Berkeley
MSSE
MOLECULAR SCIENCE &
SOFTWARE ENGINEERING

Lecture 12:

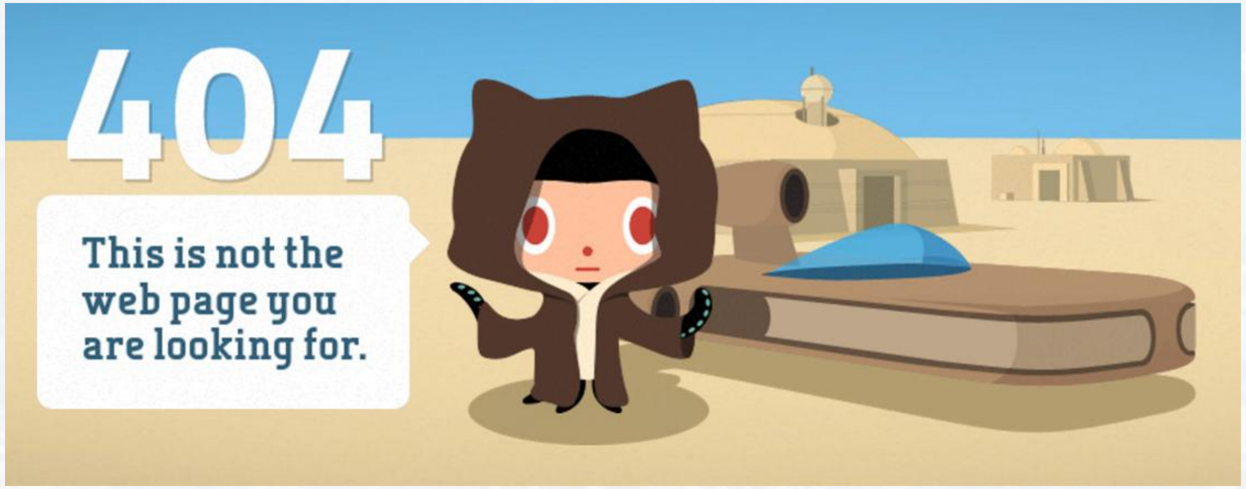
Git and GitHub

Markus Hohle

University California, Berkeley

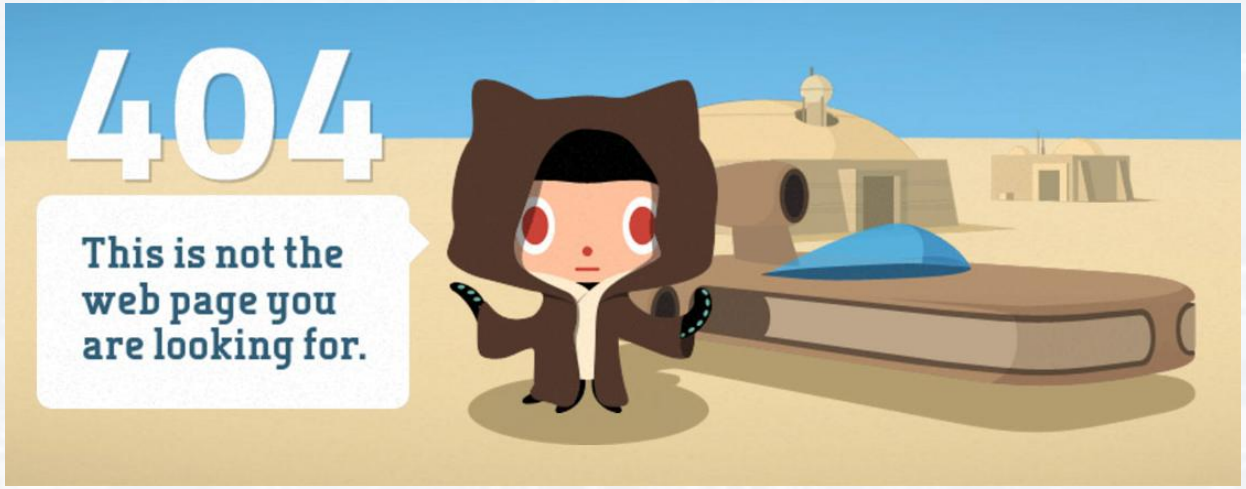
Python for Molecular Sciences

MSSE 272, 3 Units



Outline

- **Motivation**
- **Git**
  - installation and VS Code
  - status, add and commit
  - compare differences
  - restoring from index, stage and commit
- **Git Hub**
  - create your repo
  - invite a co worker
  - push
  - fetch, merge, pull

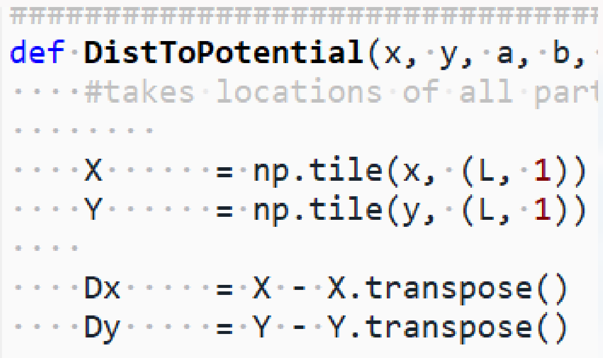


Outline

- **Motivation**
- **Git**
  - installation and VS Code
  - status, add and commit
  - compare differences
  - restoring from index, stage and commit
- **Git Hub**
  - create your repo
  - invite a co worker
  - push
  - fetch, merge, pull

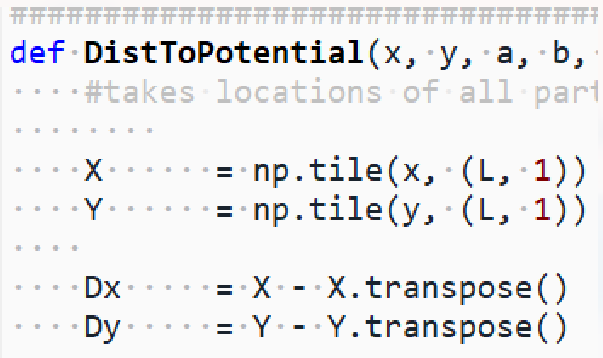
Git & GitHub

When working on a project: → many different versions



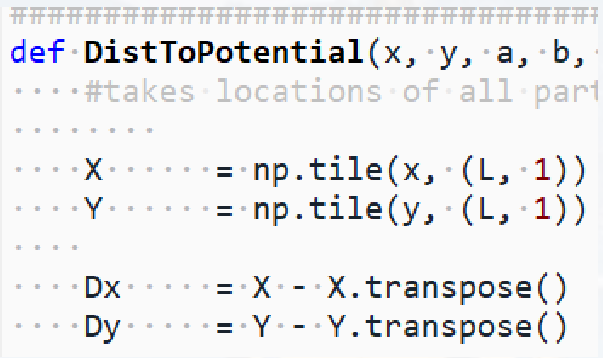
```
#####################################
def DistToPotential(x, y, a, b,
    #takes locations of all part
        
    X      = np.tile(x, (L, 1))
    Y      = np.tile(y, (L, 1))
    
    Dx     = X - X.transpose()
    Dy     = Y - Y.transpose()
```

version 1



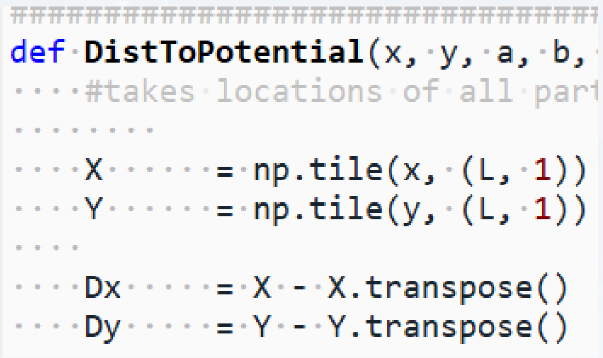
```
#####################################
def DistToPotential(x, y, a, b,
    #takes locations of all part
        
    X      = np.tile(x, (L, 1))
    Y      = np.tile(y, (L, 1))
    
    Dx     = X - X.transpose()
    Dy     = Y - Y.transpose()
```

version 1a



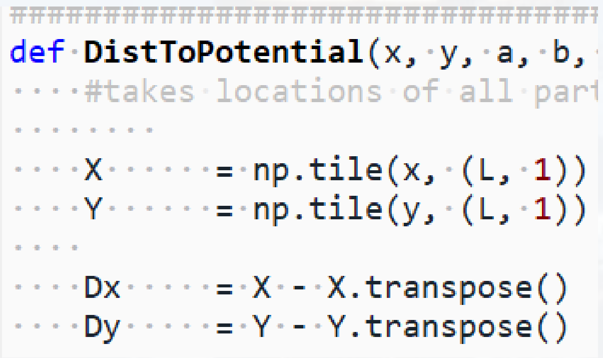
```
#####################################
def DistToPotential(x, y, a, b,
    #takes locations of all part
        
    X      = np.tile(x, (L, 1))
    Y      = np.tile(y, (L, 1))
    
    Dx     = X - X.transpose()
    Dy     = Y - Y.transpose()
```

version 2



```
#####################################
def DistToPotential(x, y, a, b,
    #takes locations of all part
        
    X      = np.tile(x, (L, 1))
    Y      = np.tile(y, (L, 1))
    
    Dx     = X - X.transpose()
    Dy     = Y - Y.transpose()
```

version 2a



```
#####################################
def DistToPotential(x, y, a, b,
    #takes locations of all part
        
    X      = np.tile(x, (L, 1))
    Y      = np.tile(y, (L, 1))
    
    Dx     = X - X.transpose()
    Dy     = Y - Y.transpose()
```

version 3

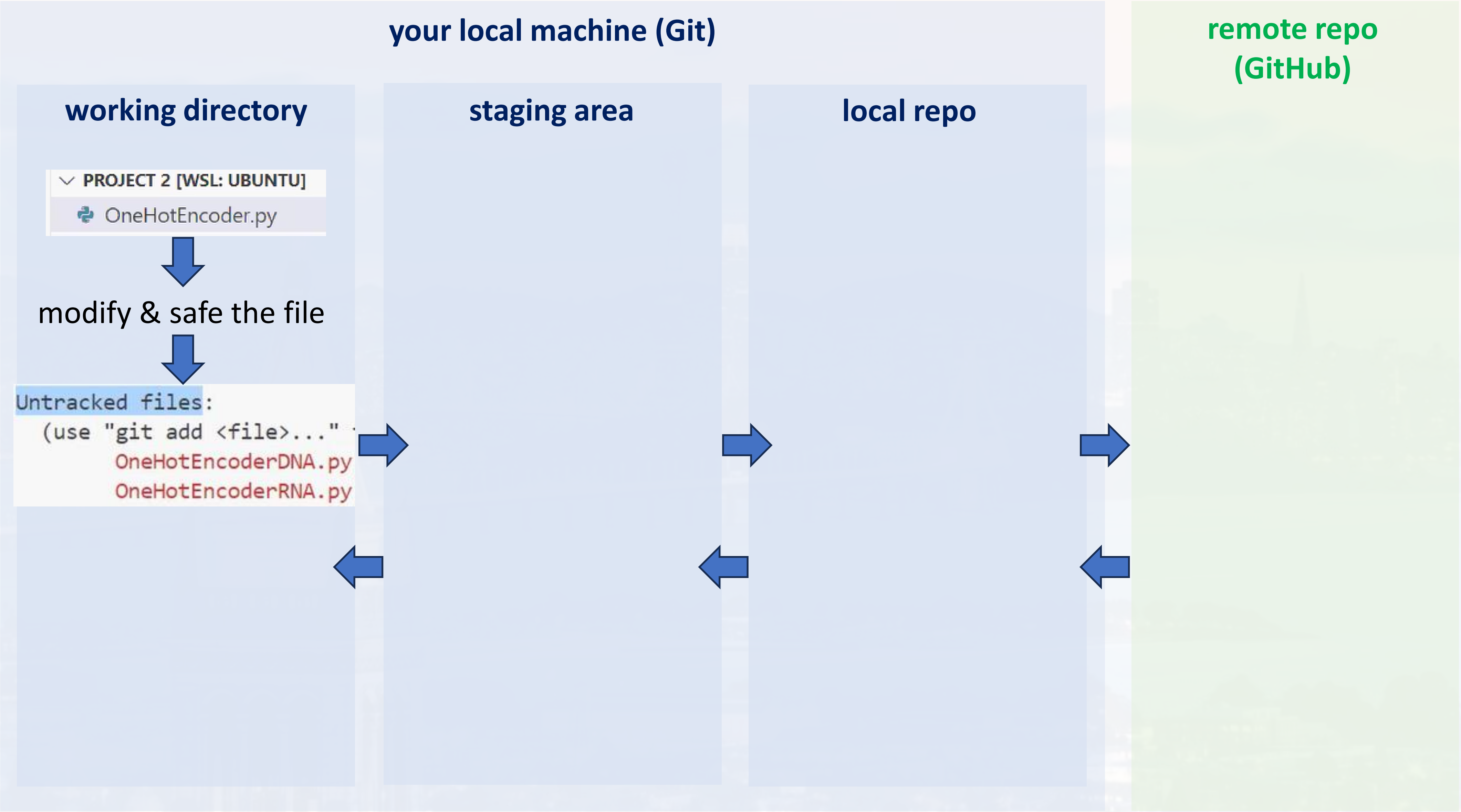
- easy to get **lost**
- What are the **differences between** the different **versions**?
- What if it turned out that **version 3c** was actually **better than 1f**?
- Sharing & exchanging with **co-workers**?

When working on a project: → many different versions

Version control system

- keeps track of **different** versions
- keeps track of **changes**
- **coordinates** work with co-workers
- **stores** data efficiently

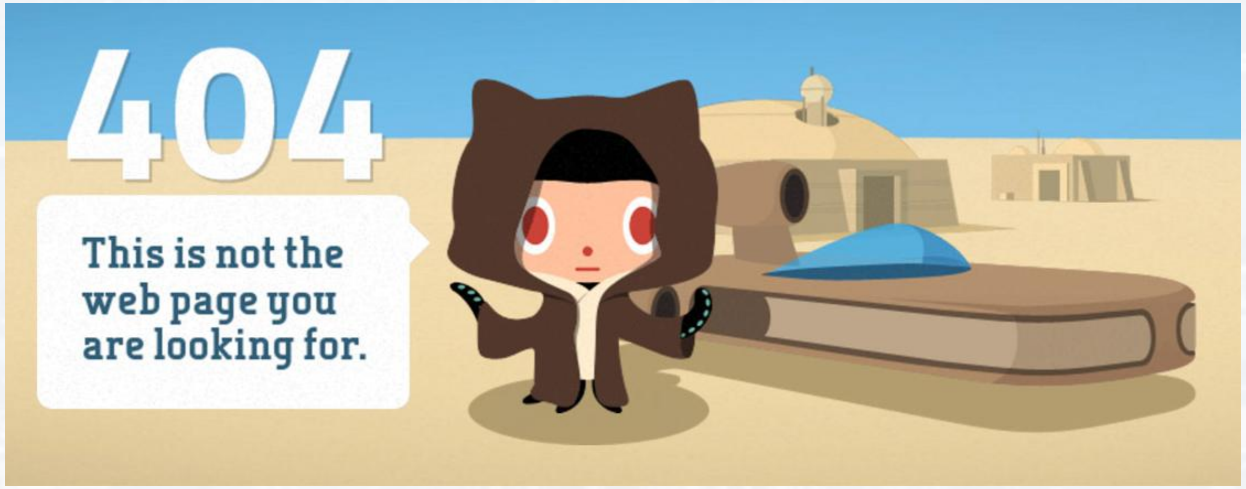
structure



PROJECT 2 [WSL: UBUNTU]
OneHotEncoder.py

modify & safe the file

```
Untracked files:
  (use "git add <file>..." 
          OneHotEncoderDNA.py
          OneHotEncoderRNA.py
```



Outline

- Motivation
- **Git**
    - **installation and VS Code**
    - status, add and commit
    - compare differences
    - restoring from index, stage and commit
- Git Hub
    - create your repo
    - invite a co worker
    - push
    - fetch, merge, pull

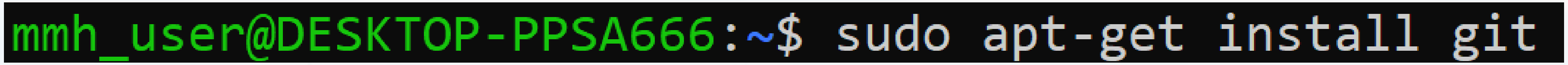
Git & GitHub

installation and VS Code

1) open WSL terminal

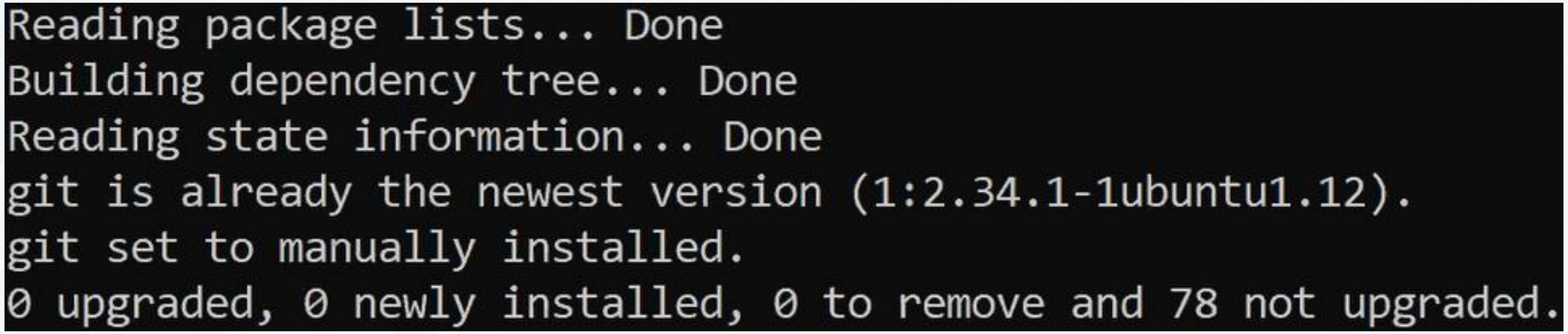
note: Git should be installed already, nevertheless, let’s run the following line:

2) install Git



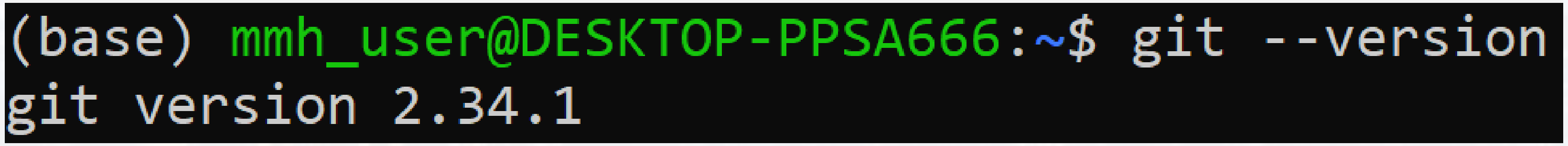
```
mmh_user@DESKTOP-PPSA666:~$ sudo apt-get install git
```

