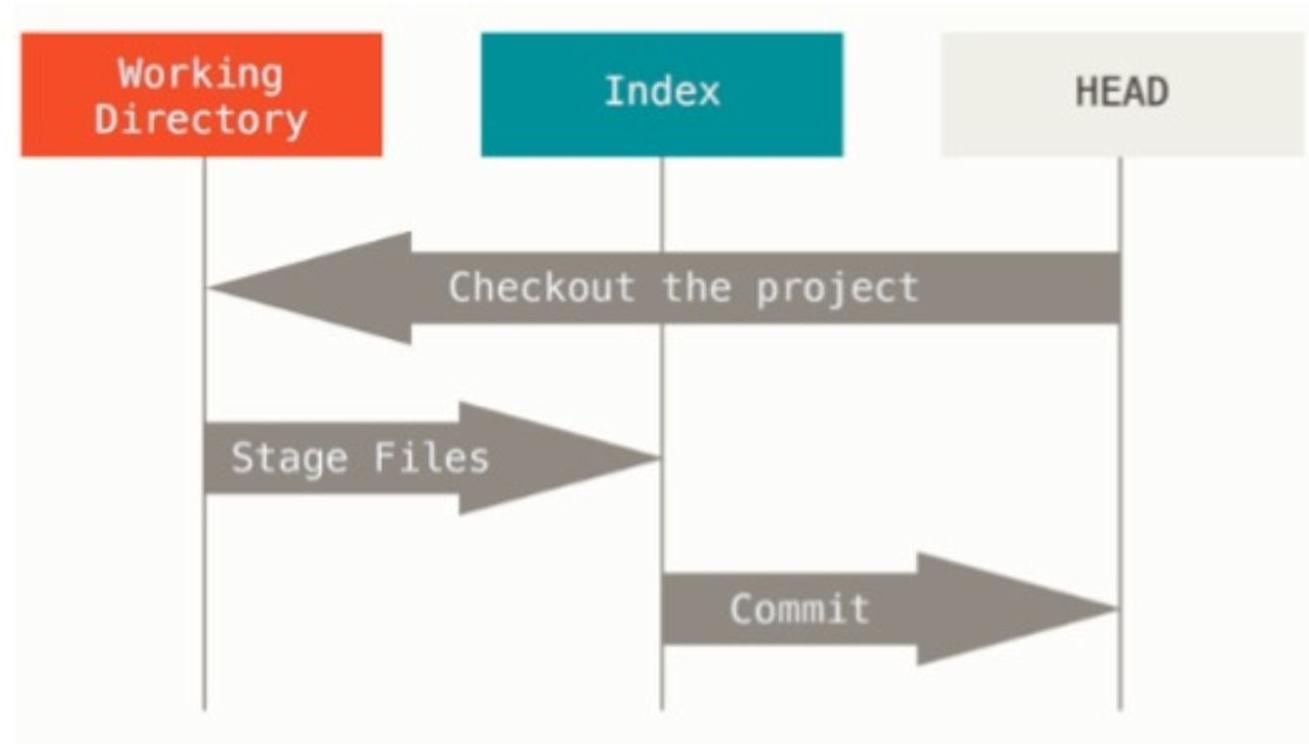
- Changed files on disk.

- **Index**

- We've told git to pay attention to these changes.

- **HEAD**

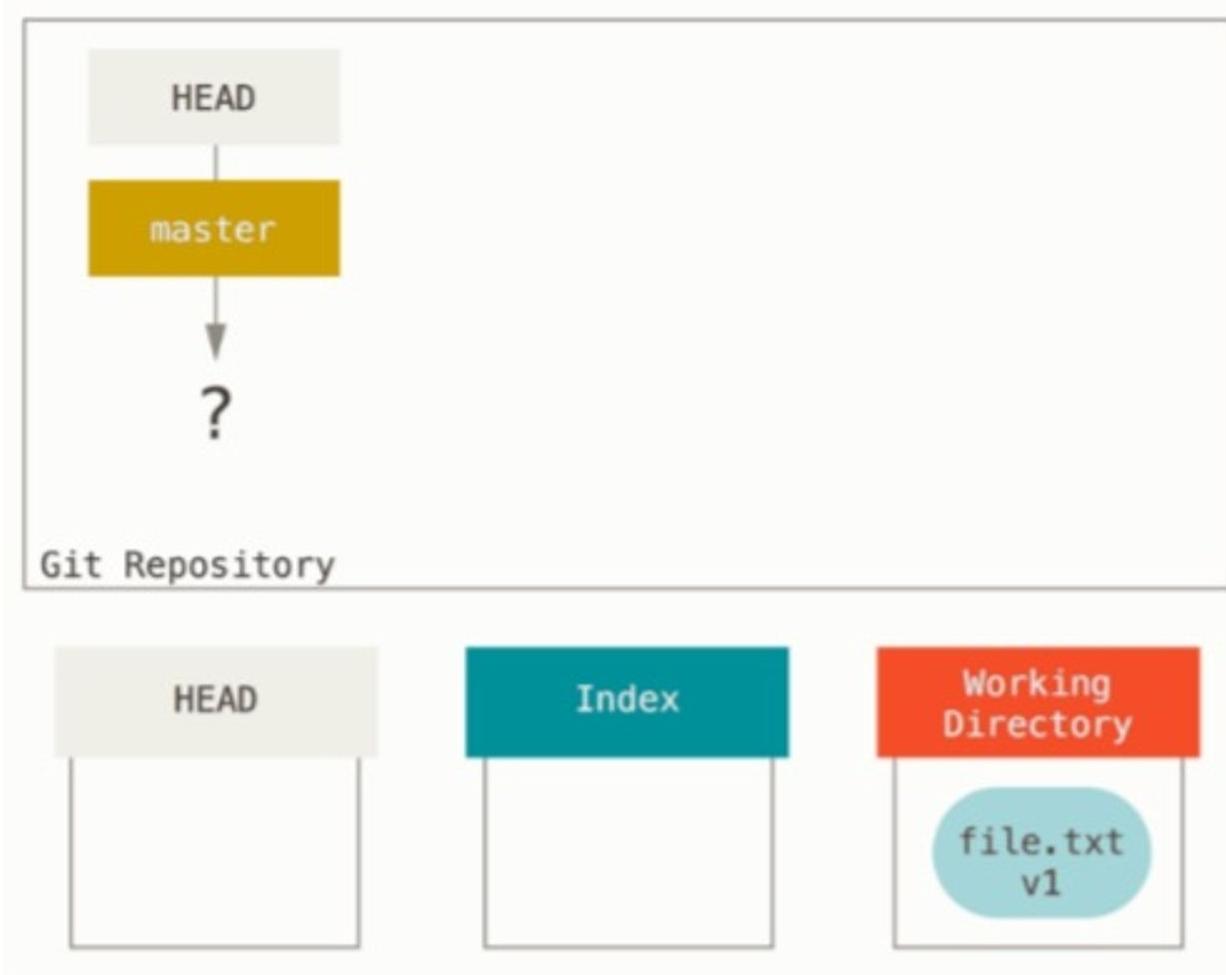
- This is just the most recent commit (snapshot). You can come back here whenever!





