IDSWG AI CODING BOOTCAMP

AGENDA

Install Python

Install local LLM through ollama

Install LlamaIdx for RAG

Python Streamlit

Git installation and Github use

INSTRUCTOR

Louise Liu, PhD MBA

Louise.liu@njstat.com/

Louise.liu@hillresearch.ai

Linkedin: LinkedIn

Tel./WhatsAPP: 475-655-9876

INSTALL PYTHON

DOWNLOAD THE ANACONDA DISTRIBUTION

- The Anaconda distribution includes many practical python systems in one bundle
  - Python
  - Conda
  - Pip
  - Common Python libraries (e.g. Numpy, Pandas…)
- https://repo.anaconda.com/archive/
- The latest version for Python 3.11 is Anaconda 2024.02-1 – pick the executable that corresponds to your system

INSTALL THE ANACONDA DISTRIBUTION

- Windows: double click the .exe, follow package instructions

- Mac: either
  - double click the .pkg
  - Command line: bash <insert_file_name>.sh

- Linux: Same as Mac command line

SETUP PYTHON ENVIRONMENT

- Open a terminal
- Update the conda environment
  - conda update conda
- Setup an environment
  - conda create --name IDSWG python=3.11
- Activate the environment
  - conda activate IDSWG
- Install necessary Python modules
  - pip install numpy scipy pandas matplotlib scikit-learn python-docx ollama streamlit spyder spyder-notebook
- Install VS Code [Visual Studio Code - Code Editing. Redefined]

INSTALL LOCAL LLM THROUGH OLLAMA

Name: Zhaohua Lu

Email: zhaohua.lu@gmail.com

zhaohualu/IDSWG-AI-Coding-Boot-Camp

INSTALL OLLAMA

- For MAC user and Other users with Nvidia and AMD GPUs: download ollama from https://ollama.com/, download and follow common software installation.
- For Windows user with integrated Intel GPU: https://github.com/intel/ipex-llm/blob/main/docs/mddocs/Quickstart/ollama_portable_zip_quickstart.md
  - Download and unzip it to a folder
  - Add the folder path to the environment variable PATH
  - Please download and install the intel driver

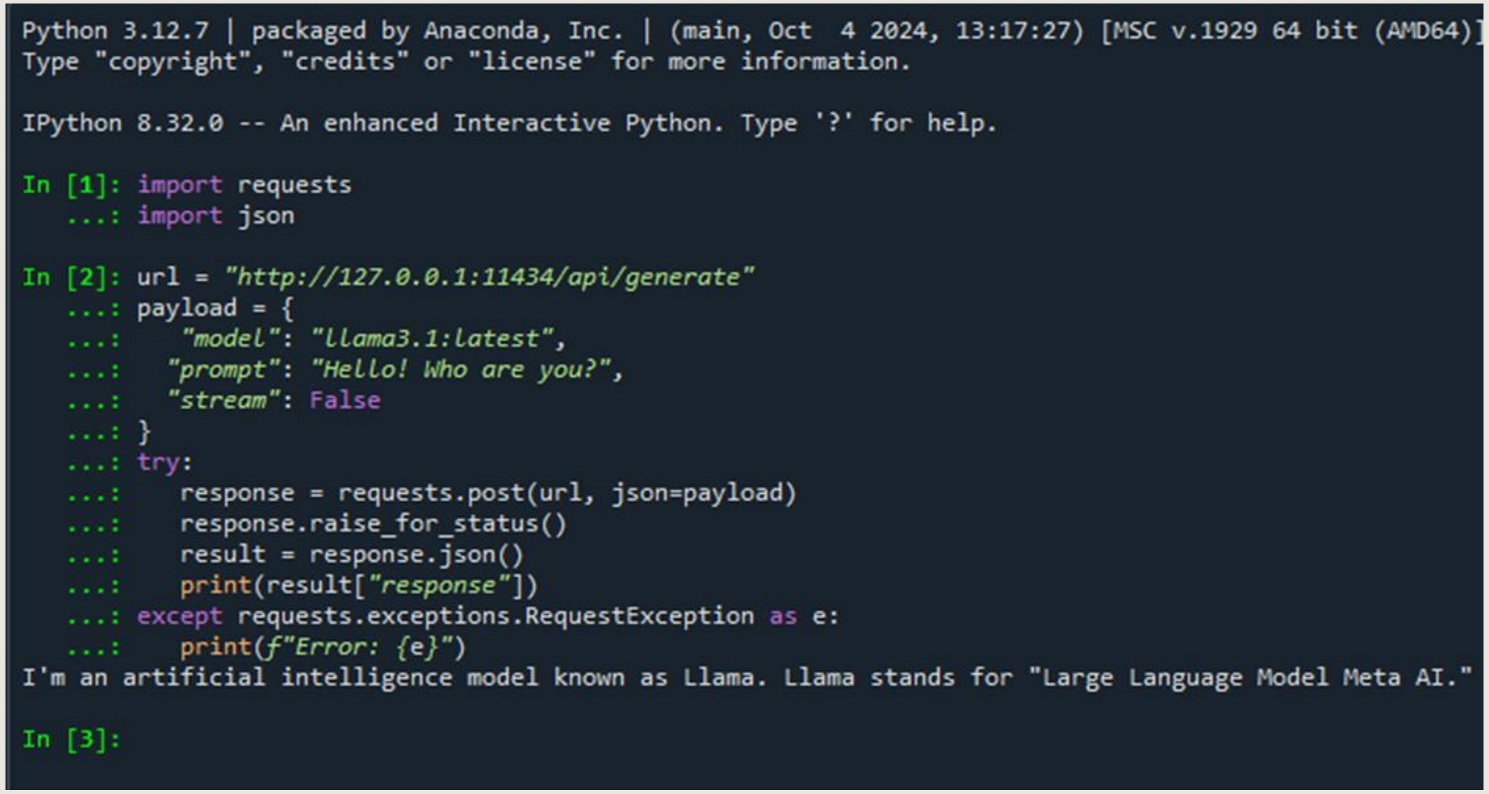
RUN OLLAMA

- Open terminal
  - Press Windows button
  - Type “cmd” and enter
- Run ollama
  - Mac: ollama serve
  - Windows: ollama-serve.bat
- Download model e.g., llama3.1, in terminal, run
  - ollama pull llama3.1
  - For interaction test, open another terminal, run
  - ollama run llama3.1

RUN PYTHON WITH OLLAMA

- Basic: Use requests module
  - E.g., use Spyder IDE or others
- Other: ollama module
  - pip install ollama

```
import requests
import json

url = "http://127.0.0.1:11434/api/generate"
payload = {
   "model": "llama3.1",
  "prompt": "Hello! Who are you?",
  "stream": False
}
try:
   response = requests.post(url, json=payload)
   response.raise_for_status()
   result = response.json()
   print(result["response"])
except requests.exceptions.RequestException as e:
   print(f"Error: {e}")
```



```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.32.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import requests
   ...: import json

In [2]: url = "http://127.0.0.1:11434/api/generate"
   ...: payload = {
   ...:     "model": "llama3.1:latest",
   ...:    "prompt": "Hello! Who are you?",
   ...:    "stream": False
   ...: }
   ...: try:
   ...:     response = requests.post(url, json=payload)
   ...:     response.raise_for_status()
   ...:     result = response.json()
   ...:     print(result["response"])
   ...: except requests.exceptions.RequestException as e:
   ...:     print(f"Error: {e}")
I'm an artificial intelligence model known as Llama. Llama stands for "Large Language Model Meta AI."

In [3]:
```

INSTALL LLAMA-INDEX FOR RAG

INSTALL LLAMA-INDEX

- Install the necessary packages in terminal:
  - pip install llama-index llama-index-llms-ollama llama-index-embeddings-huggingface llama-index-vector-stores-faiss unstructured sentence-transformers
- `Import python module in python interpreter, e.g., IDE`
  - from llama_index.llms.ollama import Ollama
  - from llama_index.embeddings.huggingface import HuggingFaceEmbedding
  - from llama_index.core import Settings
- Configure the Embedding Model and LLM
  - Settings.embed_model = HuggingFaceEmbedding(model_name="BAAI/bge-base-en-v1.5")
  - Settings.llm = Ollama(model="llama3.1", request_timeout=60.0)

LOADING DOCUMENTS AND BUILDING THE INDEX

- Load Documents, Place your text or documents in a directory, e.g., ./data/.
- Use SimpleDirectoryReader to load the documents:
  - from llama_index.core import SimpleDirectoryReader
  - documents = SimpleDirectoryReader("./data").load_data()
- Initialize the index with the loaded documents:
  - from llama_index.core import VectorStoreIndex
  - index = VectorStoreIndex.from_documents(documents)
- Prepare the query engine to handle user queries:
  - query_engine = index.as_query_engine()
- `The VectorStoreIndex uses the embedding model configured earlier to convert documents into vector representations for efficient retrieval.`

PERFORMING RAG-BASED QUESTION ANSWERING

- Use the query engine to ask a question:
  - response = query_engine.query("What is the main topic of the document?")
  - print(response)
- The query is embedded using the configured embedding model.
- Relevant documents are retrieved based on vector similarity.
- The LLM (Llama 3.1 via Ollama) generates a response grounded in the retrieved documents.

