Special Topics in Linux

- By Anas Mohammad, Experimental Data Engineer @SESAME
- (30-04-2025) session start at 09:00 AM UTC

tmux

Terminal Multiplexer:

- tmux creates multiple terminals from a single screen.
- Access several command lines in one terminal window.

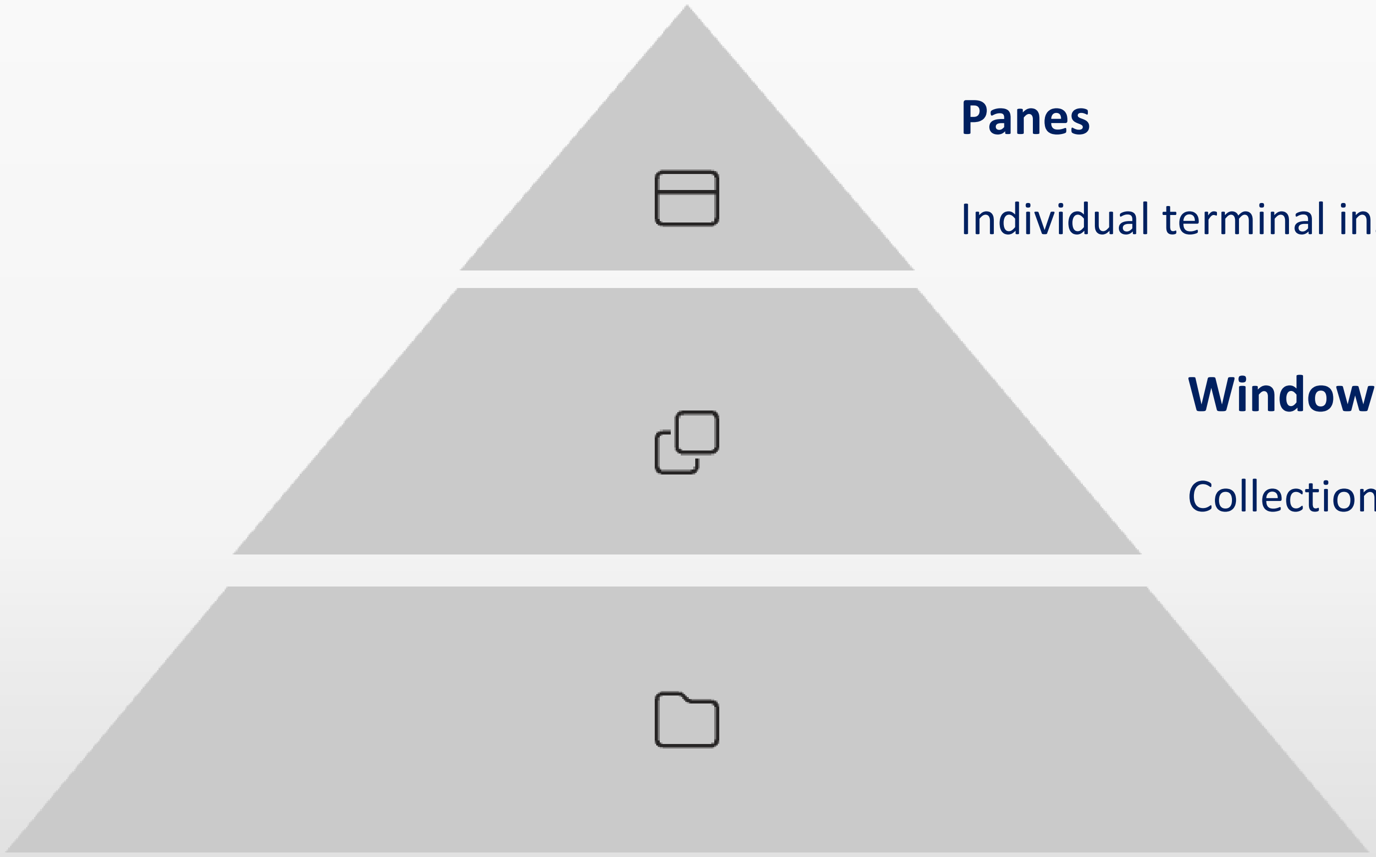
Client-Server Architecture:

- Runs as a persistent server.
- Multiple clients can attach to this server, providing flexibility and resilience.