EDITING A FILE LOCALLY

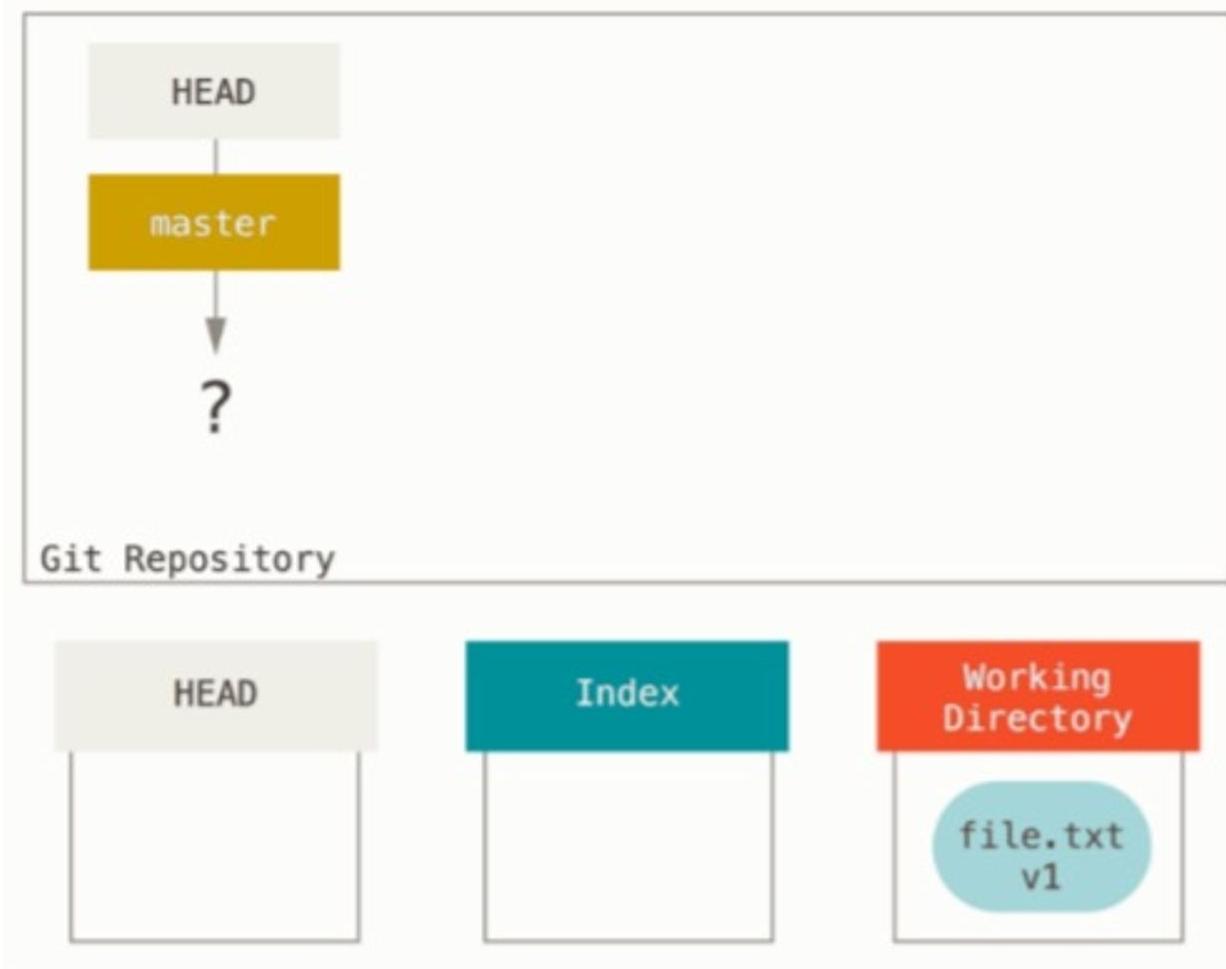
- Starting from an empty repository
 - How do we know?
 - **HEAD** doesn't point to anything (no snapshots!)
- We have edited a file in the **Working Directory**, but not yet told git to pay attention to it.





EDITING A FILE LOCALLY

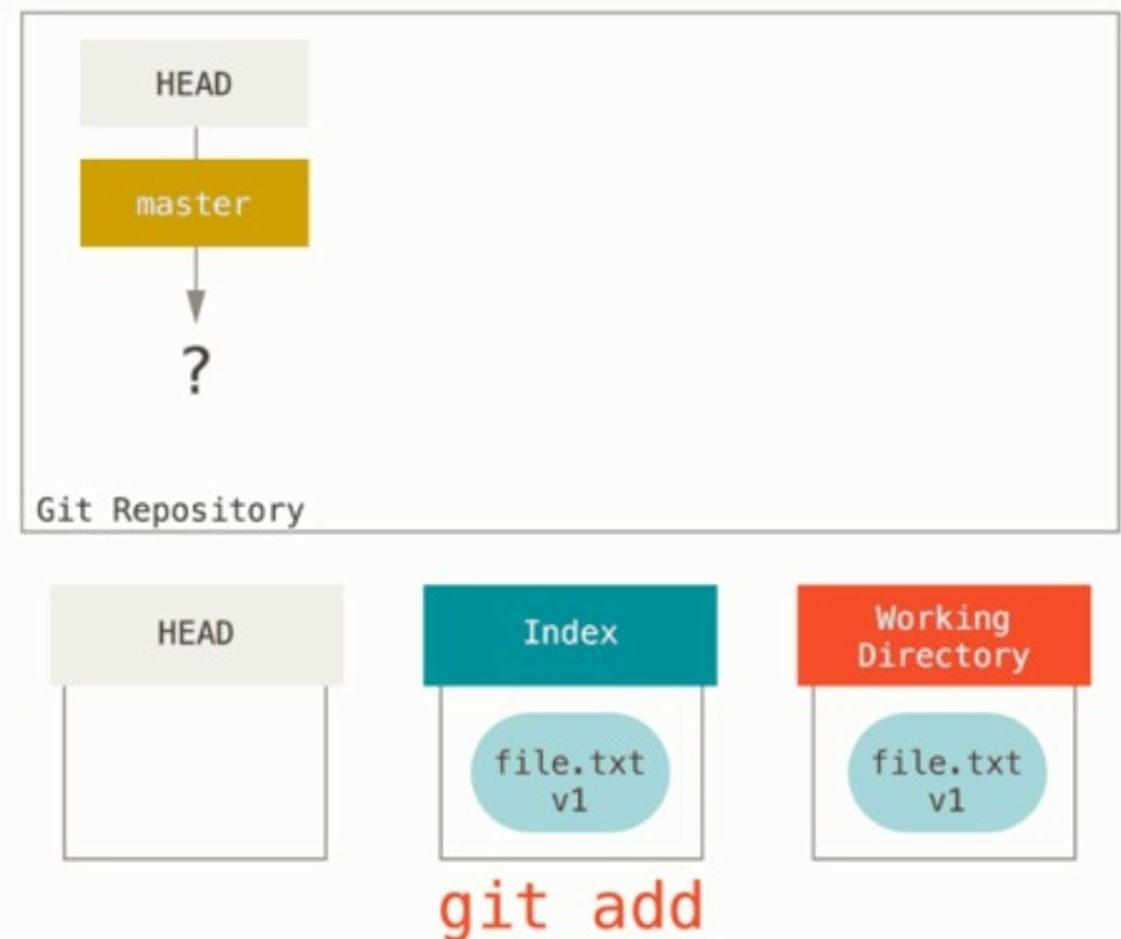
- Starting from an empty repository
 - How do we know?
 - **HEAD** doesn't point to anything (no snapshots!)
- We have edited a file in the **Working Directory**, but not yet told git to pay attention to it.





TELLING GIT ABOUT OUR CHANGES

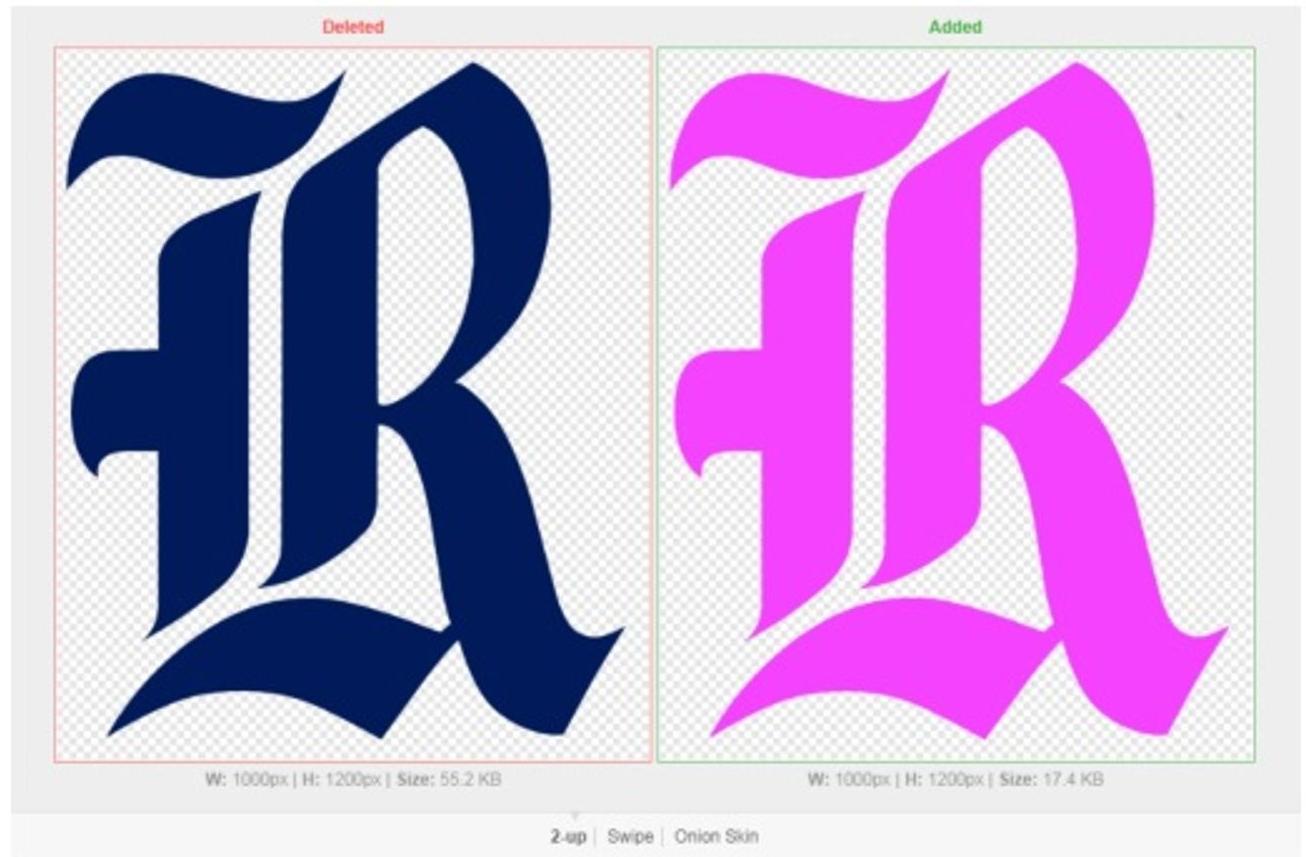
- Now that we have made file.txt, we can tell git to pay attention to the file
 - Transition between which two states for file.txt?
 - Command: `git add`
- Whatever we have added will be changed in the new snapshot. No add means it won't change!