PYTHON STREAMLIT

RUN A STREAMLIT APP

- Installing Streamlit
  - pip install streamlit
- Create a file app.py
- Launch the app
  - streamlit run app.py
- Stop Streamlit
- Streamlit App component reference https://streamlit.io/components
- Examples: https://streamlit.io/gallery

```
import streamlit as st

import pandas as pd

import numpy as np

st.title("My First Streamlit App")

st.write("Welcome to Streamlit!")

# Generate random data

data = pd.DataFrame(
np.random.randn(10, 2),
columns=['Column A', 'Column B'] )
st.line_chart(data)
```

GIT INSTALLATION AND GITHUB USE

Name: Runqiu(Rachel) Wang

Email: runqiurachelwang@gmail.com

Myweb: https://runqiuwang22.github.io/

INSTALL GIT

- Git is officially defined as a *distributed version control system* (VCS).
- Git official homepage: https://git-scm.com/

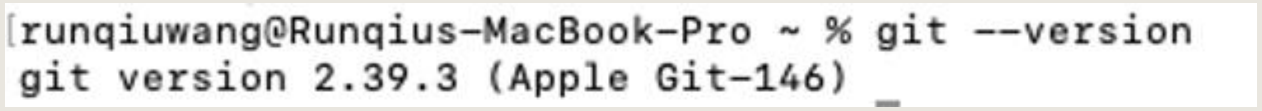
An example: Install brew:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

brew --version

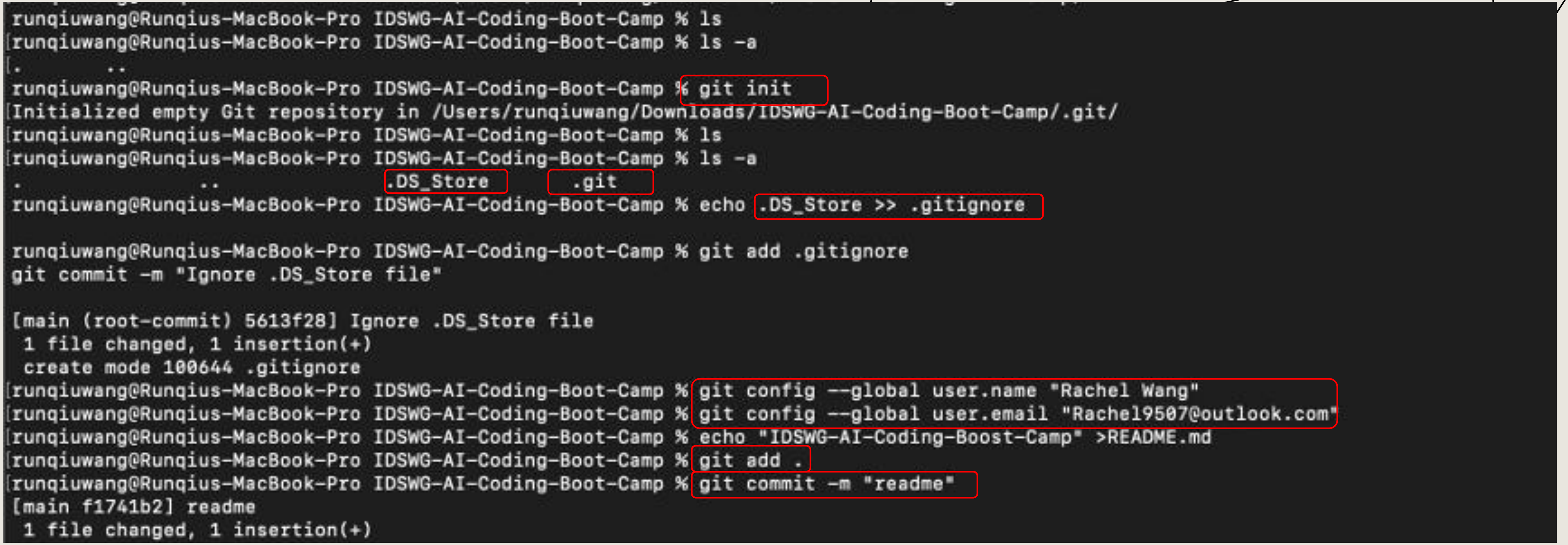
brew install git

- Verify and Check: **git --version**

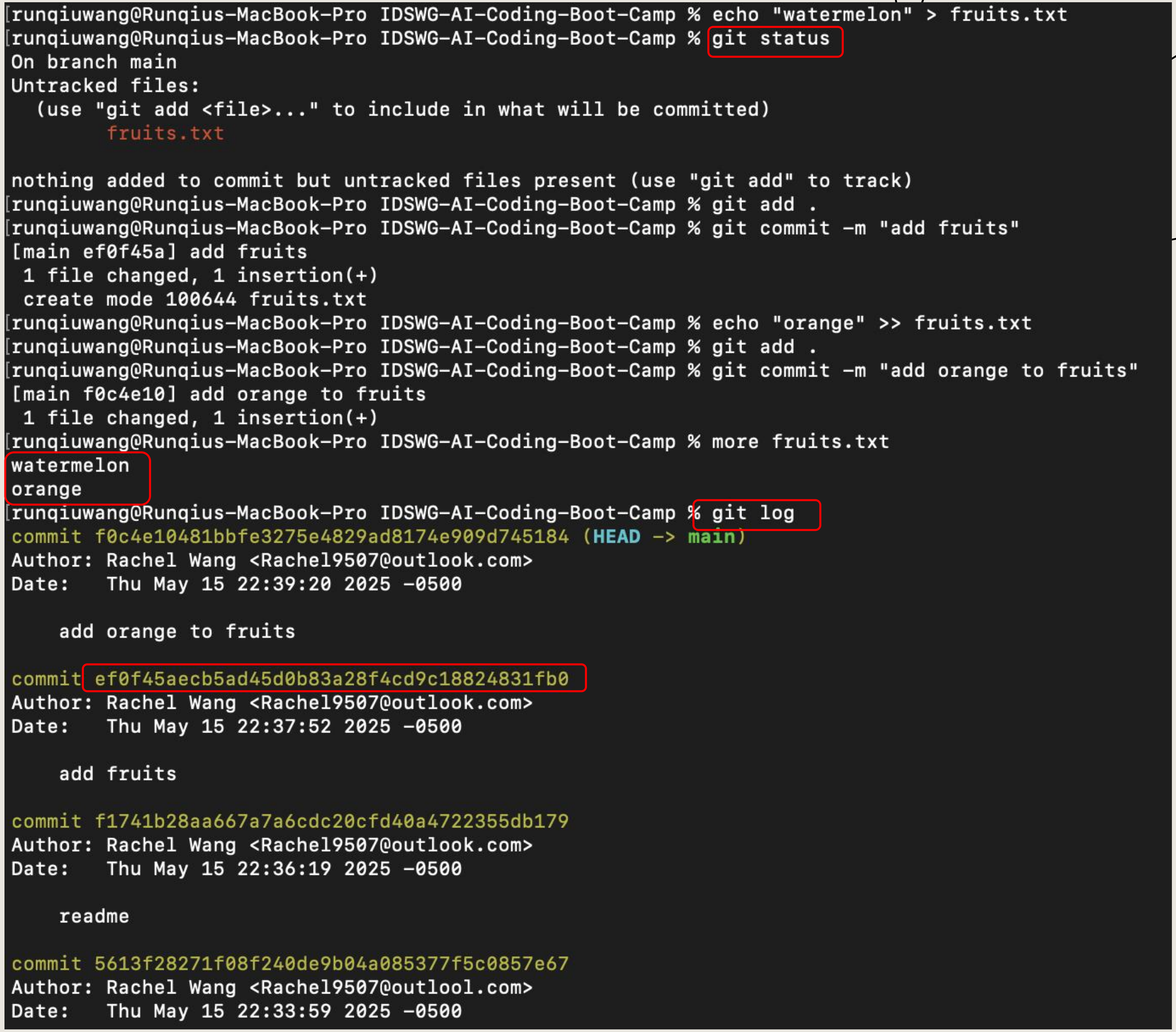


```
[runqiuwang@Runqius-MacBook-Pro ~ % git --version
git version 2.39.3 (Apple Git-146)
```

