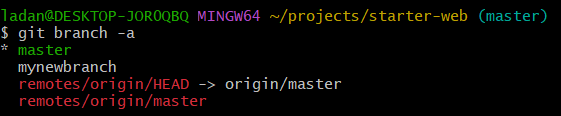
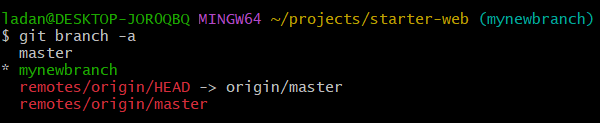
**Branching And Merging**

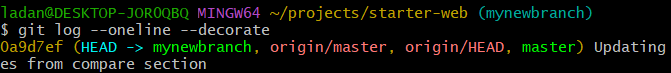
**1)Branching Basics**

* Listing the existing Branches. Command **$git branches -a** this will list both local and remote branches.
* If we want to create a new branch.
* Command **$git branch newbranchname**



* To move to new branch command **$git checkout mynewbranch**

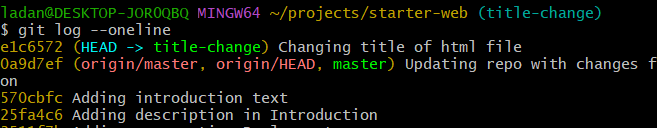


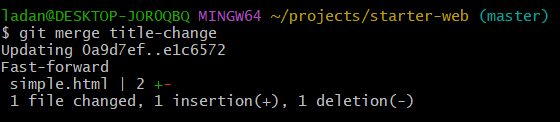
* ****Command **$git log –oneline –decorate**
* HEAD and origin/HEAD are pointers to last commit on their respective branches local and remote, master is the branch that we have been working with all the time, mynewbranch is the branch we just created.
* We can rename the branch too with command **$git branch -m mynewbranch to newbranch**
* We can delete the branch with command **$git branch -d newbranch**
* Here we need to keep in mind that if we are deleting the newbranch we should not be in the newbranch we should be in the another branch.

**2)Happy Path Fast Forward Merges**

* To create and switch to new branch directly we have command

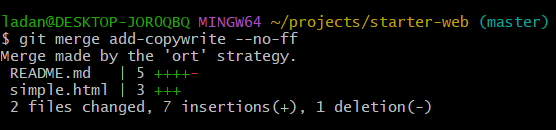
**$git checkout -b title-change**

****

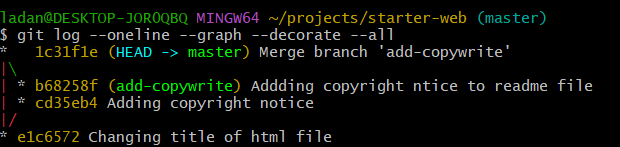
* We currently have committed change in branch title-change if we want to change it to master then command is **$git merge title-change** and this is done while we are In master branch.
* Fast forward is only possible when there are no changes made in the target branch.

**3)Happy Path without Fast Forward Merges**

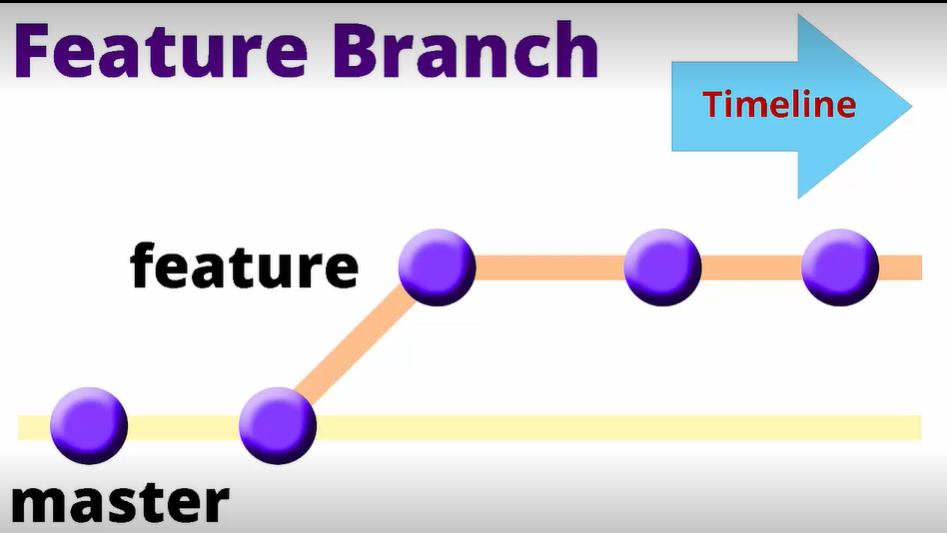
* First make changes in files when we are in copywrite branch and then switch to master branch and then command
* **$git merge add-copyright –no-ff**

