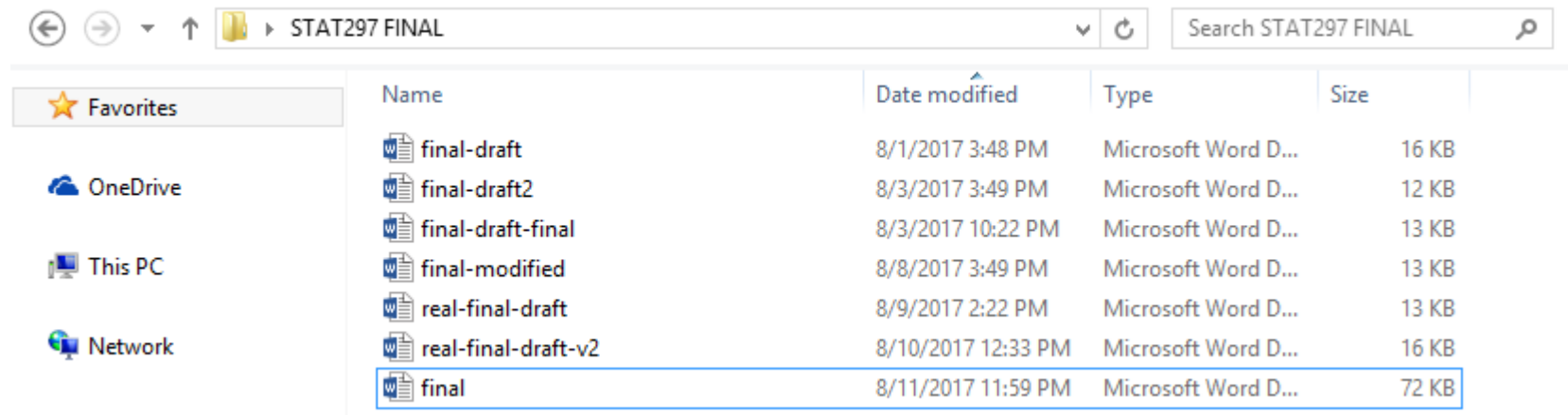


# Welcome to Git

Intro to Git and Git Fundamentals

# The need for Git



Name	Date modified	Type	Size
final-draft	8/1/2017 3:48 PM	Microsoft Word D...	16 KB
final-draft2	8/3/2017 3:49 PM	Microsoft Word D...	12 KB
final-draft-final	8/3/2017 10:22 PM	Microsoft Word D...	13 KB
final-modified	8/8/2017 3:49 PM	Microsoft Word D...	13 KB
real-final-draft	8/9/2017 2:22 PM	Microsoft Word D...	13 KB
real-final-draft-v2	8/10/2017 12:33 PM	Microsoft Word D...	16 KB
final	8/11/2017 11:59 PM	Microsoft Word D...	72 KB

Differences between all these versions?

Was real-final-draft's predecessor:

- final-draft2?
- final-modified?

Want a **history** of revisions.

# The need for Git

Want a **history** of revisions.

If we make a big mistake:

- Where is the booboo?
- What file/file version did the booboo rear its ugly head?

Revert state to point before mistake was made.

# The need for Git

Want a **history** of revisions.

Gets very important in collaboration:

## Real world scenario

Attila, Genghis, and Odoacer: building ML predictor.

### Is your kingdom ready to fall?

- Attila: wants to rebuild some data cleaning processes.
- Odoacer: still working on fine tuning predictor algorithm.
- Genghis: using current model in deployment.

Don't wanna make Genghis mad!



# The need for Git

Want a **history** of revisions.

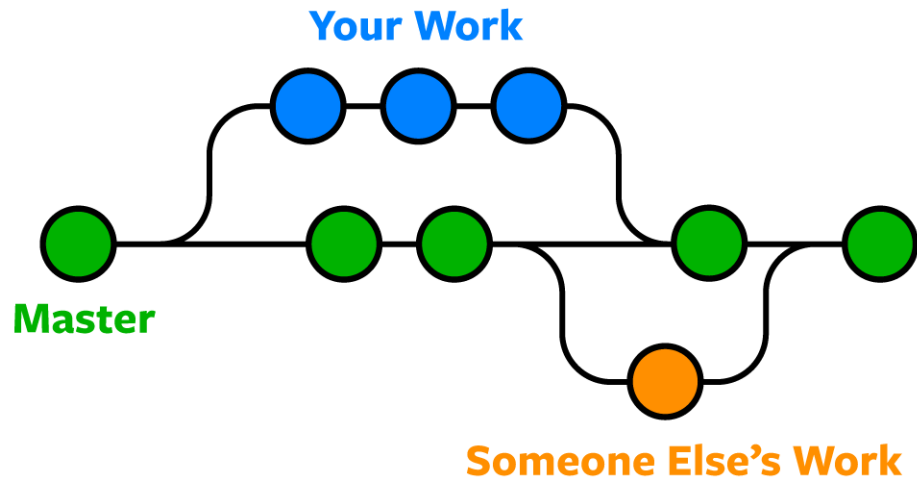
Gets very important in collaboration:

## Workflow

- Atilla and Odoacer work on their contributions in their own project space.
- Combine their work: resolve any conflicts.
- Combine with original.
  