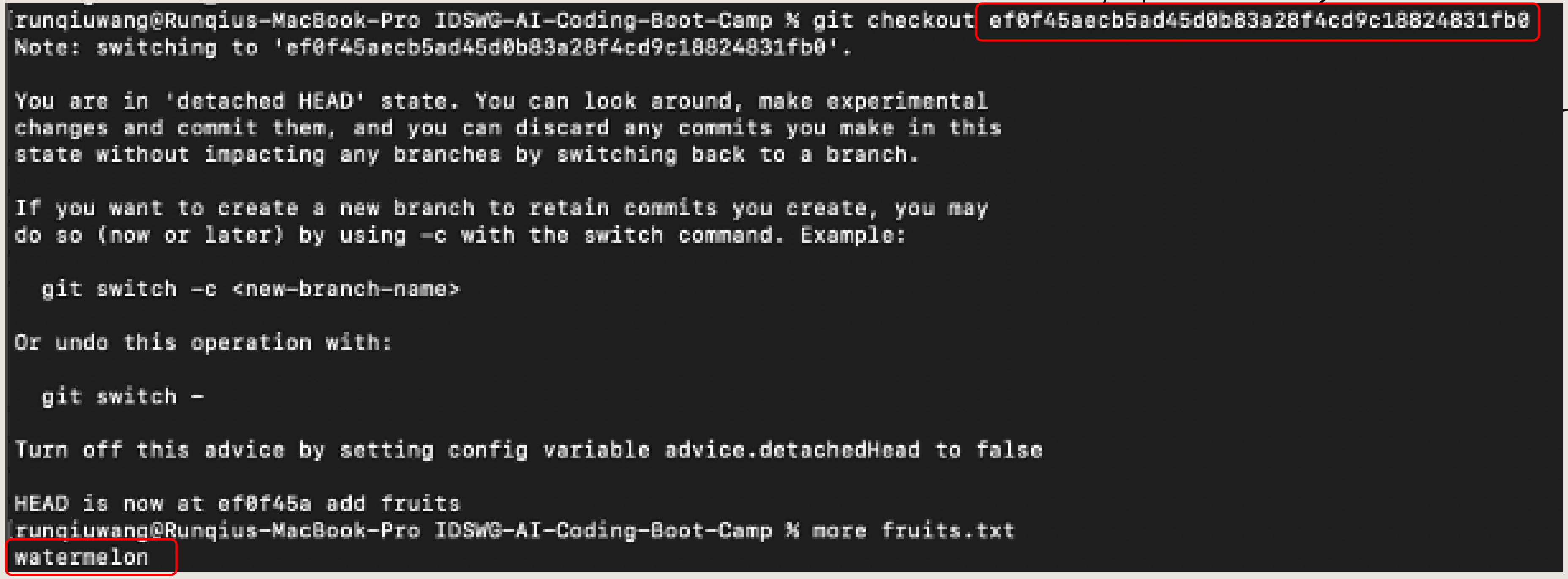
Git Basic command	
git init <directory>	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository
git config	Define author name and email to be used for all commits in current repo git config --global user.name "Your Name" git config --global user.email "your@email.com"
git add <file>	Stage all changes for a specific file for the next commit
git commit -m “<message>”	Commit the staged snapshot, but instead of launching a text editor, use as the commit message.
git status	List which files are staged, unstaged, and untracked.
git log	Display the entire commit history using the default format. For customization see additional options.
git checkout <commit-hash>	Go back to a previous state of your project code that you committed



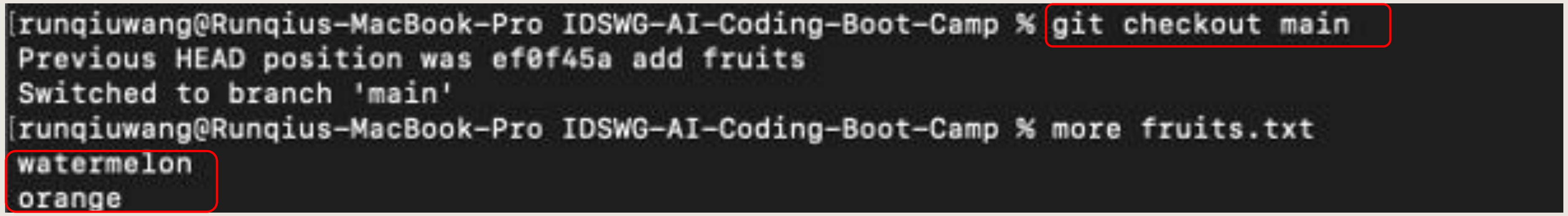
```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls -a
[.       ..
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git init
[Initialized empty Git repository in /Users/runqiuwang/Downloads/IDSWG-AI-Coding-Boot-Camp/.git/
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls -a
.               ..              .DS_Store       .git
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % echo .DS_Store >> .gitignore

runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git add .gitignore
git commit -m "Ignore .DS_Store file"

[main (root-commit) 5613f28] Ignore .DS_Store file
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git config --global user.name "Rachel Wang"
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git config --global user.email "Rachel9507@outlook.com"
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % echo "IDSWG-AI-Coding-Boost-Camp" >README.md
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git add .
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git commit -m "readme"
[main f1741b2] readme
 1 file changed, 1 insertion(+)
```



```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % echo "watermelon" > fruits.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        fruits.txt

nothing added to commit but untracked files present (use "git add" to track)
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git add .
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git commit -m "add fruits"
[main ef0f45a] add fruits
 1 file changed, 1 insertion(+)
 create mode 100644 fruits.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % echo "orange" >> fruits.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git add .
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git commit -m "add orange to fruits"
[main f0c4e10] add orange to fruits
 1 file changed, 1 insertion(+)
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % more fruits.txt
watermelon
orange
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git log
commit f0c4e10481bbfe3275e4829ad8174e909d745184 (HEAD -> main)
Author: Rachel Wang <Rachel9507@outlook.com>
Date:   Thu May 15 22:39:20 2025 -0500

    add orange to fruits

commit ef0f45aecb5ad45d0b83a28f4cd9c18824831fb0
Author: Rachel Wang <Rachel9507@outlook.com>
Date:   Thu May 15 22:37:52 2025 -0500

    add fruits

commit f1741b28aa667a7a6cdc20cfd40a4722355db179
Author: Rachel Wang <Rachel9507@outlook.com>
Date:   Thu May 15 22:36:19 2025 -0500

    readme

commit 5613f28271f08f240de9b04a085377f5c0857e67
Author: Rachel Wang <Rachel9507@outlool.com>
Date:   Thu May 15 22:33:59 2025 -0500
```