Virtual Terminal Environment:

- Creates manageable virtual terminals.
- Arrange them in various layouts for efficient multitasking.

Basic tmux Structure



Panes

Individual terminal instances

Windows

Collections of panes

Sessions

Groups of windows

tmux organizes your workspace hierarchically.

Sessions contain windows, which contain panes.

This structure enables efficient organization of complex workflows.

Start with tmux

Creating Sessions: (*Ctrl+b Deafult prefix*)

- **Basic Session Creation:** tmux new-session (or simply tmux)
- **Named Session:** tmux new -s session_name
- **Session with Working Directory:** tmux new-session -s name -c /path/to/dir
- **Detached Session:** tmux new-session -d -s background_session
- **Detaching from Sessions:** Press Ctrl+b, then d to detach

Listing Sessions: tmux ls
Within tmux: Ctrl+b s

Killing Sessions:
- **Kill Current Session:** From within: Ctrl+b then type :kill-session or Ctrl d
- **Kill Specific Session:** tmux kill-session -t session_name
- **Kill All Sessions:** tmux kill-server

Working with Windows

Create Window: Ctrl+b c

List Windows: Ctrl+b w

Previous window: Ctrl+b p

Next Window: Ctrl+b n

Rename Current Window: Ctrl+b , (comma) then type new name

Change Window Position: Ctrl+b . (period) then type new index

Find Window: Ctrl+b f then type search term

Close Current Window: Ctrl+b &

Working with panes

Vertical Split: Ctrl+b %

Horizontal Split: Ctrl+b "

Directional Navigation

- Ctrl+b ↑ (Move up)
- Ctrl+b ↓ (Move down)
- Ctrl+b ← (Move left)
- Ctrl+b → (Move right)

Numeric Navigation

- Ctrl+b q (Display pane numbers)
- Press the number quickly to select
- Ctrl+b o (Move to next pane)

Resizing Panes

- Ctrl+b Ctrl+↑ (Resize up)
- Ctrl+b Ctrl+↓ (Resize down)
- Ctrl+b Ctrl+← (Resize left)
- Ctrl+b Ctrl+→ (Resize right)

Working with panes

Close Current Pane

Ctrl+b x

Will prompt for confirmation if it's the last pane.

Convert Pane to Window

Ctrl+b !

Breaks current pane out to its own window.

Cycle Through Layouts

Ctrl+b Space

Rotates through preset pane arrangements.

Zoom Pane

Ctrl+b z

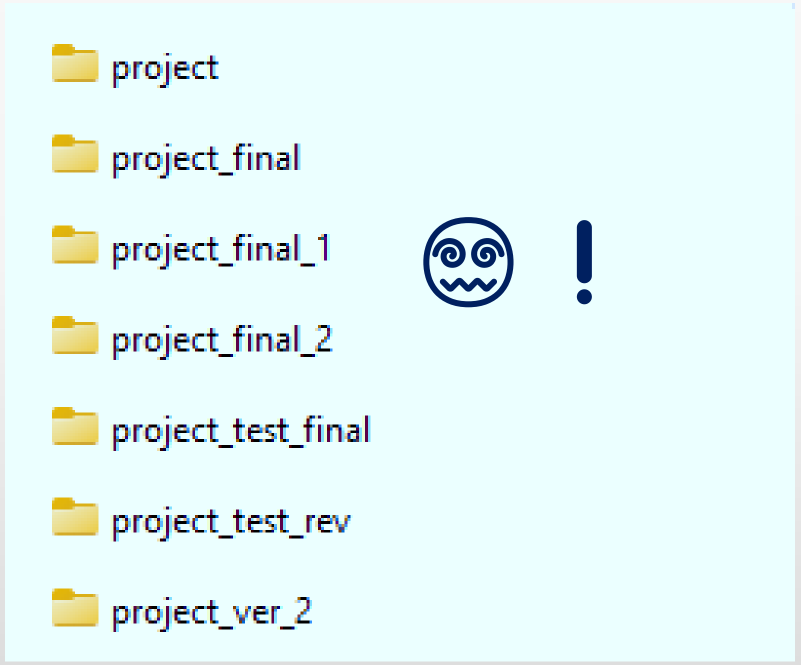
Toggles between full-screen and normal view.