- Uh-oh...Atilla's work has created a problem. Genghis starts getting mad.

Undo Atilla's changes to original, keep Odoacer's. Phew! Close call.

# What is Git?



Each node of graph: a tracked version of a project (commit).

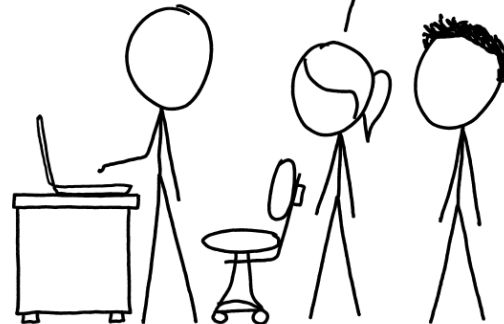
Keeps track of history: branch offs, modifications, etc.

Merging between branches and resolving conflicts.

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

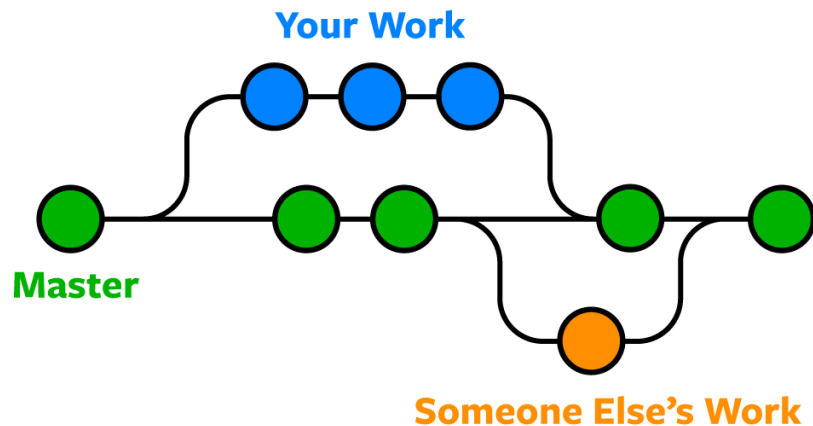
COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



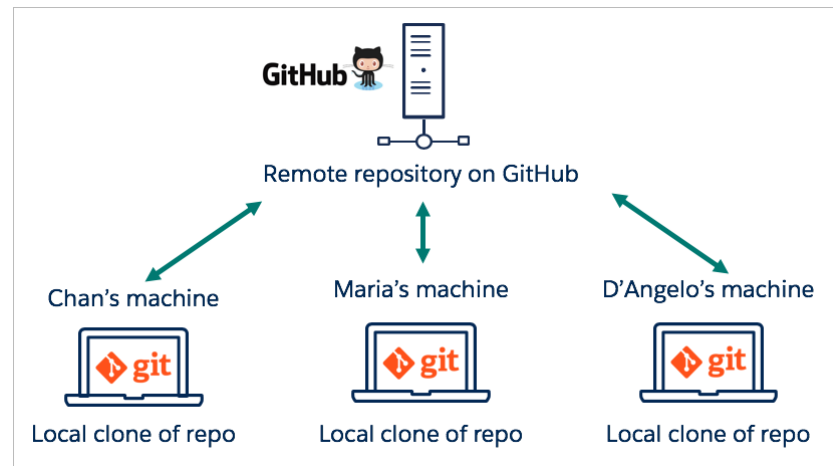
# Git and Github

## Git



Project (repository) history/workflow management.

## Github



Cloud based repository hosting. Designed with Git workflow in mind.

# Git and Github

## Git

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Flatiron_Lectures (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Phase1_Top1_Intro/MacOS_ Anaconda Installation Step-by-Step.pdf
  Phase1_Top1_Intro/Windows_ Anaconda Installation Step-by-Step.pdf
  Phase1_Top1_Intro/Windows_ Git Installation Step-by-Step.pdf
  Phase1_Topic2_TerminalGit/