VIEWING CHANGES: IMAGES

- 2-Up: side-by-side
- Swipe: left of divider is first image, right is second image
- Onion Peel: images are overlaid, slider changes transparency setting



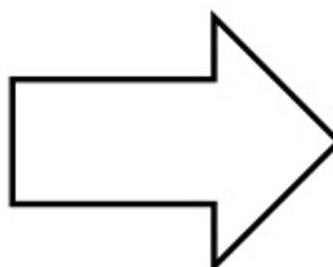


VIEWING CHANGES: DOCUMENTS

- Prose Diff: see word-by-word changes in formatting

```
... @@ -1,5 +1,9 @@
1 -# Rice Data and Donuts
2
3 Welcome to the GitHub workshop!
4
5 We're __so__ glad you could join us today.

1 +# Rice Data & Donuts
2
3 Welcome to the GitHub workshop!
4
5 We're __so__ glad you could join us today.
6 +
7 +We hope to see you again soon!
8 +
9 +:]!
```



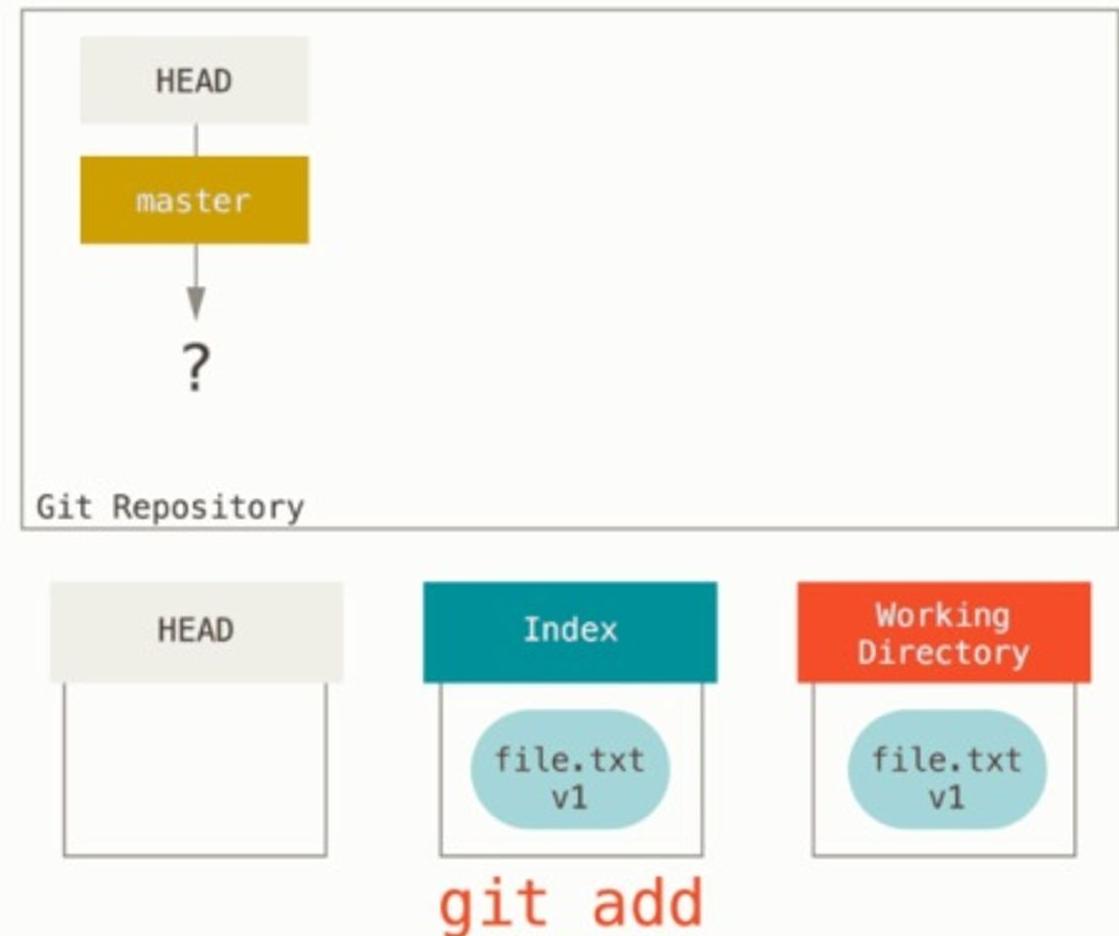
```
6 diff/sample.md
Rice Data-and & Donuts

Welcome to the GitHub workshop!
We're so glad you could join us today.
We hope to see you again soon!
:]!
```



TELLING GIT ABOUT OUR CHANGES

- Now that we have made file.txt, we can tell git to pay attention to the file
 - Transition between which two states for file.txt?
 - Command: `git add`
- Whatever we have added will be changed in the new snapshot. No add means it won't change!





SNAPSHOTS OF OUR CHANGES

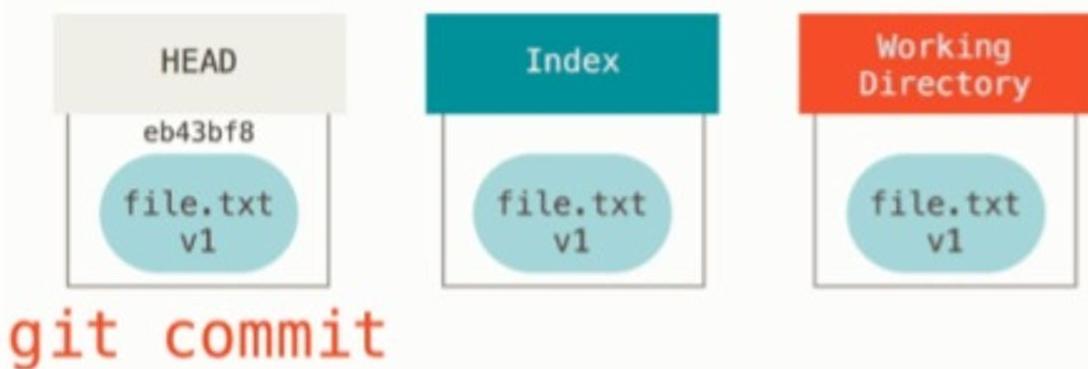
- Tell git to take a snapshot of our files!
 - Transition between which two states for file.txt?
 - Command: `git commit`
- **HEAD** now points to our most recent snapshot!





SNAPSHOTS OF OUR CHANGES

- Tell git to take a snapshot of our files!
 - Transition between which two states for file.txt?
 - Command: `git commit`
- **HEAD** now points to our most recent snapshot!





UPLOADING OUR CHANGES

- From the repository cloned earlier, upload a file with any message you like!



UPLOADING OUR CHANGES

- From the repository cloned earlier, upload a file with any message you like!



GOING BACK IN TIME

- Revert your message, then commit something different and push again!



GOING BACK IN TIME

- Revert your message, then commit something different and push again!



MAKING COMMENTS

- Make a comment on either the revision or the original commit you made!



