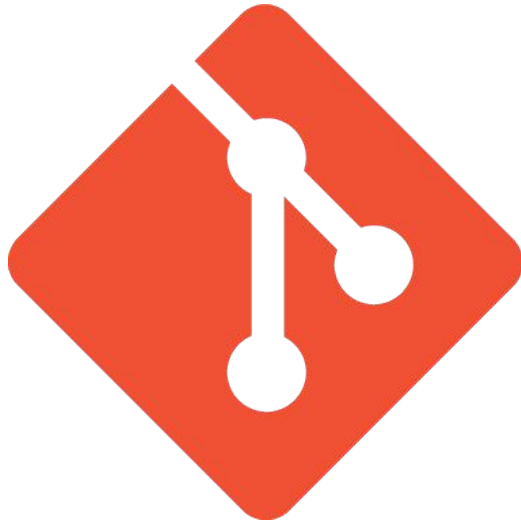# A FEW PREREQUISITES

1. Do you have a **"sandbox"** to play in?

   ❏ Make a new folder on your laptop

2. Do you have **git** installed?

   ❏ Check via command line: `$ git --version`

   ❏ https://git-scm.com/download

3. Do you have a **github** account?

   ❏ https://github.com

Eva J Herzog

linkedin.com/in/evajherzog

FIRST THINGS FIRST

`$ git --version`

Is git already installed?

git version 2.14.3 (Apple Git-98)

https://git-scm.com/download

use all the default options

Windows: Launch Git Bash
Mac: Open Terminal

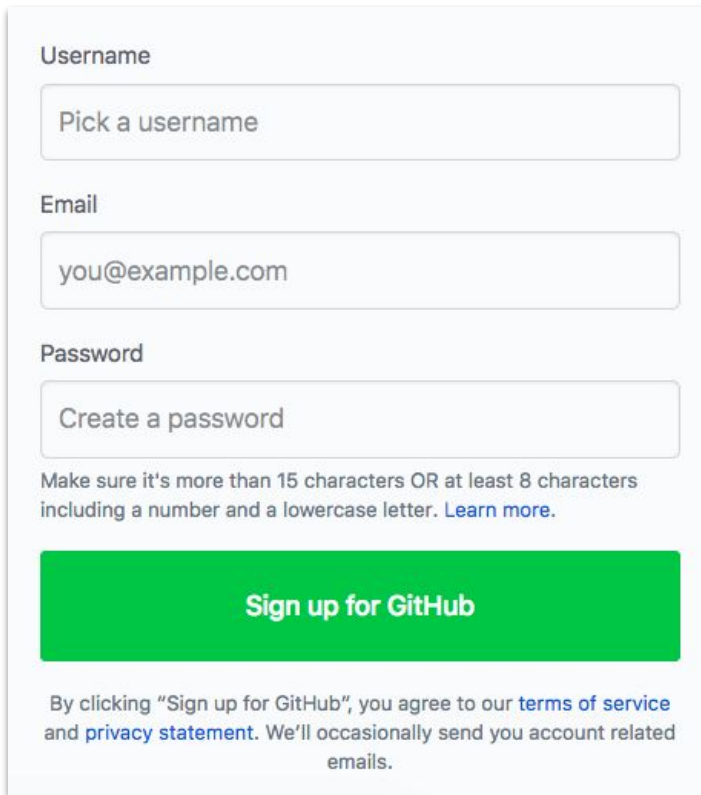Download & Install

SECOND THINGS SECOND

https://github.com

Create a github account in order to share code with your teammates

https://education.github.com/pack

Username

Pick a username

Email

you@example.com

Password

Create a password

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.

VERSION CONTROL?

Management of changes

Documents and collections of information

Ability to revert mistakes

WHAT IS GIT?

