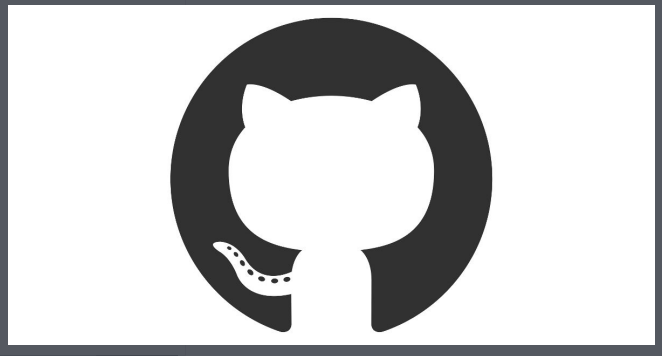
Intro to GitHub

ProvisionHacks 2021

Justin Liu

What is GitHub?



- Github is a provider of Internet hosting for software development and version control using Git
- It also provides access control and several collaboration features
  - Basic task management tools for every project
  - Websites through GitHub Pages
  - Create new versions of software without compromising current versions

Why do we use GitHub?

- Enables developers and programmers to collaboratively work on code -- no matter where they are in the world
- Even the big tech companies like FANG use GitHub today
  - It’s important to know your way around GitHub as a programmer
  - Can serve as a portfolio -- showcase your projects/past experience

What is Git?

git

- GitHub runs on Git
  - Analogous to how an iPhone is powered by iOS
- Git is a command line tool, GitHub provides a Web-based graphical interface
- Git is a version-control/history system created by computer mastermind Linus Torvald

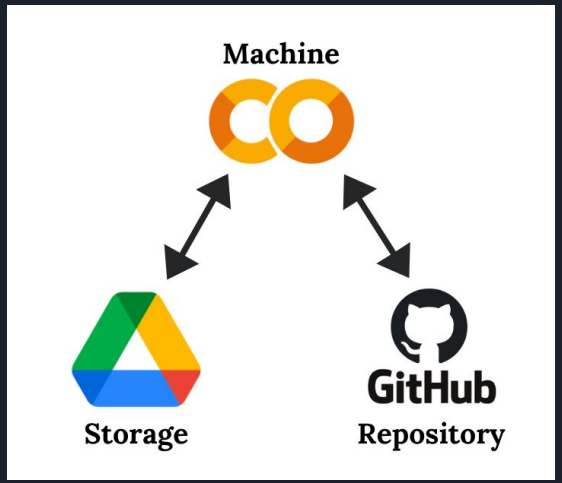
```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
 1 file changed, 1 insertion(+)
 crate mode 100644 README
$ git remote add origin git@github.com:cdown/thats.git
$ git push -u origin master
```

