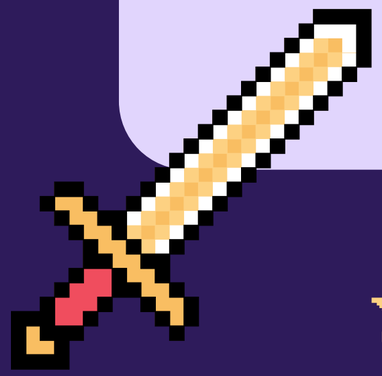
MLSA SRM

INTRODUCTION TO GIT AND GITHUB

A Workshop on Version Control and Collaboration



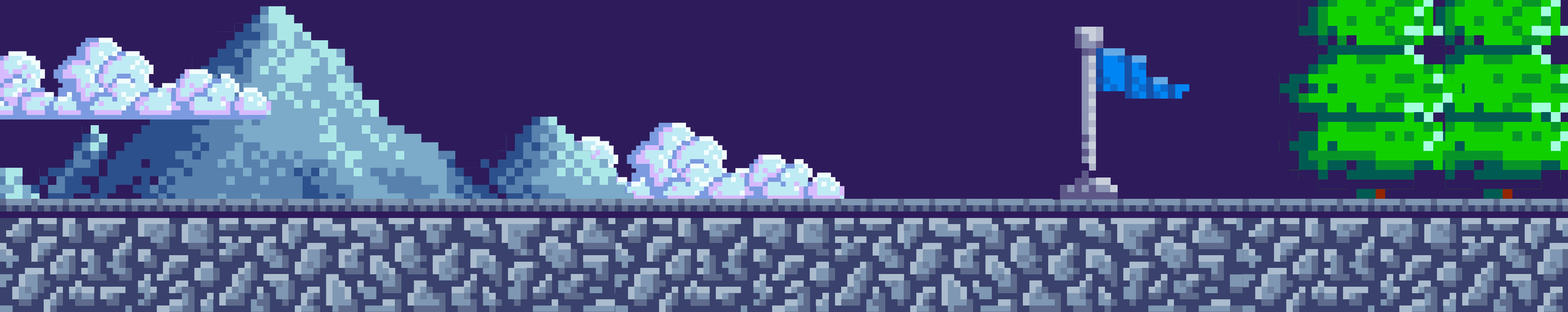

Hello World
Senior dev
New dev

WHAT IS VERSION CONTROL;

1. System to track changes in files over time
2. Maintain a history of modifications
3. Facilitate collaboration and code management
4. Key for transparency, teamwork, and code quality

INTRODUCTION TO GIT

- Distributed Version Control System
- Tracks changes to files over time
- Enables collaboration among developers
- Developed by Linus Torvalds in 2005



WHAT DOES GIT DO

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

LITERALLY NO ONE:

DEVS PUTTING A PICTURE OF THEMSELVES ON THEIR PERSONAL SITE AND/OR GITHUB:

UNDERSTANDING GITHUB

- Web-based platform for Git repositories
- Facilitates collaboration and code hosting
- Issue tracking, project management, and more
- Used by millions of developers worldwide

Basic Git and Github Terminology

- **Repository/repo:** The database storing the files. These are more like folders on our system.
- **Commit:** Commit is executing a change that you bring to your program, it can be adding, removing, and modifying code or files from your project.
- **Local and Remote:** Your project will have two independent repos, one which is offline is called Local and the one which is hosted online on a platform like GitHub is Remote.
- **Pull, Push or Fetch:** to sync your project between local and remote we use these three operations.

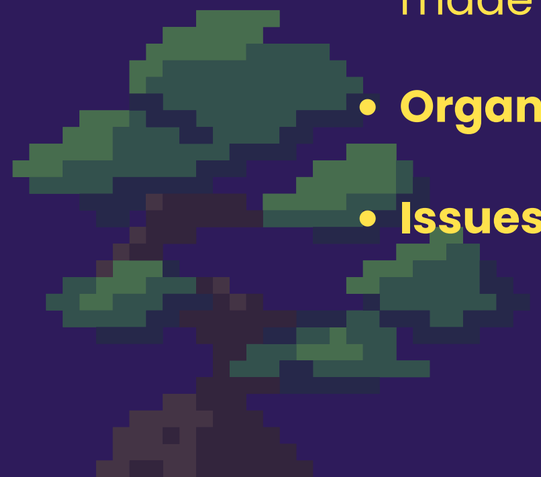
Pull - pull changes from remote to local