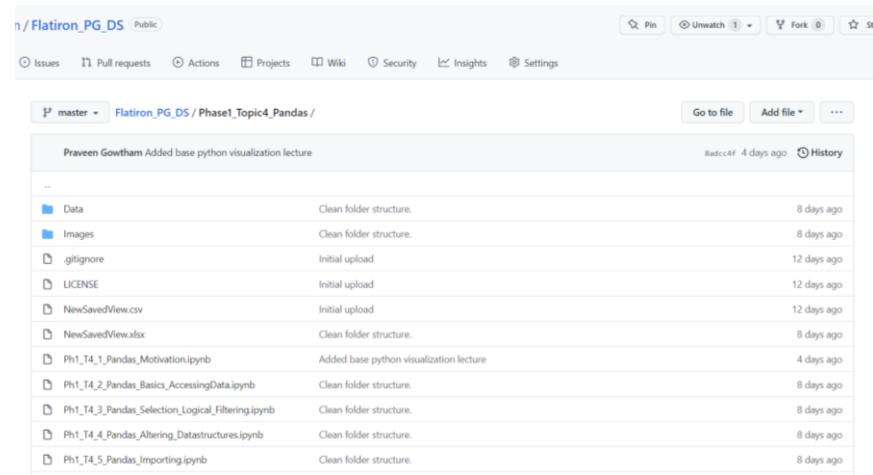
nothing added to commit but untracked files present (use "git add" to track)
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Flatiron_Lectures (master)
$ git add *
```

Track and commit changes (maintaining workflow graph).

Can submit changes to files **and** workflow graph to Github.

**Command line.**

## Github

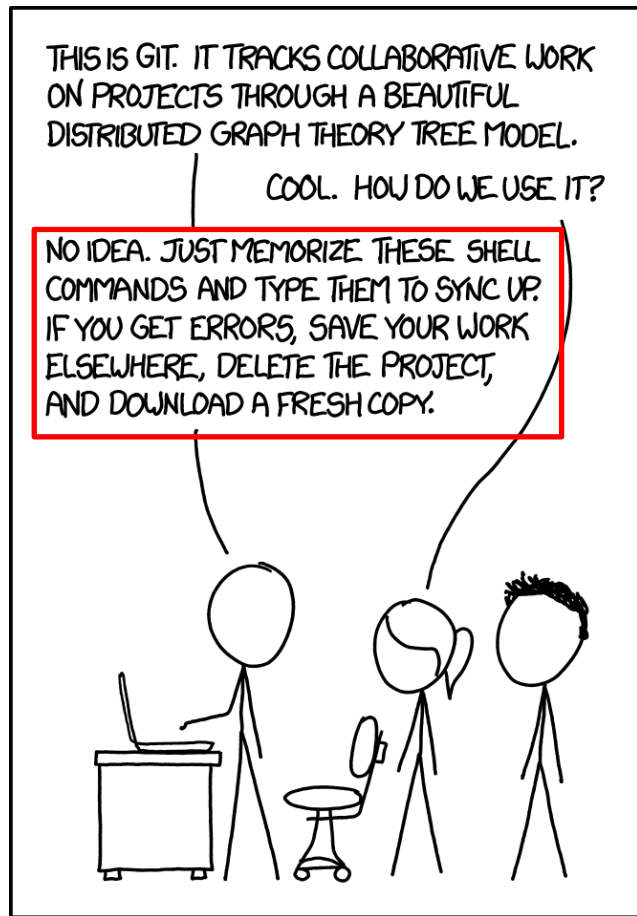


Files/workflow stored in common remote repository.

Interact via Graphical User Interface.



# Git/Github: steep learning curve



Basically what we  
are going to do.

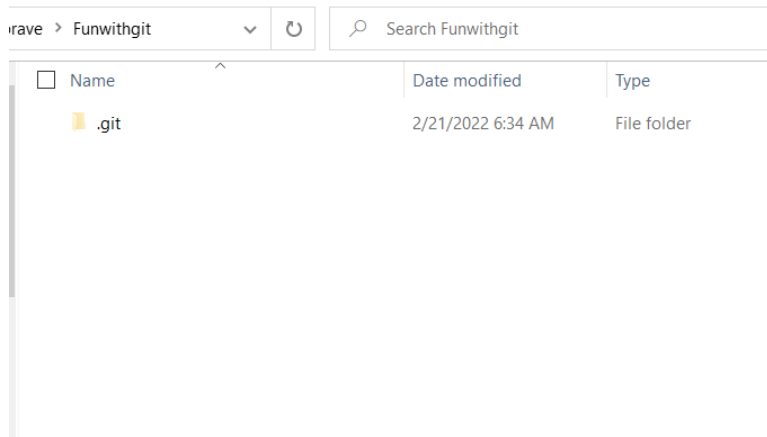
# git init

Make local directory a git repository.



```
MINGW64:/c/Users/prave/Funwithgit
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit
$ git init
Initialized empty Git repository in C:/Users/prave/Funwithgit/.git/
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$
```

Hidden .git folder



Stores workflow graph info.

# Github: make a remote repository



**Praveen Gowtham**  
admveen

Overview

Repositories 28

Projects

Packages

Stars

Find a repository...

Type ▾

Language ▾

Sort ▾

New

**Flatiron\_PG\_DS** Public

Lectures for Flatiron's DS LIVE NYC 2022

Jupyter Notebook ☆ 1 Updated 15 hours ago

☆ Star ▾

**NMF\_tutorial** Public


Some tweet topic modeling with NMF

Jupyter Notebook 🍷 1 Updated 20 days ago

☆ Star ▾

# Github: make a remote repository

Owner \*

 admveen ▾

 / 


Repository name \*

funwithgit ✓


Great repository names are short and memorable. Need inspiration? How about [symmetrical-tribble?](#)

Description (optional)

Demistifying git for the peoples

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☒ **Add a README file**


This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  `main` as the default branch. Change the default name in your [settings](#).

Create repository

# Github: make a remote repository

The screenshot shows the GitHub interface for a repository named 'funwithgit' by user 'admveen'. The repository is public. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the repository details show the 'main' branch with 1 branch and 0 tags. A commit history table lists an 'Initial commit' by 'admveen' 11 minutes ago, with a commit hash of 'a0c52b9' and 1 commit. The commit includes a file named 'README.md'. Below the commit history, the 'README.md' content is displayed, featuring the repository name 'funwithgit' and the description 'Demistifying git for the peoples.'.

admveen / funwithgit Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

admveen Initial commit a0c52b9 11 minutes ago 1 commit

File	Commit	Time
README.md	Initial commit	11 minutes ago

README.md

## funwithgit

Demistifying git for the peoples.

Contains address to  
remote repository

# Github: make a remote repository

The screenshot shows the GitHub interface for a repository named 'funwithgit' by user 'admveen'. The repository is public and has 1 branch (main) and 0 tags. The 'Code' button is highlighted in green. A dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI, with the HTTPS URL 'https://github.com/admveen/funwithgit.git' displayed. Other options include 'Open with GitHub Desktop' and 'Download ZIP'. The repository description is 'Demystifying git for the peoples.' and it has 0 stars and 0 forks. The 'About' section shows a README file and the 'Releases' section shows no releases published.

admveen / funwithgit Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

admveen Initial commit

README.md Initial commit

Clone

HTTPS SSH GitHub CLI

<https://github.com/admveen/funwithgit.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Demystifying git for the peoples.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

# Link git to github: git remote

git remote add **remote\_name** **remote\_address**

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit
$ git init
Initialized empty Git repository in C:/Users/prave/Funwithgit/.git/
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ ls
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git remote add origin https://github.com/admveen/funwithgit.git
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ |
```