3) confirm your password



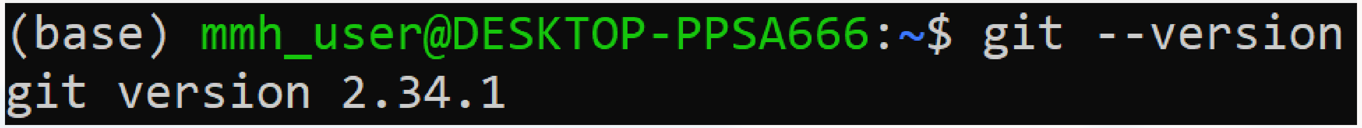
```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.12).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 78 not upgraded.
```

4) check version:



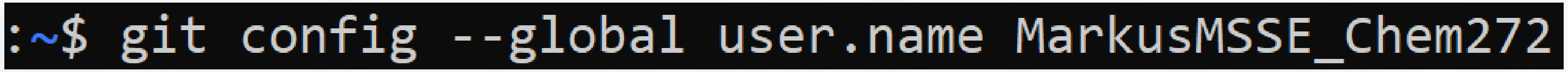
```
(base) mmh_user@DESKTOP-PPSA666:~$ git --version
git version 2.34.1
```

4) check version:

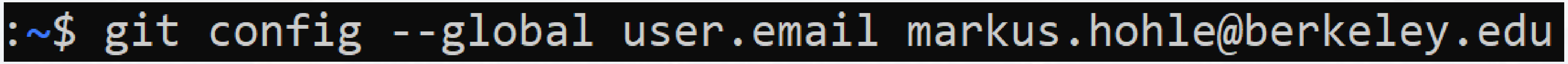


```
(base) mmh_user@DESKTOP-PPSA666:~$ git --version
git version 2.34.1
```

5) configure git: set your email address and user name

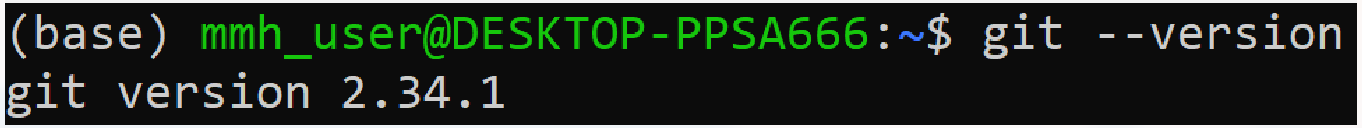


```
:~$ git config --global user.name MarkusMSSE_Chem272
```



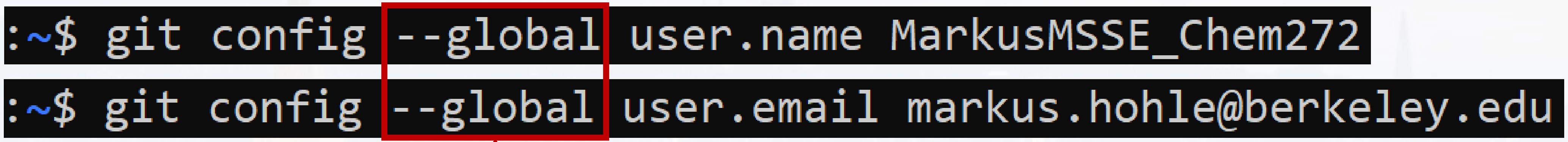
```
:~$ git config --global user.email markus.hohle@berkeley.edu
```

4) check version:



```
(base) mmh_user@DESKTOP-PPSA666:~$ git --version
git version 2.34.1
```

5) configure git: set your email address and user name

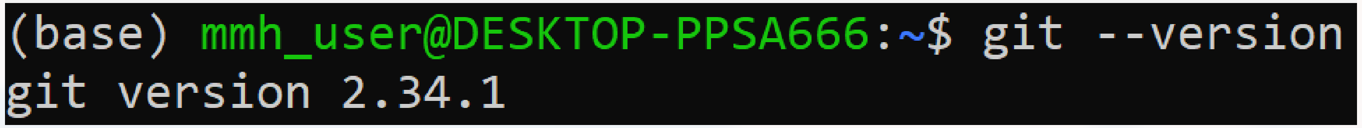


```
:~$ git config --global user.name MarkusMSSE_Chem272
:~$ git config --global user.email markus.hohle@berkeley.edu
```

Git has three levels of access:

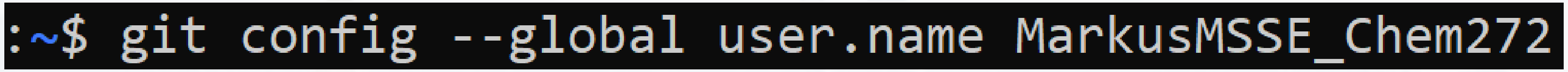
1 **Local**:	only for a **specific repo**
2 **Global**:	across all repos for **currently logged-in user** (here: *mmh_user*)
3 **System**:	**all users** and repos **on your machine** (for example if you are admin or group leader etc)

4) check version:

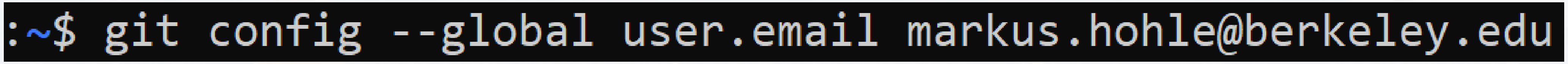


```
(base) mmh_user@DESKTOP-PPSA666:~$ git --version
git version 2.34.1
```

5) configure git: set your email address and user name

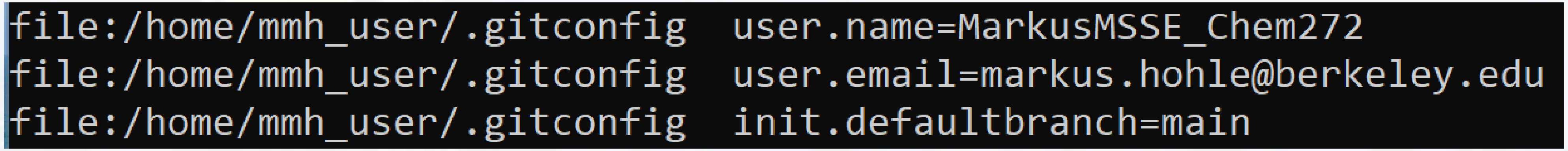


```
:~$ git config --global user.name MarkusMSSE_Chem272
```



```
:~$ git config --global user.email markus.hohle@berkeley.edu
```

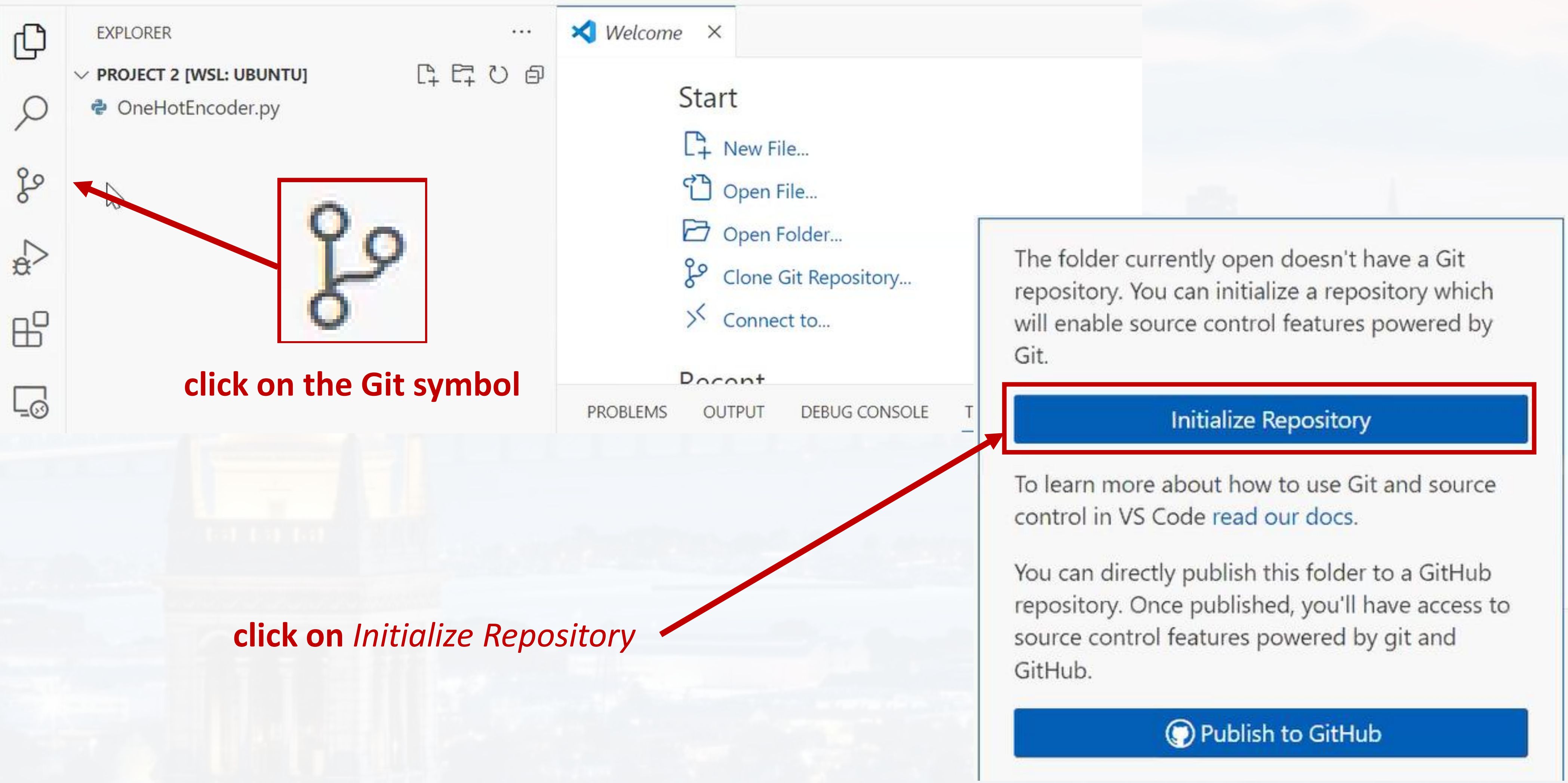
6) check config: make sure everything is correct



```
file:/home/mmh_user/.gitconfig  user.name=MarkusMSSE_Chem272
file:/home/mmh_user/.gitconfig  user.email=markus.hohle@berkeley.edu
file:/home/mmh_user/.gitconfig  init.defaultbranch=main
```

7) Git and VS Code:

- open VS code
- navigate to the folder which you want to use for your project



The folder currently open doesn't have a Git repository. You can initialize a repository which will enable source control features powered by Git.

To learn more about how to use Git and source control in VS Code read our docs.

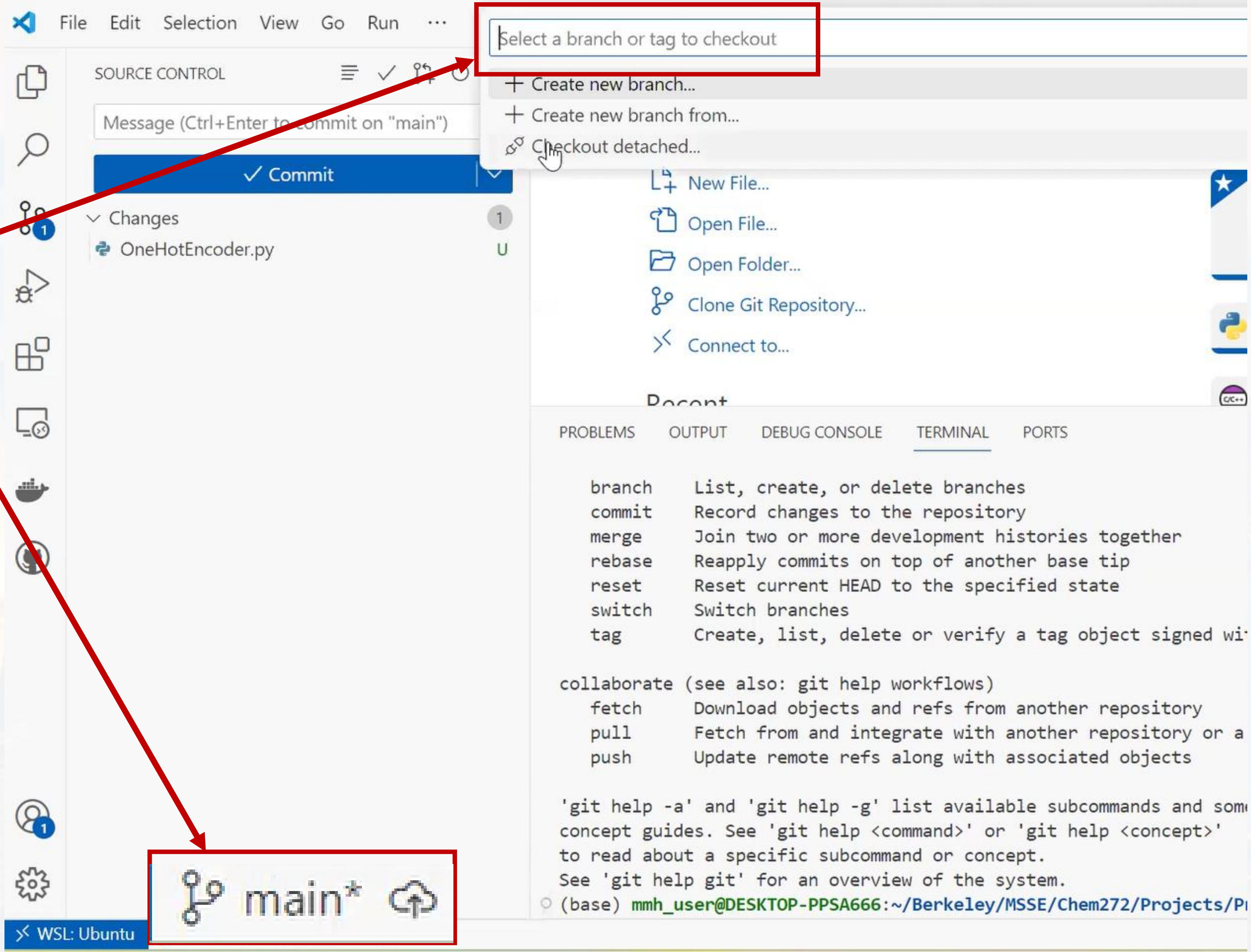
You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

click on the Git symbol

click on *Initialize Repository*

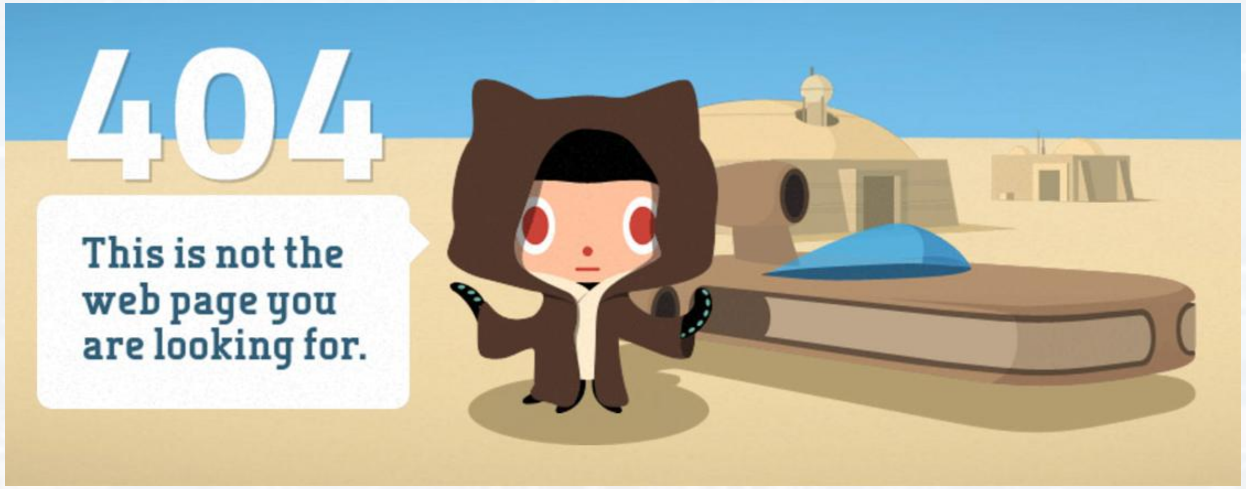
7) Git and VS Code:

default branch name is *"main"*, which can be renamed



```
   branch     List, create, or delete branches
   commit     Record changes to the repository
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   reset      Reset current HEAD to the specified state
   switch     Switch branches
   tag        Create, list, delete or verify a tag object signed wi

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and som
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
(base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/P
```

watch:
12a_Installing_Git
12b_Git_and_VS_Code_in_WSL



Outline

- Motivation
- **Git**
  - installation and VS Code
  - **status, add and commit**
  - compare differences
  - restoring from index, stage and commit
- Git Hub
  - create your repo
  - invite a co worker
  - push
  - fetch, merge, pull

structure

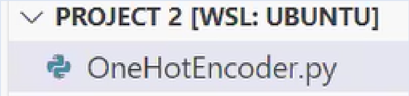
your local machine (Git)

remote repo (GitHub)

working directory

staging area

local repo



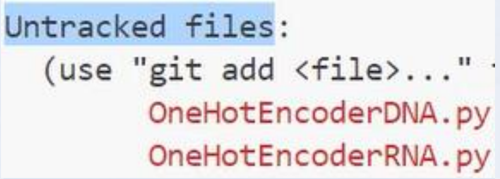
PROJECT 2 [WSL: UBUNTU]
OneHotEncoder.py

modify & safe the file

Git notices the change!

