GitHub 

GitHub is a web-based **version control**, *collaboration*, and **source code** *management platform*.

It uses **Git**, a *distributed version control system* created by Linus Torvalds, to **manage** and **track changes** in code across **different versions**.

@rkstlohchab

VERSION CONTROL

GitHub helps developers manage changes to their codebase, allowing them to track and revert changes if necessary.

It records *who* made changes and *when*, making it easy to collaborate with others.

COLLABORATION

Multiple developers can work on the same project simultaneously.

GitHub provides *tools* like pull requests, code reviews, and discussions to facilitate collaborative development.

Repositories

Projects on GitHub are **stored** in repositories.

A **repository** contains all the project **files** and the **history** of **changes** made to those files.

Repositories can be **public** or **private**.

Branching and Merging

GitHub allows developers to create branches, which are separate versions of the codebase.

This is useful for developing *features*, *fixing bugs*, or *experimenting* with new ideas without affecting the *main* codebase.

Changes from branches can be merged back into the *main* codebase when ready.

Issue Tracking

GitHub includes tools for tracking issues, *bugs*, and feature requests. This helps teams manage and prioritize their work.

Continuous Integration and Deployment

GitHub integrates with various CI/CD tools, allowing for automated testing and deployment of code changes.

Community and Open Source

GitHub hosts millions of open-source projects.

***Developers* can contribute to these projects, and organizations often use GitHub to manage and share their open-source software.**

Documentation

GitHub supports Markdown, allowing developers to create well-formatted documentation directly within their repositories.

Overall, GitHub is a crucial tool for modern *software development*, enabling efficient code management and fostering collaboration among developers *globally*.

CHEATSHEET

```
// Repository
git init                        // Initialize a new git repo
git clone <repo-url>            // Clone a repo from a URL

// Basics
git status                      // Show changes status
git add <file>                  // Add changes to staging
git commit -m "message"         // Commit changes with a message
git log                         // View commit history
```

@rkstlohchab

```
// Branching
git branch                      // List branches
git branch <branch-name>        // Create a new branch
git checkout <branch-name>      // Switch to a branch
git merge <branch-name>         // Merge changes from a branch
git branch -d <branch-name>     // Delete a branch

// Remote Repositories
git remote                      // List remotes
git remote add <name> [URL]     // Add a remote
git push <remote> [branch]      // Push changes to a remote
git pull <remote> [branch]      // Pull changes from a remote


// Undoing Changes
git pull                        // Fetch and merge changes
git fetch                       // Fetch changes without merging
git reset --hard HEAD           // Discard changes
git revert <commit-hash>        // Revert changes in a commit
```

@rkstlohchab

FOUND IT USEFUL?

Follow for more

www.linkedin.com/in/rkstlohchab

@rkstlohchab