**🟢 Basic Tasks**

| **Task** | **Brief Info** | **Answer (Command)** |
| --- | --- | --- |
| **1. Initialize a repository** | **Start tracking a project with Git.** | **git init** |
| **2. Configure user information** | **Set your name and email for commits.** | **git config --global user.name "Your Name"git config --global user.email "your@email.com"** |
| **3. Create and modify files** | **Add or edit project files.** | **echo "Hello" > file.txt** |
| **4. Check the status of changes** | **View untracked and modified files.** | **git status** |
| **5. Stage changes for commit** | **Prepare files to be committed.** | **git add <file>** |
| **6. Commit changes** | **Save changes to the repository.** | **git commit -m "Initial commit"** |
| **7. View commit history** | **View the list of past commits.** | **git log** |
| **8. View differences between changes** | **Compare changes with the last commit.** | **git diff** |
| **9. Remove files from the repository** | **Delete a file and stage the change.** | **git rm <file>** |

**🟡 Intermediate Tasks**

| **Task** | **Brief Info** | **Answer (Command)** |
| --- | --- | --- |
| **10. Create a new branch** | **Create a parallel line of development.** | **git branch <branch-name>** |
| **11. Switch between branches** | **Move between branches.** | **git checkout <branch-name>** |
| **12. Merge branches** | **Combine branch changes into another.** | **git merge <branch-name>** |
| **13. Resolve merge conflicts** | **Fix conflicts when merging.** | **Open conflicting files, fix manually, then:git add <file>git commit** |
| **14. Undo the last commit** | **Remove the last commit (soft keeps changes, hard discards).** | **git reset --soft HEAD~1git reset --hard HEAD~1** |
| **15. Revert a commit** | **Create a new commit to undo changes.** | **git revert <commit-hash>** |
| **16. View detailed commit history (graph)** | **See a visual log of branches.** | **git log --oneline --graph --all** |
| **17. Stash changes and apply stashes** | **Temporarily save changes.** | **git stashgit stash apply** |
| **18. Rename files and directories** | **Change file names within Git tracking.** | **git mv old.txt new.txt** |
| **19. Delete branches (local and remote)** | **Remove unused branches.** | **Local: git branch -d <branch-name>Remote: git push origin --delete <branch-name>** |

**🔵 Advanced Tasks**

| **Task** | **Brief Info** | **Answer (Command)** |
| --- | --- | --- |
| **20. Clone a remote repository** | **Copy an existing repo to your system.** | **git clone <repo-url>** |
| **21. Push changes to a remote repository** | **Upload commits to a remote.** | **git push origin <branch-name>** |
| **22. Pull changes from a remote repository** | **Get updates from a remote.** | **git pull origin <branch-name>** |
| **23. Set up and work with remotes** | **Link your local repo to a remote.** | **git remote add origin <repo-url>** |
| **24. Create and work with a fork** | **Copy another repo to your account.** | **Fork in GitHub UI, then clone locally.** |
| **25. Create and review pull requests** | **Suggest and review changes in GitHub.** | **Open a Pull Request on GitHub.** |
| **26. Rebase commits onto another branch** | **Apply commits on top of another branch.** | **git rebase <branch>** |
| **27. Squash multiple commits into one** | **Combine several commits into one.** | **git rebase -i HEAD~<n>** |
| **28. Cherry-pick specific commits** | **Apply a specific commit to a branch.** | **git cherry-pick <commit-hash>** |
| **29. Tagging versions and releases** | **Mark specific commits as versions.** | **git tag v1.0git push origin --tags** |
| **30. Set up a .gitignore file** | **Ignore files from Git tracking.** | **Create .gitignore file, add patterns.** |
| **31. Configure global and local Git settings** | **Set up project-specific settings.** | **Global: git config --globalLocal: git config** |
| **32. Work with submodules** | **Track external repositories.** | **git submodule add <repo-url>** |
| **33. Investigate and resolve detached HEAD state** | **Fix when not on a branch.** | **Create a branch from current state:git checkout -b <new-branch>** |
| **34. Use Git bisect to find bugs** | **Find the commit that introduced a bug.** | **git bisect startgit bisect goodgit bisect bad** |
| **35. Set up and work with Git hooks** | **Automate tasks (e.g., pre-commit checks).** | **Add shell scripts to .git/hooks/** |

**Would you like help practicing any of these?**