



A Git Primer

...

git gud

Agenda

- **What is git and github?**
- **Why you need to know it**
- **Basic commands you will see very soon, and very often**
- **What a fork is!**

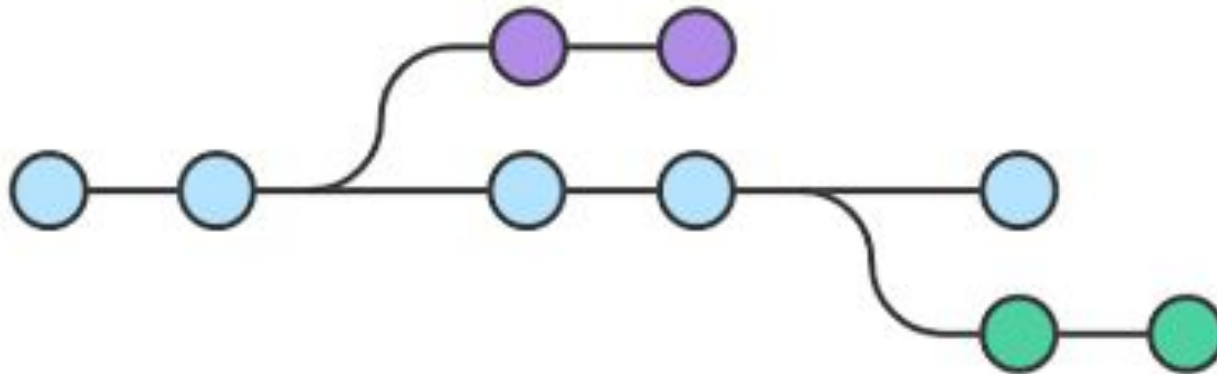
What is git and github?

- Git is a version control system!
- Github is a hosting platform for software development



Why you need to know git

- EVERY company uses some version/source control software or another - git is by far the most popular



COMMANDS WE TEACH YOU IN THE PRECOURSE

- **git clone <URL>** - clones a repository into your local machine
- **git status** - shows status of your repo, which files have been changed
- **git add <filename(s)>** - adds a specific file/files whose changes will be updated to the repo upon commit
- **git add .** - adds ALL changed files
- **git commit -m"<message>"** - saves added changes to the repo
- **git push** - updates a remote repository with local changes
- **git push <remote> <branch>** - push explicitly to a remote link and a specific branch
- **git remote -v** - view remote links. Will look like: origin <url> (fetch)
origin <url> (push)

COMMANDS TO KNOW

- **git branch** - *checks the current branches on your machine*
- **git checkout <branch name>** - *moves you to a different branch*
- **git checkout -b"<new branch name>"** - *creates AND moves you to a new branch*
- **git remote add <remote name> <URL>** - *creates a new link to a remote repository*
- **git pull** - *updates your current local branch with changes from a corresponding remote repository (this is a combination of git fetch + git merge)*
- **git pull <remote> <branch>** - *explicitly define which remote link and which branch you are grabbing from*

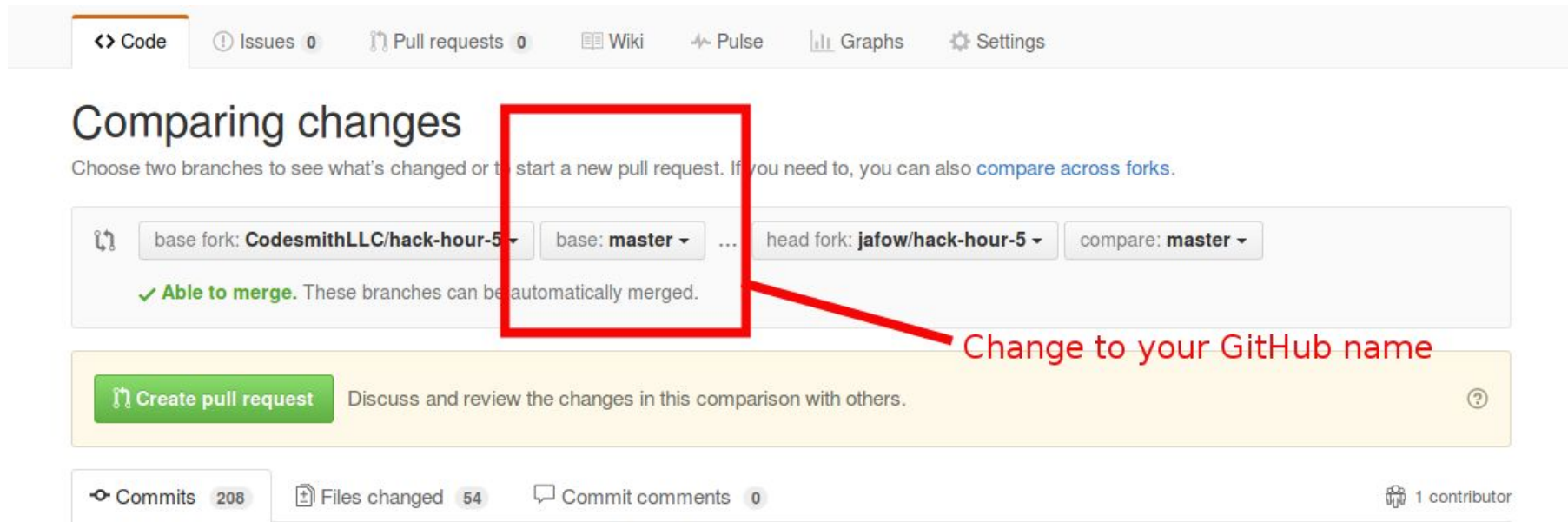
Demo time



WHEN IN DOUBT!