GitHub Desktop

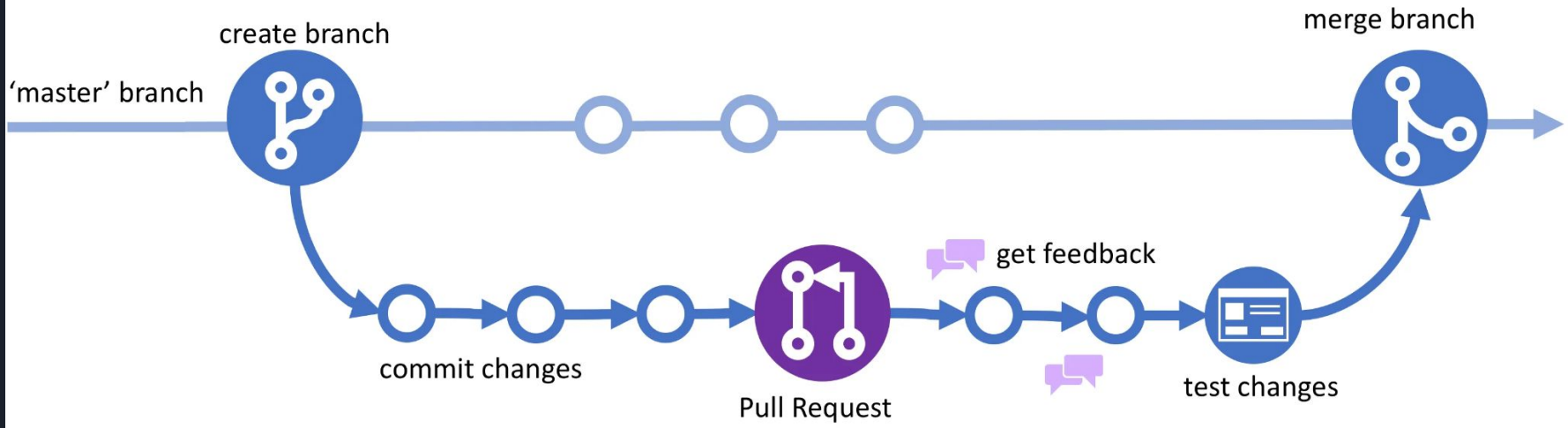
- GitHub Desktop is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser
- A GUI allows users to interact with a program using a visual interface rather than relying on text commands
- Later on, people migrate over to using the terminal and find that to be more productive

GitHub and IDE's

- GitHub is even built-in to some IDE's now!



GitHub Flow



How To Navigate GitHub

- If you want to take a look at my GitHub page for reference: https://github.com/SourLemon23
- Overview tab
  - Pinned projects
  - Contributions/activity
  - Connections/followings
  - Contact info, personal information
- Repositories tab - all their public repos
- Projects
- Packages

Your Turn!

1) Make a GitHub account
2) Create a repo
3) Create a branch
4) Create & commit changes to your branch
5) Open a pull request
6) Merge your pull request