`git status`

always check Git status
→ which files have been modified
→ which files are in the staging area

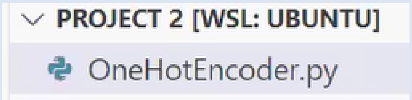


```
Untracked files:
  (use "git add <file>..." 
          OneHotEncoderDNA.py
          OneHotEncoderRNA.py
```

structure

your local machine (Git)

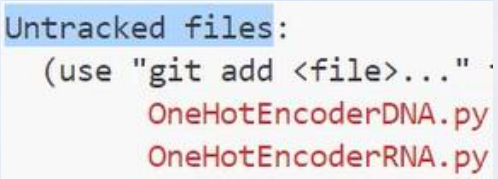
working directory



PROJECT 2 [WSL: UBUNTU]
OneHotEncoder.py

modify & safe the file

```
git status
```



```
Untracked files:
  (use "git add <file>..." 
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

staging area

adds files to staging area
→ file is ready for review

```
git add
```

```
git add OneHotEncoderDNA.py
```

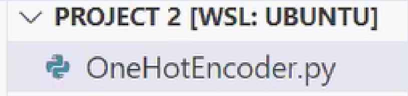
local repo

remote repo (GitHub)

structure

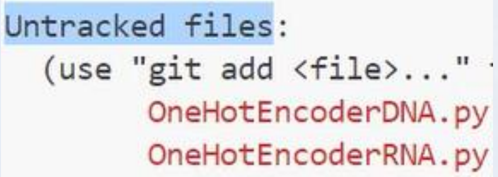
your local machine (Git)

working directory


PROJECT 2 [WSL: UBUNTU]
OneHotEncoder.py

modify & safe the file

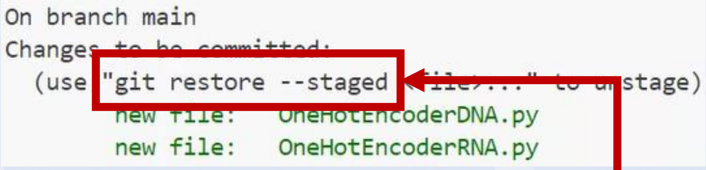
`git status`



```
Untracked files:
  (use "git add <file>..." 
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

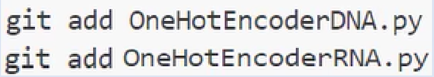
staging area

`git status`



```
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   OneHotEncoderDNA.py
        new file:   OneHotEncoderRNA.py
```

`git add`



```
git add OneHotEncoderDNA.py
git add OneHotEncoderRNA.py
```

local repo

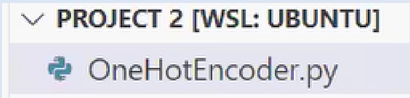
remote repo (GitHub)

tells you how to move version of staged file back to working directory!

structure

your local machine (Git)

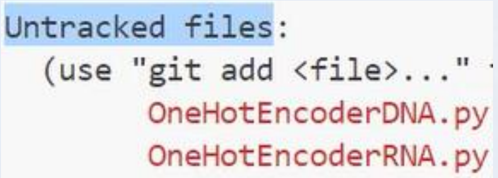
working directory



PROJECT 2 [WSL: UBUNTU]
OneHotEncoder.py

modify & safe the file

`git status`

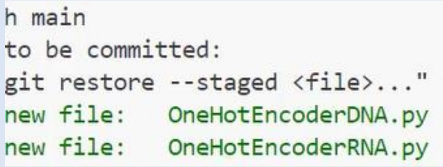


```
Untracked files:
  (use "git add <file>..." 
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

`git add`

`git restore --staged`

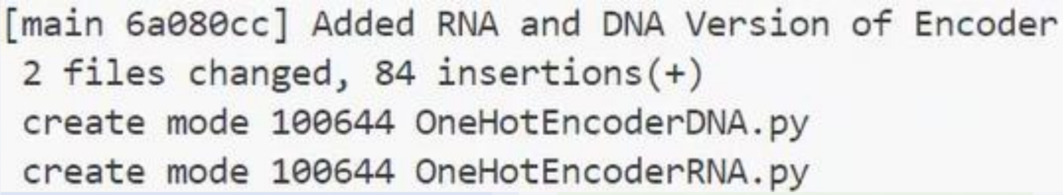
staging area



```
h main
to be committed:
git restore --staged <file>..."
new file:   OneHotEncoderDNA.py
new file:   OneHotEncoderRNA.py
```

`git commit`

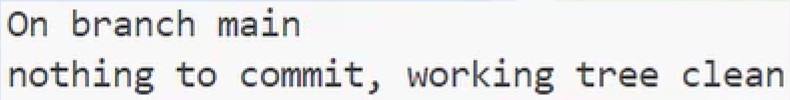
local repo



```
[main 6a080cc] Added RNA and DNA Version of Encoder
 2 files changed, 84 insertions(+)
 create mode 100644 OneHotEncoderDNA.py
 create mode 100644 OneHotEncoderRNA.py
```

files are now moved from staging area to the local repository

`git status`



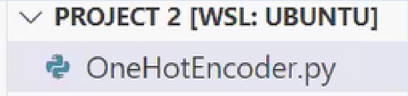
```
On branch main
nothing to commit, working tree clean
```

remote repo (GitHub)

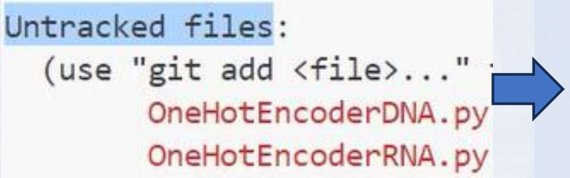
structure

your local machine (Git)

working directory



modify & safe the file

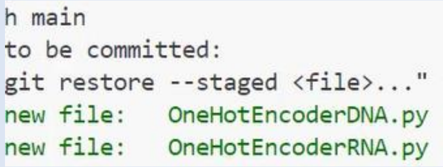


```
Untracked files:
  (use "git add <file>..."
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

`git add`

`git restore --staged`

staging area



```
h main
to be committed:
git restore --staged <file>..."
new file:   OneHotEncoderDNA.py
new file:   OneHotEncoderRNA.py
```

local repo

remote repo (GitHub)

structure

your local machine (Git)

remote repo (GitHub)

working directory

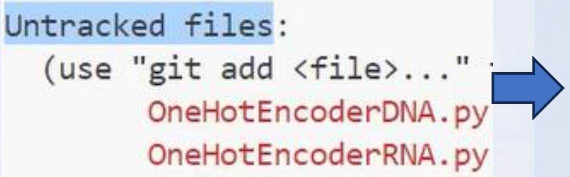
staging area

local repo

PROJECT 2 [WSL: UBUNTU]

OneHotEncoder.py

modify & safe the file



```
Untracked files:
  (use "git add <file>..."
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

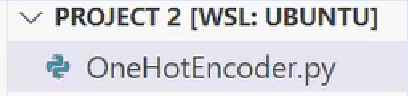
```
git add
```

```
git restore --staged
```

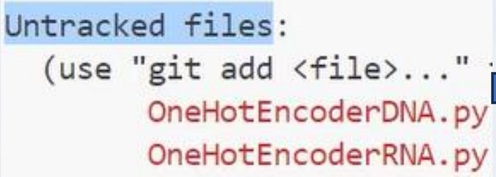
structure

your local machine (Git)

working directory



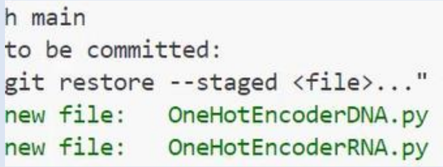
modify & safe the file



```
Untracked files:
  (use "git add <file>..."
        OneHotEncoderDNA.py
        OneHotEncoderRNA.py
```

`git add`

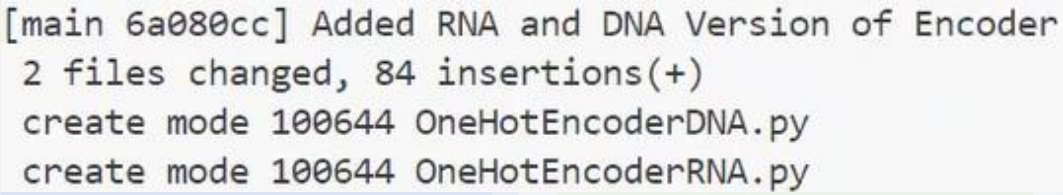
staging area



```
h main
to be committed:
git restore --staged <file>..."
new file:   OneHotEncoderDNA.py
new file:   OneHotEncoderRNA.py
```

`git commit`

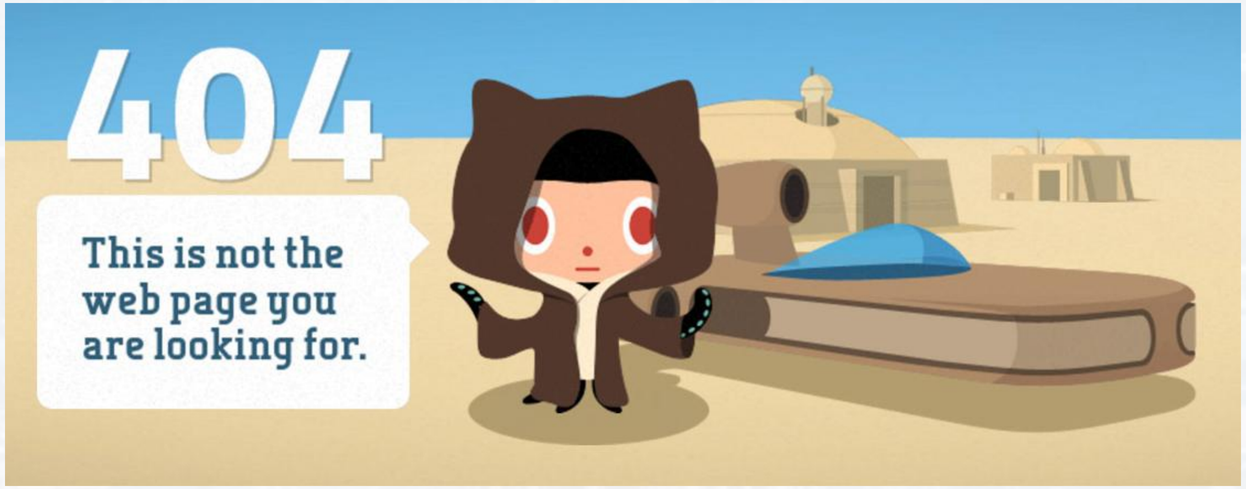
local repo



```
[main 6a080cc] Added RNA and DNA Version of Encoder
 2 files changed, 84 insertions(+)
 create mode 100644 OneHotEncoderDNA.py
 create mode 100644 OneHotEncoderRNA.py
```

remote repo (GitHub)

watch:
12c_Git_commit



Outline

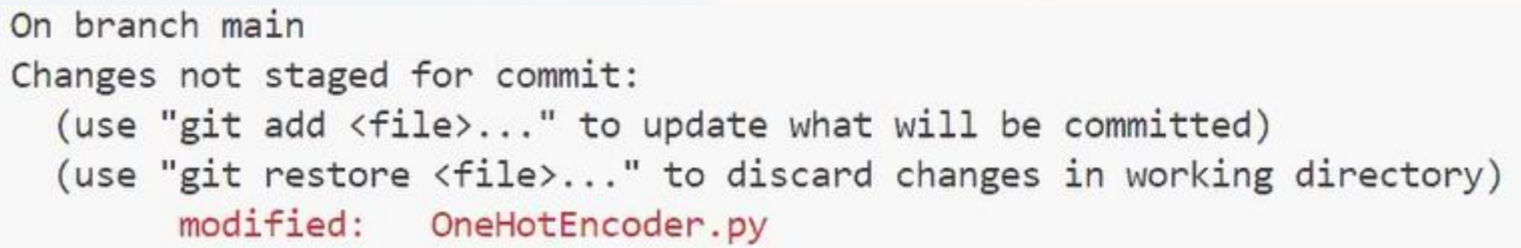
- Motivation
- **Git**
  - installation and VS Code
  - status, add and commit
  - **compare differences**
  - restoring from index, stage and commit
- Git Hub
  - create your repo
  - invite a co worker
  - push
  - fetch, merge, pull

Git & GitHub

compare differences

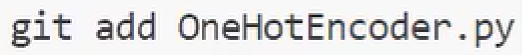
1) A new file (`OneHotEncoder.py`) in the working directory

```
git status
```



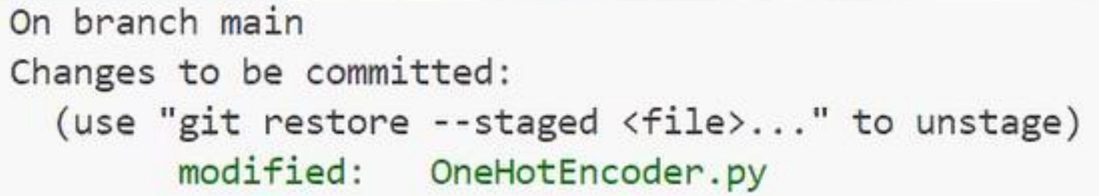
```
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   OneHotEncoder.py
```

```
git add
```



```
git add OneHotEncoder.py
```

```
git status
```



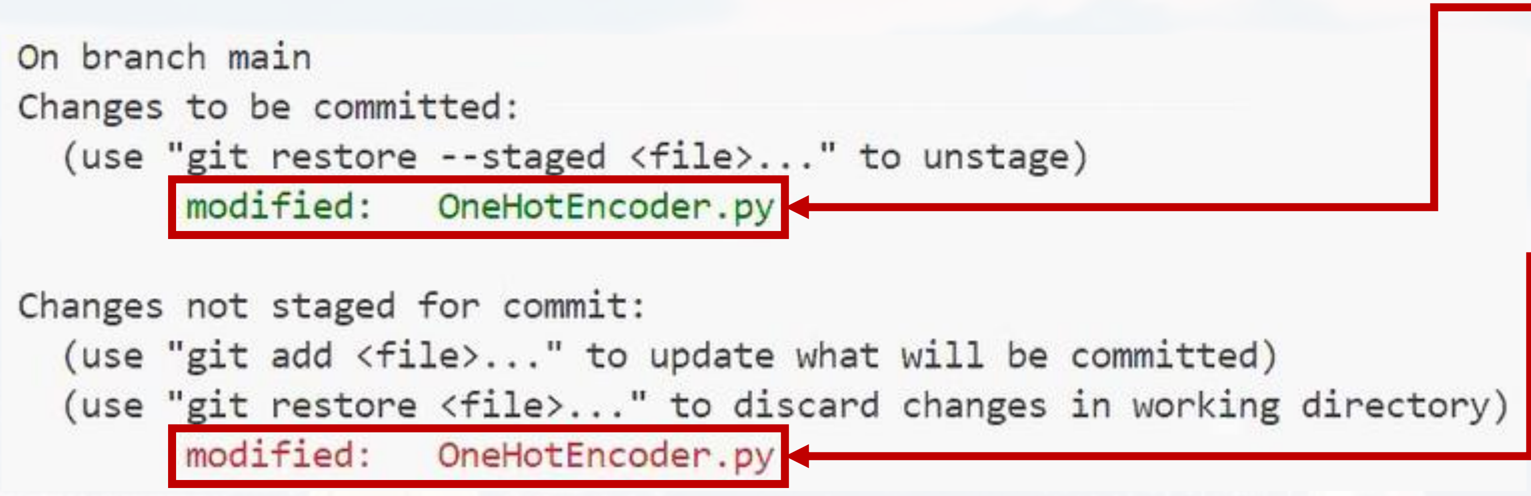
```
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   OneHotEncoder.py
```

2) We kept working on the file and have two versions now!

compare differences

2) We kept working on the file and have two versions now!

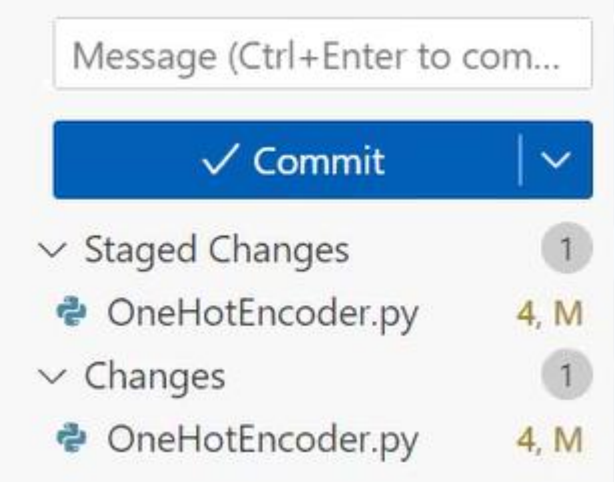
```
git status
```



```
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   OneHotEncoder.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   OneHotEncoder.py
```

the **previous** version which is located at the **stage area**

the **current** version which is located at the **work folder**



Message (Ctrl+Enter to com...

✓ Commit

Staged Changes 1

OneHotEncoder.py 4, M

Changes 1

OneHotEncoder.py 4, M

before adding or committing to the repo
→ review changes!

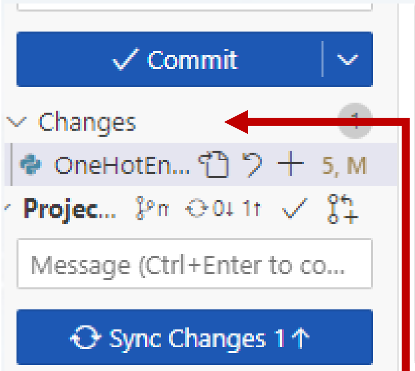
compare differences

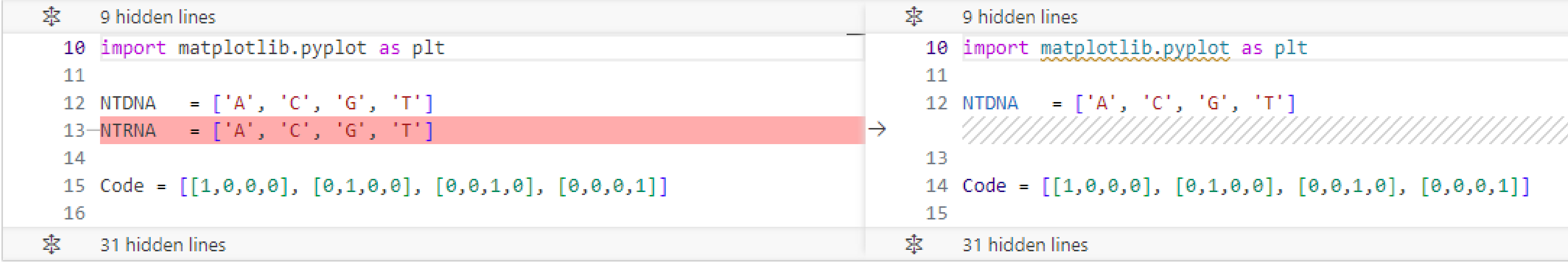
2) We kept working on the file and have two versions now!

before adding or committing to the repo
→ review changes!

staged version has one more line
(which got removed, hence highlighted in red)

folder version has one fewer line

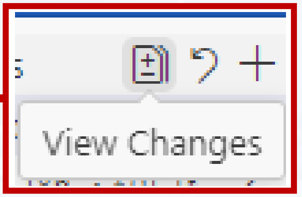




```
9 hidden lines
import matplotlib.pyplot as plt

NTDNA   = ['A', 'C', 'G', 'T']
NTRNA   = ['A', 'C', 'G', 'T']

Code = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

31 hidden lines
```

```
9 hidden lines
import matplotlib.pyplot as plt

NTDNA   = ['A', 'C', 'G', 'T']

Code = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

31 hidden lines
```

click on *Changes* or *Staged Changes*



Git & GitHub

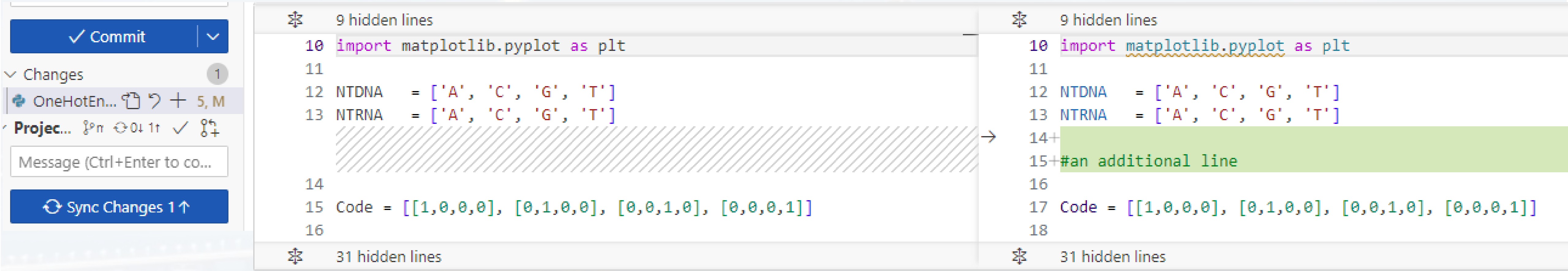
compare differences

2) We kept working on the file and have two versions now!

before adding or committing to the repo
→ review changes!

staged version

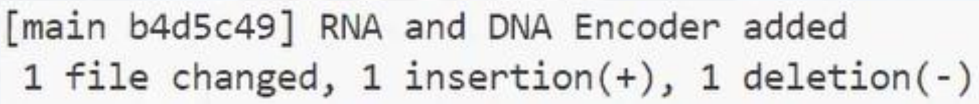
folder version has one more line
(hence highlighted in green)