GitHub

By STEVEN J. CLIPMAN

GitHub

What is version control and why we use it?


project
project_final
project_final_1
project_final_2
project_test_final
project_test_rev
project_ver_2

GitHub

Version Control System

- GitHub is a cloud-based platform built around Git. It tracks changes to code over time.

Collaboration Platform

- Developers work together on projects. Teams can coordinate from anywhere in the world.

Code Repository

- A centralized storage location. It houses millions of software projects.

EUMEDplus Bridging the Mediterranean ASREN SESAME

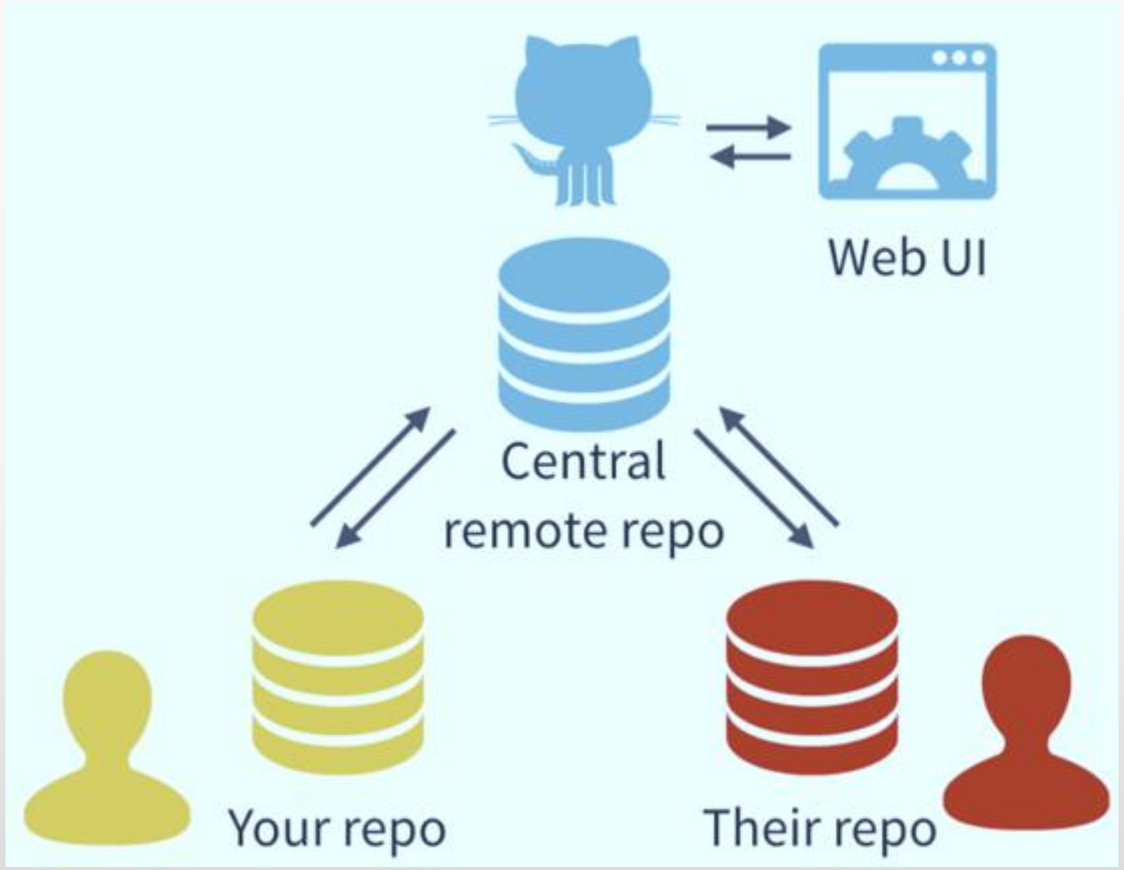
Git

- Git and GitHub are two different things.
- Git is a particular implementation of version control originally designed by Linus Torvalds in 2005 as a way of managing the Linux kernel. Git manages the evolution of a set of files– called a repository or repo.
- Essentially, the language of version control.