Git repository (local)  
and Github remote repo now linked!

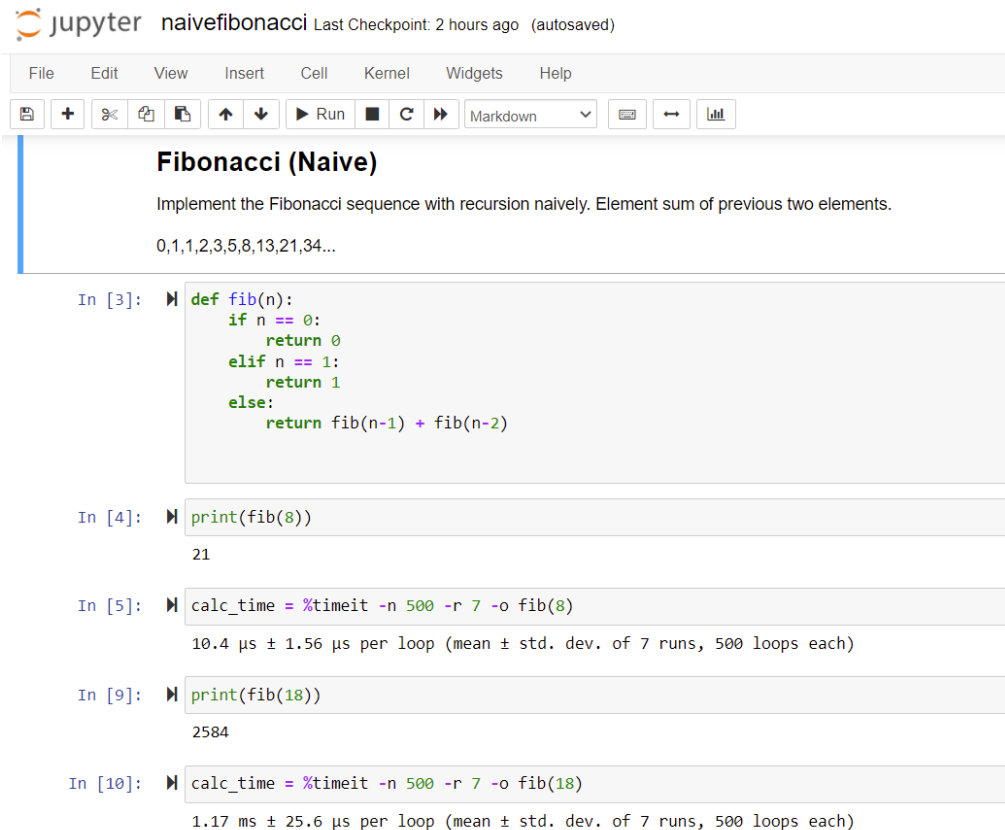
git remote -v

```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git remote -v
origin https://github.com/admveen/funwithgit.git (fetch)
origin https://github.com/admveen/funwithgit.git (push)
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$
```

# Made some changes

I created a new Jupyter notebook:

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ ls
naivefibonacci.ipynb
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ |
```



The screenshot shows a Jupyter Notebook interface with the title 'naivefibonacci' and a status bar indicating 'Last Checkpoint: 2 hours ago (autosaved)'. The notebook has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running. The notebook content includes a title 'Fibonacci (Naive)', a description 'Implement the Fibonacci sequence with recursion naively. Element sum of previous two elements.', and a list of Fibonacci numbers '0,1,1,2,3,5,8,13,21,34...'. Below this, there are five code cells. The first cell defines a recursive function 'fib(n)'. The second cell prints 'fib(8)', which outputs '21'. The third cell uses '%timeit' to measure the execution time of 'fib(8)', showing '10.4 μs ± 1.56 μs per loop (mean ± std. dev. of 7 runs, 500 loops each)'. The fourth cell prints 'fib(18)', which outputs '2584'. The fifth cell uses '%timeit' to measure the execution time of 'fib(18)', showing '1.17 ms ± 25.6 μs per loop (mean ± std. dev. of 7 runs, 500 loops each)'.

```
jupyter naivefibonacci Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

[Save] [Add Cell] [Undo] [Redo] [Up] [Down] [Run] [Stop] [Refresh] [Markdown] [List]

Fibonacci (Naive)

Implement the Fibonacci sequence with recursion naively. Element sum of previous two elements.

