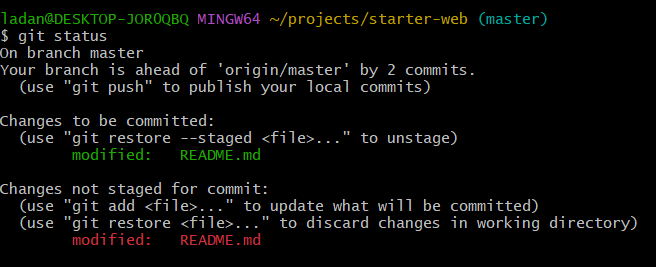
**Comparisons**

**1)Setup**

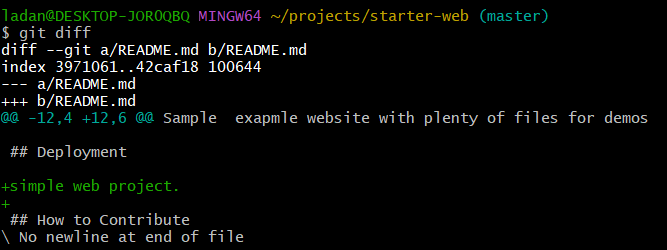
* Setting up README.md file such that the following is in every stage of Git workflow i.e., Working directory (do not add), Staging Area (only add) Repository (both add and commit), Remote (All three add, commit, push).



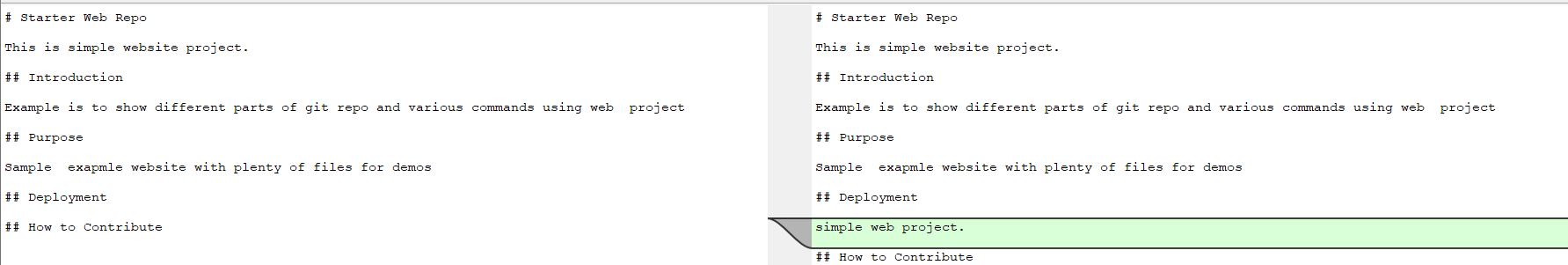
* After doing above things git status will appear like this. And we will have README.md file in GitHub repository too.

**2)Comparing Working Dir to Staging Dir**

* Git status command tells us that file is modified but doesn’t tell what has been modified so we use **$git diff** command for that.

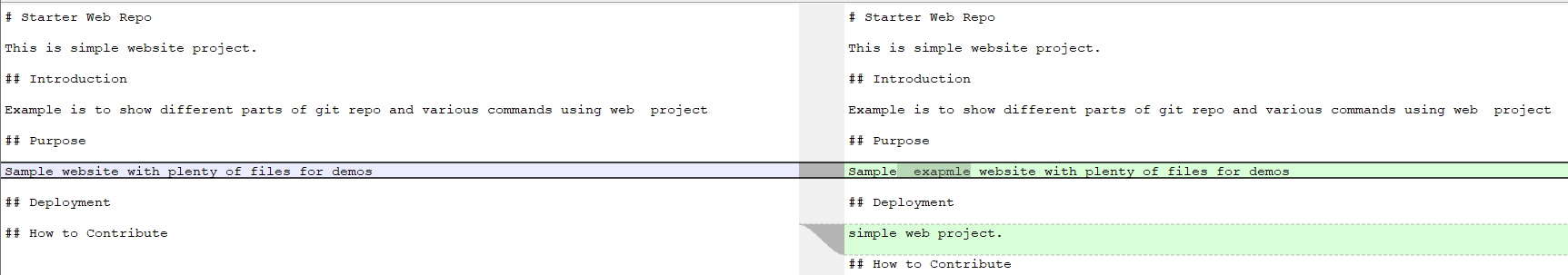


* This command shows the file difference that’s not staged vs what’s staged.
* To see the difference Visually we use command **$git difftool**