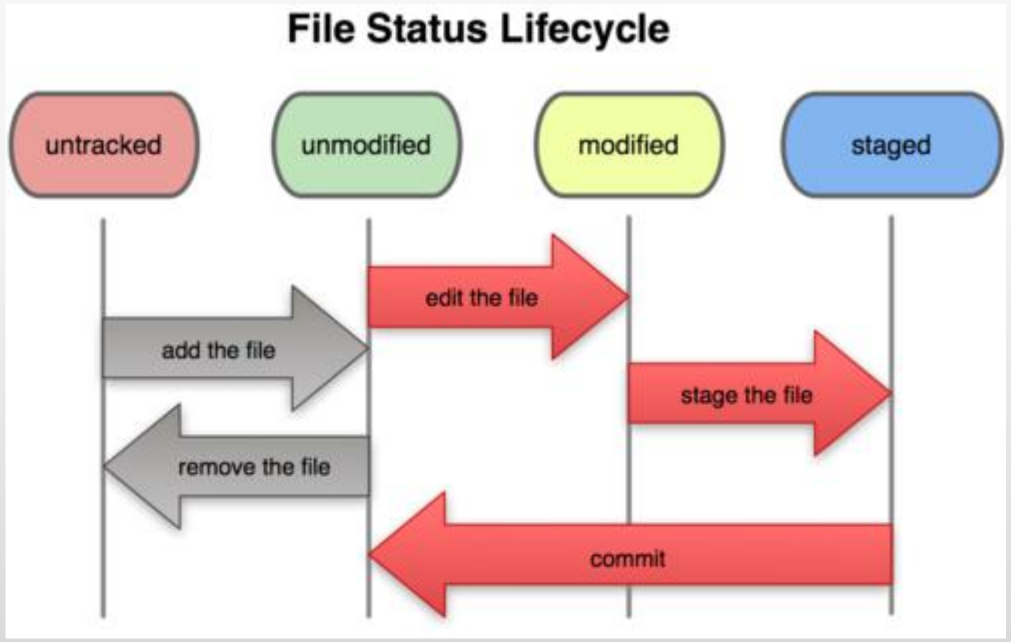
GitHub

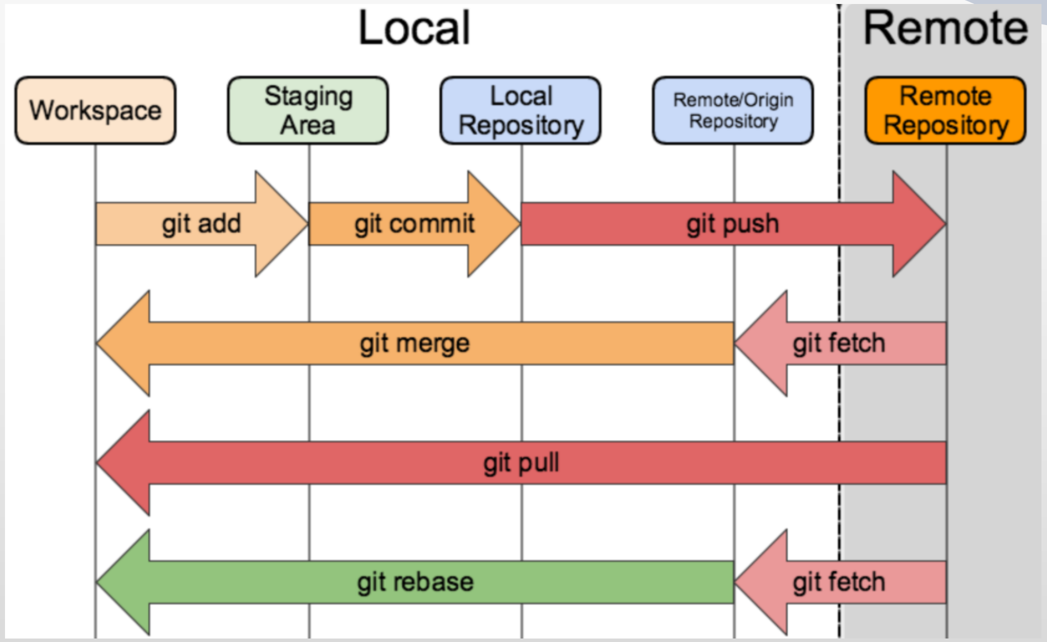
- GitHub is an online hub for hosting Git repositories and provides GUI software for using Git.
- GitHub complements Git by providing a slick user interface and distribution mechanism for repositories. Git is the software you will use locally to record changes to a set of files. GitHub is a hosting service that provides a Git-aware home for such projects on the internet.
- GitHub is like DropBoxor Google Drive, but more structured, powerful, and programmatic.

