**Git** is a distributed version control system that allows you to track software development history and collaborate on complex projects from anywhere in the world.

**Main commands**

1. **Git clone**

Git clone is a command for downloading existing source code from a remote repository (like Github, for example). In other words, Git clone basically makes an identical copy of the latest version of a project in a repository and saves it to your computer.

2. **Git branch**

Branches are highly important in the git world. By using branches, several developers are able to work in parallel on the same project simultaneously. We can use the git branch command for creating, listing and deleting branches.

3. **Git checkout**

This is also one of the most used Git commands. To work in a branch, first you need to switch to it. We use git checkout mostly for switching from one branch to another. We can also use it for checking out files and commits.

4. **Git status**

The Git status command gives us all the necessary information about the current branch.

5. **Git add**

When we create, modify or delete a file, these changes will happen in our local and won't be included in the next commit (unless we change the configurations).

6. **Git commit**

This is maybe the most-used command of Git. Once we reach a certain point in development, we want to save our changes (maybe after a specific task or issue).

7. **Git push**

After committing your changes, the next thing you want to do is send your changes to the remote server. Git push uploads your commits to the remote repository.

8. **Git pull**

The git pull command is used to get updates from the remote repo. This command is a combination of git fetch and git merge which means that, when we use git pull, it gets the updates from remote repository (git fetch) and immediately applies the latest changes in your local (git merge).

9. **Git revert**

Sometimes we need to undo the changes that we've made. There are various ways to undo our changes locally or remotely (depends on what we need), but we must carefully use these commands to avoid unwanted deletions.

10. **Git merge**

When you've completed development in your branch and everything works fine, the final step is merging the branch with the parent branch (dev or master). This is done with the git merge command.

***Commits are the main structural elements of the Git project timeline. You can think of them as snapshots or checkpoints on the Git project timeline. Commits are created using the git commit command, which takes a snapshot of the current state of the project.***