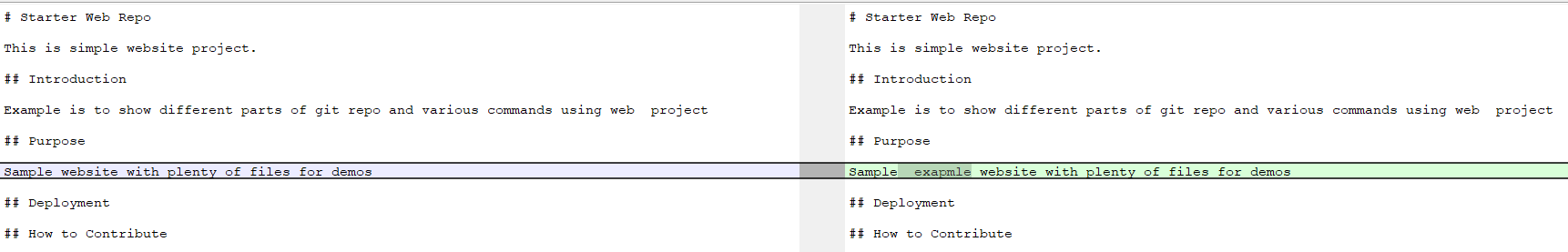


* Here the Blue line shown is the difference between the 2 files. Left one is staging dir and Right one is working dir.

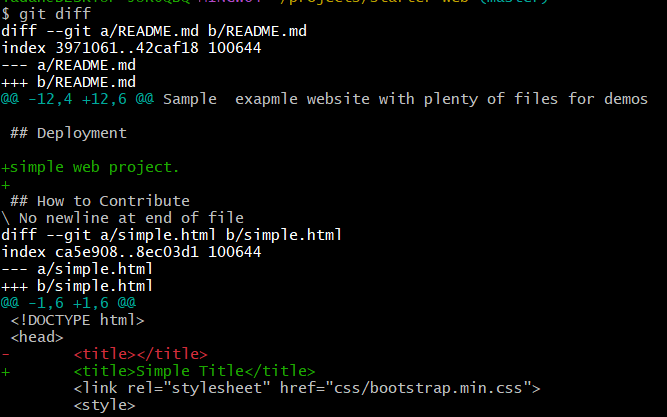
**3)Difference Between Working Dir and Local git Repository.**

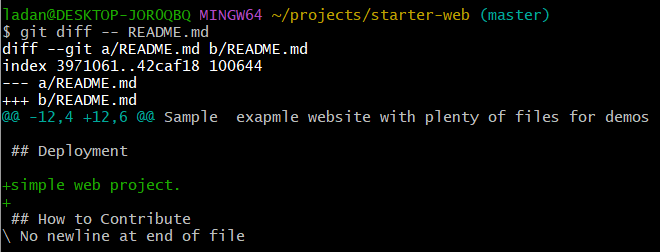
* Command is **$git diff HEAD**
* ****To see the difference Visually we use command **$git difftool HEAD**
* Left one is local git repo and Right one is working dir.

**4)Difference Between Staging Area and Local git Repository.**

* Command is **$git diff --staged HEAD**
* ****To see Visually use command **$git difftool –staged HEAD**
* Left is local git repo Right is staging area.