How it works?

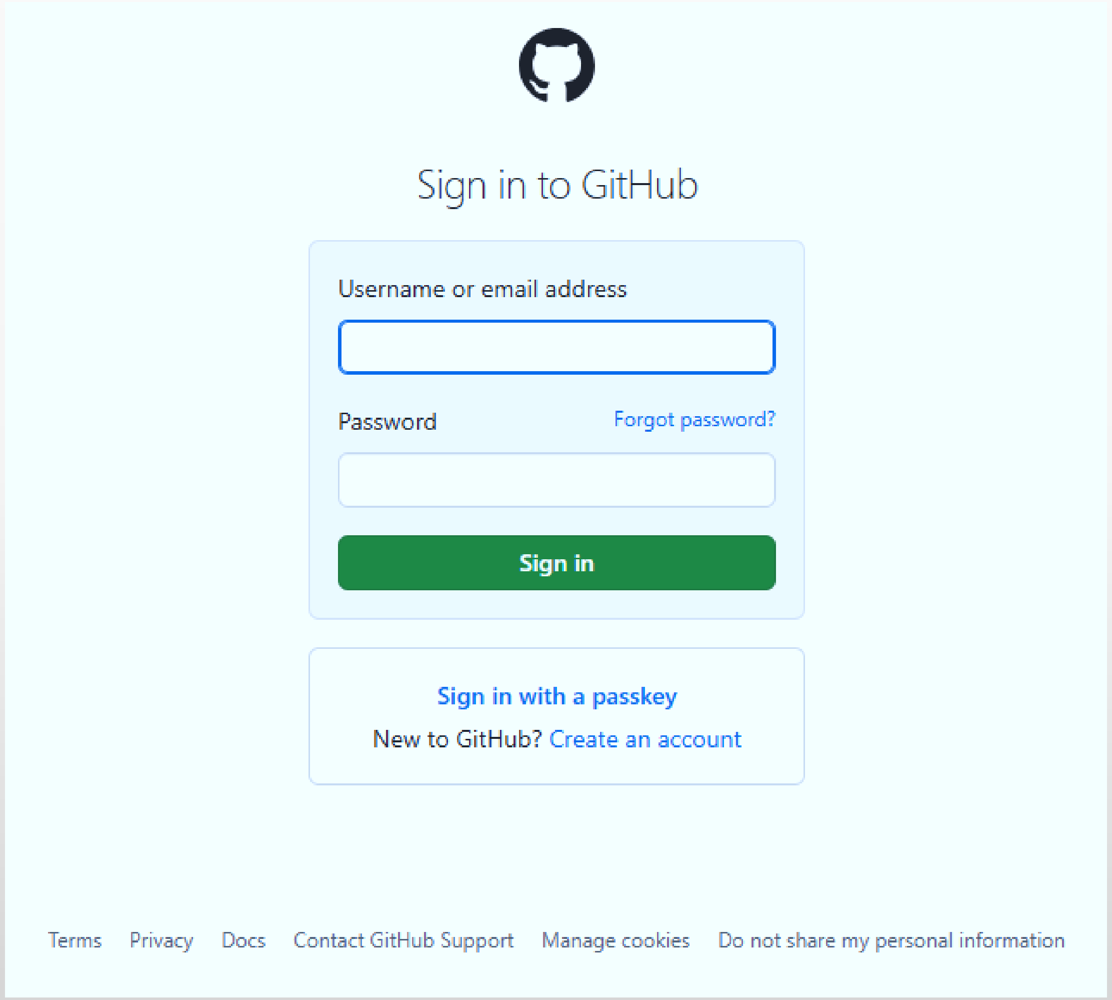


How it works?


File Status Lifecycle
untracked
unmodified
modified
staged
edit the file
add the file
stage the file
remove the file
commit


Local
Remote
Workspace
Staging Area
Local Repository
Remote/Origin Repository
Remote Repository
git add
git commit
git push
git merge
git fetch
git pull
git rebase
git fetch

Get Started with web UI


Sign in to GitHub

Username or email address

Password

Forgot password?

Sign in

Sign in with a passkey

New to GitHub? Create an account

Terms Privacy Docs Contact GitHub Support Manage cookies Do not share my personal information

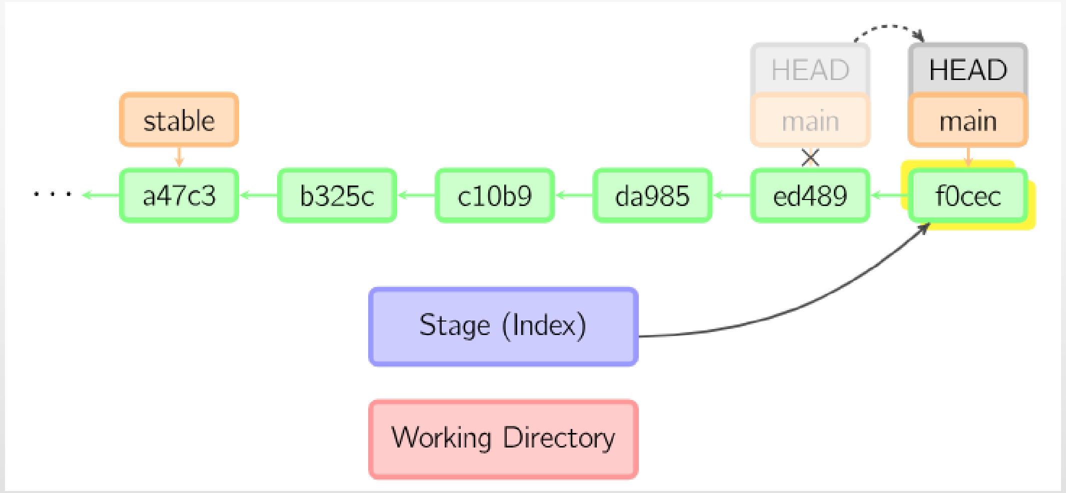
Basic Git Commands

```
git init     ### create/initialize the repository

git add      ### track the file (send the file to staging area)

git commit –m “commit”  ### save the changes to the history with a message

git status   ### check the activity

Git log      ### show the commits history

git diff     ### compare to the one in the history

### connect your local Git repository to the one hosted on GitHub, allowing you to push and pull changes.
git remote add origin https://github.com/sesame/test.git

git push origin main  ### push your local main branch to the origin remote

git pull    ### fetch changes from a remote repository and merge them into your current branch

git fetch   ### fetch changes from a remote repository without merging them into your current branch
```

Understand Commit

Commits contain 3 pieces of information:

1. Information about how the files have changed from the previous commit.
2. A reference to the commit that came before it, known as the parent commit.
3. An SHA-1 hash code (e.g. fb2d2ec5069fc6776c80b3ad6b7cbde3cade4e)



What is sha-1 hash?

SHA-1 is an algorithm and what it does is:

It takes some data as input and generates a unique 40 character string from it.