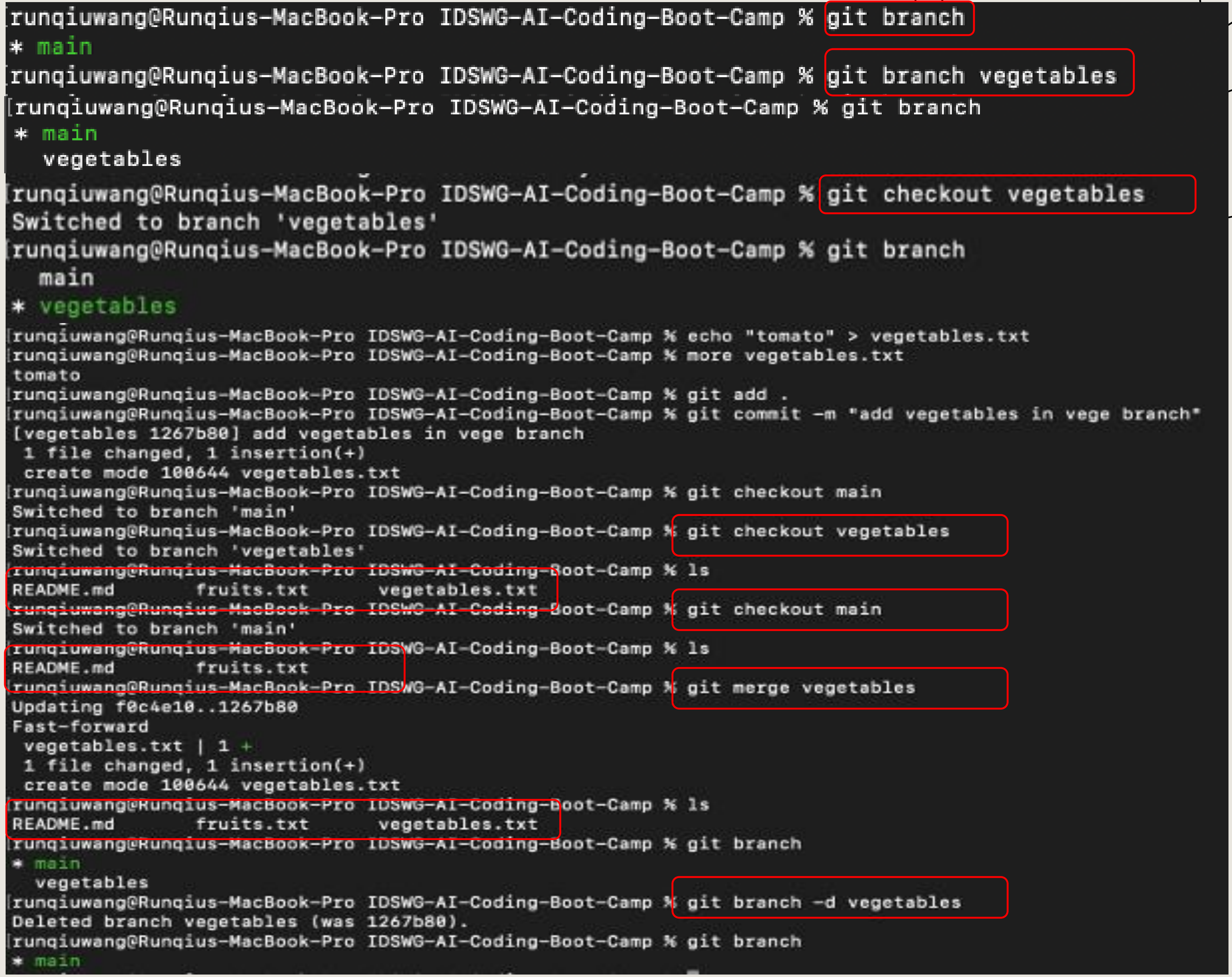


```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout ef0f45aecb5ad45d0b83a28f4cd9c18824831fb0
Note: switching to 'ef0f45aecb5ad45d0b83a28f4cd9c18824831fb0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at ef0f45a add fruits
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % more fruits.txt
watermelon
```



```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout main
Previous HEAD position was ef0f45a add fruits
Switched to branch 'main'
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % more fruits.txt
watermelon
orange
```

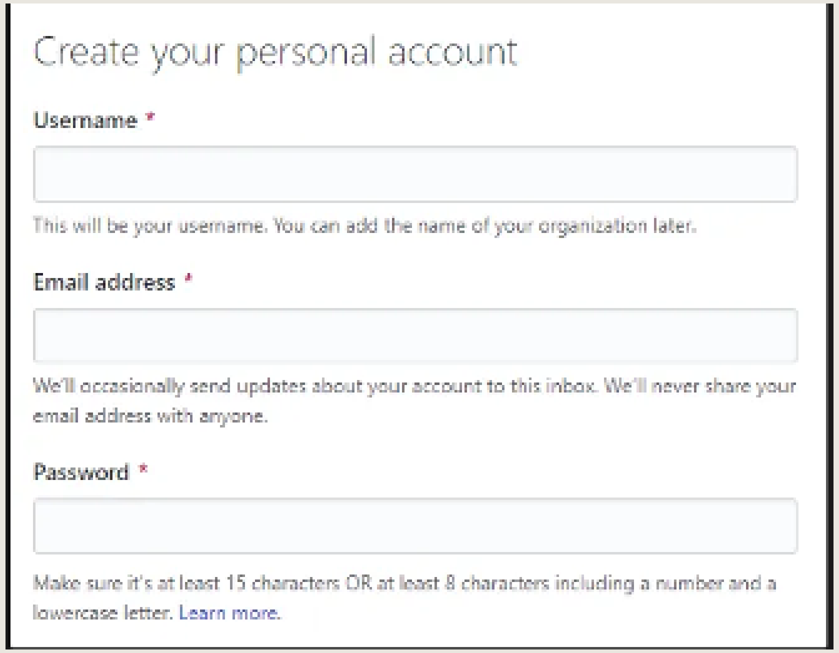
Git branches	
git branch	List all of the branches in your repo. Add a argument to create a new branch with the name .
git checkout -b	Create and check out a new branch named . Drop the -b flag to checkout an existing branch.
git merge	Merge into the current branch.
git branch -d <branch-name>	delete a branch



```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch
* main
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch vegetables
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch
 * main
   vegetables
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout vegetables
Switched to branch 'vegetables'
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch
  main
* vegetables
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % echo "tomato" > vegetables.txt
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % more vegetables.txt
tomato
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git add .
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git commit -m "add vegetables in vege branch"
[vegetables 1267b80] add vegetables in vege branch
 1 file changed, 1 insertion(+)
 create mode 100644 vegetables.txt
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout main
Switched to branch 'main'
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout vegetables
Switched to branch 'vegetables'
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls
README.md        fruits.txt        vegetables.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git checkout main
Switched to branch 'main'
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls
README.md        fruits.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git merge vegetables
Updating f0c4e10..1267b80
Fast-forward
 vegetables.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 vegetables.txt
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % ls
README.md        fruits.txt        vegetables.txt
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch
* main
  vegetables
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch -d vegetables
Deleted branch vegetables (was 1267b80).
[runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git branch
* main
```

GITHUB

GitHub homepage: github.com



Create your personal account

Username *

This will be your username. You can add the name of your organization later.