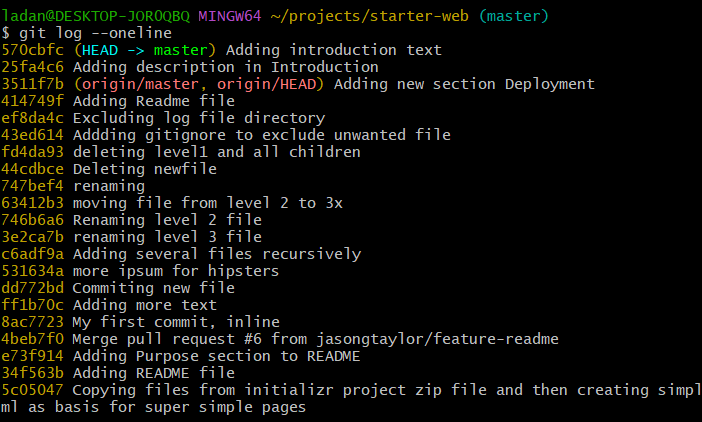
**5)Limiting Comparison to One file.**

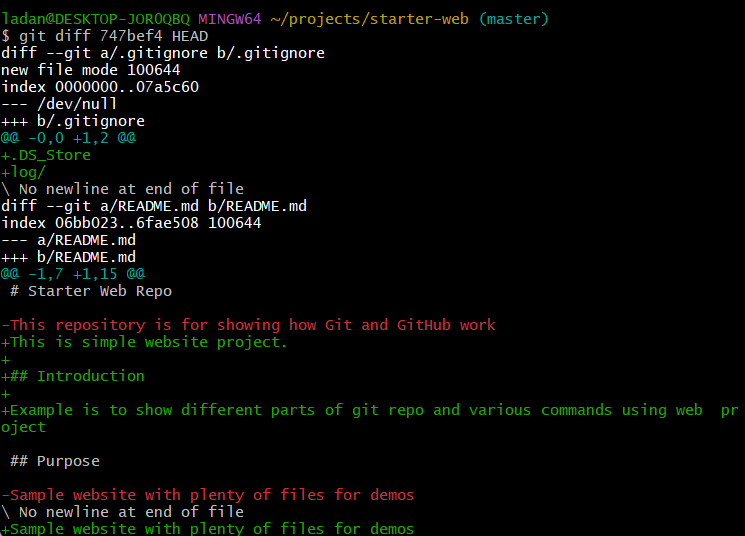
* First make a change in another file(simple.html) other than README.md.
* ****Now on git diff it will show that both files have changes. So now we will limit diff to just REAME.md file.
* Command is **$git diff – README.md**

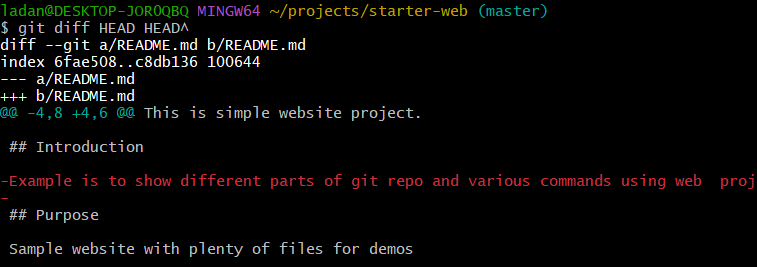
****

* Same goes for visual tool **$git difftool – README.md** this will only show visual of readme file. Otherwise, both visual will be shown one by one if we use $git difftool command.

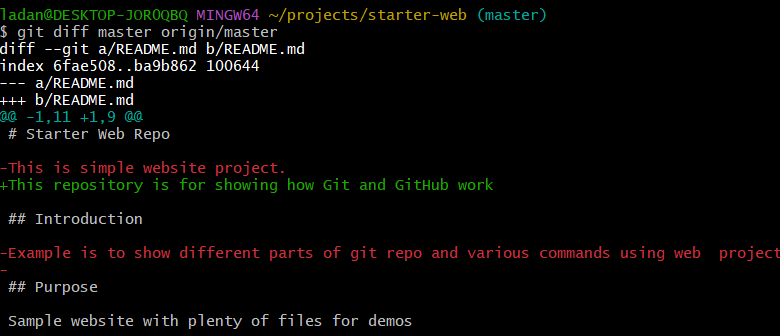
**6)Comparison between Commits.**

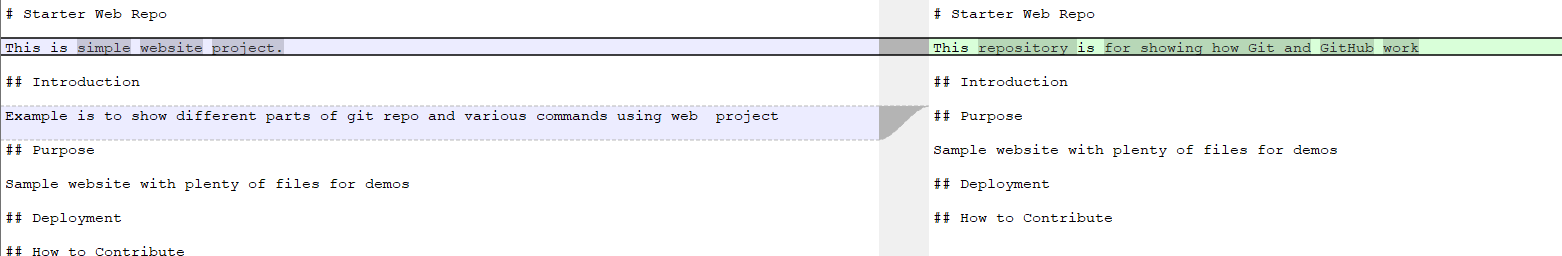
* ****First do **$git log --oneline** to see the history of commits.
* Pick any of the two and find diff **$git diff 747bef4 HEAD**



* We will see all the difference between the HEAD(first) commit and the specified commit with number 747bef4.
* Command **$git diff HEAD HEAD^** compares the head and head-1 commit.
* Same goes for visual **$git difftool HEAD HEAD^**
* Similarly, we can do for the two commits in the log in between.
* In this we see /dev/null that means the file was not existing prior to that commit.

**7)Comparison between Local and Remote master branches.**

* Command **$git diff master origin/master**
* Here the master refers to local repo and origin/master refers to GitHub repo.
* Same for visual **$git difftool master origin/master**



**8)Comparison section cleanup and just pushback to GitHub.**

* Pushing all the commits that aren’t yet.