


Additional Git Commands.

Nitin Singh



Skipping the staging area.

Staging area is the area, where all our files are ready for getting committed.

For the first time, files aren't being tracked. We can't skip the staging area for the 1st time.

If we create a new file, then we have to go through staging area.

`git add .`

(Staging area), or adding the files to staged area.

`git commit -m "first commit"`

(Now files are being tracked)

`git commit -a -m "Commit Message"`

(Skipping the staging area)

What's the difference.

If we make some changes in the tracked file, & write `git status`, then it tells us that some files are modified or that has been added.

but this (`git status`) doesn't tell what exactly the changes we made, in these files

To overcome that we have (`git diff`)

`git diff`

Tells us the diff our files that are there in working directory & the files that were already being staged in the last commit.

following command to see what we've staged that will go into your next commit.

`git diff — staged`

Git Ignore.

There are some files that we want to ignore.
Often times, we don't want git to keep track of some specific autogenerated files like logs or build files.
.DS_Store etc.

To avoid them, we can list files or a pattern for files in `.gitignore` file.

Moving & removing files.

When we move or remove files using git as a tool then it also stages the changes.
These commands stage the changes automatically.
We don't have to stage explicitly.

```
git rm filename.txt
```

```
git mv filename.txt newfilename.txt
```

Unstaging & Unmodifying Files

`git restore —staged filename.txt`

`git checkout —filename.txt`

This command takes the file to the last version / commit.

Note:

This can be a dangerous command, as it discards all our current changes in the working directory.

`git checkout -f` : Rollback to the prev. commit, losing all newly modified files.

Setting Alias.

```
git config --global alias.st status
```

Usage: `git st.`

```
git config --global alias.unstage 'restore --staged--'
```

Usage: `git unstage file.txt.`