What does unique mean in this context? Unique means that no other input data should ever produce the same hash. The same input data however should always produce exactly the same hash.

Key Concepts

1. **https vs ssh:**

SSH (Secure Shell):

- Authentication: Uses public/private key pairs.
- Setup: Requires generating an SSH key and adding it to your GitHub account.
- Usage: Once set up, it’s seamless—no need to enter your username/password every time.
- URL format:git@github.com:user/repo.git
- Best for: Developers who work with Git frequently, especially on private repos or automated scripts.

HTTPS:

- Authentication: Uses GitHub username and personal access token (PAT).
- Setup: Simple—just clone with a URL and input credentials when prompted.
- Usage: May require re-authentication unless credentials are cached.
- URL format: https://github.com/user/repo.git
- Best for: Quick access or environments where you can't use SSH keys.

Key Concepts

2. Repository: Often referred to as a repo, this a collection of all your files and the history of those files (i.e. commits) and can live on a local machine or on a remote server (e.g. GitHub)

3. README.md: Collaborate responsibly! Every repository should be initialized with a README.MD file. This is where you detail all the information about the project so that the work can be understood and recreated by another student or collaborator

4. .gitignore: A gitignore file specifies intentionally untracked files that Git should ignore. Files already tracked by Git are not affected.

Writing a good commit message

1. Use the imperative mood in the subject line

2. Limit the subject line to 50 characters

3. Capitalize the subject line

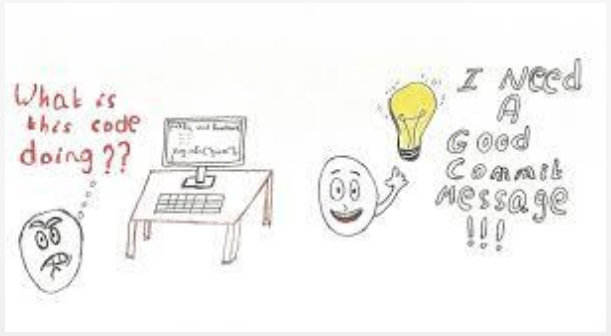
4. Do not end the subject line with a period

5. Separate subject from body with a blank line

6. Use the body to explain what and why, not how

7. Reference issues or pull requests when applicable

8. Use consistent structure and tags (if part of a team standard)



GitHub Task

1. Fork the following repository to your account: https://github.com/SESAME-Synchrotron/EUMEDPlus

2. Clone the forked repository to your terminal.

3. Create a python virtual environment.

4. Try to run the python script called `plot.py`

5. Install the necessary packages to run the script.

6. Create `requirements.txt` packages file.

7. Push the changes to remote repository.

References to learn Git

1. Official git site and tutorial: https://git-scm.com/
2. GitHub guides: https://guides.github.com/
3. Command cheatsheet: https://training.github.com/kit/ downloads/github-git-cheat-sheet.pdf
4. Mardown cheatsheet: https://github.com/adam-p/markdown-here/wiki/Markdown-Here-Cheatsheet
5. Interactive git tutorial: https://try.github.io/levels/1/challenges/1
6. Visual/interactive cheatsheet: http://ndpsoftware.com/git-cheatsheet.html

Readthedocs

What is ReadtheDocs?

Open-Source Platform: A community-driven documentation hosting service that makes technical documentation free and accessible.

Features:

1. Fully Searchable
2. Multiple Formats
3. Version Control
4. Customizable Themes

Read the Docs Architecture:

1. Python Foundation: Built with Python and Django web framework for maintainability and extensibility.
2. Sphinx Integration: Uses Sphinx documentation generator to transform RST files into documentation.
3. Cloud Infrastructure: Hosted on Amazon Web Services for reliability and scalability.

Main Components of ReadtheDocs

1. requirements.txt
2. .readthedocs.yaml
3. source/conf.py
4. source/index.rst

Get Started:

https://github.com/SESAME-Synchrotron/EUMEDPlus

Readthedocs Tutorial:

https://docs.readthedocs.com/platform/latest/tutorial/index.html