```
import matplotlib.pyplot as plt

NTDNA    = ['A', 'C', 'G', 'T']
NTRNA    = ['A', 'C', 'G', 'T']

Code = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

```

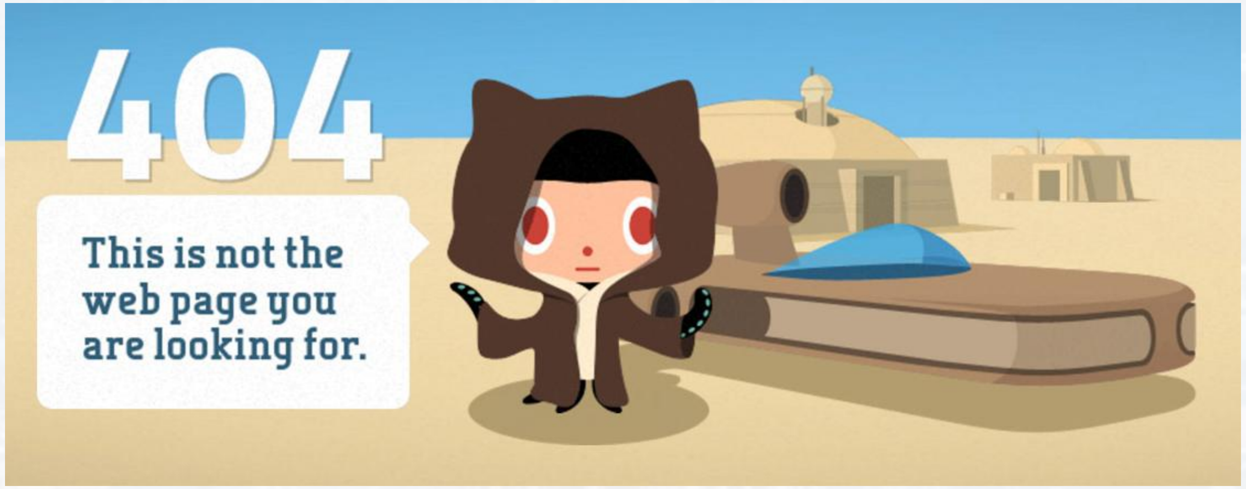
```
import matplotlib.pyplot as plt

NTDNA    = ['A', 'C', 'G', 'T']
NTRNA    = ['A', 'C', 'G', 'T']
+
+#an additional line

Code = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

```

once we add or commit, Git tells us the changes:



```
[main b4d5c49] RNA and DNA Encoder added
 1 file changed, 1 insertion(+), 1 deletion(-)
```

watch:
12d_Git_CompareDifferences



Outline

- Motivation
- **Git**
  - installation and VS Code
  - status, add and commit
  - compare differences
  - **restoring from index, stage and commit**
- Git Hub
  - create your repo
  - invite a co worker
  - push
  - fetch, merge, pull

restoring from index, stage and commit

We can always rescue any file from any state of our project!

1) changing file in working directory

   undo: `git restore`

2) `git add` file from working directory to stage area

   undo: `git restore --staged`

watch:
12e_Git_RestoreFromIndexAndStage

What if we have committed many different versions to the repo
but need to restore one particular version??

restoring from index, stage and commit

structure

your local machine (Git)

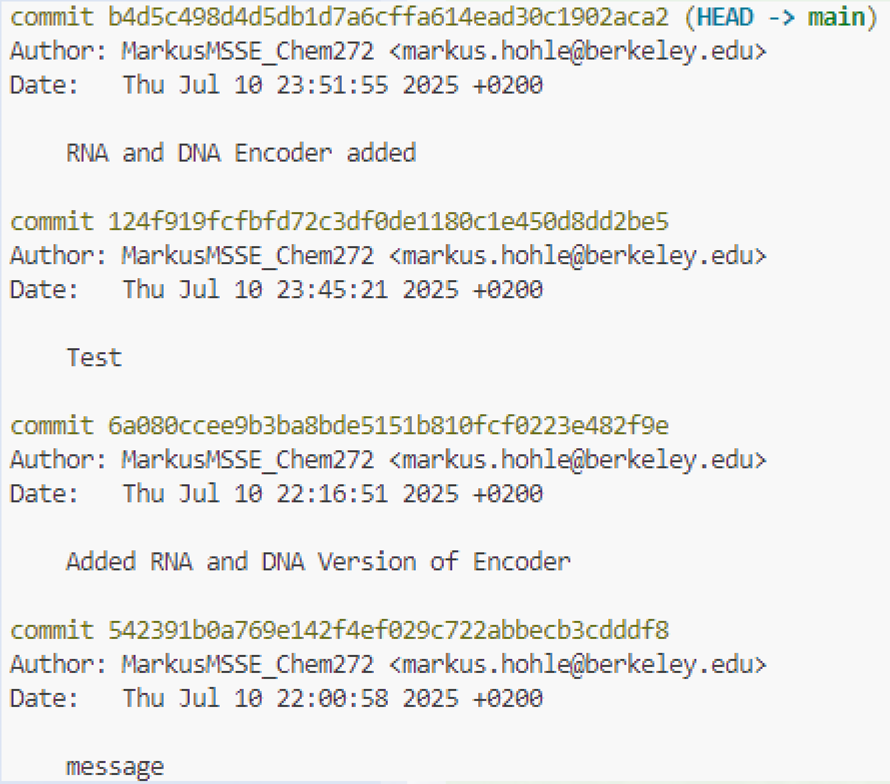
remote repo (GitHub)

working directory

staging area

local repo

git log

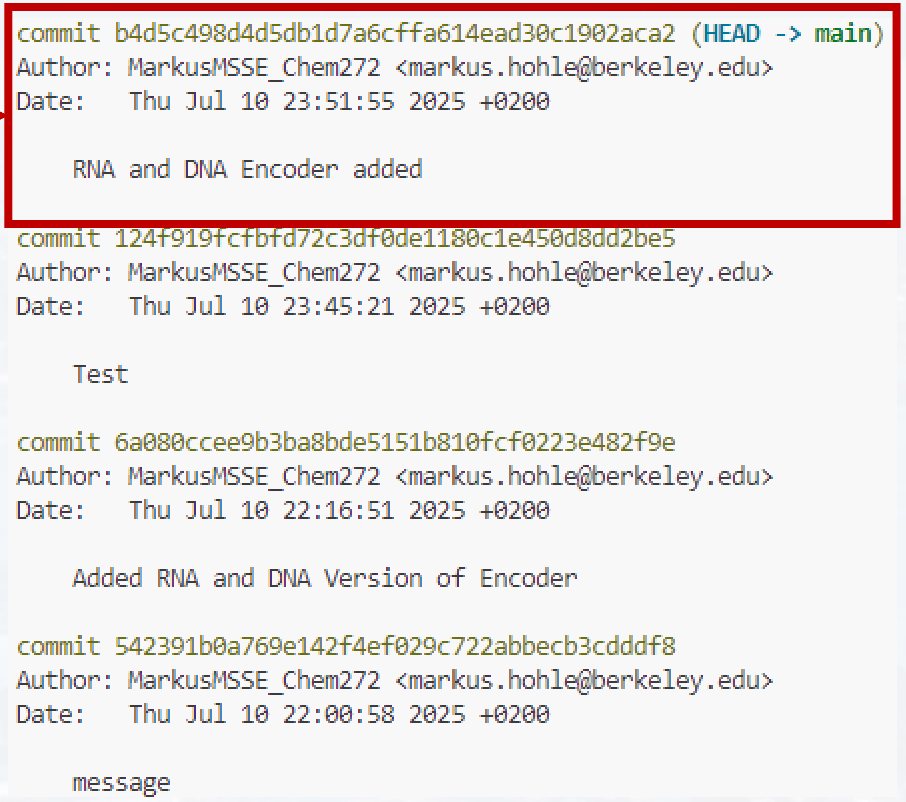


```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

restoring from index, stage and commit

The last commit (HEAD)

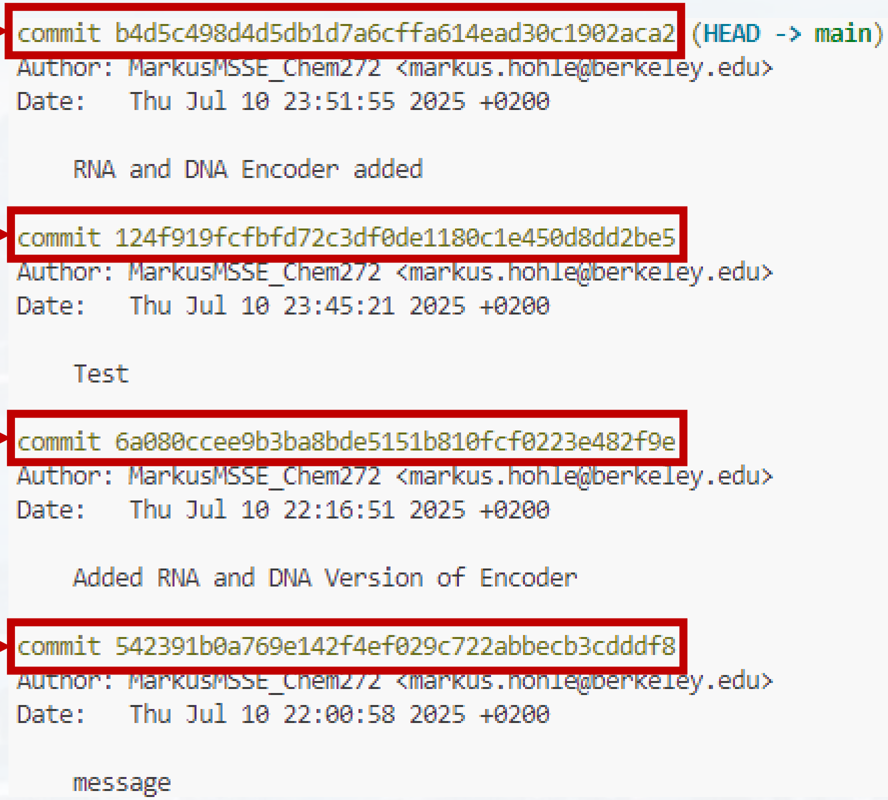
git log



```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

restoring from index, stage and commit

git log

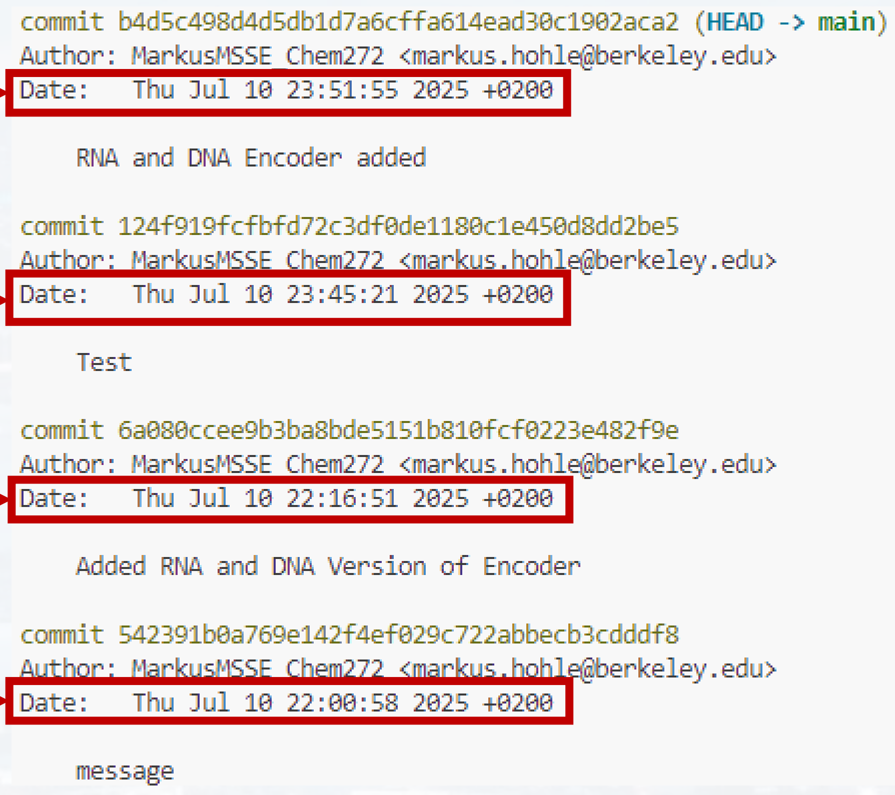


```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

each commit has an **individual ID**

restoring from index, stage and commit

```
git log
```

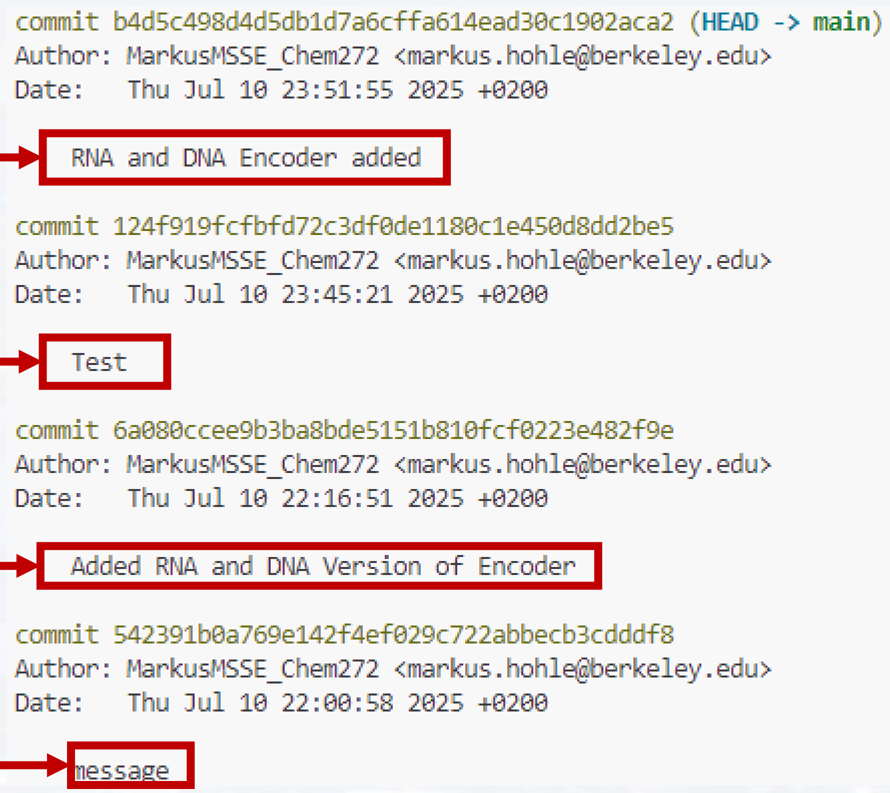


```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

each commit has a **date**

restoring from index, stage and commit

```
git log
```



```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

the **messages** you added are also stored

restoring from index, stage and commit

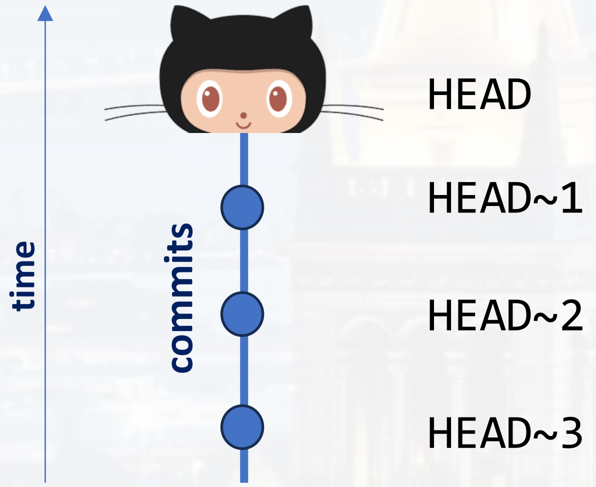
if we want to move a file from a given **commit back to the staging area**:

→ either referring to the **ID**

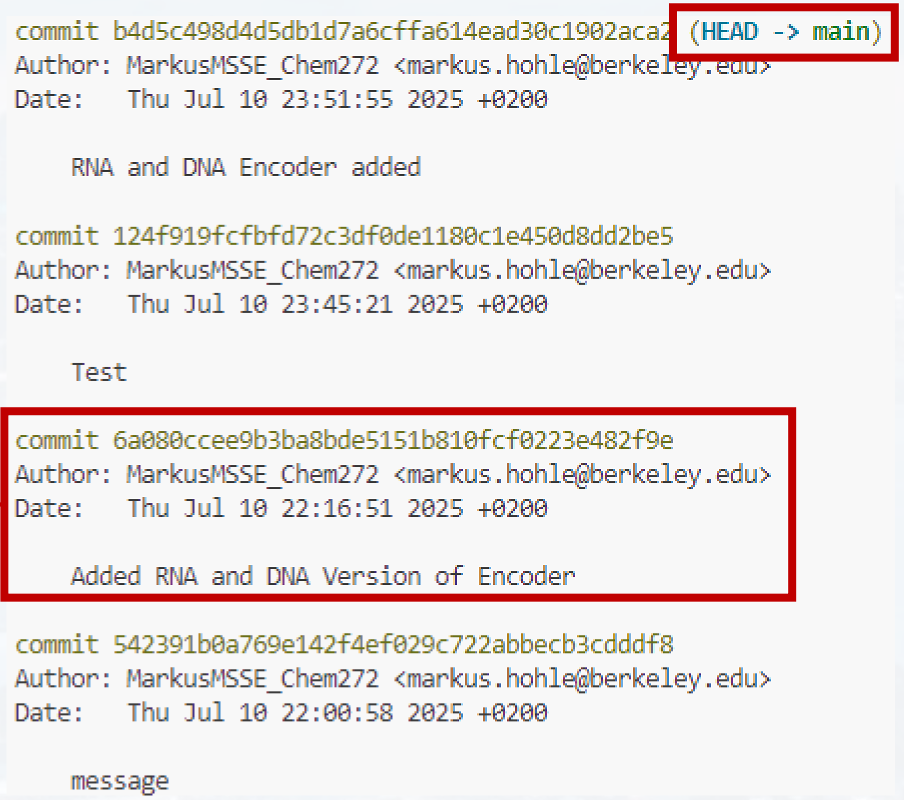
→ or to the **head**

```
git restore --source=HEAD~2 OneHotEncoder.py
```