1) **Make a GitHub account**

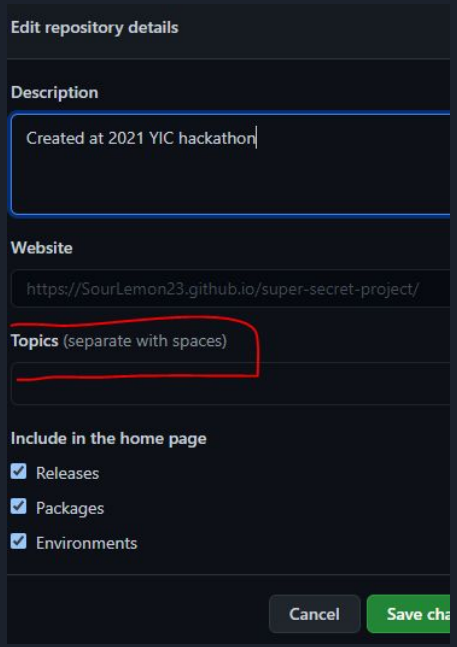
https://github.com/

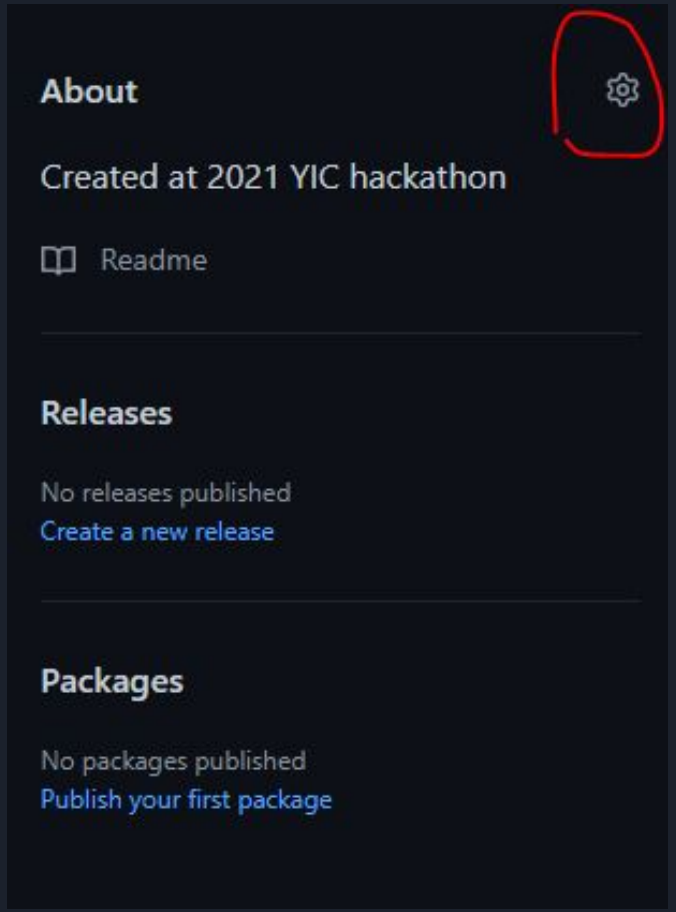
2) Creating your first repo (project)

- To do anything in GitHub, you'll need to know how to first start a repository
- A repository stores everything pertinent to a specific project including files, images, spreadsheets, and data sets
- To create a new repository, you'll hit the + sign and then "new repository" in the upper-right-hand corner. You can then name your repository, include a brief description, and check the box that says "initialize this repository with a README." Finally, you'll click "create repository."
- You may also choose to include a license or .gitignore file

Topics

- Similar to tags or hashtags that are attributed with your project





About READMEs

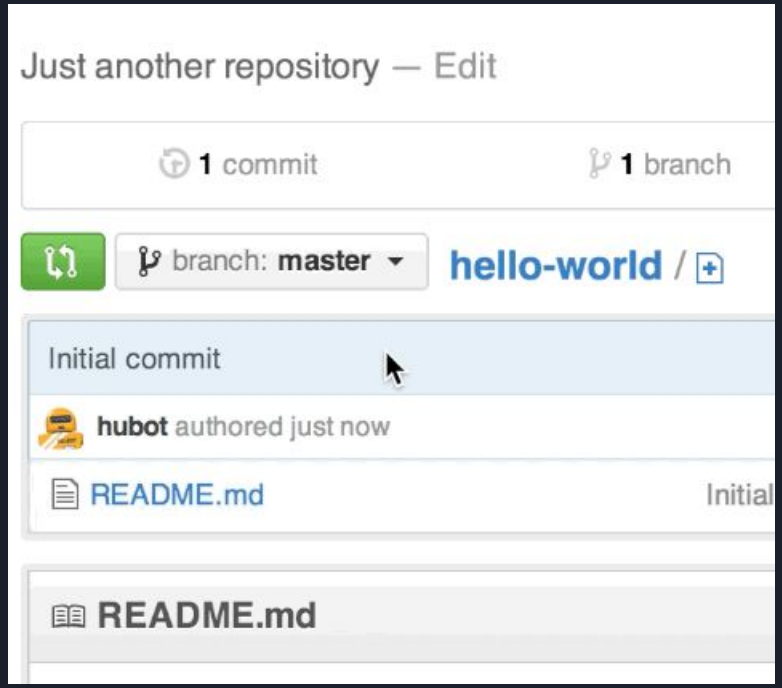
- You can add a README file to your repository to tell other people why your project is useful, what they can do with your project, and how they can use it
- People will normally skim your README before looking at anything else -- it's like a summary/ToB
- A template:
  - What the project does
  - Why the project is useful
  - How users can get started with the project
  - Where users can get help with your project
  - Who maintains and contributes to the project
- You can add images, links, and a lot more in a README
- Uses the markdown language - .md

About branches

- Projects are multi-faceted and many program versions are required when you're building
- Branching enables you to edit multiple unique versions of a repository at once -- they allow experimentation
- By creating a branch, you're making a snapshot/copy of the main branch
- Every repository automatically has a "master branch" -- the default/main branch

To create a new branch:

- Go to your new repository.
- Click the drop down at the top of the file list that says branch: main.
- Type a branch name, readme-edits, into the new branch text box.
- Select the blue Create branch box or hit “Enter” on your keyboard.
- Now you have two branches, main and readme-edits.