Email address *

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password *

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

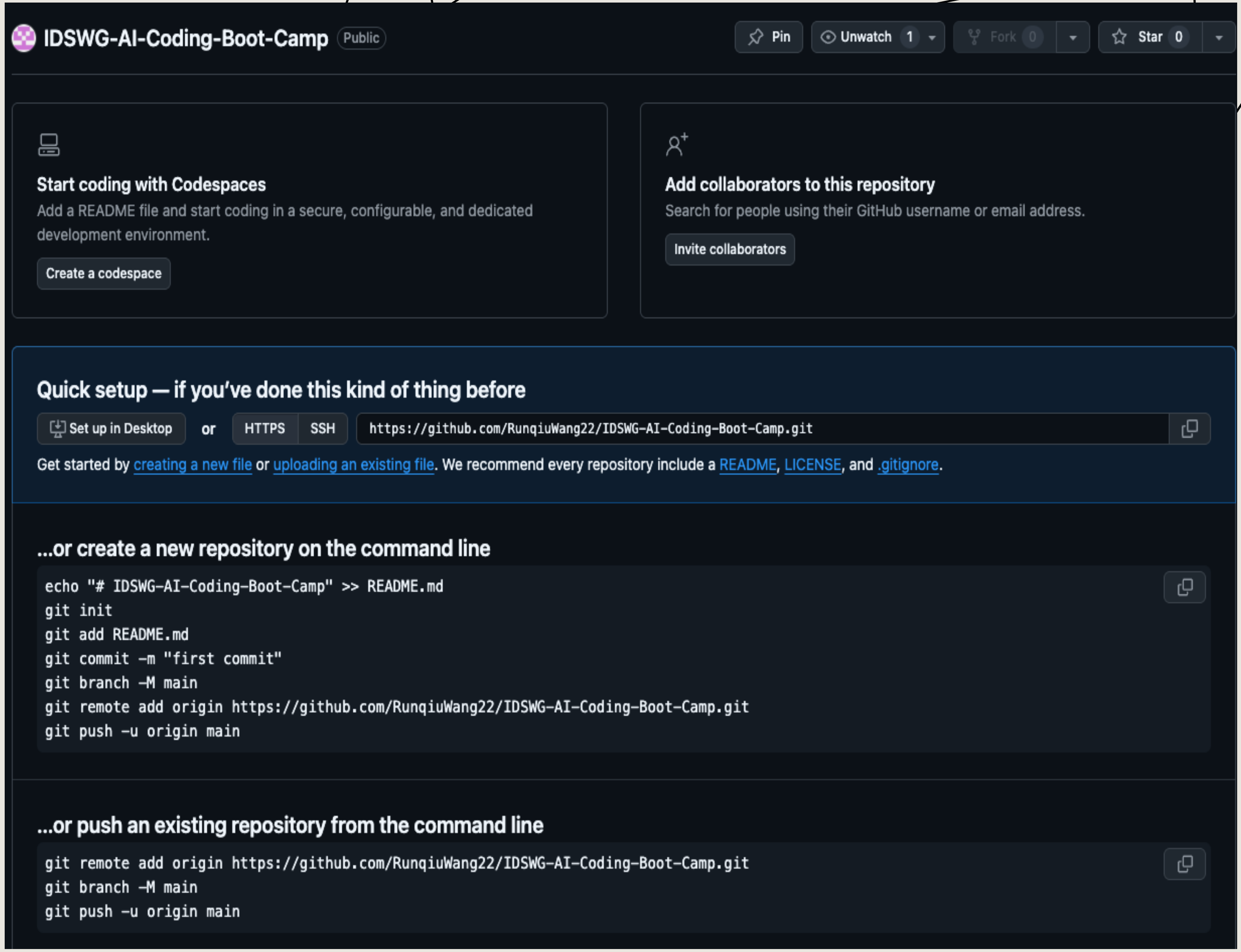
COMMON WORKFLOW: PUSH AN EXISTING REPO TO GITHUB

1. Add/commit your code locally
2. Go to Github and make a new repository
3. Connect your local repo to the github repo (add a remote)
4. Push your code up to github using the new remote

PUSHING OUR CODE TO THE GITHUB REPOSITORY

Push step:

- git remote add origin https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp.git
- git push -u origin main


IDSWG-AI-Coding-Boot-Camp Public

Pin · Unwatch 1 · Fork 0 · Star 0

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
Create a codespace

Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators

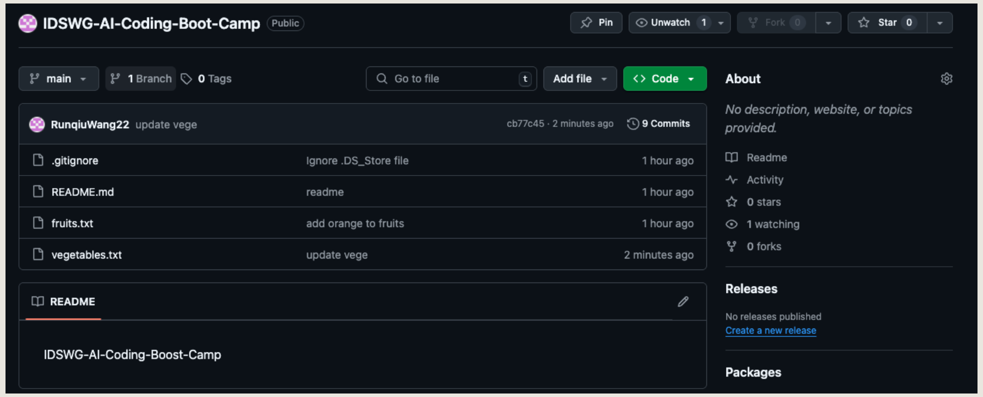
Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# IDSWG-AI-Coding-Boot-Camp" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp.git
git branch -M main
git push -u origin main
```


IDSWG-AI-Coding-Boot-Camp Public

Pin · Unwatch 1 · Fork 0 · Star 0

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

RunqiuWang22 update vege · cb77c45 · 2 minutes ago · 9 Commits

.gitignore	Ignore .DS_Store file	1 hour ago
README.md	readme	1 hour ago
fruits.txt	add orange to fruits	1 hour ago
vegetables.txt	update vege	2 minutes ago

README

IDSWG-AI-Coding-Boost-Camp

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

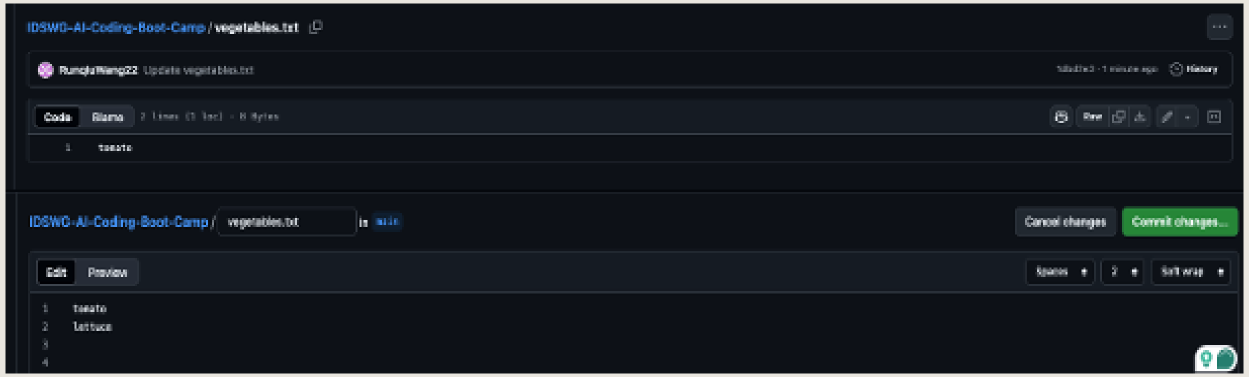
Releases

No releases published
Create a new release

Packages

PULL CHANGES FROM GITHUB

git pull origin main


IDSWG-AI-Coding-Boot-Camp / vegetables.txt

RunqiuWang22 Update vegetables.txt

Code | Blame

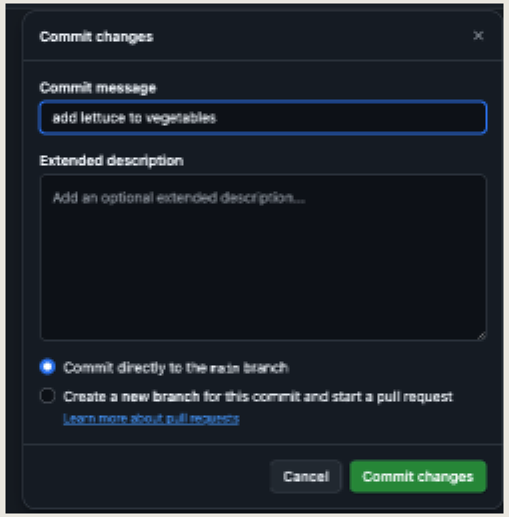
```
tomato
```

IDSWG-AI-Coding-Boot-Camp / vegetables.txt in main

Cancel changes | Commit changes...

Edit | Preview

