Index, cheatsheet, best practices, fundamental (1)

Lexic

1. **Repository**: This is analogous to a project's folder. It houses all the files relevant to the project, along with the respective revision histories.
2. **Commit**: This term signifies the act of saving the current state of your project within Git. It's equivalent to creating a snapshot of your project's state at a particular moment, which you can revisit anytime.
3. **Branch**: A branch represents a distinct version of the repository. Although it exists within the repository, it doesn't interfere with the main or master branch. Hence, you can work on it independently without disrupting the "live" version. Once your modifications are complete, you can merge your branch back into the master branch.

Recommended Practices

1. **Frequent Commits**: Breaking down the history into small, manageable segments makes it easier to comprehend.
2. **Clear Commit Messages**: Be specific about the changes you made and the reasons behind them.
3. **Branch Utilization**: Use branches for experimenting with new features. They help keep your work organized and separate from the project's stable version.
4. **Repository Maintenance**: Eliminate branches that are no longer in use. It's possible to restore a deleted branch if you need it in the future.
5. **Pull Before Push**: Always pull before pushing to ensure synchronization with the remote repository

Fundamental Git Commands

Here's a list of basic commands that every beginner should familiarize themselves with:

1. `git init`: This command initializes a new Git repository.
2. `git clone [url]`: This command is used to clone (or download) a repository that's already hosted on GitHub, including its entire version history.
3. `git add [file]`: This command adds a file to the staging area, thereby preparing it for a commit.
4. `git commit -m "[message]"`: This command commits changes to the head (though not yet to the remote repository).
5. `git status`: This command lists the files you've modified and those you still need to add or commit.
6. `git pull`: This command updates your local repository to the most recent commit.
7. `git push`: This command sends modifications to the master branch of your remote repository.
8. `git branch`: This command displays all local branches in the current repository.
9. `git checkout [branch-name]`: This command switches to the specified branch and updates the working directory.
10. `git merge [branch]`: This command integrates the specified branch’s history into the current branch.

▼ official GIT cheatsheet

https://education.github.com/git-cheat-sheet-education.pdf