- Distributed version control system
- Designed for contributors to collaborate
- Free command line tool

WHAT IS GITHUB?
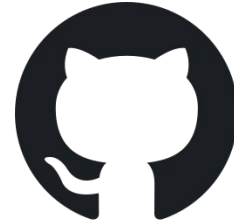
- Web-based hosting service
- Basics for collaboration, plus more
- Some features for free; education account

GETTING STARTED

List *all* files in current folder	`$ ls -a`
Change directory (move to a folder)	`$ cd <subfolder>`
What's the status of the git project?	`$ git status`
Initialize a new project	`$ git init <project>`
Download project and version history	`$ git clone <url>`

WHAT JUST HAPPENED?!

➔ Initialized a project
➔ Created a file
➔ Edited the file
➔ Saved the file
➔ *Staged* the file
➔ "On branch master"?
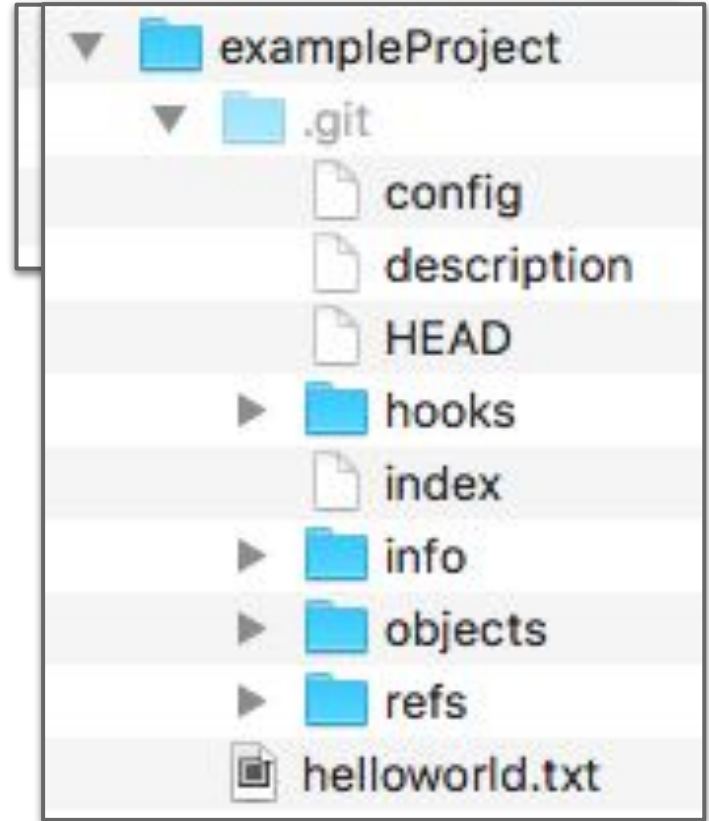
WHAT'S NEXT?

Look through history of the project	`$ git log`
Take a snapshot of the current work	`$ git commit -m <msg>`
What's changed since last commit?	`$ git diff (--staged)`

QUICK DETOUR!

Mac:

`$ git config --global core.editor "open -W -n"`

Windows:

`$ git config --global core.editor notepad`

Changing the default git editor will allow us to avoid editing text files with things like vi, vim, emacs.

WHAT'S NEXT?

Look through history of the project	`$ git log`
Take a snapshot of the current work	`$ git commit -m <msg>`
What's changed since last commit?	`$ git diff (--staged)`
Creates a new commit, records in log	`$ git revert <commit>`
Undoes later commits, keeps changes	`$ git reset <commit>`

