Sure! I can help you make your notes concise and organized into a clean **Git & GitHub Cheat Sheet**. Here's a polished version of your learning notes:

Git & GitHub Cheat Sheet

Date: 16.1.2026

Why Git?

- Enables collaboration from anywhere.
- Tracks full history of code.
- Easy to fix mistakes or revert changes.

What is Git?

- Distributed Version Control System (DVCS).
- Open-source tool for tracking code and changes in projects.

What is GitHub?

- Git is the tool; GitHub is a cloud-based hosting service for Git repositories.

SSH Key Concept

- Provides secure, password-less authentication.
- After generating SSH key, add it to GitHub to clone private repos locally.

Basic Git Commands

Command	Purpose
`git --version`	Check installed Git version

Command	Purpose
`git config --global user.name "name"` `git config --global user.email "email"`	Set author info for commits
`git init`	Initialize a Git repository in the current folder
`git status`	View untracked, modified, and staged files
`git add .`	Stage all changes for commit
`git commit -m "message"`	Create a snapshot of the project
`git checkout -b branch_name`	Create & switch to a new branch
`git merge branch_name`	Merge another branch into current branch
`git push`	Send local commits to remote repository
`git pull`	Update local branch with changes from remote
`git restore <file>`	Undo changes in a file or unstage it

Advanced Git Commands

Command	Purpose
`git cherry-pick <commit>`	Apply a specific commit from one branch to another
`git rebase <branch>`	Move your branch's commits on top of another branch
Merge vs Rebase	Merge: Creates a merge commit, branched history Rebase: No merge commit, linear history

Handling Merge Conflicts

- Create two branches (e.g., `dev` & `dev2`).
- Edit same file in both branches.

- When merging, Git will show conflicts to resolve manually.

```
GIT CLONE ...
```