```
tomato
lettuce
```


Commit changes

Commit message: add lettuce to vegetables

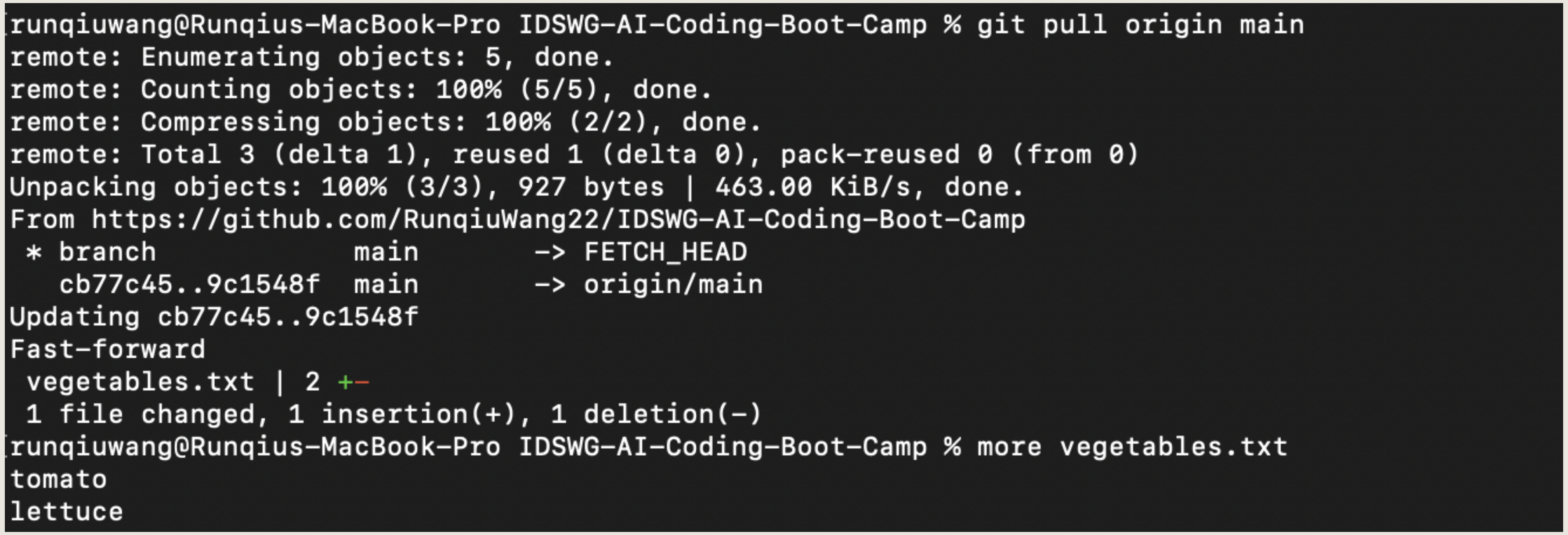
Extended description: Add an optional extended description...

[✓ Commit directly to the main branch]

Create a new branch for this commit and start a pull request

Learn more about pull requests

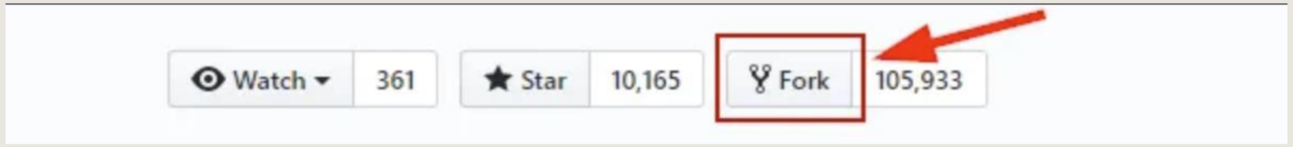
Cancel | Commit changes

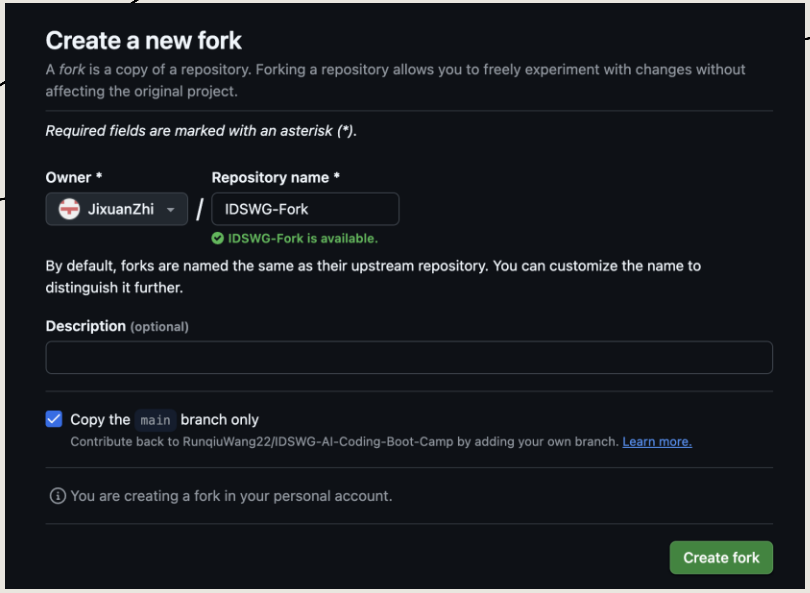