BRANCHES
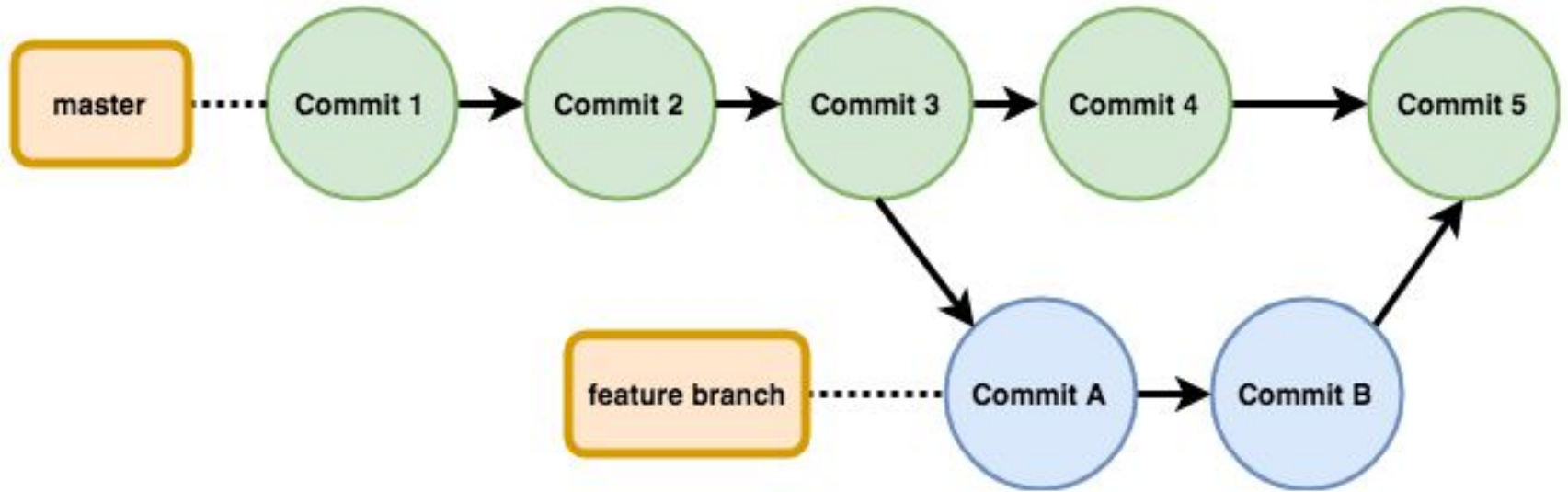
BRANCHES

`git branch`
   List out all the branches

`git checkout (-b) <name>`
   Switch to the specified branch