- **git status** - *shows status of your repo, which files have been changed*
- **git branch** - *checks the current branches on your machine*
- **git remote -v** - *view remote links. Will look like: origin <url> (fetch)
origin <url> (push)*
- **git log** - *view previous commit history*

Pull requests



The screenshot shows the GitHub 'Comparing changes' interface. At the top, there's a navigation bar with tabs for Code, Issues (0), Pull requests (0), Wiki, Pulse, Graphs, and Settings. Below this, the title 'Comparing changes' is followed by a subtitle: 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).' The main comparison area has two dropdown menus: 'base fork: CodesmithLLC/hack-hour-5' and 'base: master'. These are enclosed in a red box. To the right, there's a 'head fork: jafow/hack-hour-5' and a 'compare: master' dropdown. Below the dropdowns, a green checkmark indicates 'Able to merge. These branches can be automatically merged.' A red arrow points from the text 'Change to your GitHub name' to the 'base fork' dropdown menu. At the bottom, there's a green button labeled 'Create pull request' and a yellow box with the text 'Discuss and review the changes in this comparison with others.' The footer shows 'Commits 208', 'Files changed 54', 'Commit comments 0', and '1 contributor'.

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: CodesmithLLC/hack-hour-5 base: master ... head fork: jafow/hack-hour-5 compare: master

✓ Able to merge. These branches can be automatically merged.

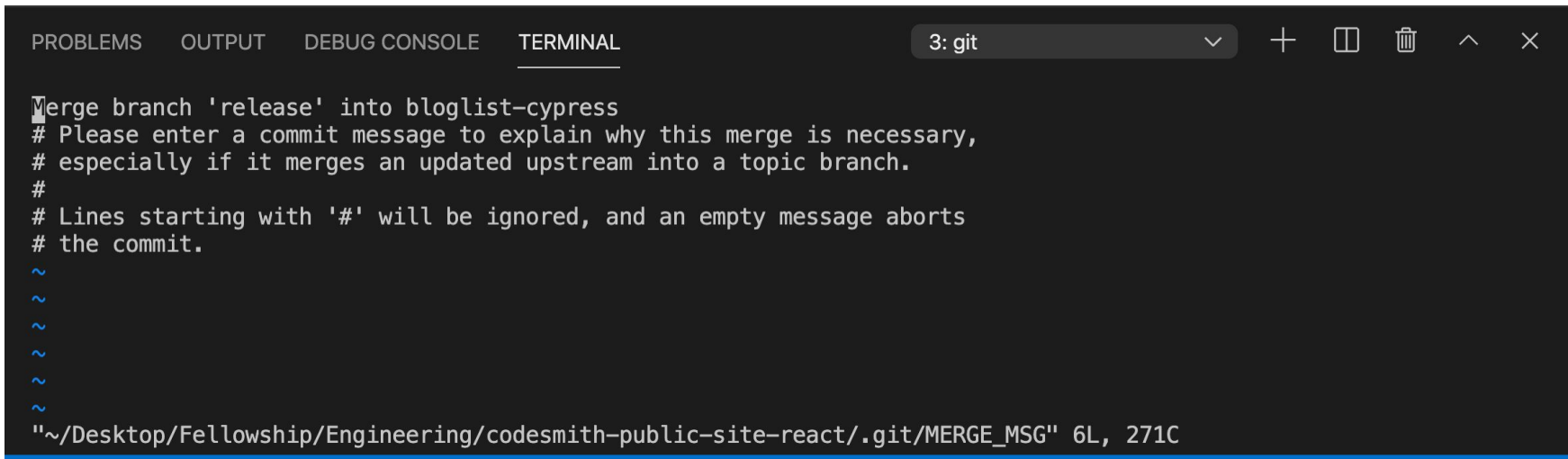
Create pull request Discuss and review the changes in this comparison with others.

Commits 208 Files changed 54 Commit comments 0 1 contributor

Change to your GitHub name

Notes on git pull

You may find yourself facing a vim editor like this



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. To the right of the tabs, there is a dropdown menu showing '3: git' and several icons for window management. The terminal content shows a git merge prompt in vim editor. The text is as follows:

```
Merge branch 'release' into bloglist-cypress
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~/Desktop/Fellowship/Engineering/codesmith-public-site-react/.git/MERGE_MSG" 6L, 271C
```

Notes on git pull

Just write :q or :qa in the terminal and then press ENTER!

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
3: git
Merge branch 'release' into bloglist-cypress
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
```

What the fork?

- What's the difference between a fork and a branch?
 - *A fork is an ENTIRELY NEW repository, based off a parent*
 - *A branch is a temporary workspace, meant to be deleted once the additions have been merged into a main branch*

<https://www.pluralsight.com/blog/software-development/the-definitive-guide-to-forks-and-branches-in-git>



Additional resources

- Git documentation: <https://git-scm.com/doc>
- Github's git guide: <https://github.com/git-guides>
- Github's docs: <https://docs.github.com/en>
- Nodeschool git workshop: <https://github.com/jlord/git-it-electron>
- Learn git branching sandbox: <https://learngitbranching.js.org/>
- Fireship videos on git:
 - Git explained in 100 seconds:
<https://youtu.be/hwP7WQkmECE>
 - Git It? How to use git and github:
<https://youtu.be/HkdAHXoRtos>