```
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 1 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 927 bytes | 463.00 KiB/s, done.
From https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp
 * branch            main       -> FETCH_HEAD
   cb77c45..9c1548f  main       -> origin/main
Updating cb77c45..9c1548f
Fast-forward
 vegetables.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
runqiuwang@Runqius-MacBook-Pro IDSWG-AI-Coding-Boot-Camp % more vegetables.txt
tomato
lettuce
```

FORKING PROJECTS ON GITHUB

Step 1: Forking a repository:

https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp





Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *: JixuanZhi / Repository name *: IDSWG-Fork

IDSWG-Fork is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

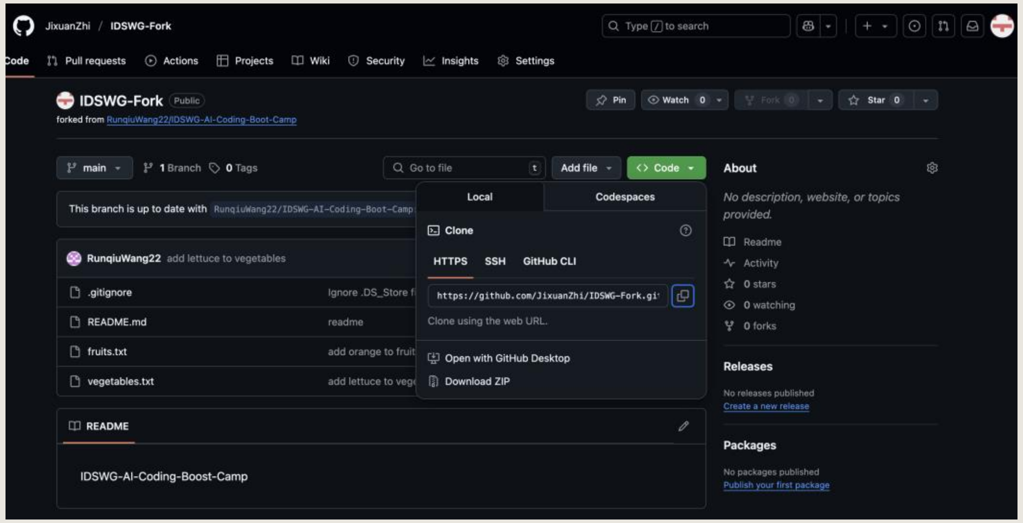
☑ Copy the `main` branch only

Contribute back to RunqiuWang22/IDSWG-AI-Coding-Boot-Camp by adding your own branch. Learn more.

You are creating a fork in your personal account.

Create fork

Step 2: Working with a forked repository on your local machine



JixuanZhi / IDSWG-Fork

IDSWG-Fork Public

forked from RunqiuWang22/IDSWG-AI-Coding-Boot-Camp

main · 1 Branch · 0 Tags

This branch is up to date with RunqiuWang22/IDSWG-AI-Coding-Boot-Camp

RunqiuWang22 add lettuce to vegetables

.gitignore	Ignore .DS_Store f
README.md	readme
fruits.txt	add orange to fruit
vegetables.txt	add lettuce to veg

Local · Codespaces

Clone: HTTPS · SSH · GitHub CLI

`https://github.com/JixuanZhi/IDSWG-Fork.gi`

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

README

IDSWG-AI-Coding-Boost-Camp

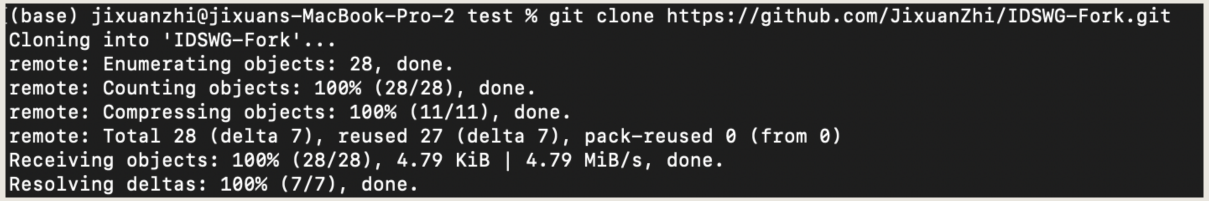
About

No description, website, or topics provided.

Readme · Activity · 0 stars · 0 watching · 0 forks

Releases: No releases published. Create a new release

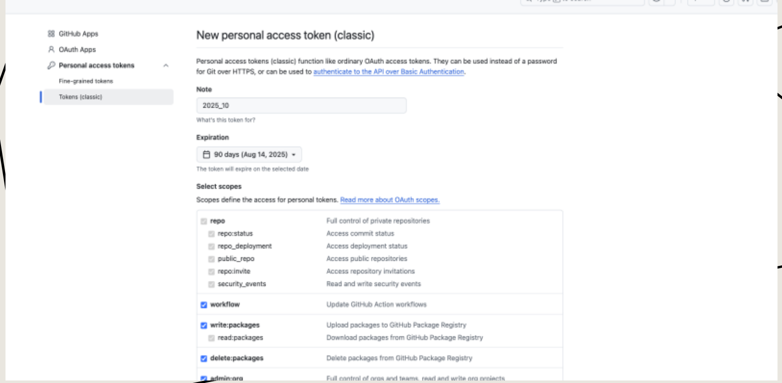
Packages: No packages published. Publish your first package



```
(base) jixuanzhi@jixuans-MacBook-Pro-2 test % git clone https://github.com/JixuanZhi/IDSWG-Fork.git
Cloning into 'IDSWG-Fork'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 28 (delta 7), reused 27 (delta 7), pack-reused 0 (from 0)
Receiving objects: 100% (28/28), 4.79 KiB | 4.79 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

git clone https://github.com/<your-username>/<project-name>.git

FORKING PROJECTS ON GITHUB



GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note
2025_10
What's this token for?

Expiration
90 days (Aug 14, 2025)
The token will expire on the selected date

Select scopes
Scopes define the access for personal tokens. Read more about OAuth scopes.

☐ repo	Full control of private repositories
☐ repo:status	Access commit status
☐ repo_deployment	Access deployment status
☐ public_repo	Access public repositories
☐ repo:invite	Access repository invitations
☐ security_events	Read and write security events
☑ workflow	Update GitHub Action workflows
☑ write:packages	Upload packages to GitHub Package Registry
☐ read:packages	Download packages from GitHub Package Registry
☑ delete:packages	Delete packages from GitHub Package Registry
☑ admin:org	Full control of orgs and teams, read and write org projects

Step 3: Add the upstream repository (if not already added)
git remote add upstream https://github.com/RunqiuWang22/IDSWG-AI-Coding-Boot-Camp.git

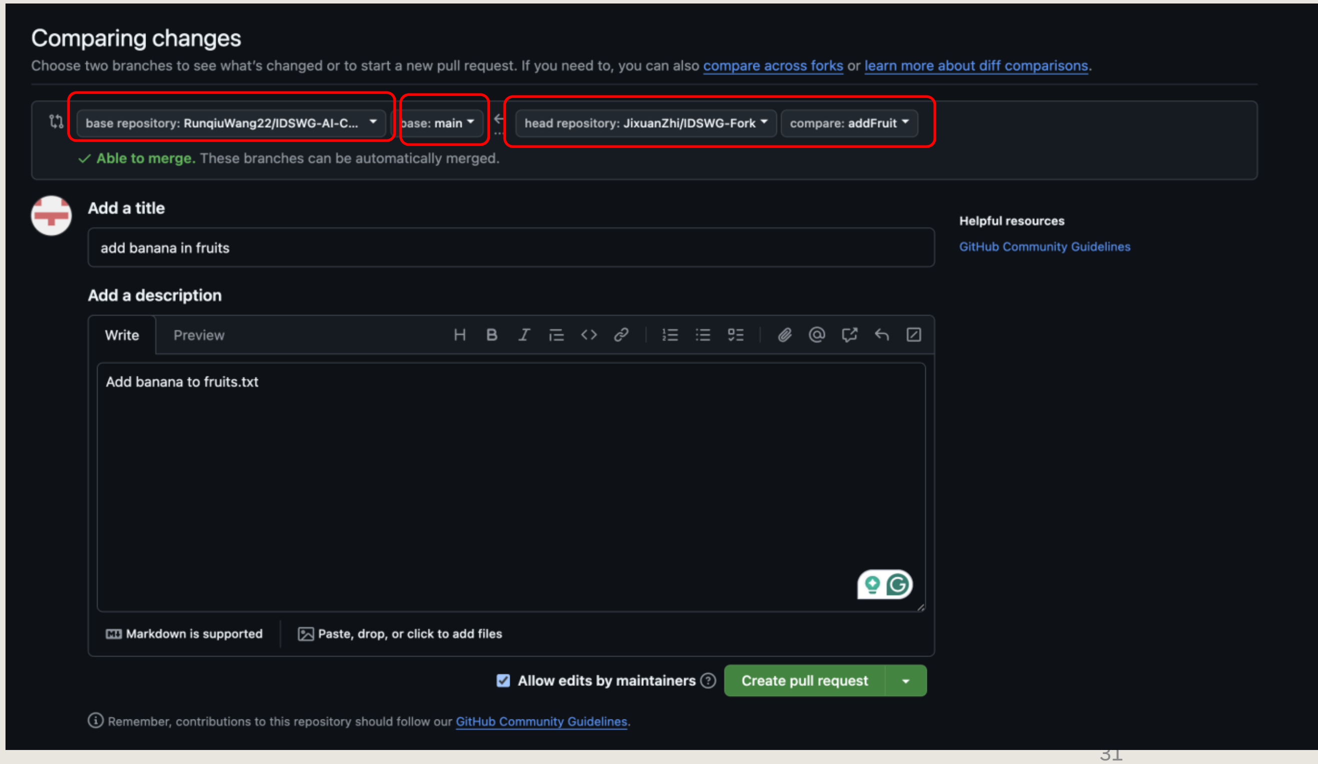
Step 4: List your branch for your changes
git checkout
git checkout -b <your branch name>

Step 5: Make your changes
git add .
git commit -m "Describe your change"

Step 6: Push your changes to your fork
git push origin <your branch name>

Step 7: Create a Pull Request on GitHub

- Go to your fork on GitHub
- Click "Compare & pull request" button
- Choose:
- **Base repo**: the *original* repository (upstream)
- **Base branch**: `main` on the upstream repo
- **Compare**: your branch on your fork



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base repository: RunqiuWang22/IDSWG-AI-C... base: main ← head repository: JixuanZhi/IDSWG-Fork compare: addFruit

✓ Able to merge. These branches can be automatically merged.