   (with `-b`, create new branch)
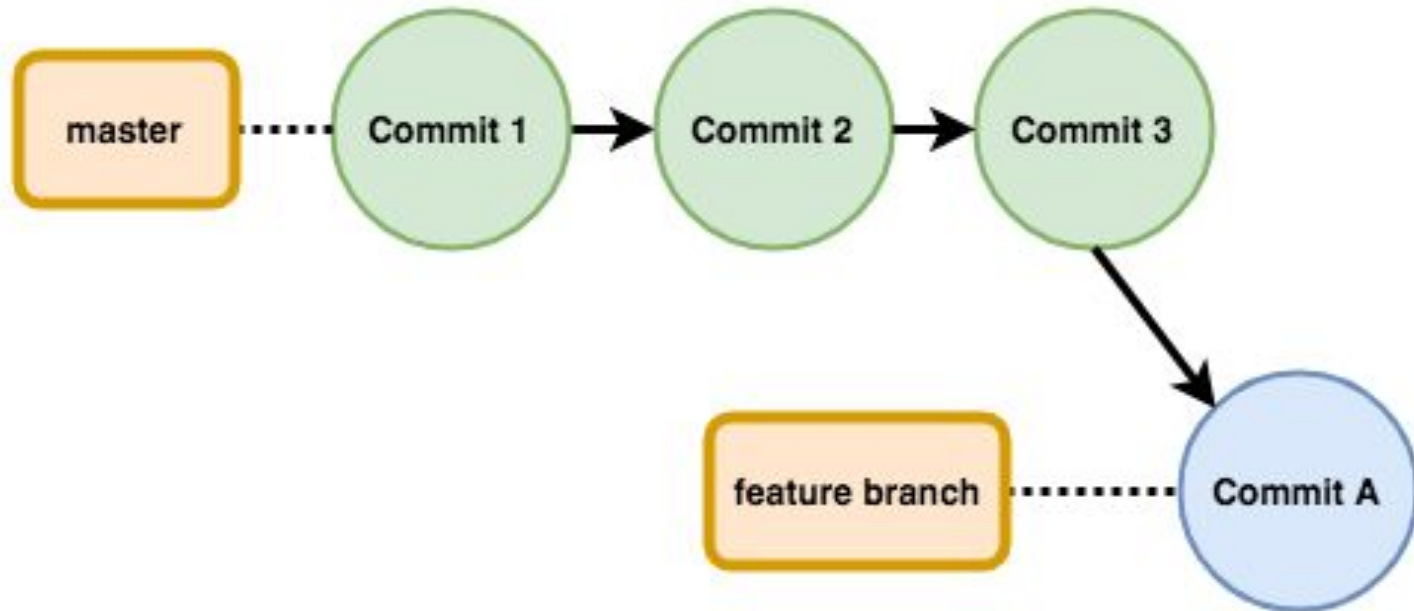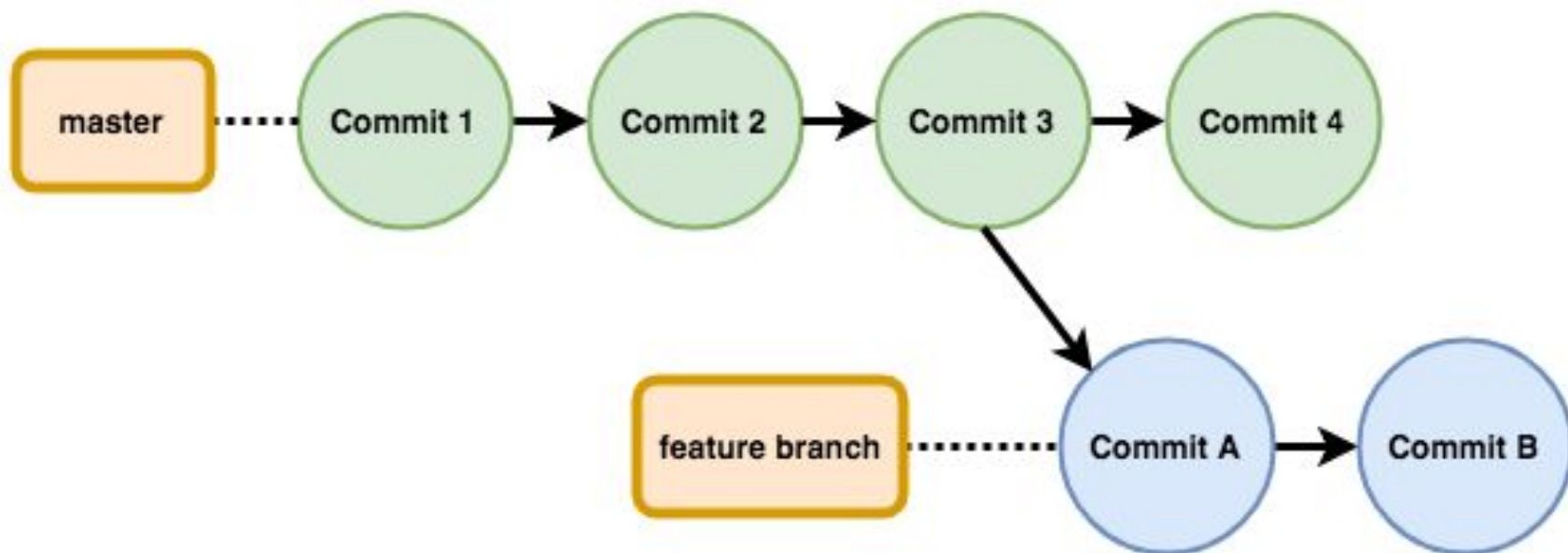
`git merge <name>`