About commits

On GitHub, saved changes are called commits

Each commit has an associated commit message, which is a description explaining why a particular change was made

Commit messages capture the history of your changes, so other contributors can understand what you've done and why

Make and commit changes

- Click the README.md file.
- Click the pencil icon in the upper right corner of the file view to edit.
- In the editor, write a sentence about yourself.
- Write a commit message that describes your changes.
- Click Commit changes button.

About pull requests

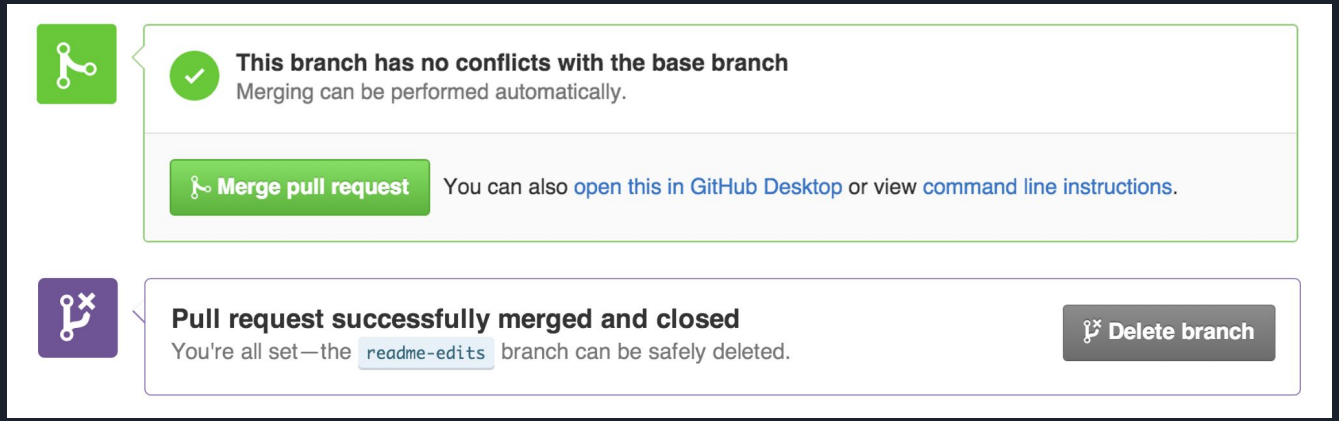
- Now that you have changes in a branch off of main, you can open a pull request.
- Pull Requests are the heart of collaboration on GitHub
- When you open a pull request, you’re proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch
- The changes, additions, and subtractions are shown in green and red
- By using GitHub’s @mention system in your pull request message, you can ask for feedback from specific people or teams, whether they’re down the hall or 10 time zones away.

Open a pull request

- Click the Pull Request tab, then from the Pull Request page, click the green “New pull request” button.
- In the Example Comparisons box, select the branch you made, readme-edits, to compare with main (the original).
- Look over your changes in the diffs on the Compare page, make sure they’re what you want to submit.
- When you’re satisfied that these are the changes you want to submit, click the big green Create Pull Request button.
- Give your pull request a title and write a brief description of your changes.
- You can preview your new changes by switching over to the preview

The last step - merge your pull request

- It’s time to bring your changes together – merging your readme-edits branch into the main branch
- Click the green Merge pull request button to merge the changes into main.
- Click Confirm merge.
- Go ahead and delete the branch, since its changes have been incorporated, with the Delete branch button in the purple box.



Thank You!