0,1,1,2,3,5,8,13,21,34...

In [3]: def fib(n):
        if n == 0:
            return 0
        elif n == 1:
            return 1
        else:
            return fib(n-1) + fib(n-2)

In [4]: print(fib(8))
21

In [5]: calc_time = %timeit -n 500 -r 7 -o fib(8)
10.4 μs ± 1.56 μs per loop (mean ± std. dev. of 7 runs, 500 loops each)

In [9]: print(fib(18))
2584

In [10]: calc_time = %timeit -n 500 -r 7 -o fib(18)
1.17 ms ± 25.6 μs per loop (mean ± std. dev. of 7 runs, 500 loops each)
```

Yikes! Not scalable. I need to change this!



# Make change part of history

File created (in folder)

Not yet in workflow graph (git repo)

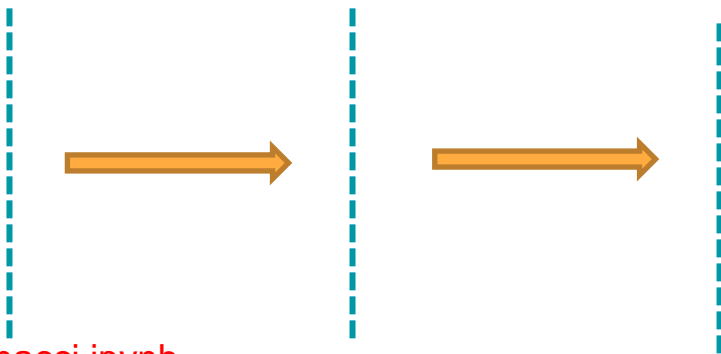
```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git log
fatal: your current branch 'master' does not have any commits yet
```

Want to commit changes: add to graph/make part of our history.

Local file system

Staging Area

Local git repo



naivefibonacci.ipynb

# git status

Compares files in directory to local git repo:

- Are there changes? Which files?
- Are they in staging area or not (**tracked/untracked**)?

```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .ipynb_checkpoints/
        naïvefibonacci.ipynb

nothing added to commit but untracked files present (use "git add" to track)
```

# git add

git add .

This adds all untracked changes to staging area

git add filename

Adds file with untracked changes to staging area

```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git add *
warning: LF will be replaced by CRLF in naivefibonacci.ipynb.
The file will have its original line endings in your working directory
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   naivefibonacci.ipynb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .ipynb_checkpoints/
```

# git commit

When ready: add changes to the official record/history.

Transfer changes from staging area to the git graph / repo.

`git commit -m "Message"`

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git commit -m "Add naive implementation of Fibonacci"
[master (root-commit) 3bc1064] Add naive implementation of Fibonacci
1 file changed, 135 insertions(+)
create mode 100644 naivefibonacci.ipynb
```

`git log` (see history of graph)

```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git log
commit 3bc106469af06625930ff9fa6ac0762f1170fee3 (HEAD -> master)
Author: Praveen Gowtham <praveen.gowtham@flatironschool.com>
Date: Mon Feb 21 12:53:04 2022 -0500

    Add naive implementation of Fibonacci
```

# commit messages

Commit messages are important:

- Tells you/others what commit is about

`git commit -m "meaningful stuff"`

**Rule 1:** One line description.

**Rule 2:** Should be meaningful.

**Style guideline:**

Start with a verb in the present tense,  
imperative mood.

*"Implement dynamic programming solution."*

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT  
MESSAGES GET LESS AND LESS INFORMATIVE.

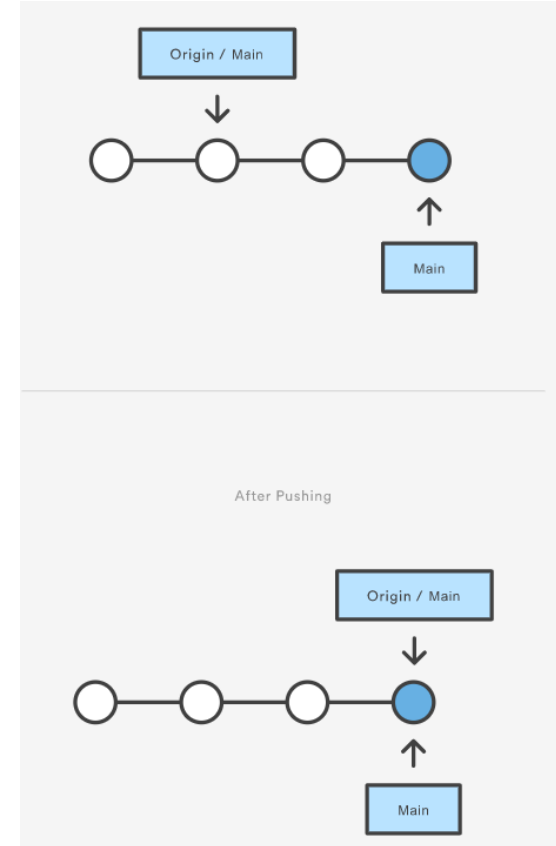
# git push

Previously: linked local git to github remote repo.