   Merge *name*'s changes into the
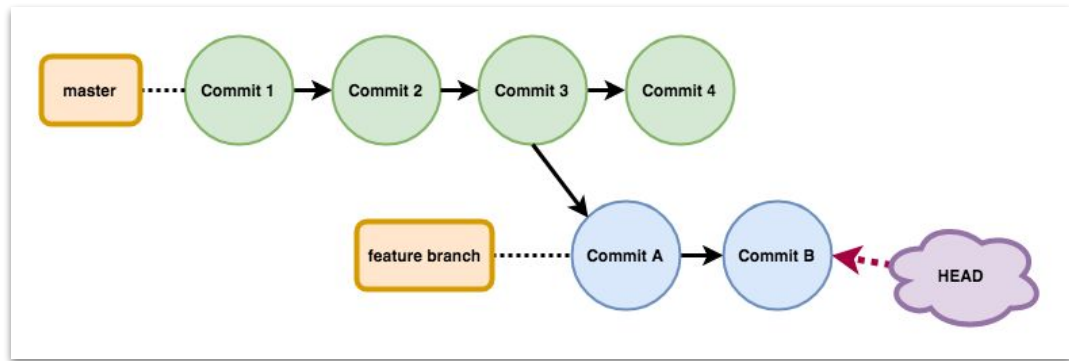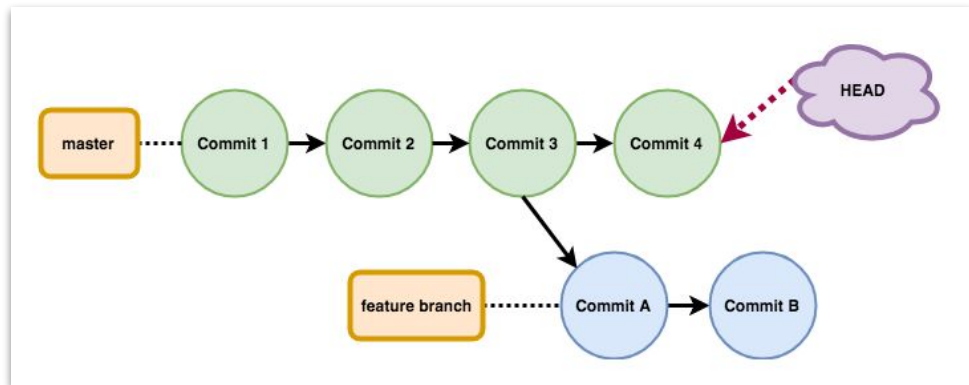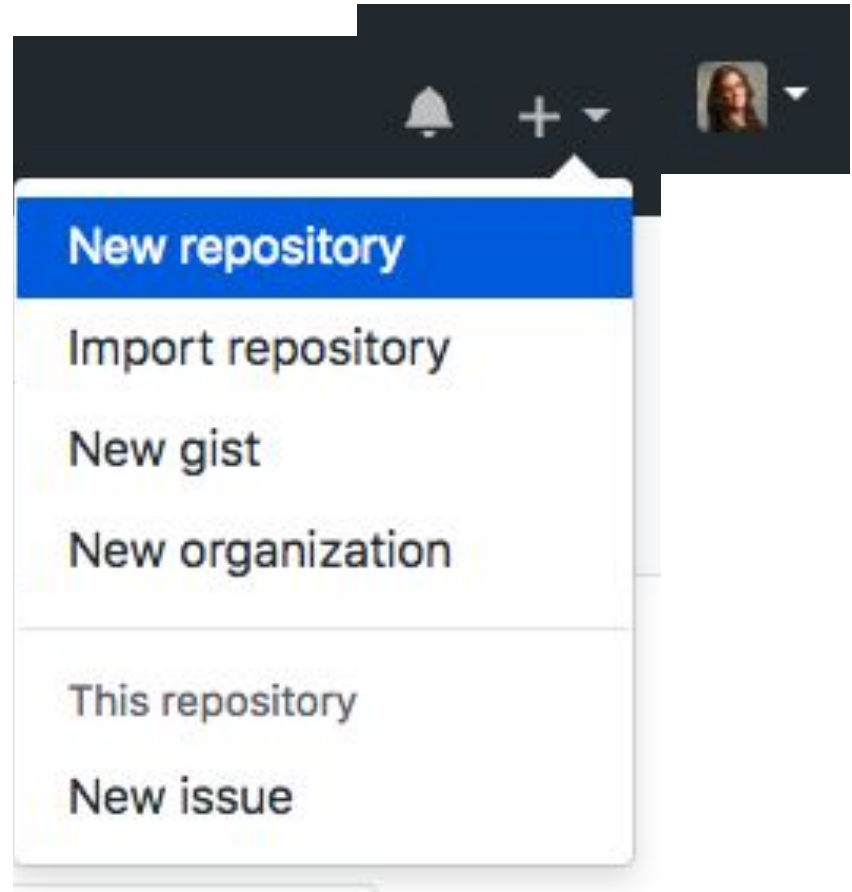   current branch