MAKING COMMENTS

- Make a comment on either the revision or the original commit you made!

BRANCHES & MERGING

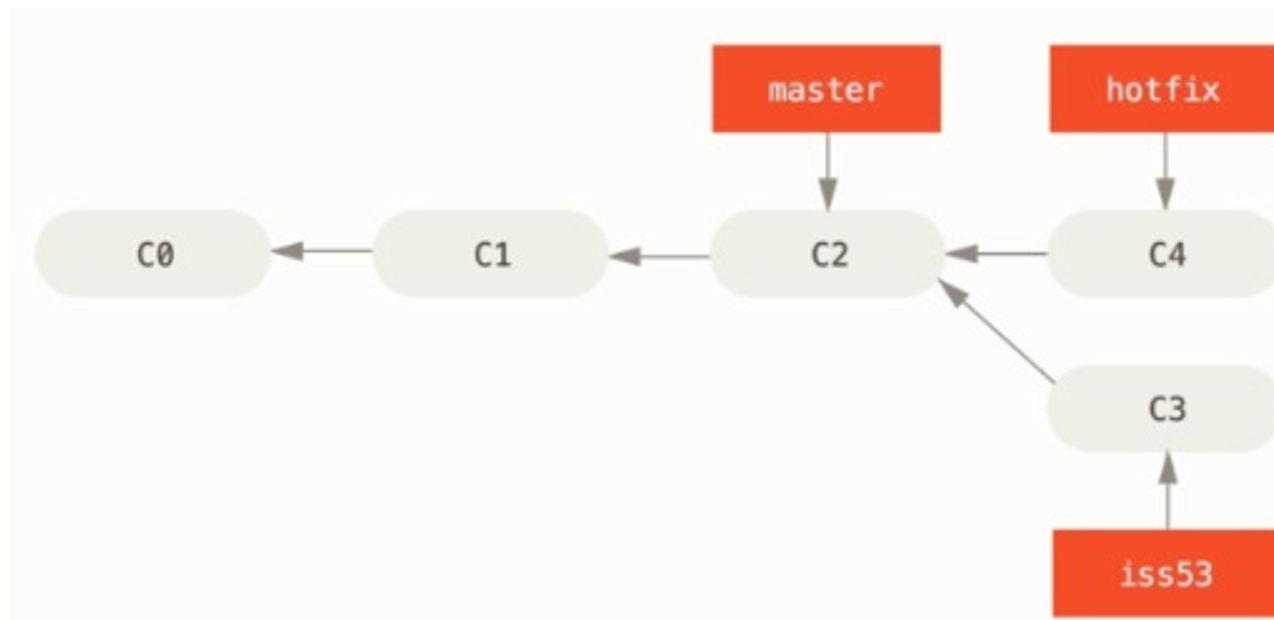
Creating, working on, combining, and switching branches





BRANCHES

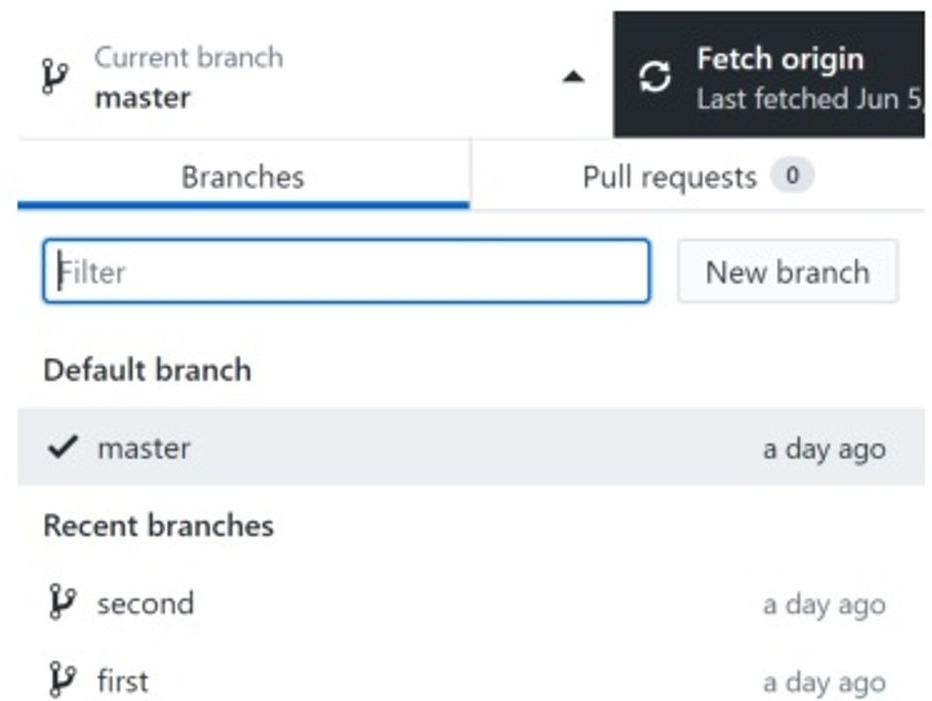
- If you're working on different features or as part of a larger team
- Divergent history
- Independent files
 - Avoids unintended consequences!
- Easy integration





CREATING & SWITCHING BRANCHES

- Click on the “Current branch” tab
- If a branch isn’t listed, you may have to search for it
- Make sure to publish the branch to GitHub!



The screenshot shows a GitHub repository interface. At the top, there is a navigation bar with tabs for 'Current branch' (which is 'master'), 'Fetch origin' (last fetched Jun 5), and other repository details. Below the navigation bar, there are two tabs: 'Branches' (which is selected and highlighted in blue) and 'Pull requests' (0). A 'Filter' input field and a 'New branch' button are also present. The main content area shows a list of branches under the heading 'Default branch'. The 'master' branch is selected, indicated by a checked checkbox and a timestamp 'a day ago'. Below this, there is a section for 'Recent branches' with entries for 'second' and 'first', both also timestamped 'a day ago'. A large black callout box with a white arrow points to the 'Publish branch' button.

Branch	Last Published
master	a day ago
second	a day ago
first	a day ago

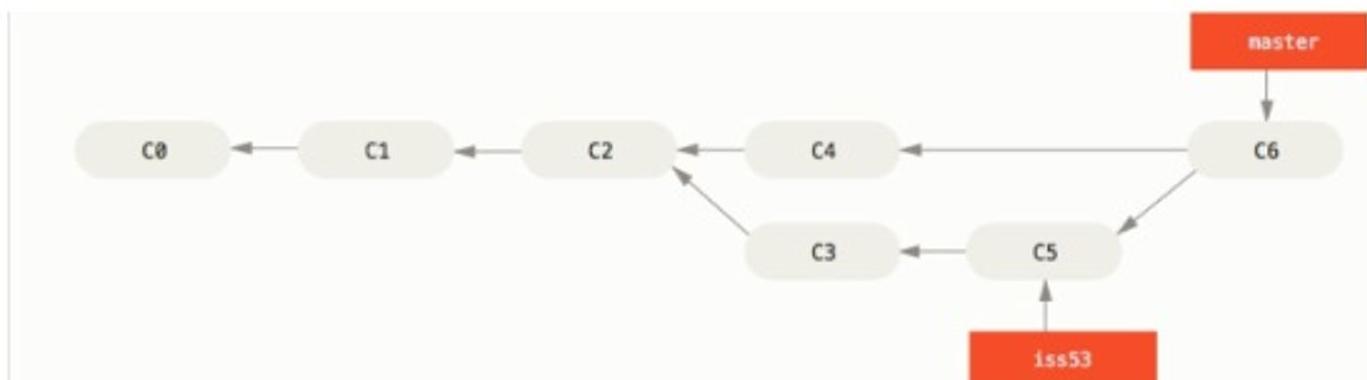
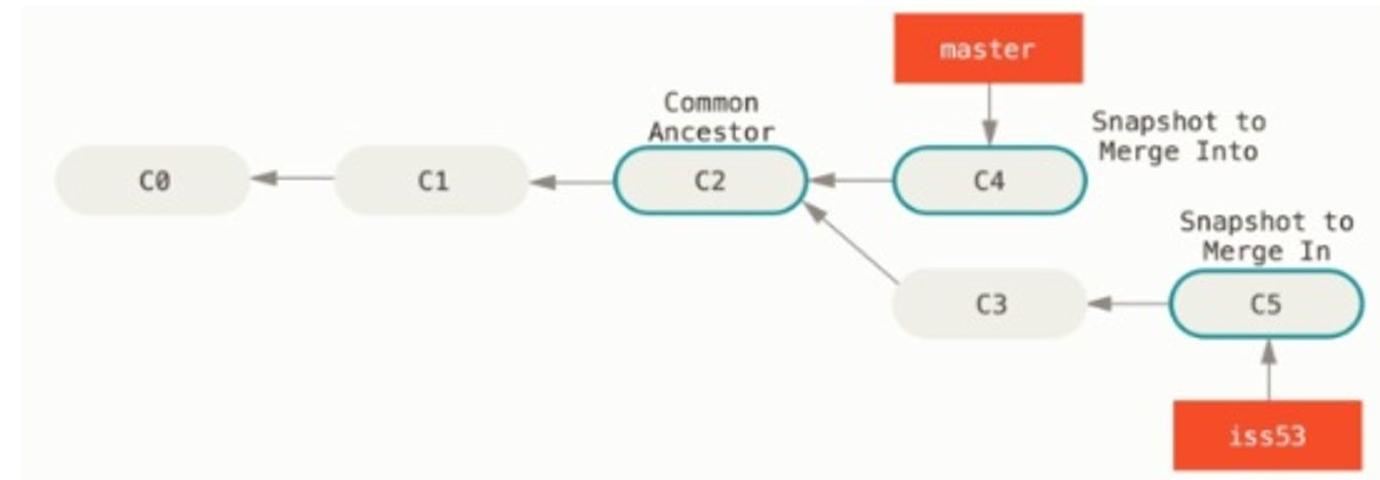