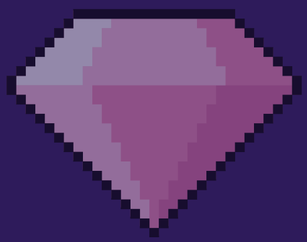
Push - push changes from local to remote

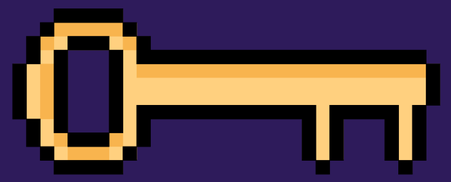
Fetch - only downloads new data but doesn't integrate it to your working project in local

Basic Git and Github Terminology

- **Branching :** Branching out or creating a new branch is having another copy of your code on the same repository and they basically divert you from the mainline of development, so that you can fix the bug or build a new feature and then merge it back without messing up the main code.
- **Pull Request:** it is simply a way of telling people that you want the changes you made in the branch to get included in Main Code.
- **Organizations:** Organizations are group-owned repos.
- **Issues:** Keep track of enhancements, and bugs for your project.





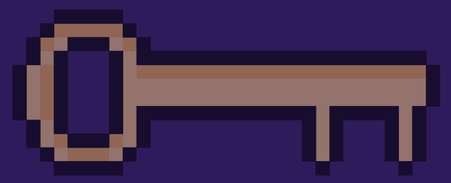


HANDS ON PRACTICE

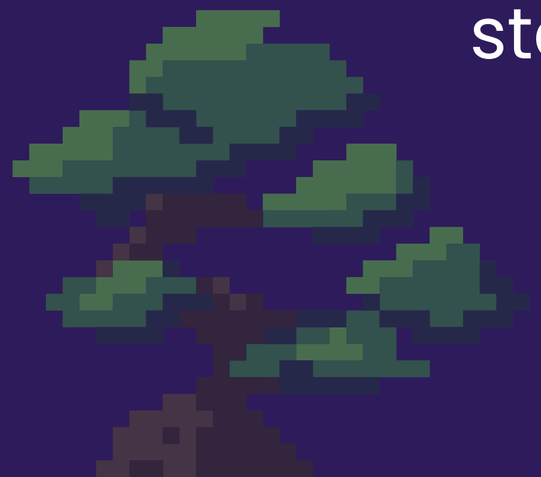
Let's get started

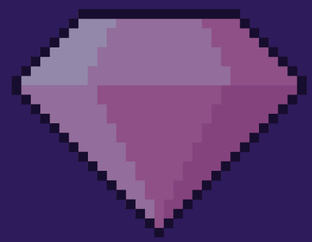
https://t.ly/JM4hK

Git Commands

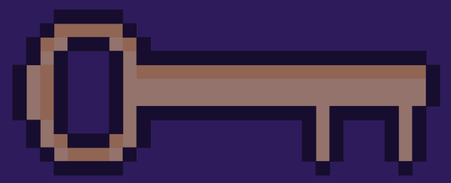


- **git init**: Initializes a new Git repository.
- **git clone <repository_url>**: Copies a repository from GitHub to your local machine.
- **git add <file(s)>**: Stages changes for the next commit.
- **git commit -m "Commit message"**: Commits staged changes with a descriptive message.