Now: send new changes + git history to remote repo/branch.


git push remote\_name local\_branch

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git remote -v
origin https://github.com/admveen/funwithgit.git (fetch)
origin https://github.com/admveen/funwithgit.git (push)
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ |
```





# git push


Bam! There it is on github.

[/ funwithgit](#) Public 

[Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 master ▾

 2 branches

 0 tags


[Go to file](#)

[Add file ▾](#)

[Code ▾](#)

**Praveen Gowtham** Add naive implementation of Fibonacci

3bc1064 35 minutes ago ⌚ 1 commit

 naivefibonacci.ipynb

Add naive implementation of Fibonacci

35 minutes ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)

# Make better implementation.

- Created new one (good\_fibonacci).
- Deleted old one (naive\_fibonacci).
- Also created new text file.

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ ls
good_fibonacci.ipynb  untracked_data.txt
```

## Fibonacci with Dynamic Programming

(a.k.a don't repeat work)

```
In [2]: # Lookup table for results
result_dict = {0: 1, 1: 1}

def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        # if already computed, use value in lookup table
        if (n - 1) in result_dict.keys():
            fibn1 = result_dict[n-1]
        else:
            # otherwise, compute
            fibn1 = fib(n-1)
        # if already computed, use value in lookup table
        if (n - 2) in result_dict.keys():
            fibn2 = result_dict[n-2]
        else:
            # otherwise, compute
            fibn2 = fib(n-2)

        fibn = fibn1 + fibn2

        # store new result in lookup table
        result_dict[n] = fibn

    return result_dict[n]
```

```
In [3]: calc_time = %timeit -n 500 -r 7 -o fib(18)
```

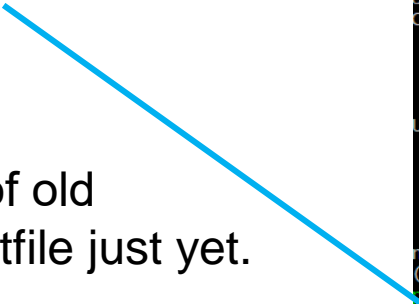


# You can be a little fancy

Stage/commit new one  
(good\_fibonacci)

**But:**

Don't commit delete of old  
implementation or textfile just yet.



```
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    naivefibonacci.ipynb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .ipynb_checkpoints/
        good_fibonacci.ipynb
        untracked_data.txt

no changes added to commit (use "git add" and/or "git commit -a")
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git add good_fibonacci.ipynb
warning: LF will be replaced by CRLF in good_fibonacci.ipynb.
The file will have its original line endings in your working directory
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   good_fibonacci.ipynb

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    naivefibonacci.ipynb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .ipynb_checkpoints/
        untracked_data.txt

(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
```

# Push!

```
(base)
prave@LAPTOP-JA891BK6 MINGW64 ~/Funwithgit (master)
$ git commit -m "Add dynamic programming solution."
[master 847e758] Add dynamic programming solution.
1 file changed, 99 insertions(+)
 create mode 100644 good_fibonacci.ipynb
(base)
prave@LAPTOP-JA891BK6 MINGW64 ~/Funwithgit (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.10 KiB | 1.10 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/admveen/funwithgit.git
   3bc1064..847e758  master -> master
(base)
prave@LAPTOP-JA891BK6 MINGW64 ~/Funwithgit (master)
$ git log
commit 847e7582b0d7deb7bf0e3258256c2e88beb822cb (HEAD -> master, origin/master)
Author: Praveen Gowtham <praveen.gowtham@flatironschool.com>
Date:   Mon Feb 21 14:13:13 2022 -0500

    Add dynamic programming solution.

commit 3bc106469af06625930ff9fa6ac0762f1170fee3
Author: Praveen Gowtham <praveen.gowtham@flatironschool.com>
Date:   Mon Feb 21 12:53:04 2022 -0500

    Add naive implementation of Fibonacci
(base)
prave@LAPTOP-JA891BK6 MINGW64 ~/Funwithgit (master)
$
```

admveen / funwithgit Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master ▾

2 branches

0 tags

Go to file

Add file ▾

Code ▾

Praveen Gowtham Add dynamic programming solution.		847e758 5 minutes ago 2 commits
good_fibonacci.ipynb	Add dynamic programming solution.	5 minutes ago
naivefibonacci.ipynb	Add naive implementation of Fibonacci	1 hour ago

# Do rest of changes

After git add . and commit:

- Commit delete naivefibonacci in repo.
- Commit add text file

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git add .
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Funwithgit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .ipynb_checkpoints/good_fibonacci-checkpoint.ipynb
    renamed:    naivefibonacci.ipynb -> .ipynb_checkpoints/naivefibonacci-checkpoint.ipynb
    new file:   untracked_data.txt
```

Then push:

The screenshot shows the GitHub repository page for 'admveen / funwithgit'. The repository is public. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there are buttons for 'Go to file', 'Add file', and 'Code'. The main content area shows a list of commits. The most recent commit is by Praveen Gowtham, titled 'Add text file. Confirm delete old implementation.', made 31 minutes ago. This commit includes three files: '.ipynb\_checkpoints' (Add text file. Confirm delete old implementation., 31 minutes ago), 'good\_fibonacci.ipynb' (Add dynamic programming solution., 1 hour ago), and 'untracked\_data.txt' (Add text file. Confirm delete old implementation., 31 minutes ago).

admveen / funwithgit Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 2 branches 0 tags Go to file Add file Code

Praveen Gowtham Add text file. Confirm delete old implementation. 99fc055 31 minutes ago 3 commits	
.ipynb_checkpoints	Add text file. Confirm delete old implementation. 31 minutes ago
good_fibonacci.ipynb	Add dynamic programming solution. 1 hour ago
untracked_data.txt	Add text file. Confirm delete old implementation. 31 minutes ago

# Forking/Cloning

Want to work with someone else's project:

Step 1: Fork

Step 2: Clone

## Forking

- Create copy of another's repo in your Github account.
- Any changes (commit/push) are made to ***your*** copy.
- Doesn't mess their project up!

# Forking

avbrown / music\_to\_my\_ears Public

Watch 1

Fork 0

Star 0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Forks repo

main 1 branch 0 tags

Go to file Add file Code

avbrown Add files via upload 7e418a2 10 days ago 15 commits

images	Delete playlist_distribution_2.jpg	10 days ago
tmp	Create .gitkeep	10 days ago
.gitignore	Adding .gitignore	10 days ago
README.md	Updated files that assumed tmp would have temporary files available t...	10 days ago
main_notebook.ipynb	Updated files that assumed tmp would have temporary files available t...	10 days ago
modeling_notebook.ipynb	Add files via upload	10 days ago
preprocessing_notebook.ipynb	Updated files that assumed tmp would have temporary files available t...	10 days ago
presentation.pdf	Add files via upload	10 days ago

README.md

Music to My Ears

About

This project develops a content-based filtering recommendation system for musical tracks by utilizing a multilabel binarizer as a preprocessing tool on a million user playlists. This creates a large scale utility matrix (1,000,000 by 2,262,292) comprised of playlists and tracks, which is used with Non-negative Matrix Factorization (NMF) along si...

Readme

0 stars

1 watching


0 forks

Releases

No releases published



# After forking


Now a copy of repo in my github


 **admveen / music\_to\_my\_ears** Public  
forked from avbrown/music\_to\_my\_ears

[Pin](#) [Watch 0](#) [Fork 1](#) [Star 0](#)

[Code](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)



 **main** 


 1 branch









 0 tags



[Go to file](#) [Add file](#) [Code](#)

This branch is up to date with avbrown:main.

 Contribute  Fetch upstream


 **avbrown** Add files via upload 7e418a2 10 days ago 15 commits


 images	Delete playlist_distribution_2.jpg	10 days ago
 tmp	Create .gitkeep	10 days ago
 .gitignore	Adding .gitignore	10 days ago
 README.md	Updated files that assumed tmp would have temporary files available t...	10 days ago
 main_notebook.ipynb	Updated files that assumed tmp would have temporary files available t...	10 days ago
 modeling_notebook.ipynb	Add files via upload	10 days ago
 preprocessing_notebook.ipynb	Updated files that assumed tmp would have temporary files available t...	10 days ago
 presentation.pdf	Add files via upload	10 days ago


 **README.md** 


### About

This project develops a content-based filtering recommendation system for musical tracks by utilizing a multilabel binarizer as a preprocessing tool on a million user playlists. This creates a large scale utility matrix (1,000,000 by 2,262,292) comprised of playlists and tracks, which is used with Non-negative Matrix Factorization (NMF) along si...

 [Readme](#)

 0 stars

 0 watching

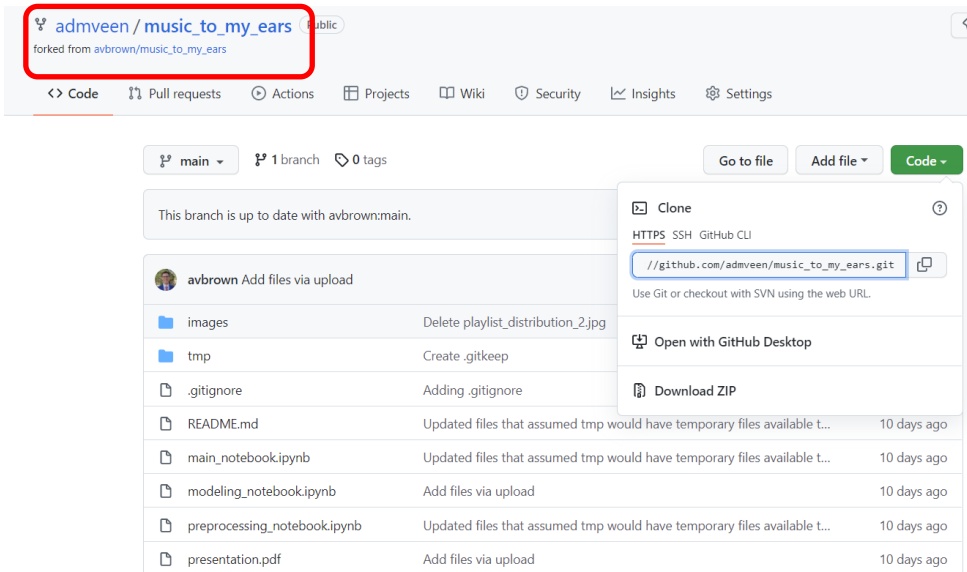
 1 fork

### Releases

No releases published  
[Create a new release](#)

# Cloning

Now a copy of repo in my github



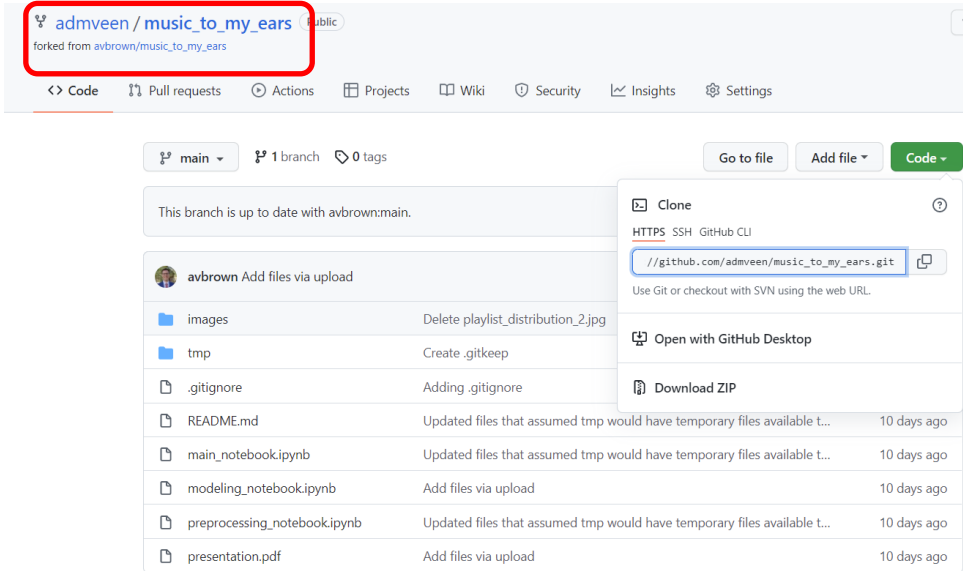
git clone address\_forked\_repo

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ git clone https://github.com/admveen/music_to_my_ears.git
Cloning into 'music_to_my_ears'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 50 (delta 25), reused 8 (delta 5), pack-reused 0
Receiving objects: 100% (50/50), 8.78 MiB | 10.96 MiB/s, done.
Resolving deltas: 100% (25/25), done.
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ ls
music_to_my_ears/
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ cd music_to_my_ears
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun/music_to_my_ears (main)
$ ls
README.md  main_notebook.ipynb  preprocessing_notebook.ipynb  tmp/
images/    modeling_notebook.ipynb  presentation.pdf
```