Publish branch

Publish this branch to GitHub

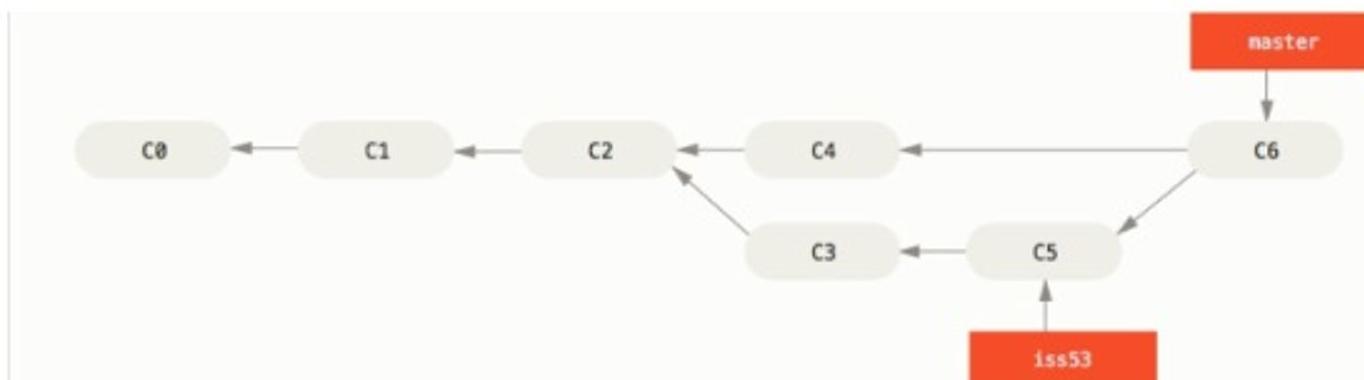
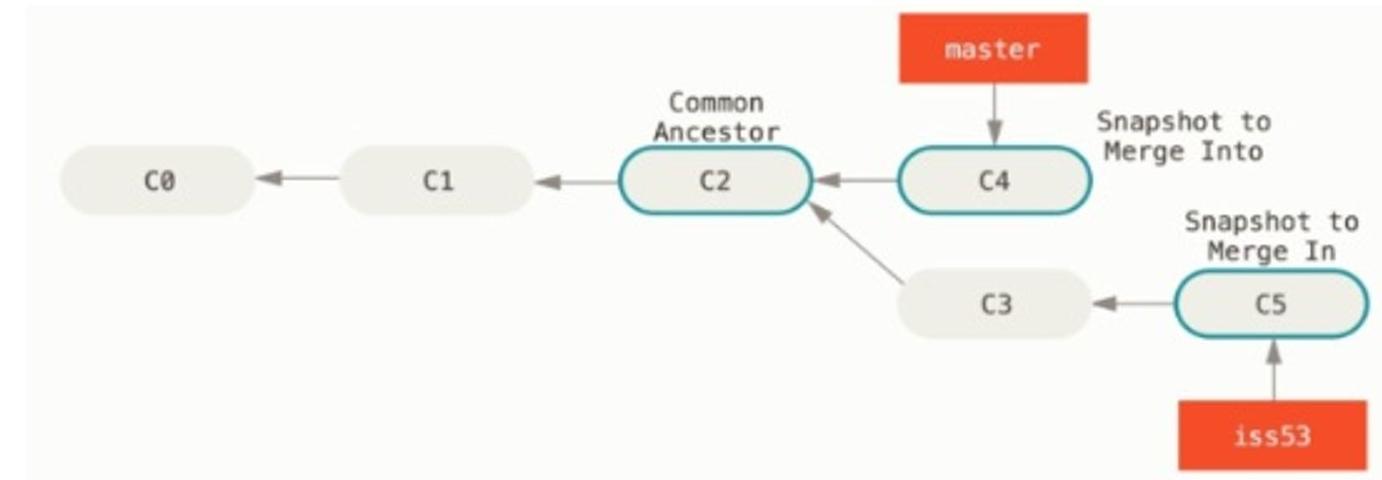