two commits before the current one



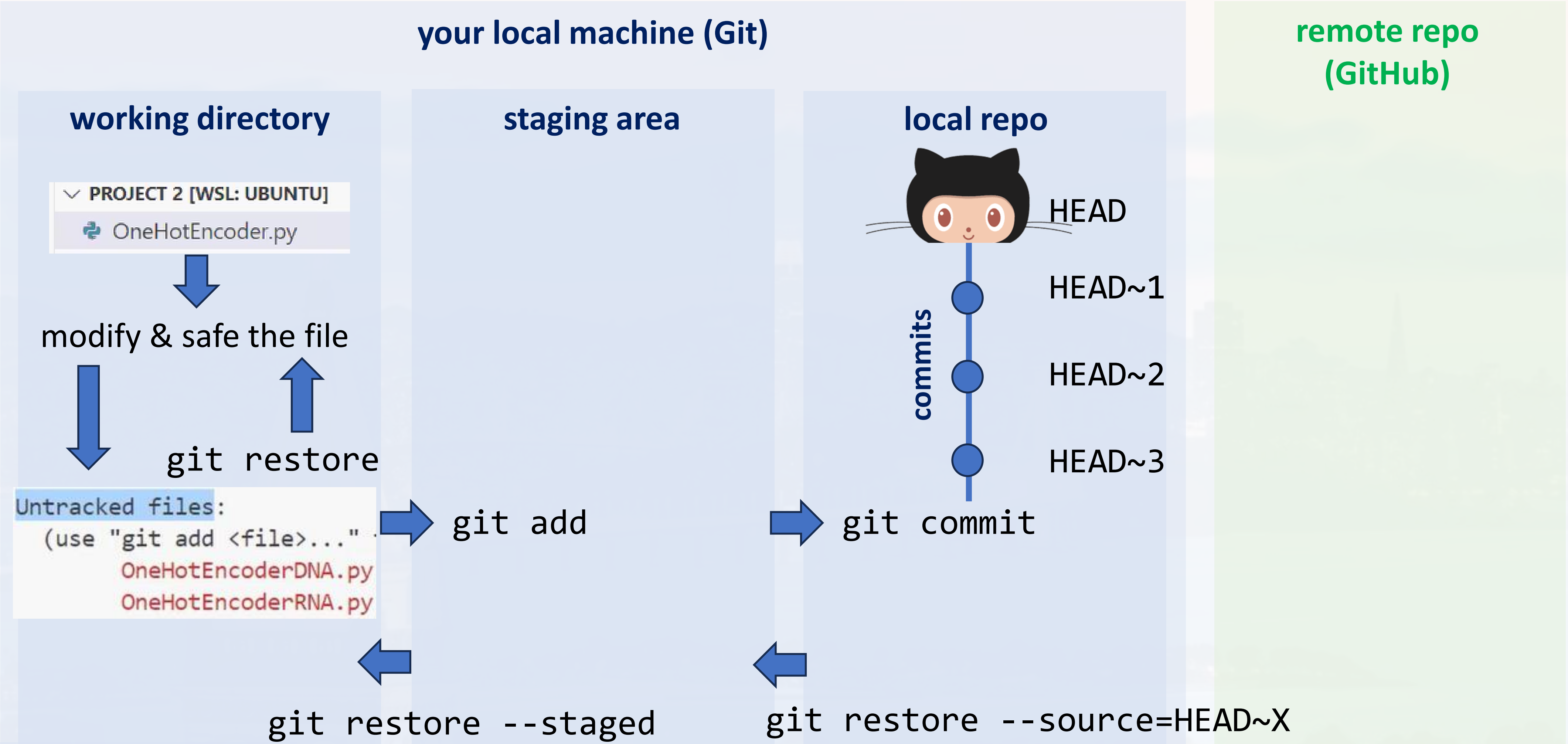
```
git log
```



```
commit b4d5c498d4d5db1d7a6cffa614ead30c1902aca2 (HEAD -> main)
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:51:55 2025 +0200

    RNA and DNA Encoder added

commit 124f919fcfbfd72c3df0de1180c1e450d8dd2be5
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 23:45:21 2025 +0200

    Test

commit 6a080ccee9b3ba8bde5151b810fcf0223e482f9e
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:16:51 2025 +0200

    Added RNA and DNA Version of Encoder

commit 542391b0a769e142f4ef029c722abbecb3cdddf8
Author: MarkusMSSE_Chem272 <markus.hohle@berkeley.edu>
Date:   Thu Jul 10 22:00:58 2025 +0200

    message
```

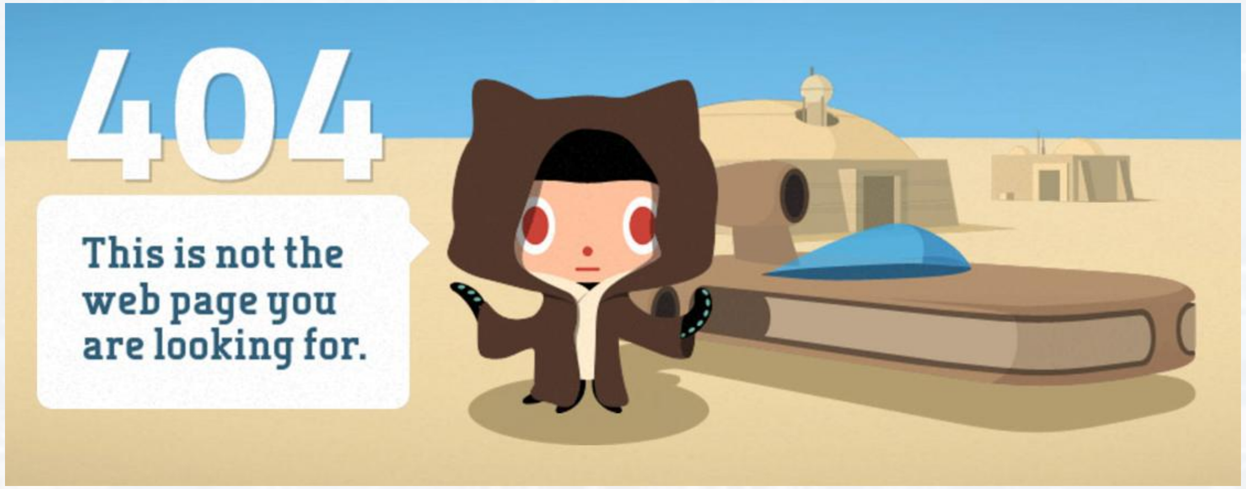
restoring from index, stage and commit

structure



```
Untracked files:
  (use "git add <file>..." 
          OneHotEncoderDNA.py
          OneHotEncoderRNA.py
```

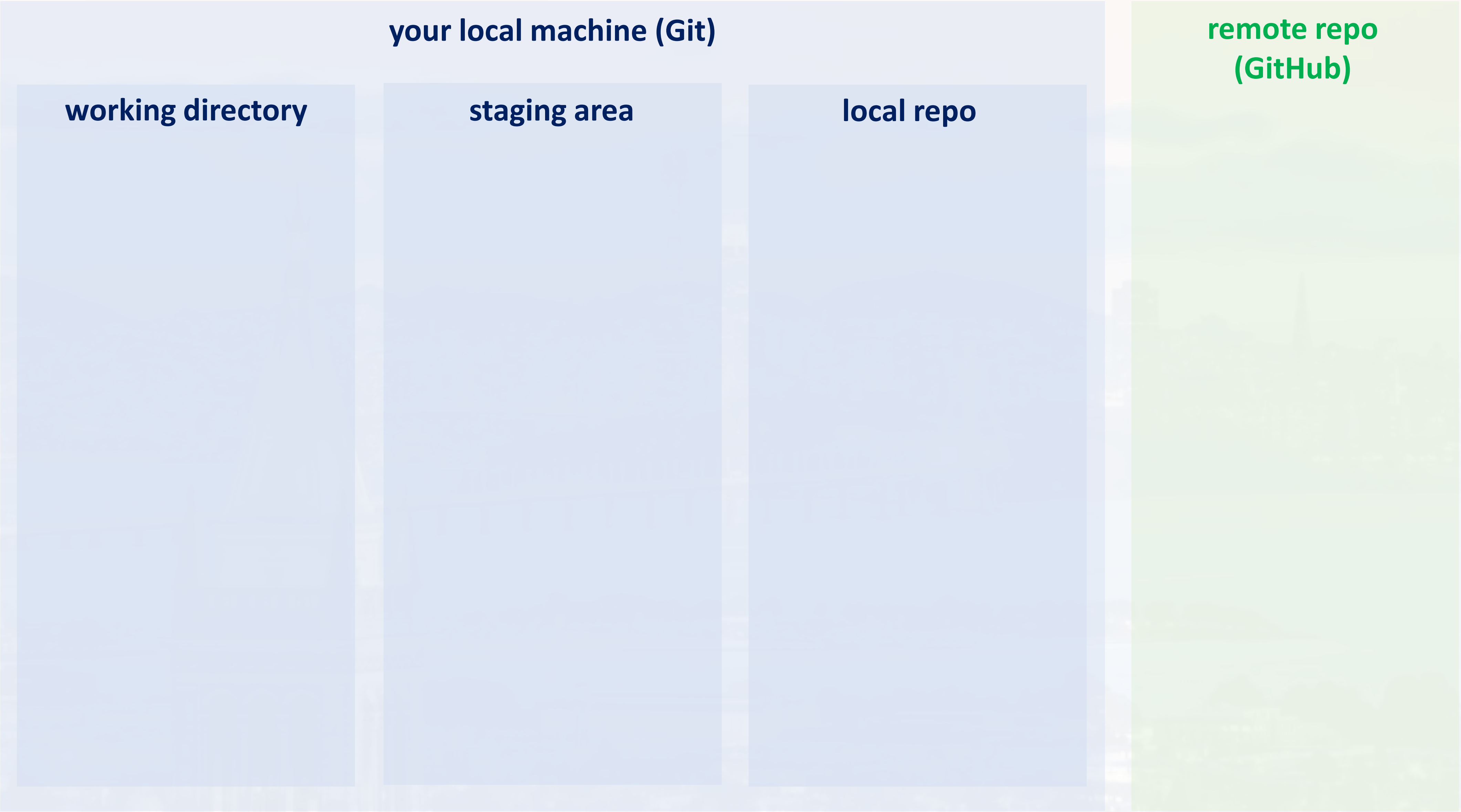
watch:
12e_Git_RestoreFromIndexAndStage
12f_Git_RestoreFromAnyCommit



Outline

- Motivation
- Git
  - installation and VS Code
  - status, add and commit
  - compare differences
  - restoring from index, stage and commit
- **Git Hub**
  - **create your repo**
  - invite a co worker
  - push
  - fetch, merge, pull

structure



your local machine (Git)

working directory

staging area

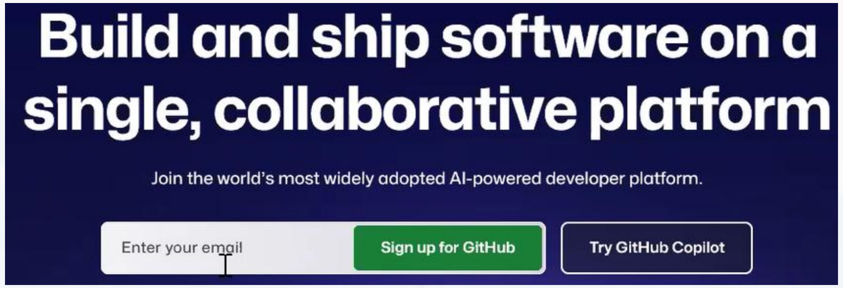
local repo

remote repo (GitHub)

Git & GitHub

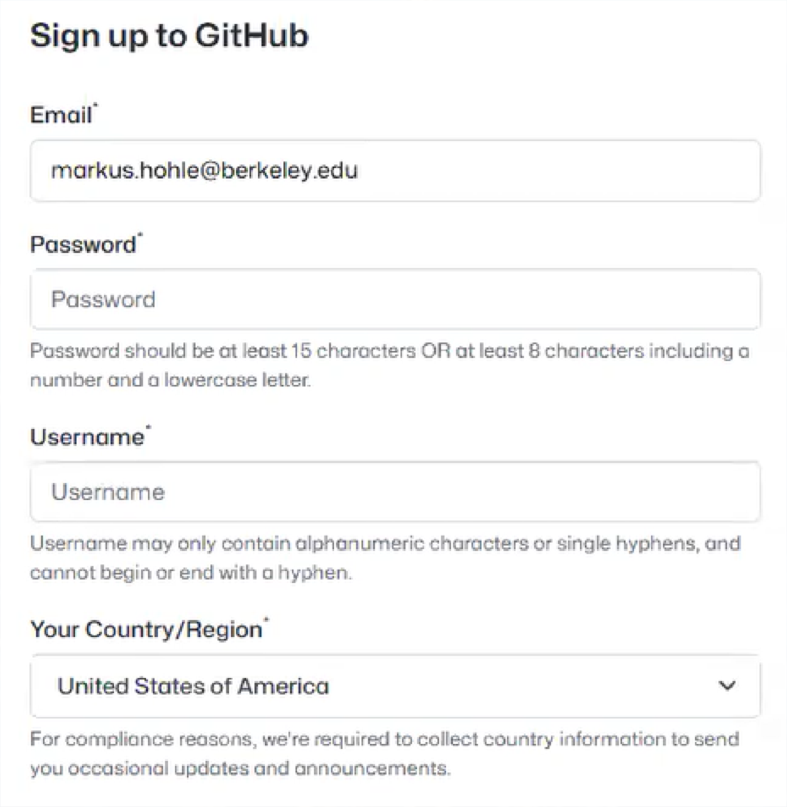
create your repo

got to **github.com**



Build and ship software on a single, collaborative platform

Join the world's most widely adopted AI-powered developer platform.



Sign up to GitHub

Email*

markus.hohle@berkeley.edu

Password*

Password

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

Username

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

United States of America

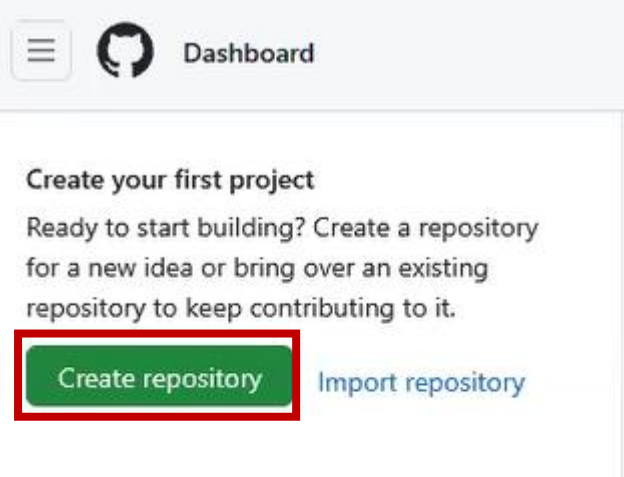
For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

register and confirm email address

remote repo (GitHub)

Git & GitHub

create new repository



Dashboard

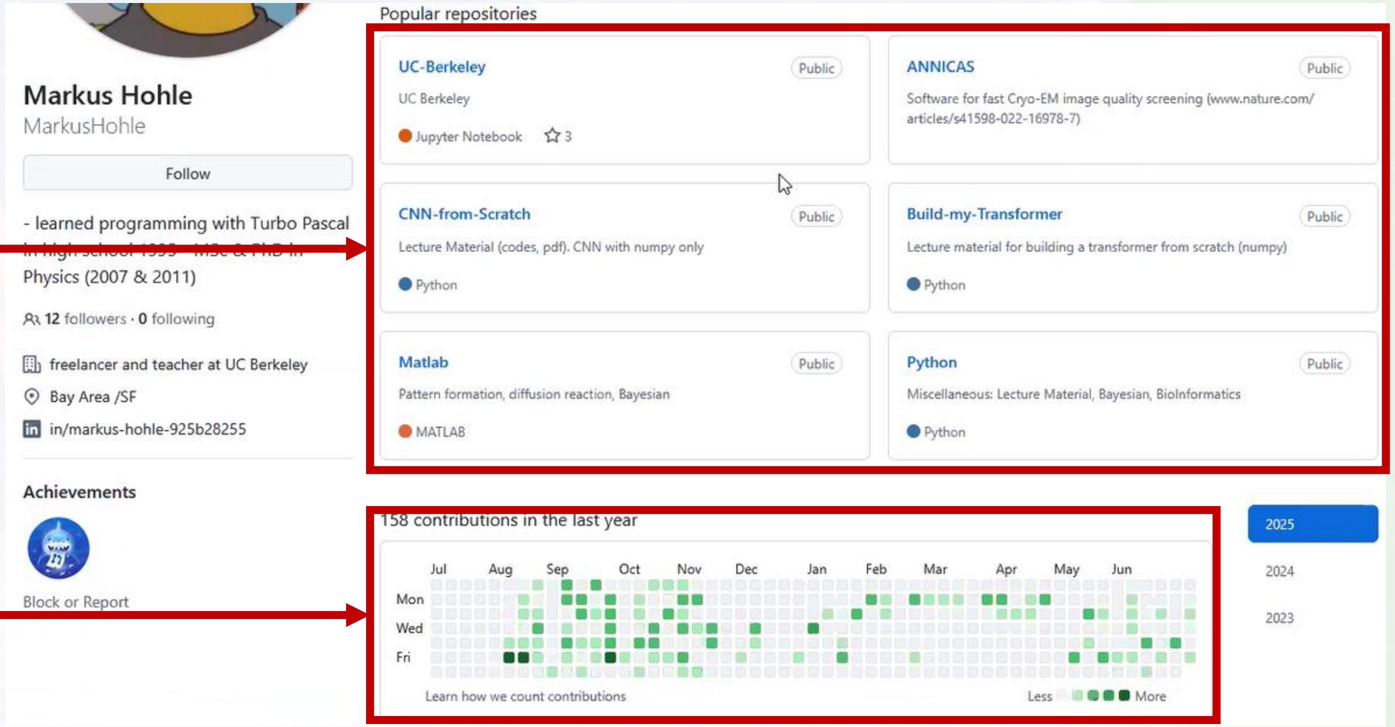
Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

remote repo (GitHub)

list of repos

your **garden** shows your activity: **keep it green!**



Markus Hohle
MarkusHohle

- learned programming with Turbo Pascal in high school 1995 - MSc & PhD in Physics (2007 & 2011)

12 followers · 0 following

freelancer and teacher at UC Berkeley

Bay Area /SF

in/markus-hohle-925b28255

Achievements

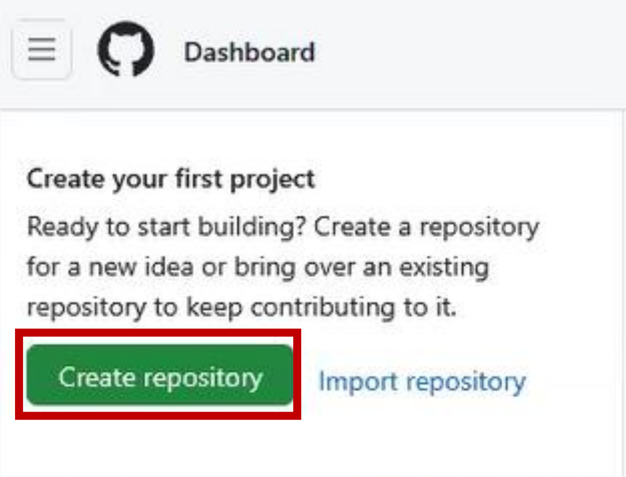
Block or Report

Popular repositories

UC-Berkeley Public UC Berkeley ● Jupyter Notebook ☆ 3	ANNICAS Public Software for fast Cryo-EM image quality screening (www.nature.com/articles/s41598-022-16978-7)
CNN-from-Scratch Public Lecture Material (codes, pdf). CNN with numpy only ● Python	Build-my-Transformer Public Lecture material for building a transformer from scratch (numpy) ● Python
Matlab Public Pattern formation, diffusion reaction, Bayesian ● MATLAB	Python Public Miscellaneous: Lecture Material, Bayesian, BioInformatics ● Python

