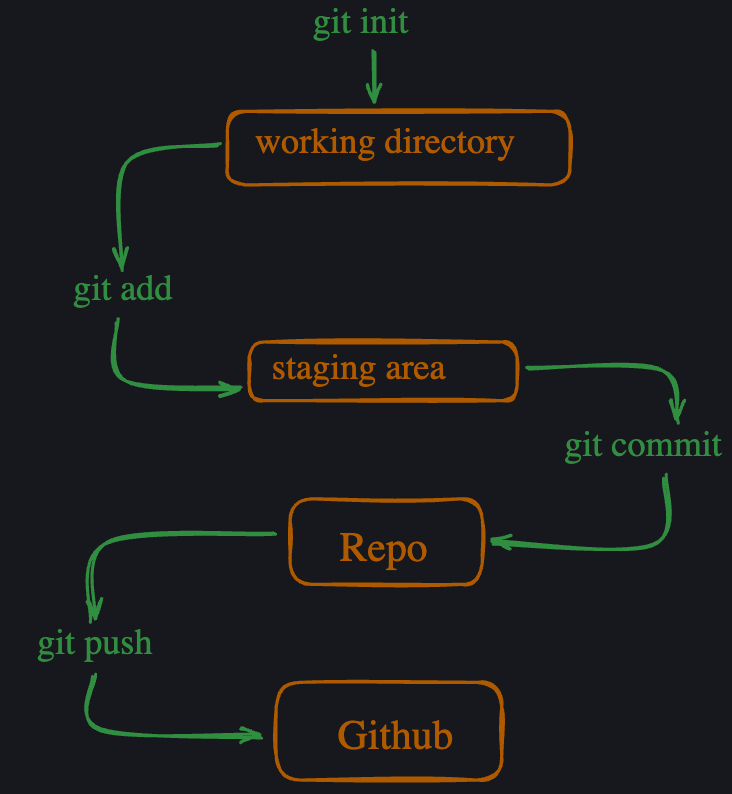
**git init-**

* Convert an existing, unversioned project to a Git repository. Run only once for a project.
* It creates a hidden .git folder which keep track of all files and sub-folders.
* To initialize git project with some other branch names ishan: git init -b ishan

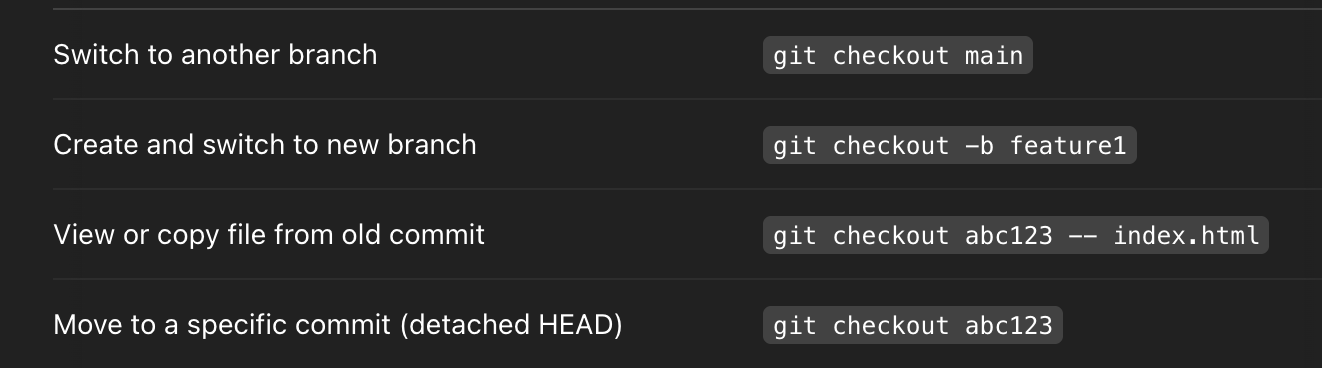


**Head**

We can see all the heads inside .git/refs/heads, it will show names of all branches. Hence every branch head is the tip of that branch

Current head always points to the tip of current branch, it can also be seen in Head file inside .git

**git checkout**



If you commit changes in detached HEAD state, the commits are not saved to any branch. To keep these changes, you must create a new branch using "git checkout -b new-branch"

Q1. What is .gitkeep file?

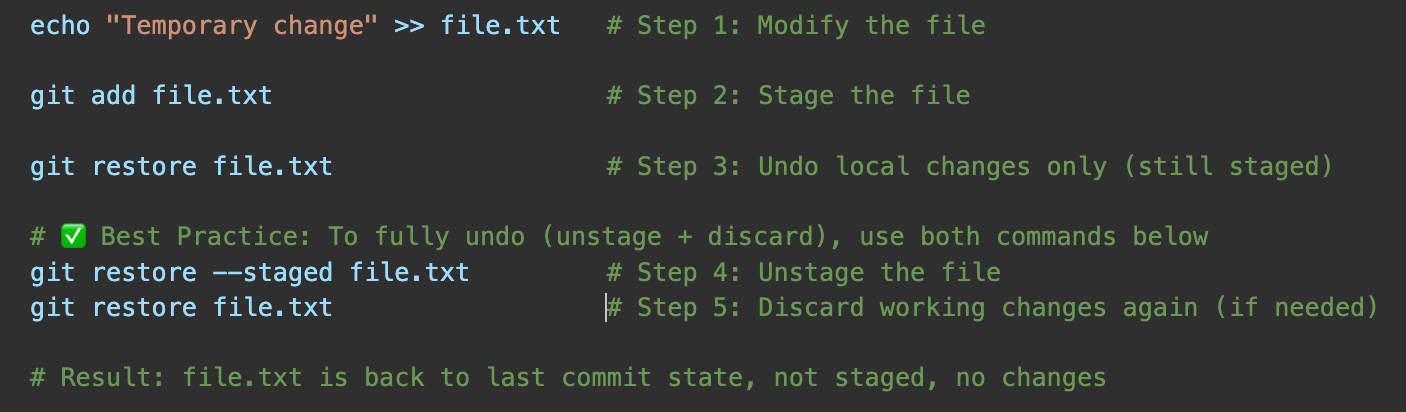
Ans- Git cannot add a completely empty directory. People who want to track empty directories in Git have created the convention of putting files called .gitkeep in these directories.

Q2. What is git restore & git restore --staged v/s git add?

Ans-

git restore : Reset file to the last committed state, hence it undo the changes you made in file, takes back to last committed version in local.

git restore --staged : It is completely opposite of git add. Hence after doing git add, if we want to unstaged a file, we use git restore --staged. It does not undo the changes you made in the file, it only unstages it.



Q3. What happens when we use “git diff” before git add & “git diff --staged” after git add ?

Ans-

git diff : Shows the changes you made in your working directory compared to the last staged. You can see what is modified but not yet staged.

git diff --staged : Shows the staged changes, yet to be committed.