Clones files/commit graph in local repository

# git clone

Now a copy of repo in my github



git clone address\_forked\_repo

```
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ git clone https://github.com/admveen/music_to_my_ears.git
Cloning into 'music_to_my_ears'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 50 (delta 25), reused 8 (delta 5), pack-reused 0
Receiving objects: 100% (50/50), 8.78 MiB | 10.96 MiB/s, done.
Resolving deltas: 100% (25/25), done.
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ ls
music_to_my_ears/
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun
$ cd music_to_my_ears
(base)
prave@LAPTOP-JA891BKF MINGW64 ~/Cloning_Repo_Fun/music_to_my_ears (main)
$ ls
README.md  main_notebook.ipynb  preprocessing_notebook.ipynb  tmp/
images/    modeling_notebook.ipynb  presentation.pdf
```

Clones files/commit graph in local repository

Now, can work on project safely!



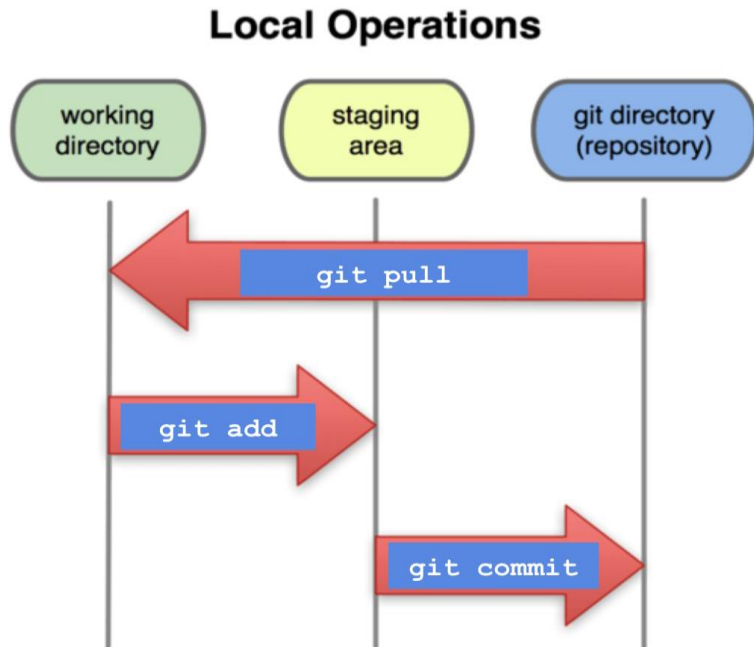
# git pull

Someone pushes change to your remote repo.

You want to update your local repo.

`git pull remote_name local_branch`

Fetches changes in remote branch and merges in your local branch.



# So...

- That was a lot.
- We'll get some practice in subsequent labs.
- Can only learn git by doing it.
- Later: will learn collaborating with git, branch management, reverting git state.