158 contributions in the last year

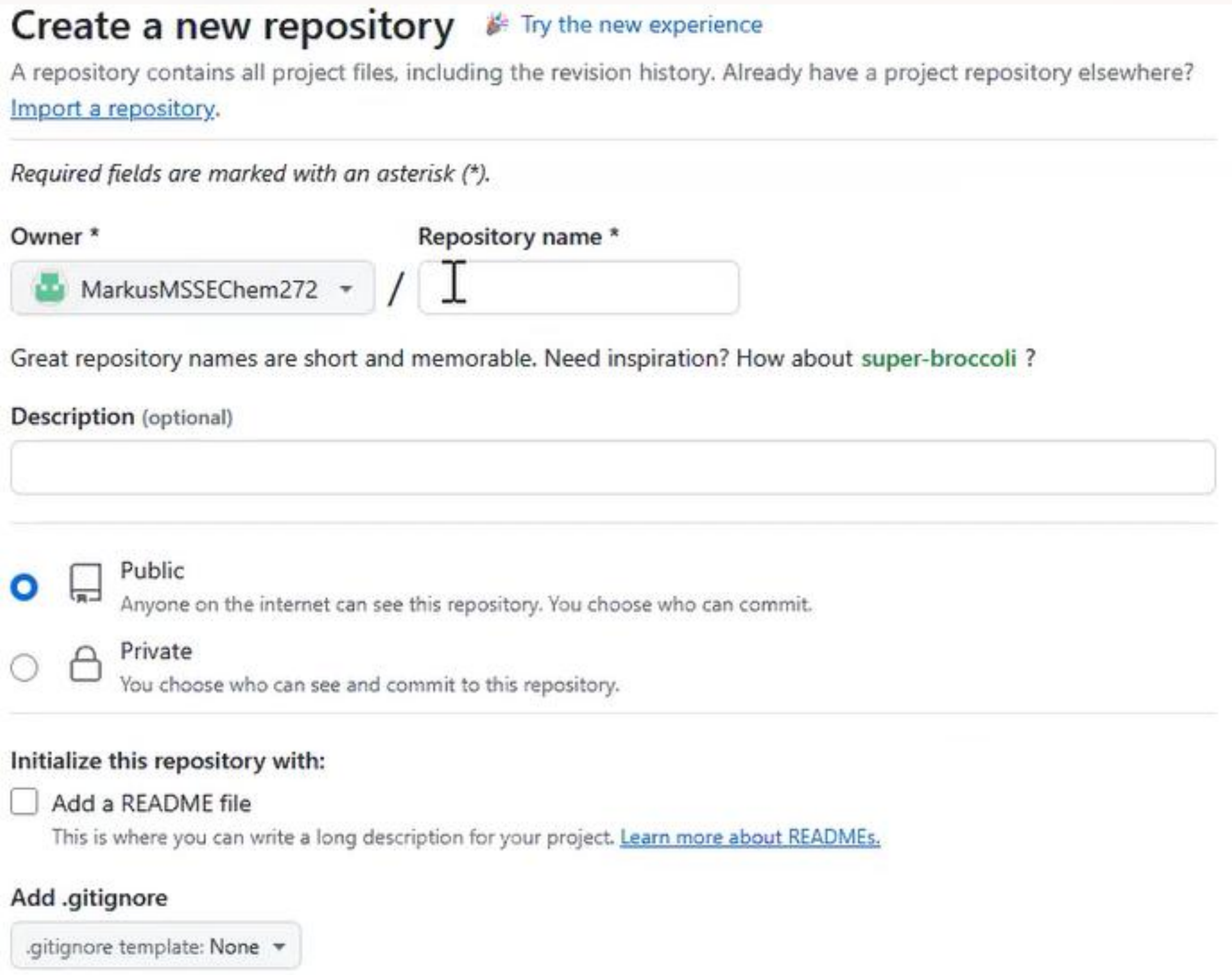
create new repository



Dashboard

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * MarkusMSSEChem272 / Repository name *

Great repository names are short and memorable. Need inspiration? How about super-broccoli ?

Description (optional)

● 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:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

create new repository

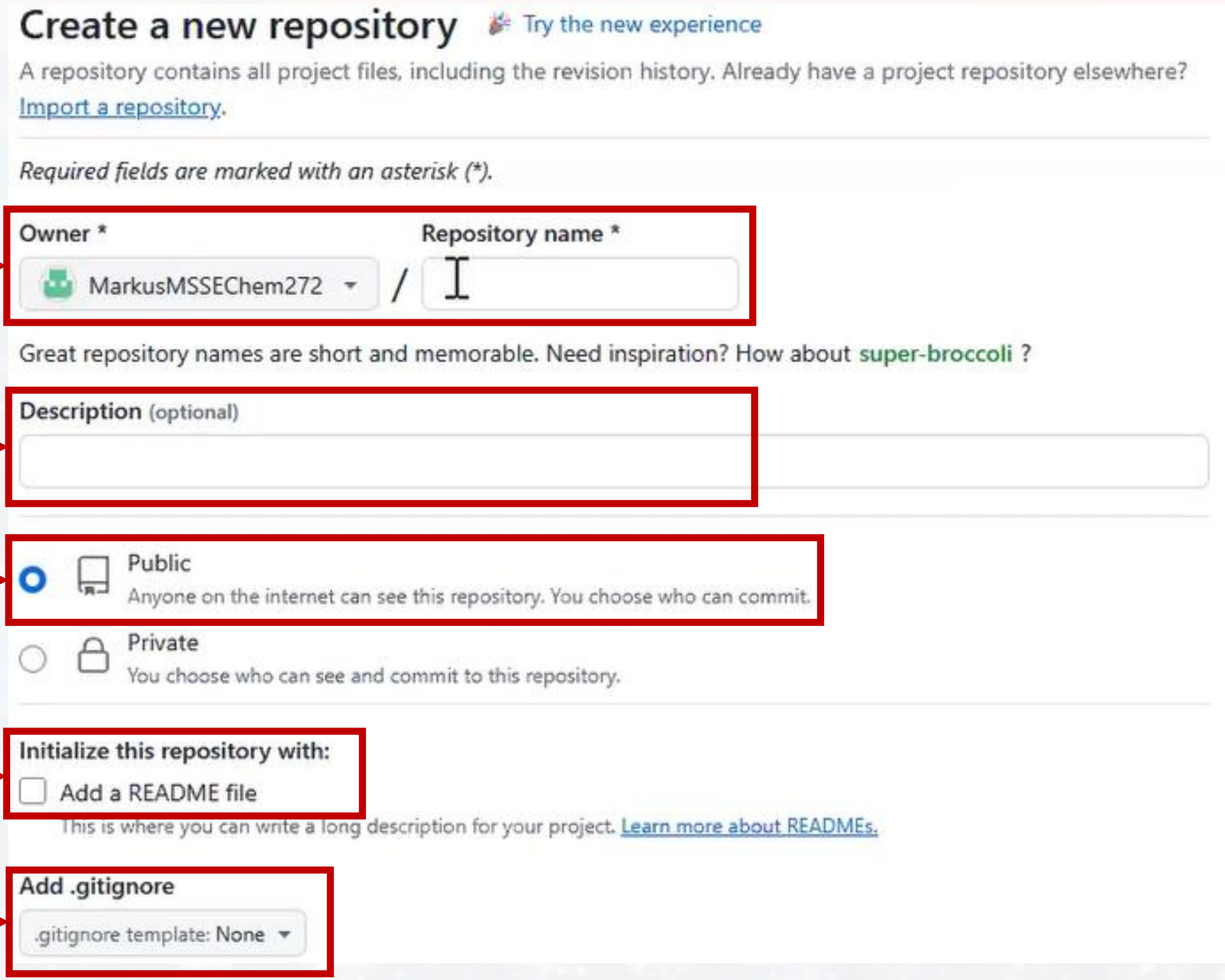
add a meaningful name…

…and discription…

public means **anyone** can see your repo

you can add more info in a ReadMe file

gitignore lists files and directories that will not be committed if you use the option -a



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * MarkusMSSEChem272 / Repository name *

Great repository names are short and memorable. Need inspiration? How about super-broccoli ?

Description (optional)

● 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:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

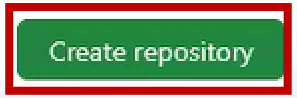
Add .gitignore

.gitignore template: None

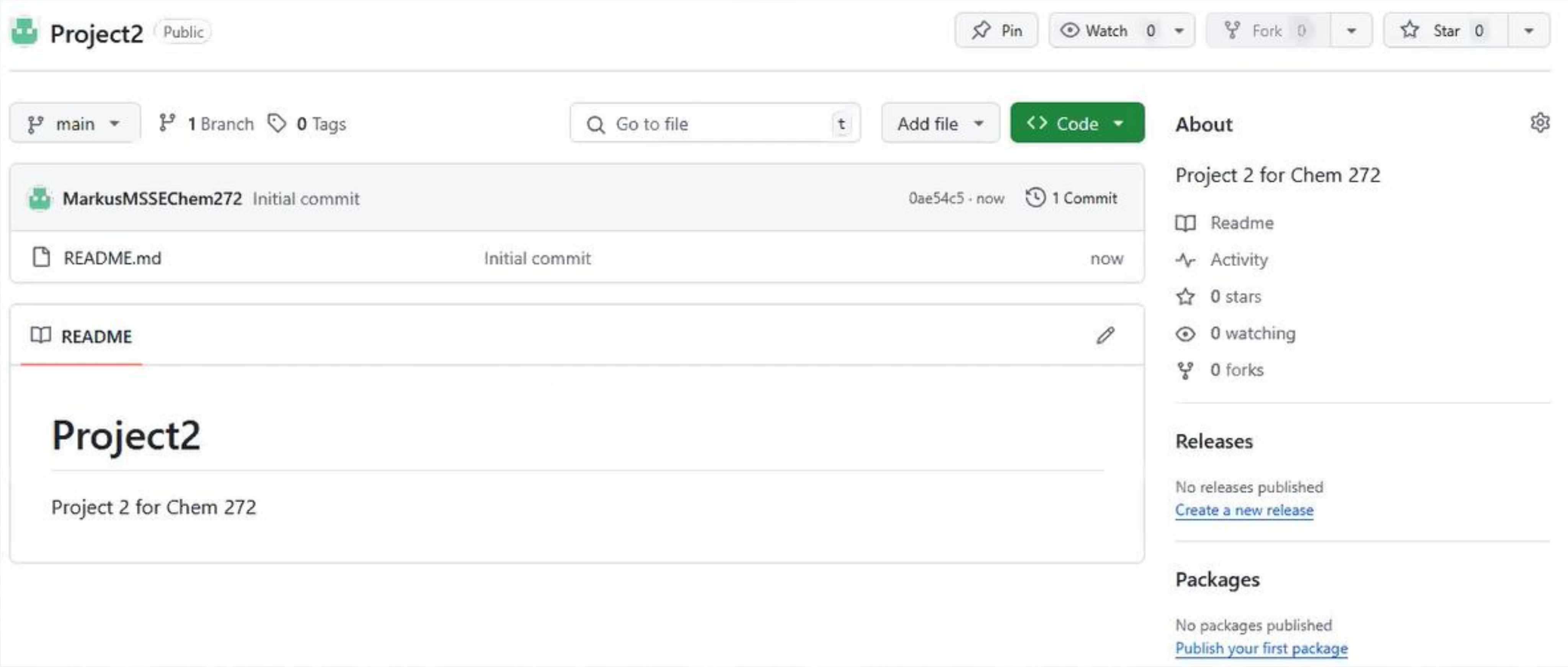
Git & GitHub

create your repo

create new repository



Congratulations! You have created your first repository!



Project2 Public

Pin | Watch 0 | Fork 0 | Star 0

main | 1 Branch | 0 Tags | Go to file | Add file | Code

MarkusMSSEChem272 Initial commit — 0ae54c5 · now — 1 Commit

README.md — Initial commit — now

README

Project2

Project 2 for Chem 272

About

Project 2 for Chem 272

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

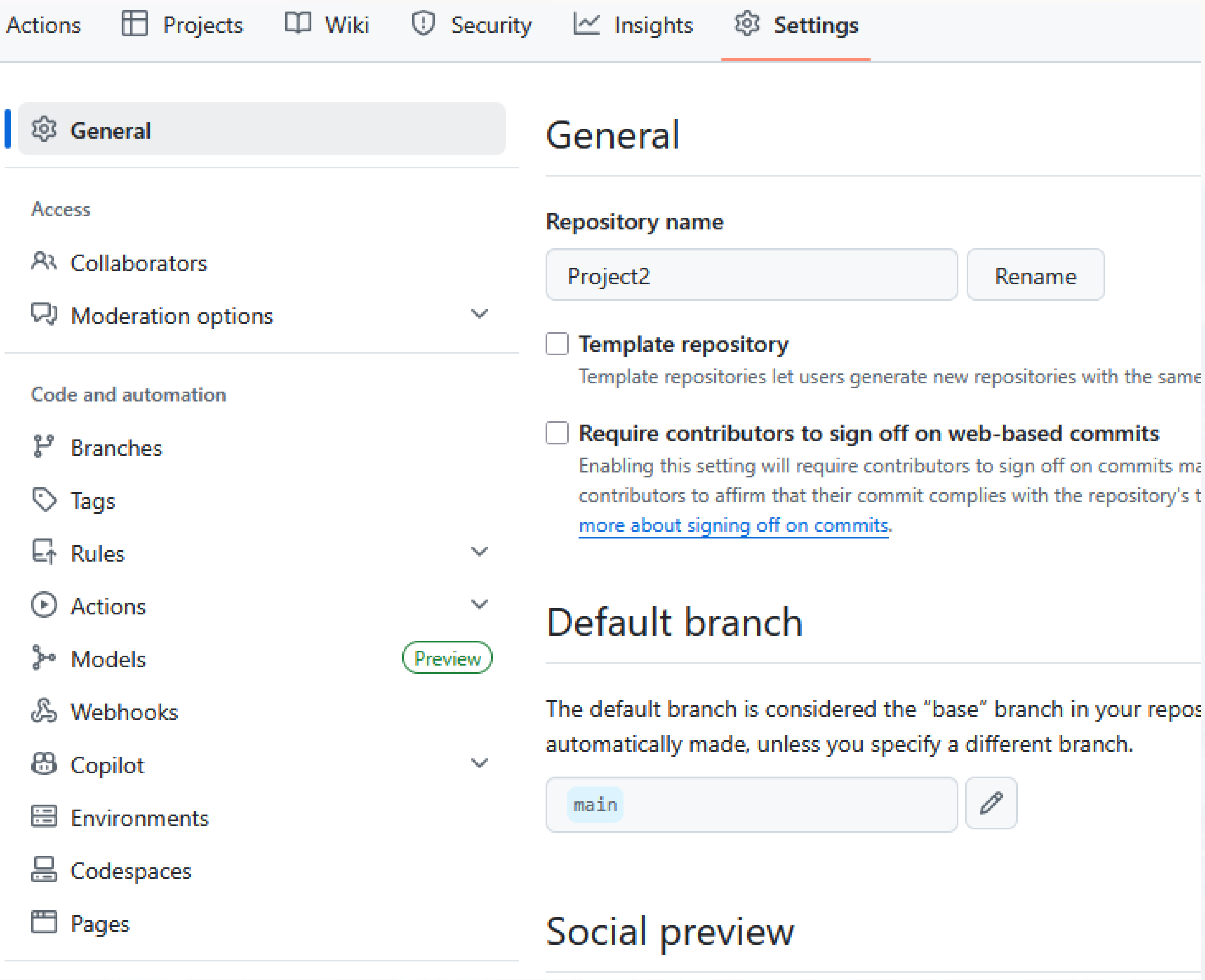
Packages

No packages published

Publish your first package

Git & GitHub

Note there are way more settings you can adjust



General

Repository name

Project2 Rename

□ **Template repository**

Template repositories let users generate new repositories with the same

□ **Require contributors to sign off on web-based commits**

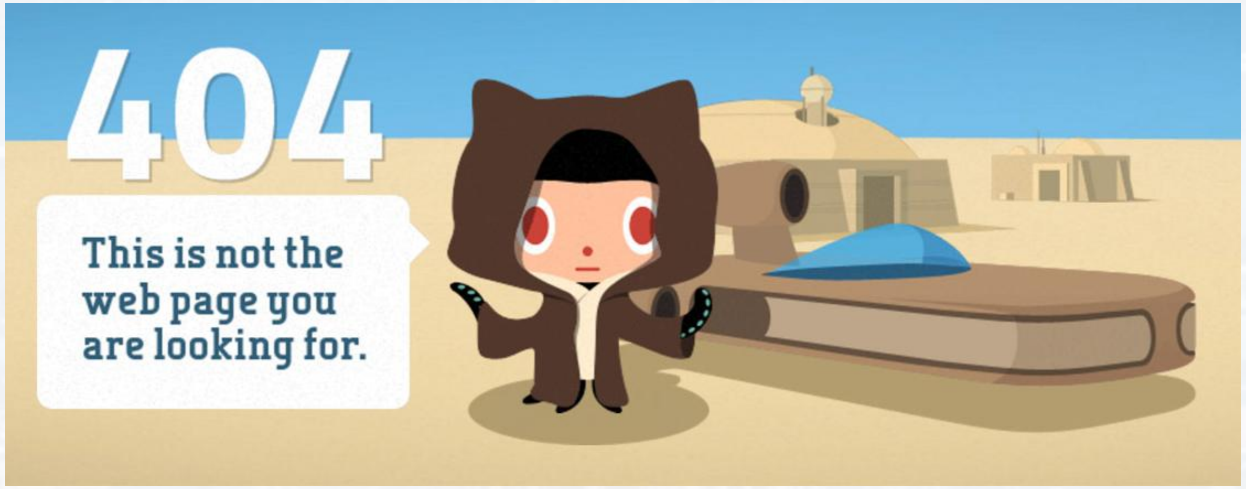
Enabling this setting will require contributors to sign off on commits ma contributors to affirm that their commit complies with the repository's t more about signing off on commits.

Default branch

The default branch is considered the “base” branch in your repos automatically made, unless you specify a different branch.