Add a title
add banana in fruits

Add a description
Write Preview

Add banana to fruits.txt

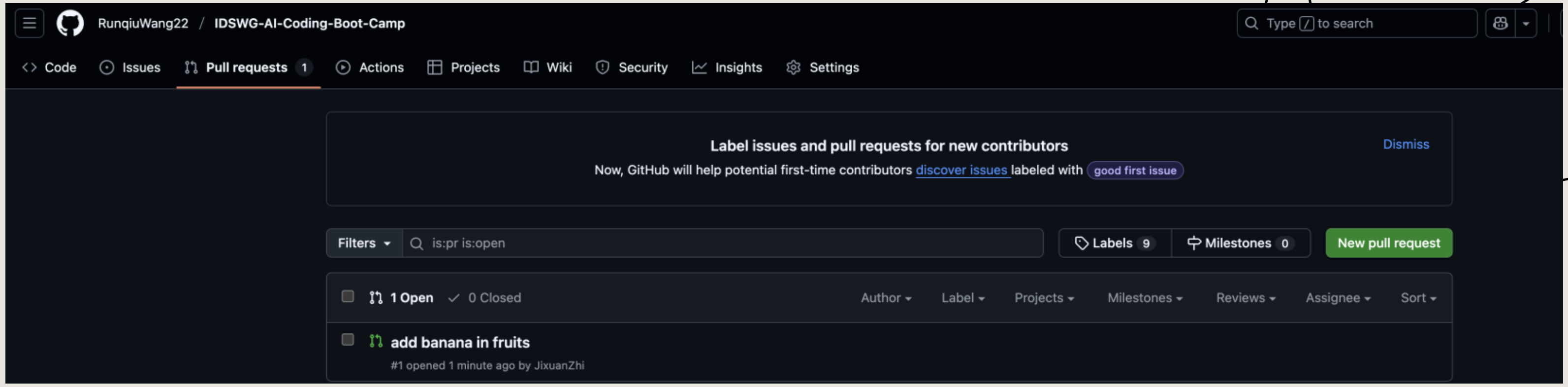
Markdown is supported Paste, drop, or click to add files

☑ Allow edits by maintainers Create pull request

Helpful resources
GitHub Community Guidelines

Remember, contributions to this repository should follow our GitHub Community Guidelines.

REVIEW PULL QUEST AND COMMIT



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

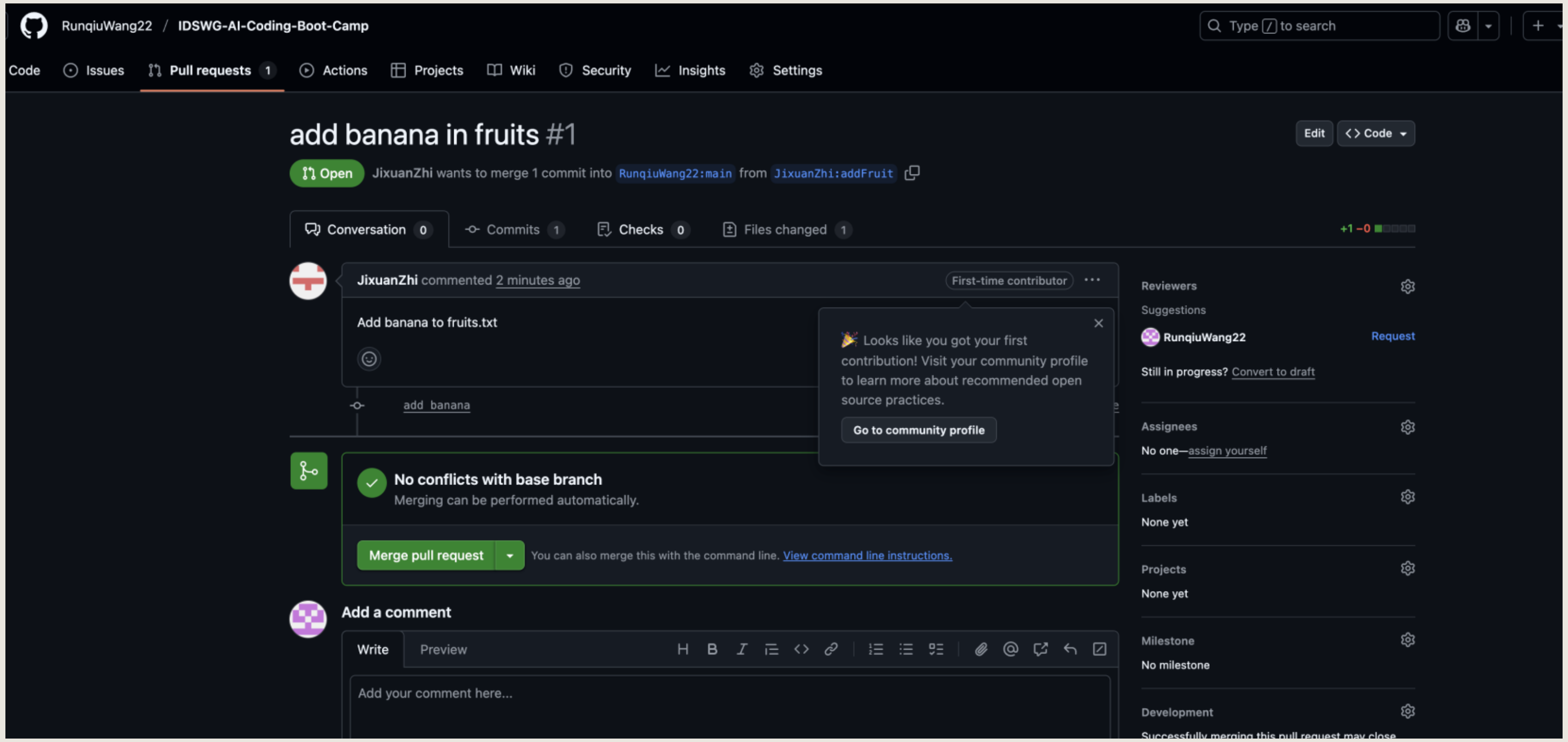
Dismiss

Filters is:pr is:open Labels 9 Milestones 0 New pull request

1 Open 0 Closed Author Label Projects Milestones Reviews Assignee Sort

add banana in fruits

#1 opened 1 minute ago by JixuanZhi



add banana in fruits #1

Open JixuanZhi wants to merge 1 commit into RunqiuWang22:main from JixuanZhi:addFruit

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 −0

JixuanZhi commented 2 minutes ago First-time contributor

Add banana to fruits.txt

add banana

Looks like you got your first contribution! Visit your community profile to learn more about recommended open source practices.

Go to community profile

No conflicts with base branch

Merging can be performed automatically.

Merge pull request You can also merge this with the command line. View command line instructions.

Add a comment

Write Preview

Add your comment here...

Reviewers

Suggestions

RunqiuWang22 Request

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close

SYNCING A FORK TO KEEP IT UP-TO-DATE WITH THE UPSTREAM REPOSITORY.

Step 1: Fetch from upstream

```
git fetch upstream
```

Step 2: Update your local main

```
git checkout main
git merge upstream/main
```

Step 3: Push to your fork if needed

```
git push origin main
```

Useful link: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/syncing-a-fork

PROJECTS

PROJECTS

- demographics table summary;
- patient narratives;
- waterfall plot data extraction;
- swimmer plot data extraction;
- create one table shell

INSTALL LOCAL LLM THROUGH OLLAMA

INSTALL OLLAMA

- For MAC user and Other users with Nvidia and AMD GPUs: download ollama from https://ollama.com/, download and follow common software installation.
- For Windows user with integrated Intel GPU: https://github.com/intel/ipex-llm/blob/main/docs/mddocs/Quickstart/ollama_portable_zip_quickstart.md
  - Download and unzip it to a folder
  - Add the folder path to the environment variable PATH
  - Please download and install the intel driver

THANK YOU

Brita Tamm

502-555-0152

brita@firstupconsultants.com

www.firstupconsultants.com