BRANCHES

BRANCHES

Terminology: head

- Symbolic reference

- Last known state of working directory

- Parent of your next commit

LET'S COLLABORATE!

- Go to github.com

- Choose 1 team member (ONLY 1) who will create a new repository

- Use the + in the upper right corner

LET'S COLLABORATE!

Create a new repository

A repository contains all project files, including the revision history.

Owner

ejherzog ▾ /

Repository name *

sister_hacks ✓

Great repository names are short and memorable. Need inspiration? How about **fantastic-train**?

Description (optional)

Sample Project

🔵 📘 **Public**
   Anyone can see this repository. You choose who can commit.

⚪ 🔒 **Private**
   You choose who can see and commit to this repository.

LET'S COLLABORATE!

LET'S COLLABORATE!

LET'S COLLABORATE!

- Go to github.com/<username>/<repositoryName>

- Click green 'Clone or Download' button

- Copy URL to clipboard

- `$ git clone <URL>`

- `$ cd <repositoryName>`

DO SOME BRANCH WORK

- $ git pull
- $ git status
- $ git checkout -b <yourname>
- Create at least one new file (.txt, .csv, .md, .py)
- $ git add .
- $ git commit -m "<meaningful message>"

WHERE ARE YOUR CHANGES?

- `$ git status`
- Look at the github page for your repo: no branches??
- `$ git pull`
- `$ git push`
- `$ git push --set-upstream origin <branch>`
- Look back at the github page for your repo

PUSHING AND PULLING

Get remote updates to local machine	`git fetch`
git fetch + incorporate those changes	`git pull`
Upload all local branches to Github	`git push`

`git push --set-upstream origin <branch name>`

Push a local branch to the remote repo for the first time

OPEN A PULL REQUEST

Last words of wisdom

❖ About to push changes? Pull first.

❖ Trying out an idea? Create a new branch.

❖ Avoid erasing history. Leave the evidence.

❖ "Take chances, make mistakes, & get messy!"

-Ms. Frizzle

Sources

1. Git Logos: Jason Long [CC BY 3.0 (https://creativecommons.org/licenses/by/3.0)]

2. Seven Sisters Page History: https://en.wikipedia.org/wiki/Seven_Sisters_(colleges)

3. Computer ClipArt: AJ from openclipart.org [CC0]

4. Server Icon: The Oxygen Team, KDE; [LGPL (https://www.gnu.org/copyleft/lgpl.html)]

5. Github Logo: GitHub [MIT (http://opensource.org/licenses/mit-license.php)]

6. Branch diagrams: Eva Herzog using draw.io