`main`

Social preview



Outline

- Motivation
- Git

  installation and VS Code

  status, add and commit

  compare differences

  restoring from index, stage and commit

- **Git Hub**

  create your repo

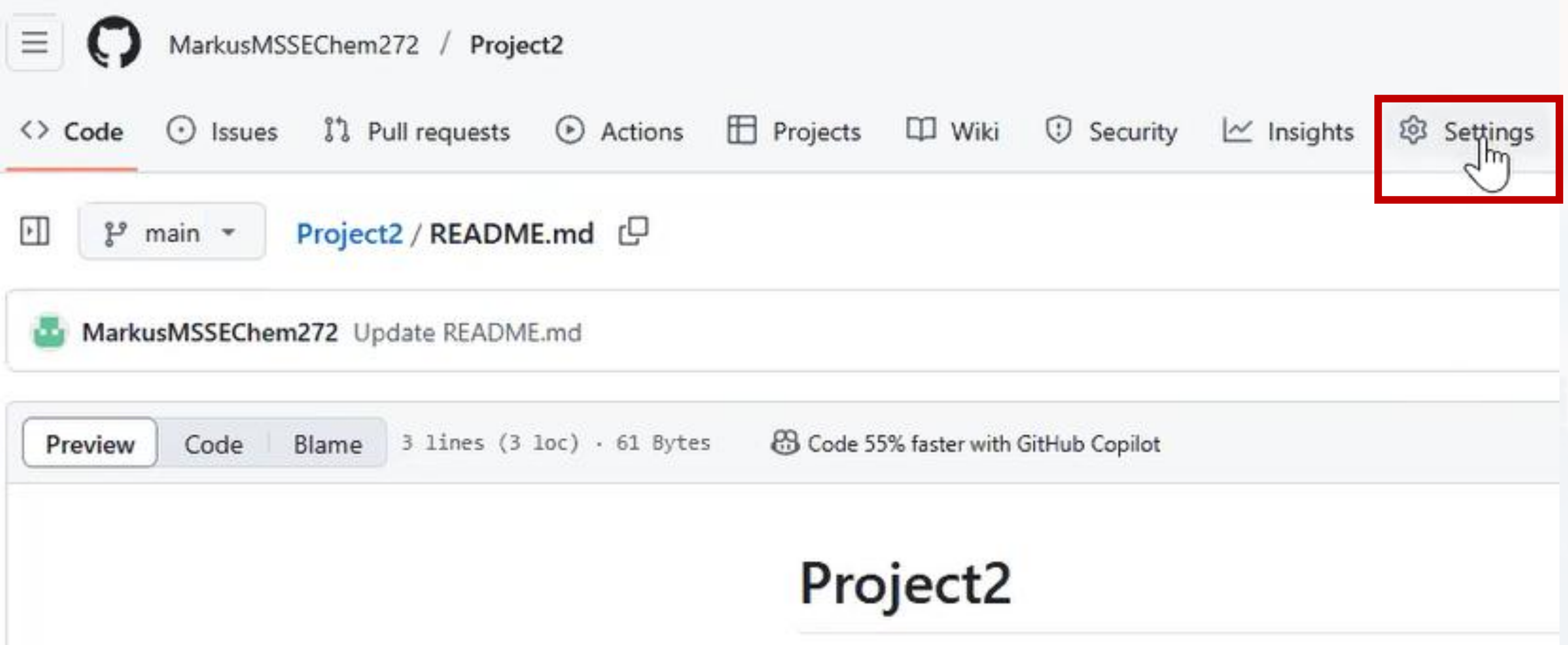
  invite a co worker

  push

  fetch, merge, pull

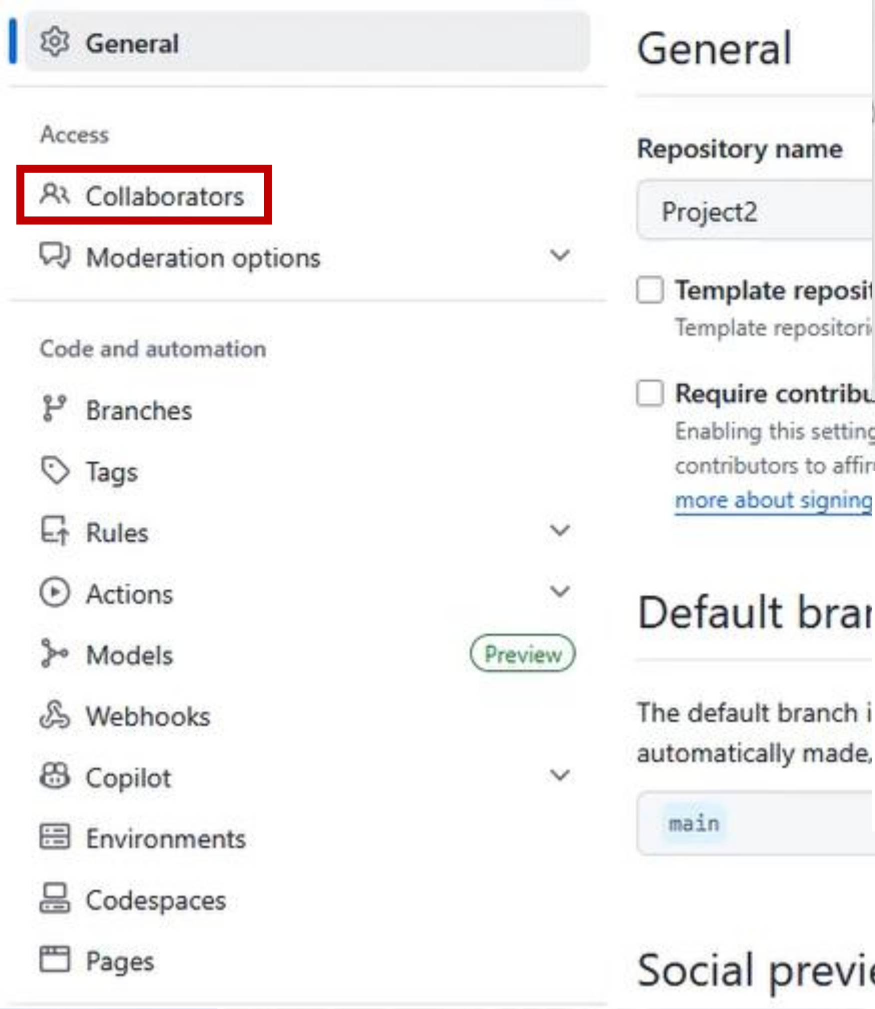
invite a co-worker

go to **settings**


MarkusMSSEChem272 / Project2
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
Project2 / README.md
MarkusMSSEChem272 Update README.md
Preview
Code
Blame
3 lines (3 loc) · 61 Bytes
Code 55% faster with GitHub Copilot
Project2

invite a co-worker

go to **collaborators**



General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models Preview

Webhooks

Copilot

Environments

Codespaces

Pages

General

Repository name

Project2

□ Template reposit

Template repositori

□ Require contribu

Enabling this setting will require contributors to sign off on c

contributors to affirm that their commit complies with the re

more about signing

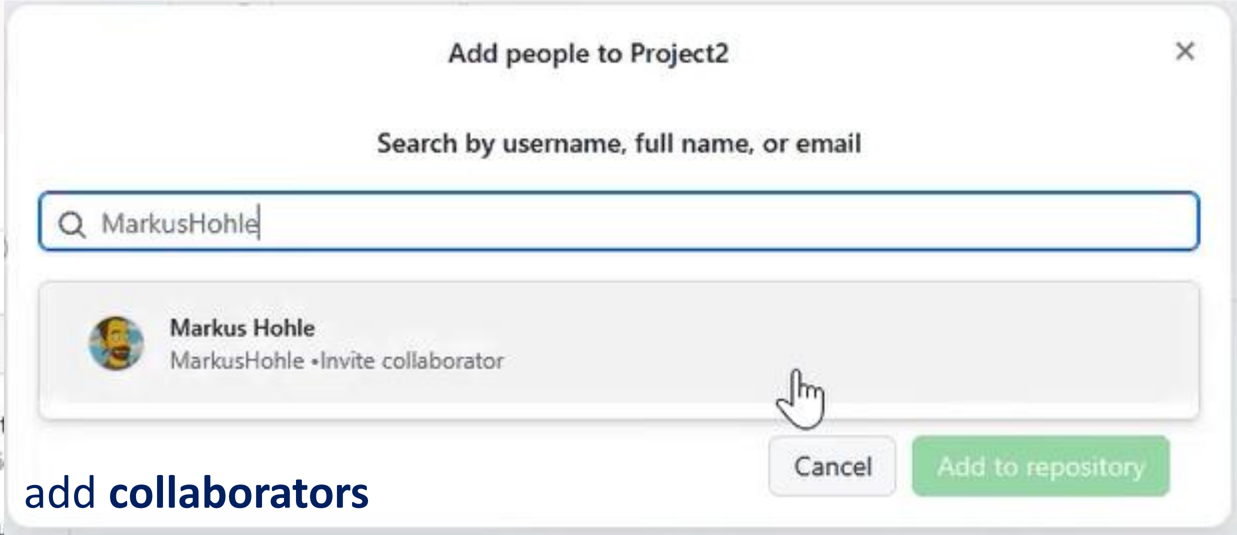
Default bra

The default branch i

automatically made,

main

Social preview



Add people to Project2

Search by username, full name, or email

MarkusHohle

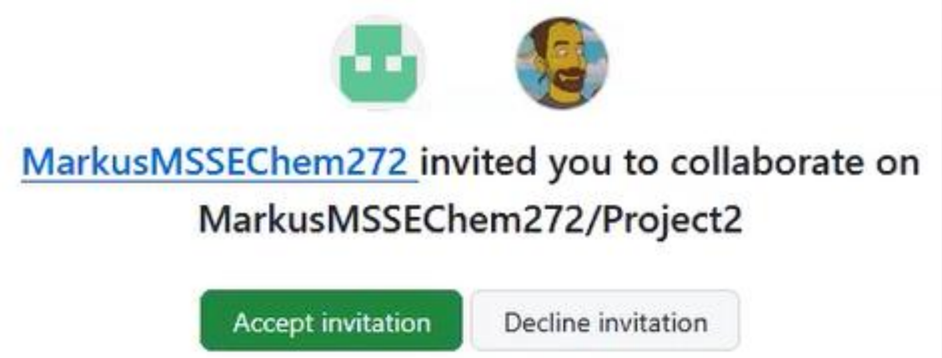
Markus Hohle

MarkusHohle •Invite collaborator

Cancel

Add to repository

add **collaborators**



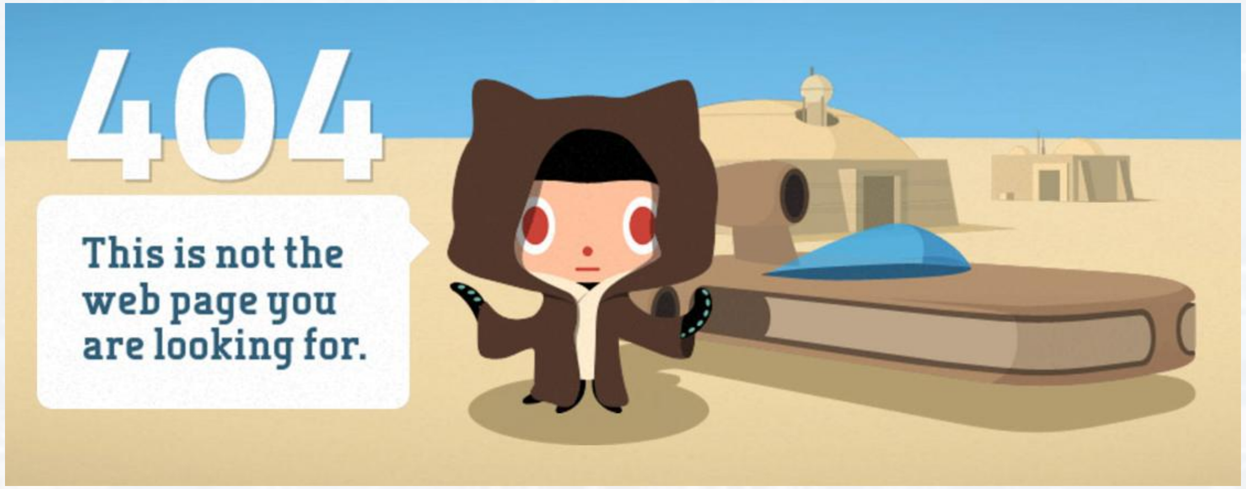
MarkusMSSEChem272 invited you to collaborate on MarkusMSSEChem272/Project2

Accept invitation

Decline invitation

your **collaborator** will get an invitation and needs to confirm it

watch:
12h_GitHub_InviteCoWorker



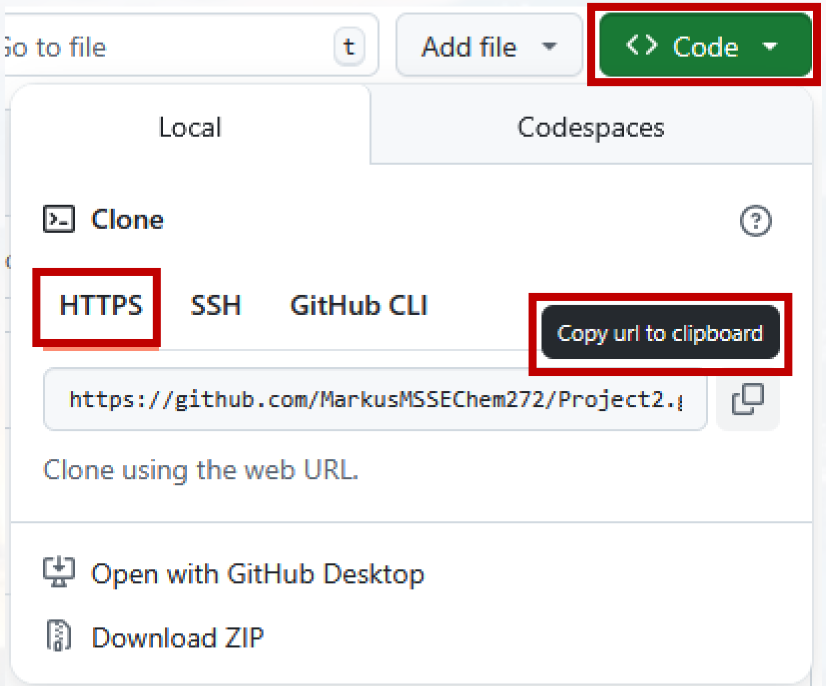
Outline

- Motivation
- Git
    - installation and VS Code
    - status, add and commit
    - compare differences
    - restoring from index, stage and commit
- **Git Hub**
    - create your repo
    - invite a co worker
    - **push**
    - fetch, merge, pull

Git & GitHub

1) copying the remote repo to your local machine (`git clone`)

- copy the **url** of your repo



https://github.com/MarkusMSSEChem272/Project2.g

Clone using the web URL.

a) go to your repo and click *code*

b) select an option (e. g. HTTPS)

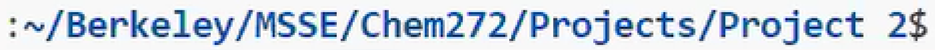
c) copy the url

next: open VS Code and navigate to the directory in which you want to copy the repo

Git & GitHub

push

next: open VS Code and navigate to the directory in which you want to copy the repo

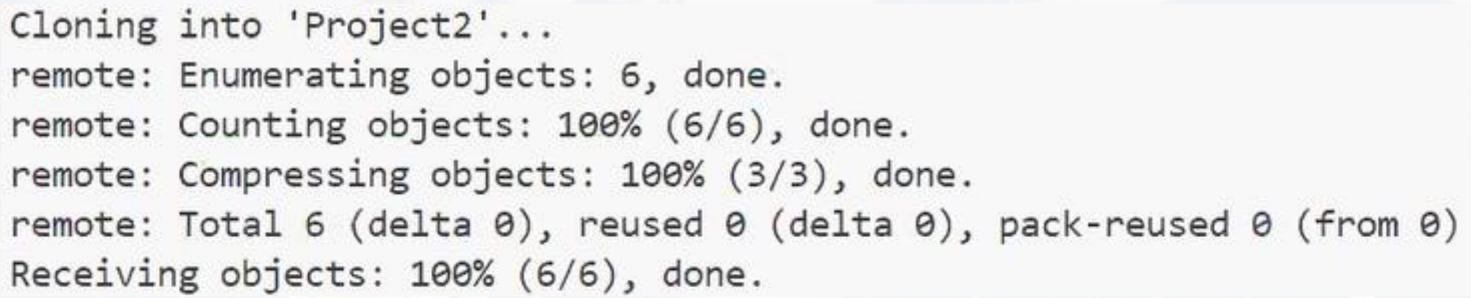


```
:~/Berkeley/MSSE/Chem272/Projects/Project 2$
```

clone the repo:

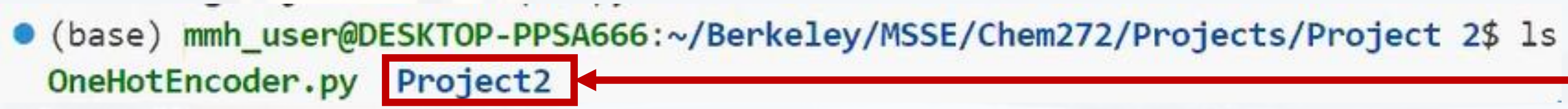
```
git clone https://github.com/MarkusMSSEChem272/Project2.git
```

the url we copied earlier

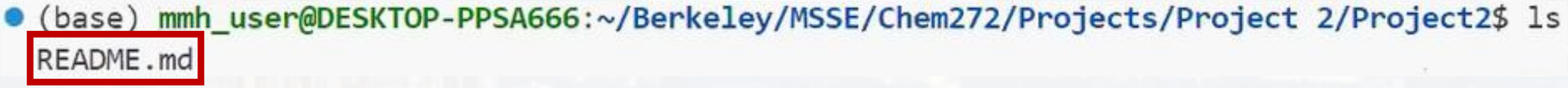


```
Cloning into 'Project2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
```

the entire repo is being copied to your local machine now!



```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2$ ls
  OneHotEncoder.py  Project2
```



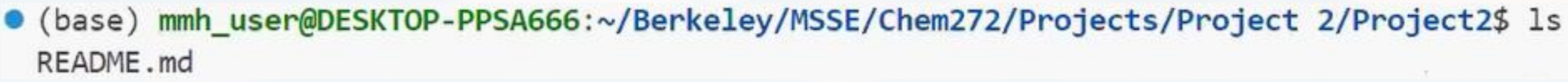
```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ ls
 README.md
```

Git & GitHub

push

2) workflow

a) navigate to your project folder



```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ ls
  README.md
```

b) check `git status`

either Git tells you that to local repo is up to date, or

it tells you that the **remote repo is X commits ahead**
→ run `git fetch` to update your local repo (**see later!**)!