MERGING



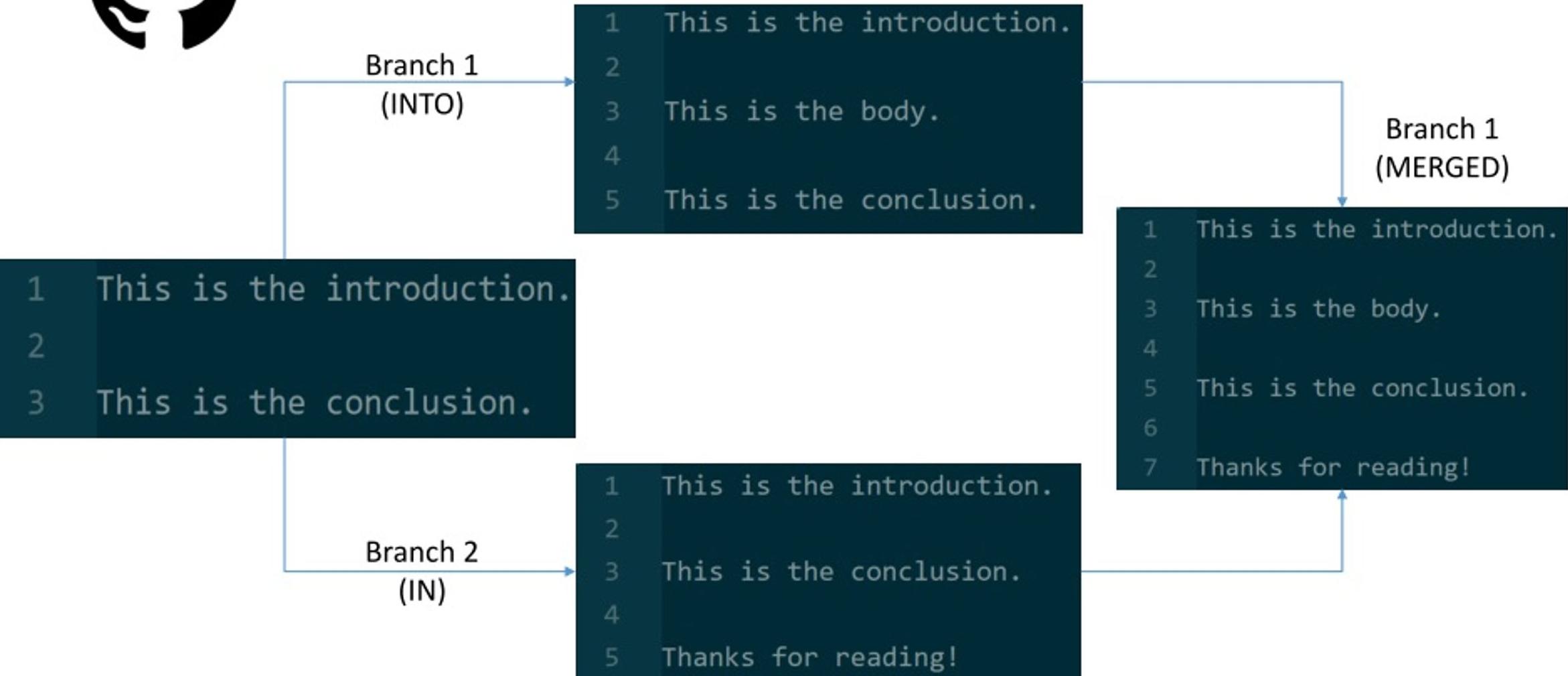


MERGING





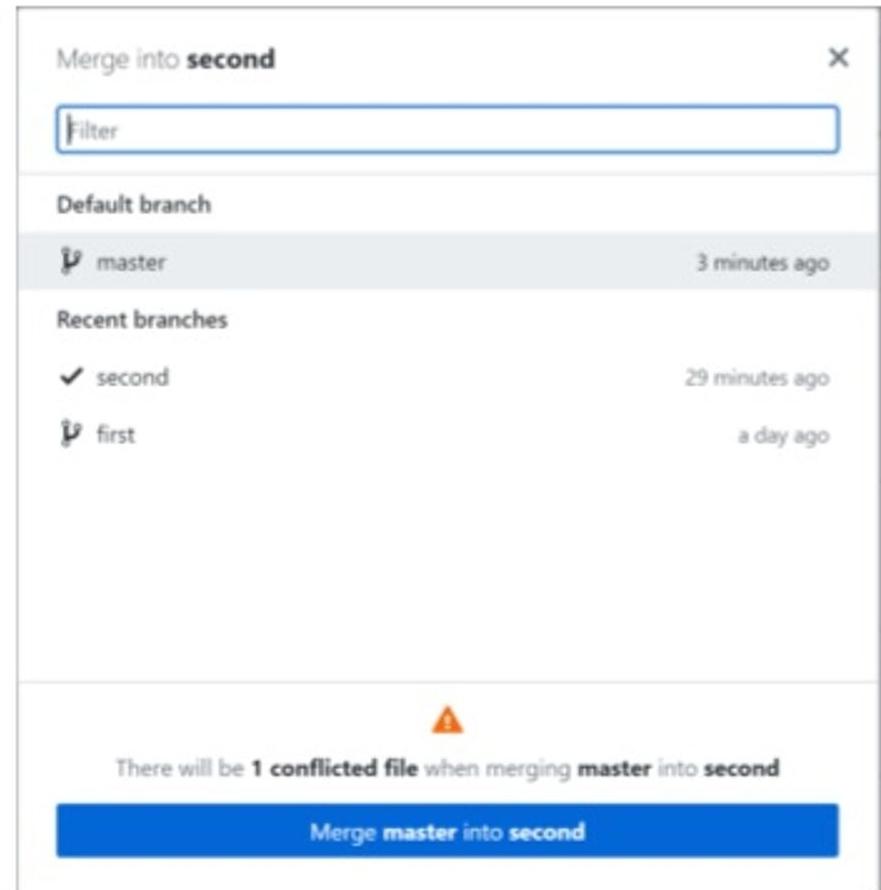
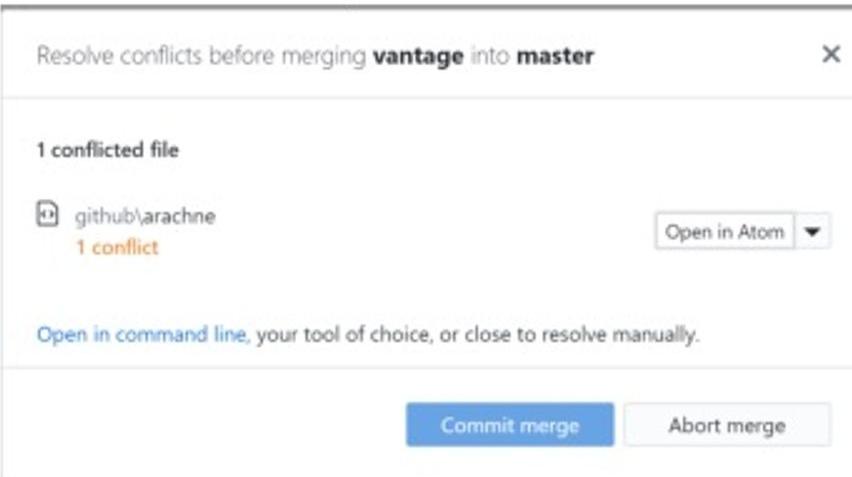
MERGING IN ACTION





CONFLICTS

- Occasionally, two people may have edited the same part of the same file on different branches
- Which version do we take?
- Must manually resolve conflicts

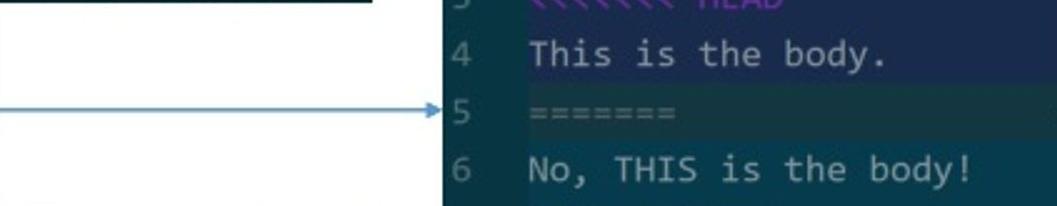




CONFLICTS IN ACTION

```
1 This is the introduction.  
2  
3 This is the body.  
4  
5 This is the conclusion.
```

```
1 This is the introduction.  
2  
3 No, THIS is the body!  
4  
5 This is the conclusion.
```



```
1 This is the introduction.  
2  
3 <<<<< HEAD  
4 This is the body.  
5 =====  
6 No, THIS is the body!  
7 >>>>> veck  
8  
9 This is the conclusion.
```

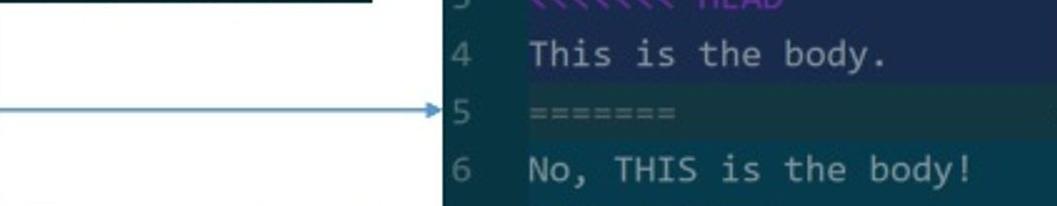
```
1 This is the introduction.  
2  
3 These are both the body.  
4  
5 This is the conclusion.
```



CONFLICTS IN ACTION

```
1 This is the introduction.  
2  
3 This is the body.  
4  
5 This is the conclusion.
```

```
1 This is the introduction.  
2  
3 No, THIS is the body!  
4  
5 This is the conclusion.
```



```
1 This is the introduction.  
2  
3 <<<<< HEAD  
4 This is the body.  
5 =====  
6 No, THIS is the body!  
7 >>>>> veck  
8  
9 This is the conclusion.
```

```
1 This is the introduction.  
2  
3 These are both the body.  
4  
5 This is the conclusion.
```



CONFLICTS AND MERGING

- Create a new branch.
- Edit `community.txt` to include only a one-line message.
- Try to merge it.
- If it fails, merge by combining the two messages. Don't delete anyone else's work!

COLLABORATING ON GITHUB



Extensions to collaborate within teams, organizations, and more.



GITHUB EXTENSIONS TO MERGING

- Fork <https://github.com/bakitybacon/rice-data-and-donuts-github>
- Edit community.txt one more time
- Make a pull request and comment on it!



GITHUB EXTENSIONS TO MERGING

- Fork <https://github.com/bakitybacon/rice-data-and-donuts-github>
- Edit community.txt one more time
- Make a pull request and comment on it!



GISTS

- Snippets of code, text, or anything.
- Linked to your account, not a repository
- Useful if you only have one file!
- Shareable by URL (even in secret mode!)

