## 🔰 1. ****Git Setup****

git config --global user.name "Your Name"

git config --global user.email "your@email.com"

git config --global core.editor code # or nano/vim

## 📁 2. ****Create a Git Repository****

git init

📌 Initializes a new Git repository in the current directory.

## 🗂️ 3. ****Clone a Repository****

git clone <repo-url>

📌 Example:

git clone https://github.com/user/repo.git

## 📌 4. ****Check Status****

git status

📌 Shows the status of changes (staged, unstaged, untracked files).

## 🧠 5. ****Track and Commit Changes****

git add filename # Track one file

git add . # Track all files in the current directory

git commit -m "Message" # Commit the changes

## ♻️ 6. ****Amend Last Commit****

git commit --amend -m "New commit message"

## 🧾 7. ****View Commit History****

git log # Full history

git log --oneline # One line per commit

git log --graph --oneline --all # Graph view

## 🧭 8. ****Branching****

git branch # List all branches

git branch new-branch # Create new branch

git checkout new-branch # Switch to a branch

git checkout -b new-branch # Create and switch

## 🔄 9. ****Merging Branches****

git checkout main

git merge new-branch

## 💥 10. ****Delete Branch****

git branch -d new-branch # Delete local branch

git push origin --delete branch-name # Delete remote branch

## 🔁 11. ****Pull Latest Changes****

git pull origin main

## 📤 12. ****Push Changes****

git push origin branch-name

## 🔗 13. ****Add Remote Repository****

git remote add origin <repo-url>

git remote -v

## 📥 14. ****Fetch from Remote****

git fetch origin

## 🔧 15. ****Stash Changes****

git stash # Stash uncommitted changes

git stash pop # Reapply the latest stashed changes

git stash list # List all stashes

## ✅ 16. ****Check Difference****

git diff # All unstaged changes

git diff --staged # Staged vs latest commit

## 🧹 17. ****Remove File from Git but Not Local****

git rm --cached filename

## 🗑️ 18. ****Undo Changes****

git restore filename # Undo uncommitted changes

git restore --staged filename # Unstage a file

## 🔐 19. ****Tag a Commit****

git tag v1.0 # Lightweight tag

git tag -a v1.0 -m "v1.0 release" # Annotated tag

git push origin v1.0

## 🌍 20. ****Working with .gitignore****

Create a file named .gitignore and add files/folders to ignore:

node\_modules/

.env

\*.log

Then:

git rm -r --cached .

git add .

git commit -m "Apply .gitignore"

## 🧼 21. ****Clean Untracked Files****

git clean -f # Remove untracked files

git clean -fd # Remove untracked files & directories

## 🛠️ Bonus: Useful Aliases

git config --global alias.co checkout

git config --global alias.br branch

git config --global alias.cm commit

git config --global alias.st status

Now you can use:

git co main

git br

git cm -m "message"

git st

Would you like a PDF version of this cheat sheet or a downloadable .sh script to practice all these commands locally?

##### You said:

git and git hub all commends with expalin give chat cheat notes

###### ChatGPT said:

Here’s a **Git & GitHub Cheat Sheet** in a chat-friendly format, covering **all essential commands** with **explanations and examples** – perfect for interviews, practice, or daily use.

# ✅ ****GIT + GITHUB COMPLETE CHEAT SHEET****

## 🔧 GIT CONFIGURATION

| **Command** | **Explanation** |
| --- | --- |
| git config --global user.name "Your Name" | Set your Git username |
| git config --global user.email "you@example.com" | Set your Git email |
| git config --list | Show all Git config settings |

## 📁 INIT, CLONE, REMOTE

| **Command** | **Explanation** |
| --- | --- |
| git init | Initialize a new Git repository |
| git clone <repo-url> | Copy a remote repo to your system |
| git remote -v | Show remote URLs |
| git remote add origin <repo-url> | Add a remote repository |

🔸 **Example:**

git clone https://github.com/user/repo.git

## 📄 STAGING & COMMITTING

| **Command** | **Explanation** |
| --- | --- |
| git status | Show current status of files |
| git add file.txt | Stage a file |
| git add . | Stage all changes |
| git commit -m "message" | Commit staged changes |
| git commit --amend | Modify the last commit |

## 🔍 LOGS & DIFFS

| **Command** | **Explanation** |
| --- | --- |
| git log | Show commit history |
| git log --oneline | Compact view of commits |
| git diff | View unstaged changes |
| git diff --staged | View staged changes |

## 🔁 BRANCHING

| **Command** | **Explanation** |
| --- | --- |
| git branch | List all branches |
| git branch new-branch | Create a new branch |
| git checkout new-branch | Switch to a branch |
| git checkout -b new-branch | Create & switch to new branch |
| git merge branch | Merge into current branch |
| git branch -d branch | Delete a local branch |

## 🔄 PULL, PUSH, FETCH

| **Command** | **Explanation** |
| --- | --- |
| git pull origin main | Pull latest code from remote |
| git push origin branch-name | Push branch to GitHub |
| git fetch | Get remote changes but don’t merge |

## 🧹 STASHING

| **Command** | **Explanation** |
| --- | --- |
| git stash | Save uncommitted changes |
| git stash pop | Apply latest stash |
| git stash list | List all stashes |

## 🗑️ UNDOING CHANGES

| **Command** | **Explanation** |
| --- | --- |
| git restore file.txt | Undo changes in file |
| git restore --staged file.txt | Unstage file |
| git reset HEAD~1 | Undo last commit (keep changes) |
| git reset --hard HEAD~1 | Delete last commit (lose changes) |

## 🌟 TAGGING

| **Command** | **Explanation** |
| --- | --- |
| git tag v1.0 | Create a lightweight tag |
| git tag -a v1.0 -m "msg" | Create annotated tag |
| git push origin v1.0 | Push tag to GitHub |

## 🚫 .GITIGNORE

| **Description** |
| --- |
| Used to ignore files like node\_modules, .env, etc. |

🔸 Example content of .gitignore:

node\_modules/

.env

\*.log

🔸 Apply .gitignore:

git rm -r --cached .

git add .

git commit -m "Apply .gitignore"

## 🧼 CLEAN UNTRACKED FILES

git clean -f # Remove untracked files

git clean -fd # Remove untracked files and folders

## 📌 GITHUB WORKFLOW (Basic)

1. ✅ **Create a GitHub repo**
2. 💻 git init (if not already initialized)
3. 📂 git add . and git commit -m "first commit"
4. 🔗 git remote add origin <repo-url>
5. 🚀 git push -u origin main

## 🛠️ ALIASES (Shortcuts)

git config --global alias.st status

git config --global alias.co checkout

git config --global alias.br branch

git config --global alias.cm commit

Now you can do:

git st # instead of git status

git co main # instead of git checkout main

git br # instead of git branch

git cm -m "..." # instead of git commit -m

## 💡 TIPS

* Always **pull before pushing**
* Commit often with clear messages
* Use **branches** for features/bugs
* Use .gitignore to avoid uploading sensitive or unnecessary files

Would you like me to generate this as a **PDF file** or **interactive markdown** for offline use?

##### You said:

can you give pdf or wor fomrate with interviw questions

###### ChatGPT said:

Here is your downloadable file:

📄 Git\_and\_GitHub\_CheatSheet\_with\_Interview\_Questions.docx

This Word document includes:

* A full Git & GitHub command cheat sheet with explanations
* Top Git & GitHub interview questions