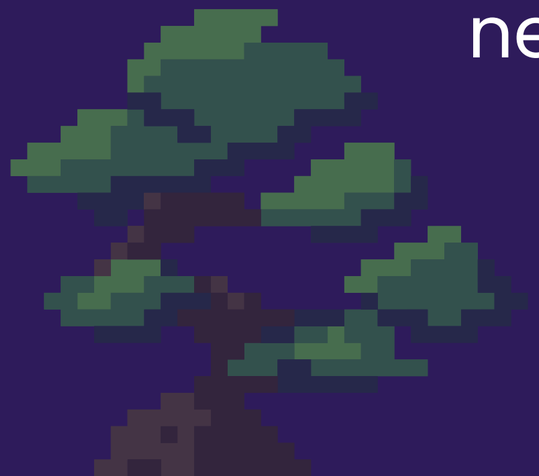


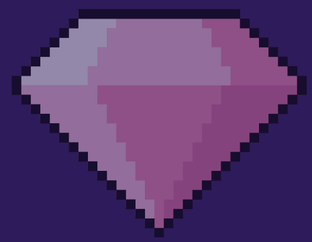


Git Commands

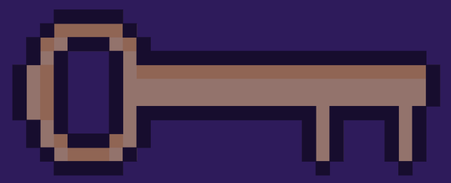


- **git status**: Shows the status of your working directory and staged changes.
- **git log**: Displays a history of commits.
- **git branch**: Lists all branches in the repository.
- **git checkout -b <branch_name>**: Creates a new branch and switches to it.



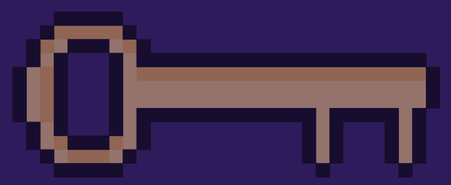


Git Commands

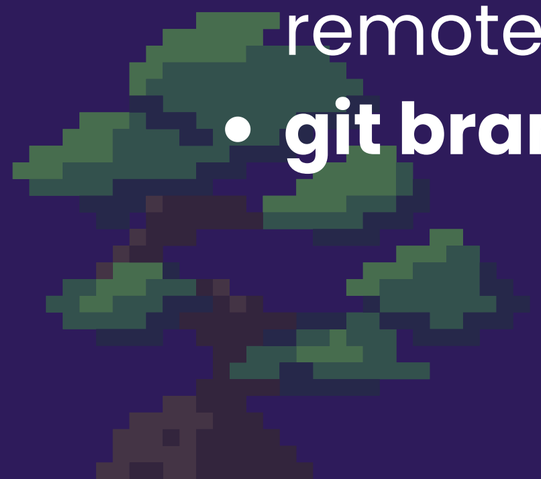


- **git pull:** Fetches changes from a remote repository and integrates them into the current branch.
- **git push:** Pushes local changes to a remote repository.
- **git remote -v:** Lists the remote repositories associated with your local repository.
- **git fetch:** Fetches changes from a remote repository but doesn't integrate them.

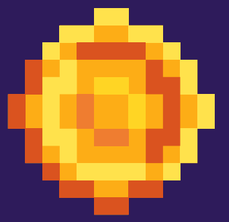
Git Commands



- **git merge <branch_name>:** Merges changes from one branch into the current branch.
- **git pull origin <branch_name>:** Fetches and merges changes from a remote branch.
- **git push origin <branch_name>:** Pushes changes to a remote branch.
- **git branch -d <branch_name>:** Deletes a local branch.

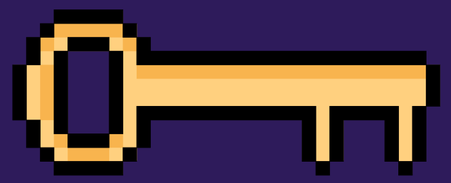


fun fact



The name "git" was given by Linus Torvalds when he wrote the very first version. He described the tool as "the stupid content tracker"

The Emoji License: There's an open-source software license called the "Beerware License," which states that you can use the software under the condition that if you meet the author and you like the software, you should buy them a beer.



GIT != GOOGLE DRIVE


HACKTOBER
FEST00010

WHAT IS HACKTOBERFEST

Hacktoberfest is an annual month-long event that encourages people to contribute to open source projects on platforms like GitHub.

RESOURCES

Resources to get started

Git

Atlassian Docs on Git

Video on Git and Github by Kunal Kushwaha

Commits Naming Conventions

Pull Request Naming Conventions

Git Cheatsheet

What is Open Source

Register for Hacktoberfest

Hacktoberfest

Resources to find issues where you could contribute to

Finder by EddieHub Community

GitHub's advanced search

GitHub Topics

Good First Issue label

Scoutflo Atlas

THANKYOU