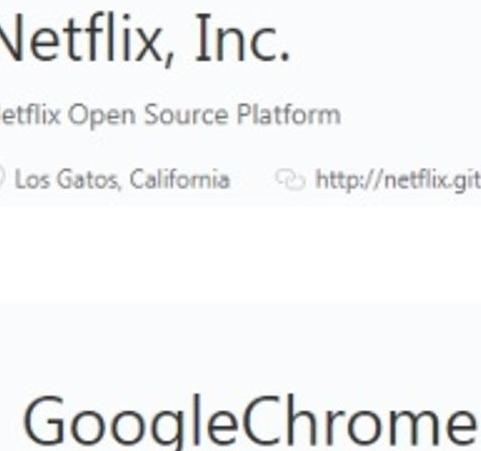
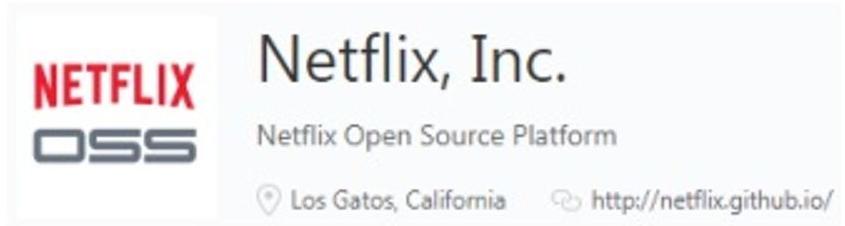
 [grej / enumerate_letters.py](#)
Created 30 seconds ago

```
1  def enumerate_letters(count):  
2      """  
3          Generates a list of unique capital letter strings accordingly  
4          eg. ['A', 'B', ... , 'AA', 'BB', 'CC', ...]  
5      """  
6      :param df: number of unique letter strings to return  
7      :returns: list <str>: list of capital letter strings  
8      """  
9  
10     lettes_in_alphabet = 26
```



ORGANIZATIONS

- In GitHub, a group of users and projects
- Organizations can make repositories
- Organization owner can manage permissions for all users





TEAMS

- Subset of an organization
- Generally given an area of expertise (such as design) or a task

 [frc461/robot](#)

 [frc461/website](#)



ISSUES

- Pointing out what's wrong
- Hope someone can figure it out!
 - Even if they're not the owner of the repository!
 - The benefit of open source
- Great way to get involved
- Supports discussions, fixes, etc

Compute metrics on "slices" of the data?
#547 opened 2 days ago by jayleverett



Want to contribute to pytorch/ignite?

If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue.
If you're ready to tackle some open issues, we've collected [some good first issues](#) for you.



RELEASES

- Basically just a tag for a commit
- Can write a description
 - Markdown, again!
- Link files (pdf of report, for example)

0.23.2
F24c958

0.23.2

 bazel-io released this on Mar 11 · 2058 commits to master since this release

Cherry picks:

- [6ca7763](#) : Fix a typo
- [2310b1c](#) : Ignore SIGCHLD in test setup script
- [f9eb1b5](#) : Complete channel initialization in the event loop
- [f8a1597](#) : remote: properly reset state when using remote cache. Fixes [#7555](#)
- [56366ee](#) : Set non-empty values for msvc_env_* when VC not installed

Notice: Bazel installers contain binaries licensed under the GPLv2 with Classpath exception. Those installers should always be redistributed along with the source code.

Some versions of Bazel contain a bundled version of OpenJDK. The license of the bundled OpenJDK and other open-source components can be displayed by running the command `bazel license`. The vendor and version information of the bundled OpenJDK can be displayed by running the command `bazel info java-runtime`. The binaries and source-code of the bundled OpenJDK can be [downloaded from our mirror server](#).

Security: All our binaries are signed with our [public key 48457EE0](#).



PROJECTS

- Tracking things to do
 - Individual or organizational scale
- Can have issues, pull requests
- Small informational cards

The image displays three separate GitHub Projects boards side-by-side. Each board is a vertical list of items, each represented by a card.

- Upstream issues to track**:
 - Card 1: <https://github.com/git-lfs/git-lfs/issues/2627> (Add card, Cancel)
 - Card 2: Git LFS 2.3.1 seems to break Windows (#2627 opened by larsxschneider)
 - Card 3: docker build limit io disk (#36012 opened by sztworok)
 - Card 4: repl: allow 'await' in REPL (#13209 opened by benjamingr)
- New things to check out**:
 - Card 1: Implement split diffs (#866 opened by BinaryMuse, work-in-progress)
 - Card 2: Change license and remove references to PATENTS (#10804 opened by sophiebits, CLA Signed)
 - Card 3: "Clone in Desktop" flow now recognizes gists (#2939 opened by shiftkey, ready-for-review)
- Fixes to upgrade for**:
 - Card 1: #3311 opened by kdzwinel (audit)
 - Card 2: Error: Undefined variable: "\$ht-size-mobile" (#229 opened by kaelig)
 - Card 3: util: use faster -O check (#16726 opened by mscode, performance, util)
 - Card 4: Git LFS 2.3.1 seems to break Windows (#2627 opened by larsxschneider)

DISTRIBUTED WORK

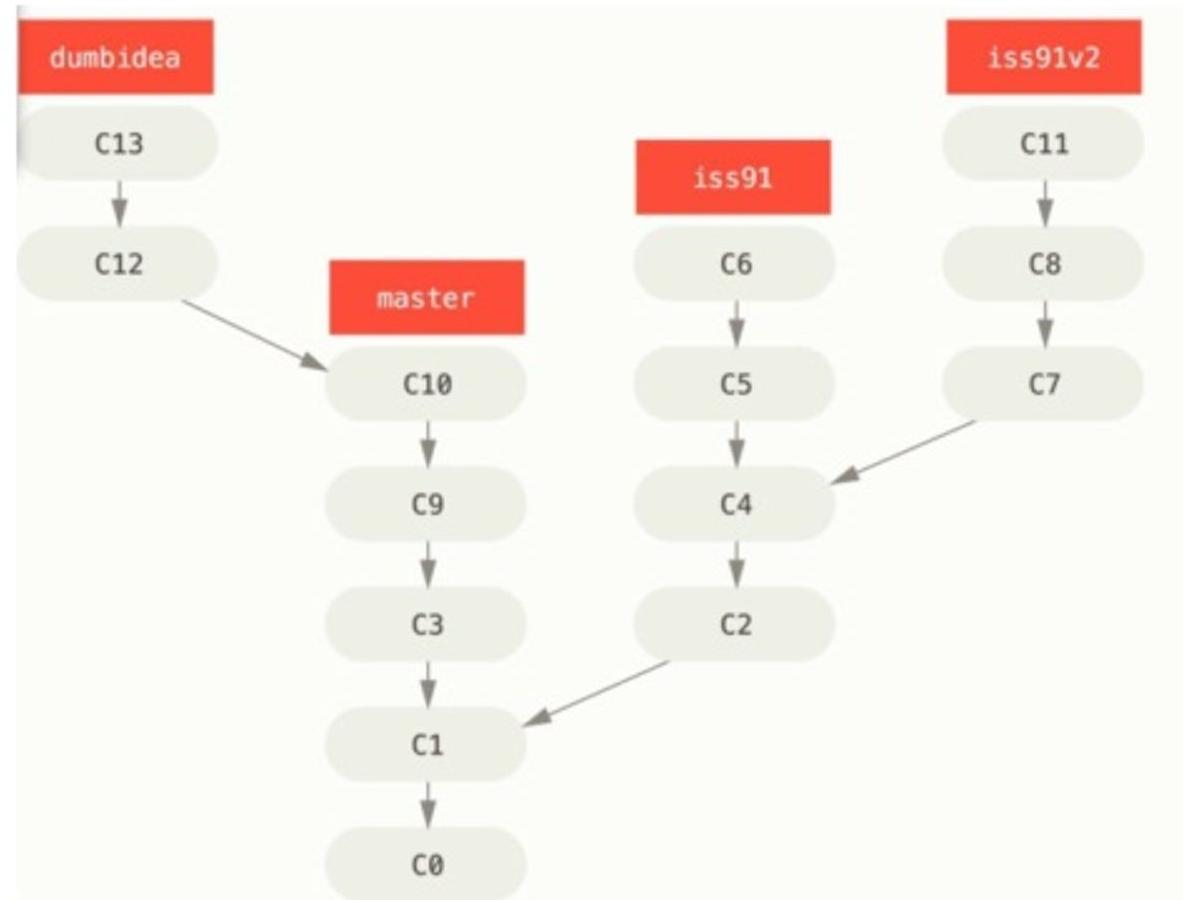
Paradigms for getting the most out of working as a team





