Alaina Kafkes
@alainakafkes

git init

A Beginner’s Guide to Git Workflow

What is Git?

Less fun explanation

Git is a **version-control system** that holds the entire (saved) history of your code.

Git allows you to **create save points** in your code to go back to later if need be.

Git gives you the freedom to **pursue new features & ideas** without ruining your code.

The takeaway?

Git makes messing up

INCREDIBLY DIFFICULT.

Git… while alone

Some terminology

Git (not the same as Github!)

Repository (aka “repo”)

Commit

Local versus remote (and why it matters)

Git commands while local

git init

git status

git add

git commit

git log

git diff

git revert

`git init`

Init means **initialize** – but what does this do?

Start to **keep track of local changes** within a directory. How can we visualize these?

`git status`

Visualize the state of tracked files in the initialized Git repo.

Possible file statuses: **staged for commit**, **unstaged but modified, unmodified, untracked**

Use often to gain a visual model of Git workflow.

`git add [FILENAME]`

Stages files that were previously untracked or unstaged but modified.

Two use cases:

`git add main.py` stages the file entitled main.py

`git add -A` stages all untracked & unstaged but modified files.

`git commit -m “MESSAGE”`

Saves a snapshot of all staged files alongside the old version of all files that were not staged.

Use the -m flag to **add a message to your commit** describing its purpose – why?

`git log`

Displays the **entire history** of your code, **commit-by-commit**.

Includes commit message, timestamp, and commit identifier.

Use to **revert to old commits**.

`git diff`

See **line-by-line changes made in individual files** since the last commit.

`git revert`

Undo the changes from a previous commit.

Demo: Introducing local repos

`Local repos are cool`

What was the point of discussing **remote repos** then?

Why remote repos matter

Remote repos allow you to **back up your code** outside of your computer.

Remote repos **support team collaboration**

Why remote repos matter

Open source. 😥💦

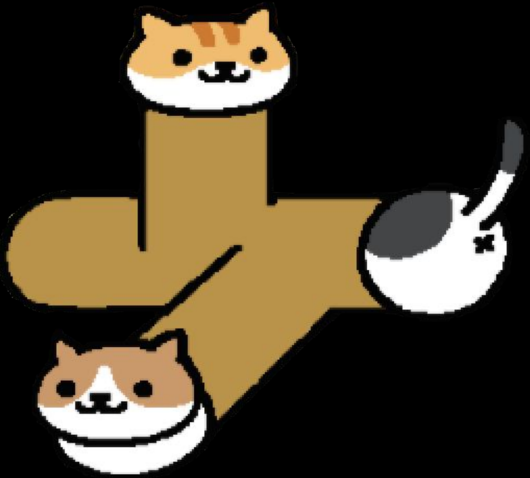
`git push origin [BRANCH]`

Update a remote repo with the commit(s) from a local repo.

But… how do we connect local and remote repositories?

Demo: Connecting remote + local

Collaborating with Git



More terminology

Clone

Branch

Fork (GitHub term)

Pull request (GitHub term)

Remote + local commands

git pull

git clone

git branch

git checkout

git fetch

git merge

`git pull`

Update a local repo with a collaborator’s commit(s) from a remote repo.

Combination of **git fetch + git merge**.

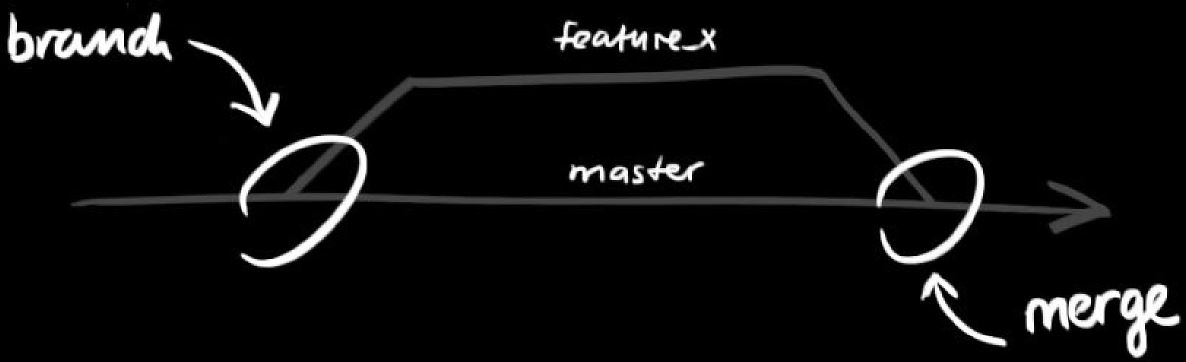
`git clone`

Initializes a local repo containing the current contents of a remote repo, and **creates a connection between local and remote**.

Branching

Used to **develop features in isolation**, and then merge them back into the main code.

Default branch is always **master**.



Branching cheat sheet

git checkout -b [BRANCH] : create a new branch

git branch : list all local repo branches

git checkout [BRANCH] : switch to a branch

git merge [BRANCH] : merges branch & master

git branch -d [BRANCH] : deletes a branch

Important branching note

Local branches are **not seen in the remote repo** until git push.

git fetch

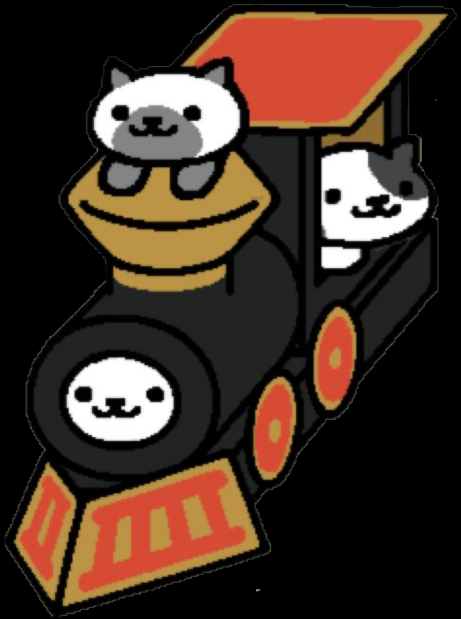
Updates local branches with tracking information about the remote branches.

`git merge`

Combines the history of local and remote branches.

Can lead to scary messages like “CONFLICT” that you may have to sort out by hand.

Demo: Collaborating remotely



Forking repositories

A **fork** is a copy of a repository.

It allows you to freely experiment with a project without making changes to the original repo

Making changes w/ fork

1. Fork the repository
2. Make your changes
3. Submit a **pull request** to the project owner

Demo: What the fork?

What’s next?

Git-ing started

Check **git status** liberally.

Search all error messages on StackOverflow and Google.

Work through tutorials, **make cheat sheets**, and **ask for help**!

Helpful Git resources

Git: The Simple Guide (cheat sheet)

Try Git (tutorial)

GitHub's YouTube Account (tutorial)

99 [PR]oblems (tutorial by me!)

Time to git started!