c) you work on your code and want to add it to the remote repo now

```
git add
git commit -m “my message”
```

```
git push https://github.com/MarkusMSSEChem272/Project2.git main
```

pushes the new code to the remote repo

Git & GitHub

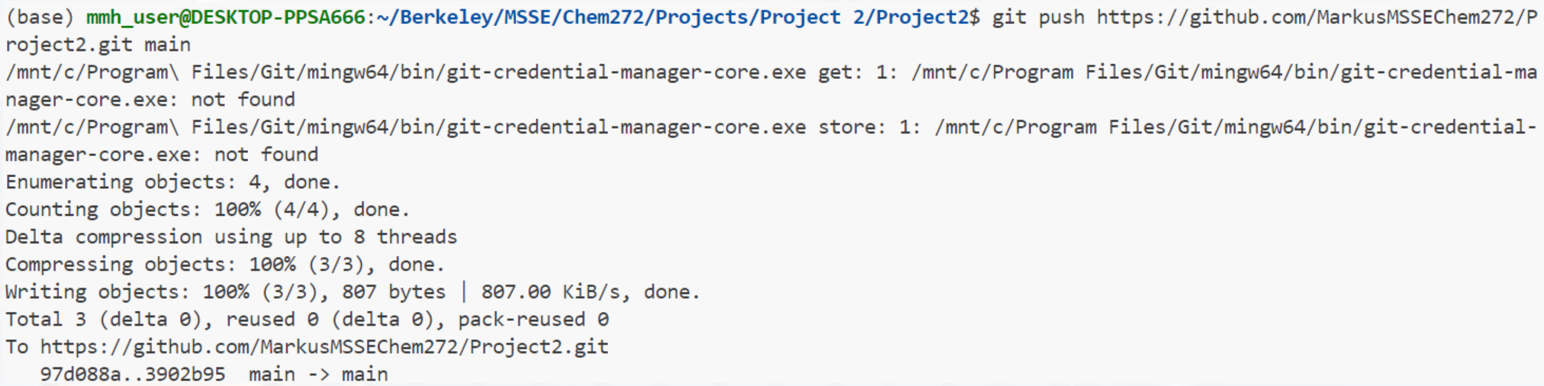
push

2) workflow

```
git add
git commit -m “my message”
```

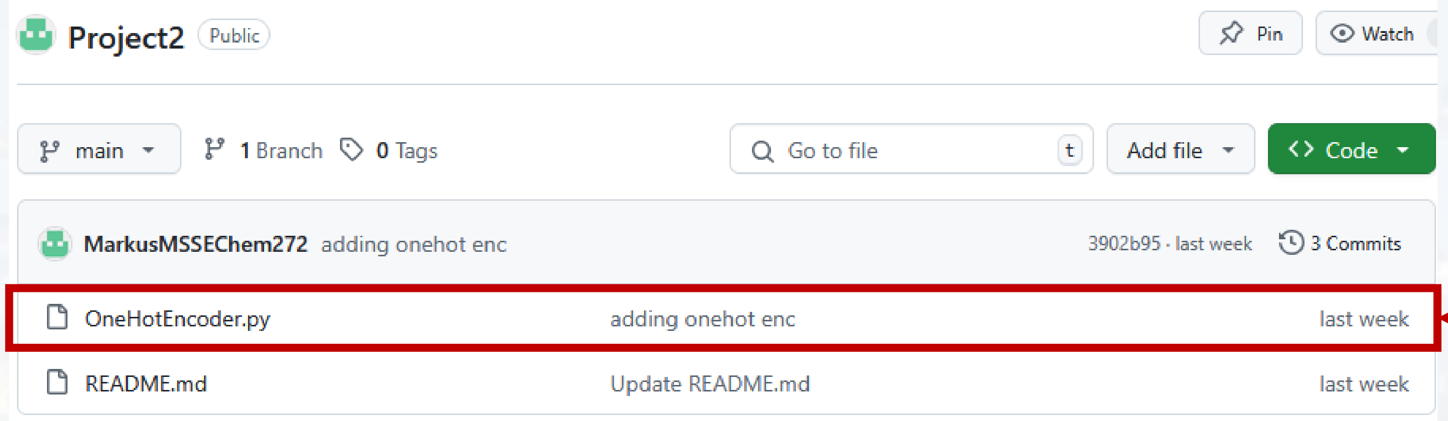
```
git push https://github.com/MarkusMSSEChem272/Project2.git main
```

pushes the new code to the remote repo



```
(base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ git push https://github.com/MarkusMSSEChem272/P
roject2.git main
/mnt/c/Program\ Files/Git/mingw64/bin/git-credential-manager-core.exe get: 1: /mnt/c/Program Files/Git/mingw64/bin/git-credential-ma
nager-core.exe: not found
/mnt/c/Program\ Files/Git/mingw64/bin/git-credential-manager-core.exe store: 1: /mnt/c/Program Files/Git/mingw64/bin/git-credential-
manager-core.exe: not found
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), 807 bytes | 807.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MarkusMSSEChem272/Project2.git
   97d088a..3902b95  main -> main
```

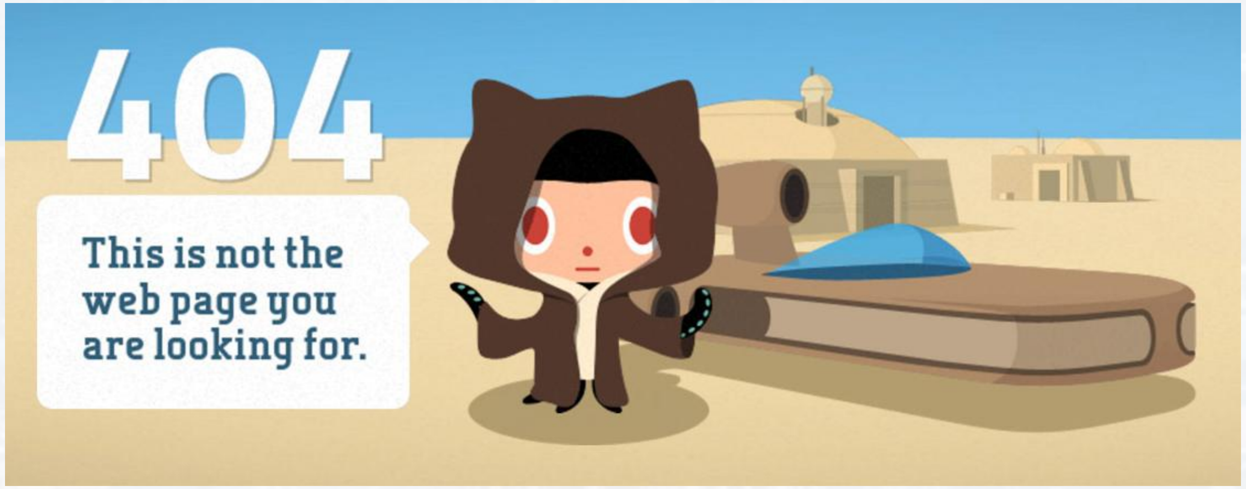
code is online!



MarkusMSSEChem272 adding onehot enc 3902b95 · last week 3 Commits

OneHotEncoder.py	adding onehot enc	last week
README.md	Update README.md	last week

watch:
12i_GitHub_CloneRepo
12j_GitHub_Push



Outline

- Motivation
- Git
  - installation and VS Code
  - status, add and commit
  - compare differences
  - restoring from index, stage and commit
- **Git Hub**
  - create your repo
  - invite a co worker
  - push
  - **fetch, merge, pull**

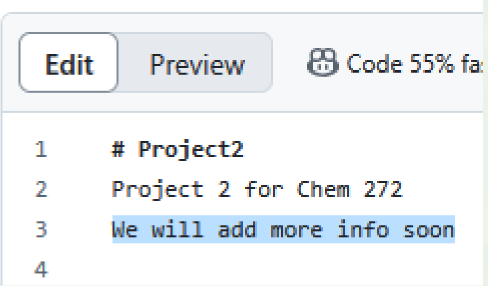
Git & GitHub

fetch, merge, pull

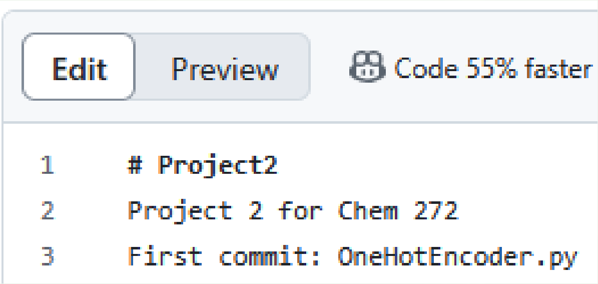
if your co-workers have pushed new code to the **remote repo**

→ update your **local repo/directory** accordingly

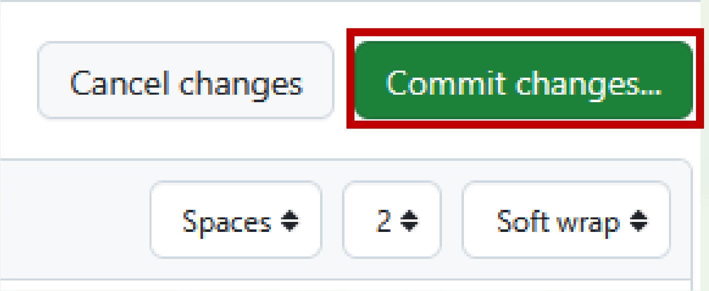
remote repo (GitHub)



```
# Project2
Project 2 for Chem 272
We will add more info soon
```

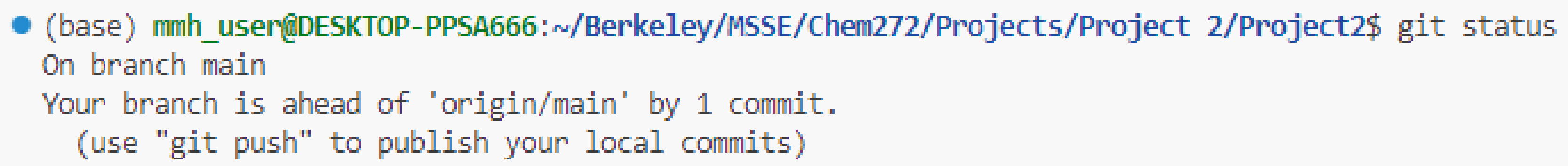


```
# Project2
Project 2 for Chem 272
First commit: OneHotEncoder.py
```



local repo

```
git status
```



```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ git status
  On branch main
  Your branch is ahead of 'origin/main' by 1 commit.
    (use "git push" to publish your local commits)
```

Git & GitHub

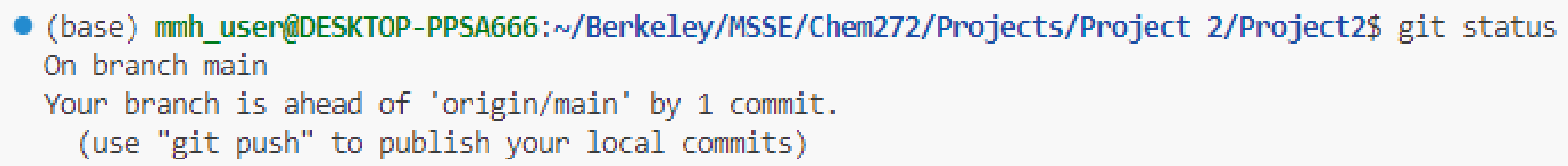
fetch, merge, pull

if your co-workers have pushed new code to the **remote repo**

→ update your **local repo/directory** accordingly

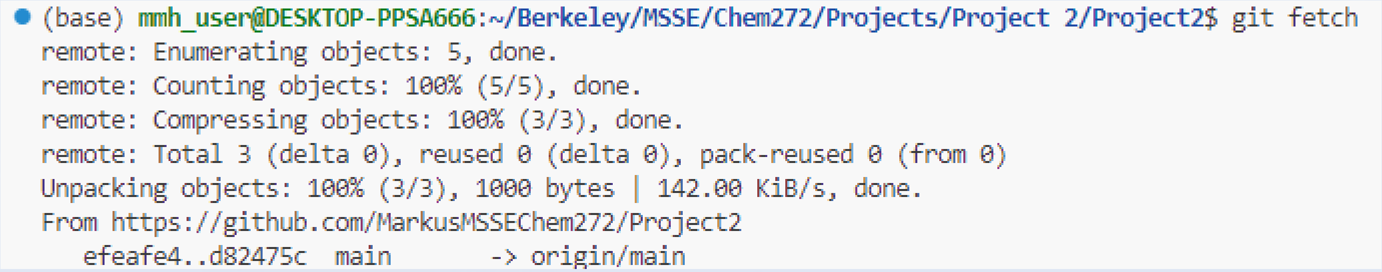
local repo

`git status`



```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ git status
  On branch main
  Your branch is ahead of 'origin/main' by 1 commit.
    (use "git push" to publish your local commits)
```

`git fetch`



```
● (base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ git fetch
  remote: Enumerating objects: 5, done.
  remote: Counting objects: 100% (5/5), done.
  remote: Compressing objects: 100% (3/3), done.
  remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
  Unpacking objects: 100% (3/3), 1000 bytes | 142.00 KiB/s, done.
  From https://github.com/MarkusMSSEChem272/Project2
     efeafe4..d82475c  main       -> origin/main
```

`git fetch`

- **does not** update your working directory
- updates only commits, tags etc → **safe to inspect changes**

fetch, merge, pull

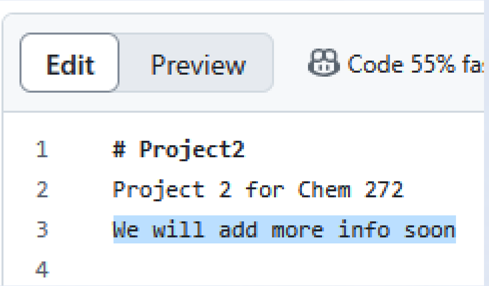
if your co-workers have pushed new code to the **remote repo**

→ update your **local repo/directory** accordingly

local repo

```
git fetch
```

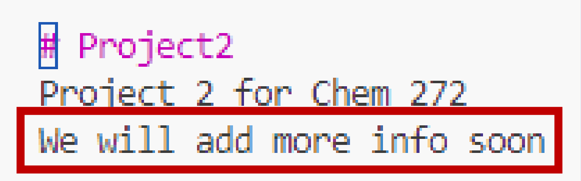
- **does not** update your working directory
- updates only commits, tags etc → **safe to inspect changes**



```
# Project2
Project 2 for Chem 272
We will add more info soon
```

old version is still stored in working directory

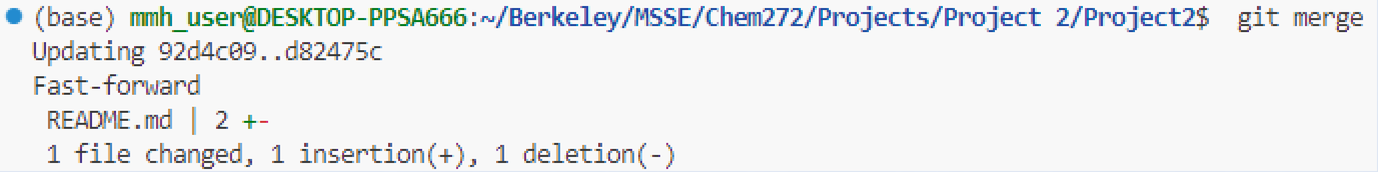
working directory



```
# Project2
Project 2 for Chem 272
We will add more info soon
```

```
git merge
```

- **does** update your working directory



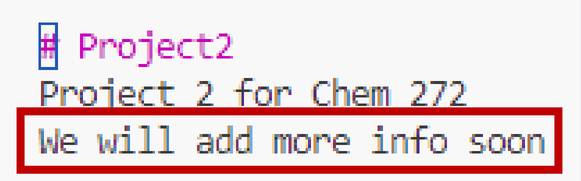
```
(base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$  git merge
Updating 92d4c09..d82475c
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Git & GitHub

fetch, merge, pull

if your co-workers have pushed new code to the **remote repo**
→ update your **local repo/directory** accordingly

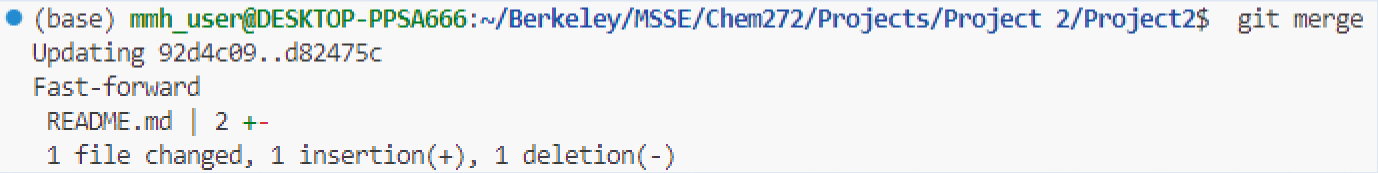
working directory



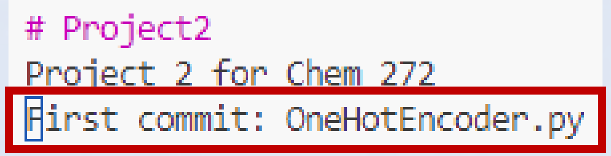
```
# Project2
Project 2 for Chem 272
We will add more info soon
```

`git merge`

- **does** update your working directory



```
(base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$  git merge
Updating 92d4c09..d82475c
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

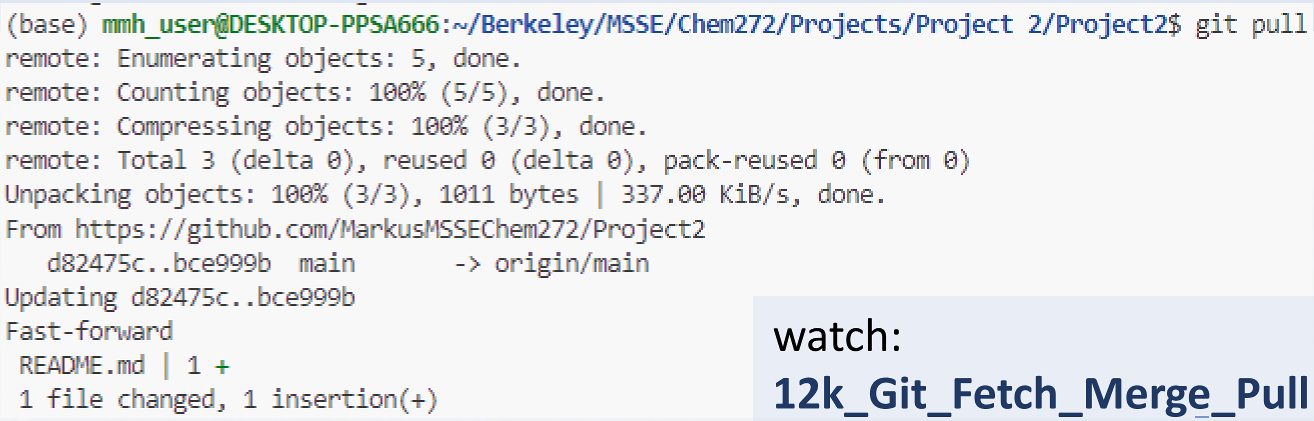


```
# Project2
Project 2 for Chem 272
First commit: OneHotEncoder.py
```

Now, also the **working directory** has been updated

alternatively: `git pull`

runs `git fetch` **and** `git merge`



```
(base) mmh_user@DESKTOP-PPSA666:~/Berkeley/MSSE/Chem272/Projects/Project 2/Project2$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1011 bytes | 337.00 KiB/s, done.
From https://github.com/MarkusMSSEChem272/Project2
   d82475c..bce999b  main       -> origin/main
Updating d82475c..bce999b
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

watch:
12k_Git_Fetch_Merge_Pull

summary

`git add <my_file>`	adds a file to staging area
`git commit –m “my_message”`	commits files from staging area to local repo
`git push  <remote_repo_name>  <branch_name>`	updates remote repo
`git restore <my_file>`	undoes changes in indexed file
`git restore --staged <my_file>`	moves file from staged area to index
`git restore --source=HEAD~X`	moves commit (X commits before current one) back to staged area
`git fetch`	updates local commit from remote repo
`git merge`	updates local directory after `fetch`
`git pull`	runs `fetch` **and** `merge`
`git clone`	clones entire remote repo (has to be done only **once**!)

fetch, merge, pull

summary

`git add <my_file>`	adds a file to staging area
`git commit –m “my_message”`	commits files from staging area to local repo
`git push  <remote_repo_name>  <branch_name>`	updates remote repo
`git restore <my_file>`	undoes changes in indexed file
`git restore --staged <my_file>`	moves file from staged area to index
`git restore --source=HEAD~X`	moves commit (X commits before current one) back to staged area
`git fetch`	updates local commit from remote repo
`git merge`	updates local directory after `fetch`
`git pull`	runs `fetch` **and** `merge`
`git clone`	clones entire remote repo (has to be done only **once**!)

note, there is way more you could do

check out different **branches**
moving **head**

We hope you have enjoyed the course!