BRANCHING WORKFLOWS

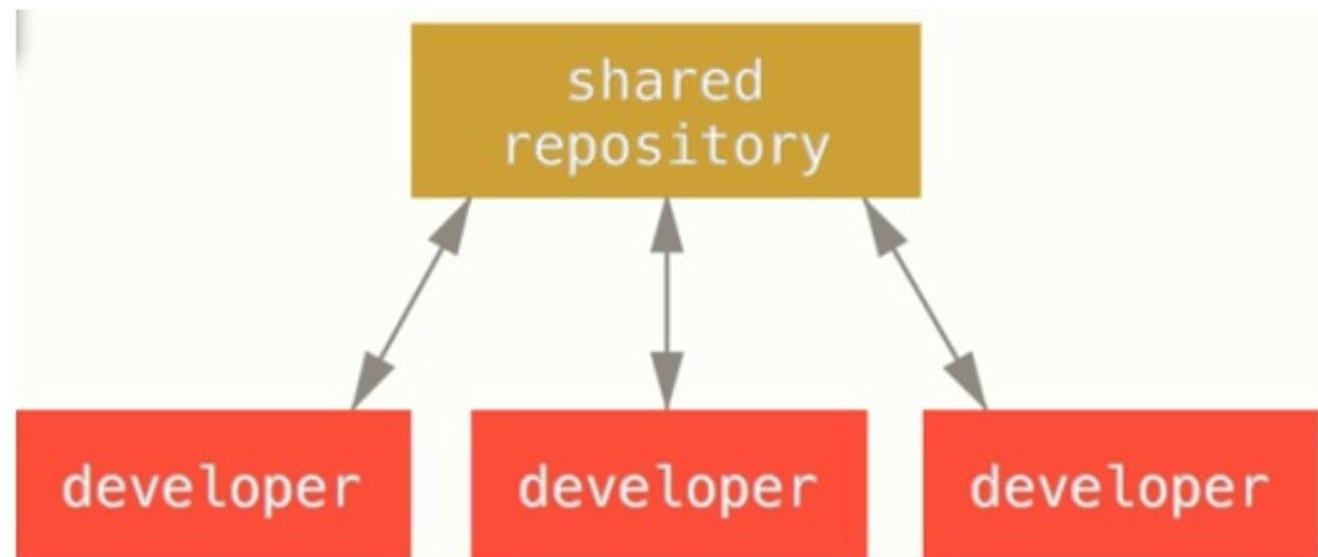
- + Not technically distributed
- + Try a lot of things out on separate branches & merge!
- + Can work on many problems at the same time





CENTRALIZED WORKFLOW

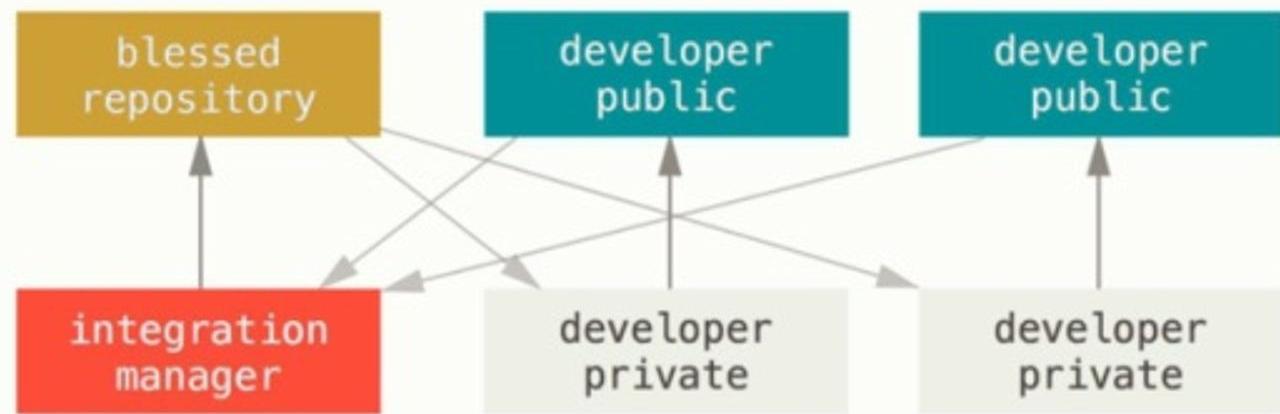
- + Everyone is equal
- + Leverages branches
- + Everyone must merge their own work!
 - + What if you don't like how someone else merged?
- + No overhead from dedicated integration manager





INTEGRATION MANAGER WORKFLOW

- + Developers have copies of the “blessed” repository
- + They make changes, then ask integration manager to incorporate them
- + More overhead
- + More accountability





INTEGRATION MANAGER WORKFLOW IN GITHUB

- + You can see this workflow in GitHub all the time
- + Fork/Merge & Pull Request:
 - + Make a copy of a repository
 - + Make a change
 - + Ask the original repository's owner to “pull” your change into his project

Hwk4 skeleton #7

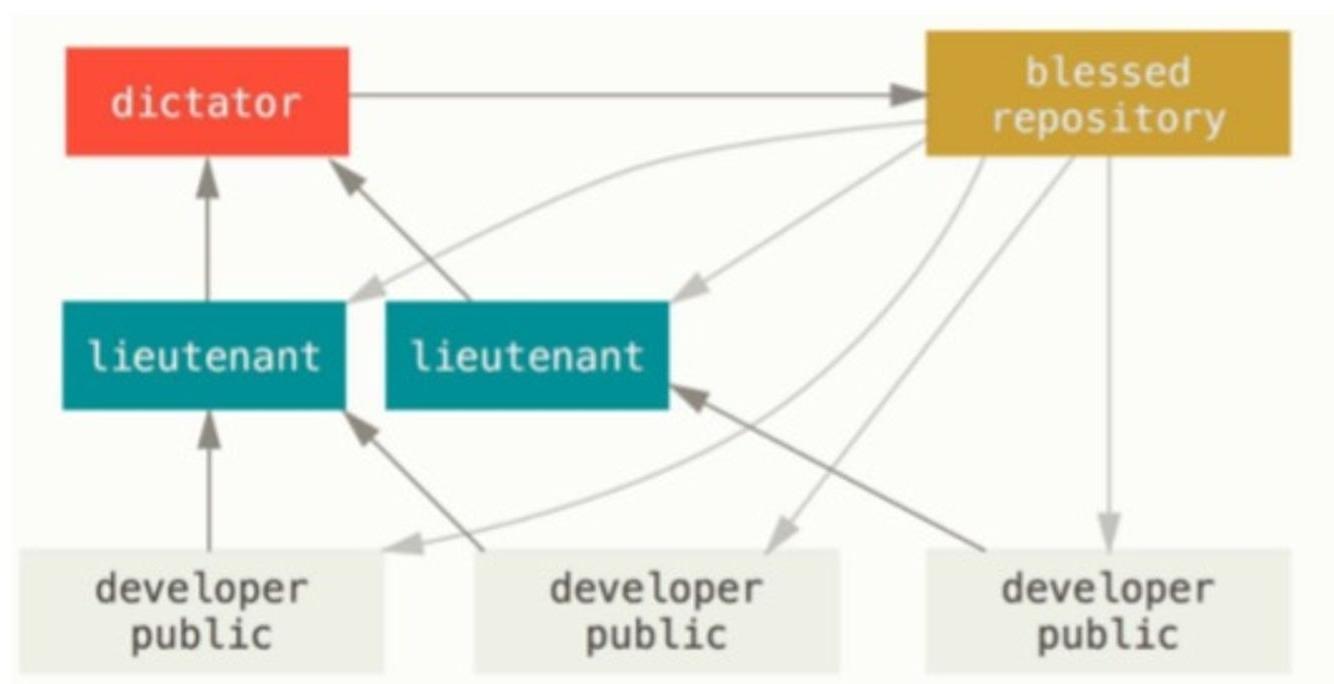


bakitybacon wants to merge 17 commits into `master` from `hwk4_skeleton`



LIEUTENANT & DICTATOR WORKFLOW

- + Essentially the same as integration manager with one more layer to go through
- + More overhead, but can use expertise effectively
 - + Lieutenant 1 expert in programming, Lieutenant 2 in design, for example





ACKNOWLEDGMENTS

- The figures are taken from Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, used under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
- The GitHub Mark and Octocat are used under their official license (<https://github.com/logos>)
- The Git logo is used under the Creative Commons Attribution 3.0 Unported License.
- Screenshots are taken from GitHub Desktop and Atom.



ABOUT RESEARCH DATA SERVICES

- Data @ Rice Workshops on Python, R, Excel, and many more
- Consulting on managing data, acquiring data, use of R, Python, SPSS, Excel, and more
- Review draft data management plans
- Teach custom workshops on request
- **Data Office Hours**
 - Prefer a one-on-one consultation? Drop by our office hours [3:00 p.m.-4:00 p.m. on Tuesdays](#) or [12:30-1:30 p.m. on Thursdays](#) in the GIS-Data Center classroom (basement of Fondren Library), or schedule an appointment at <https://library.rice.edu/data-assistance>.