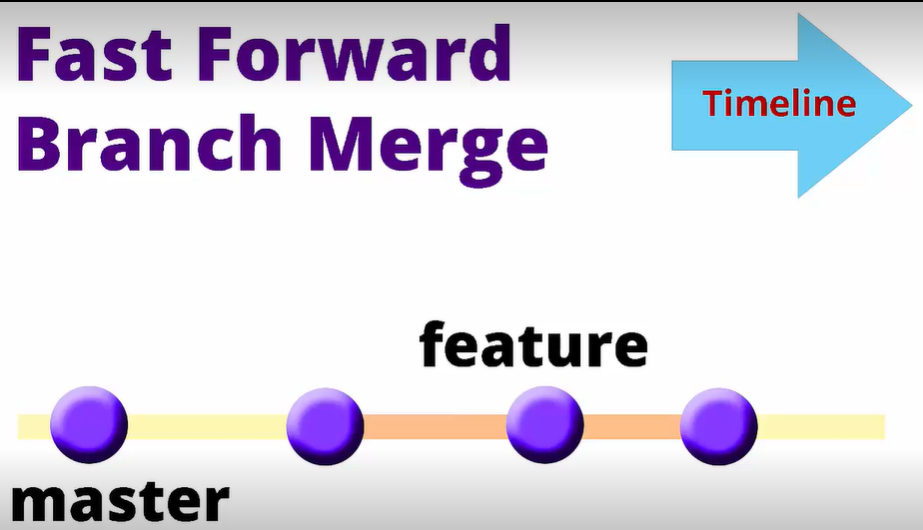
****

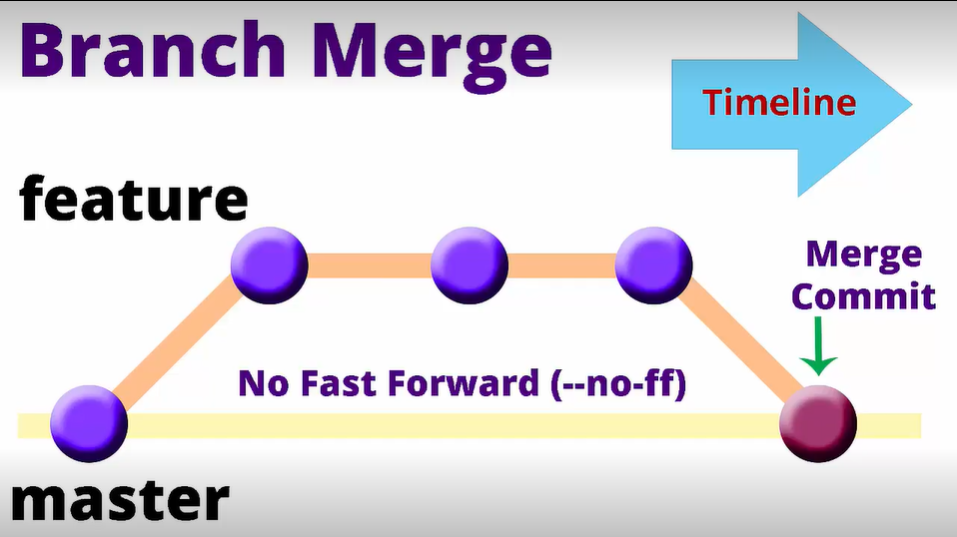
* Now if we do $git log –oneline --graph --decorate –all



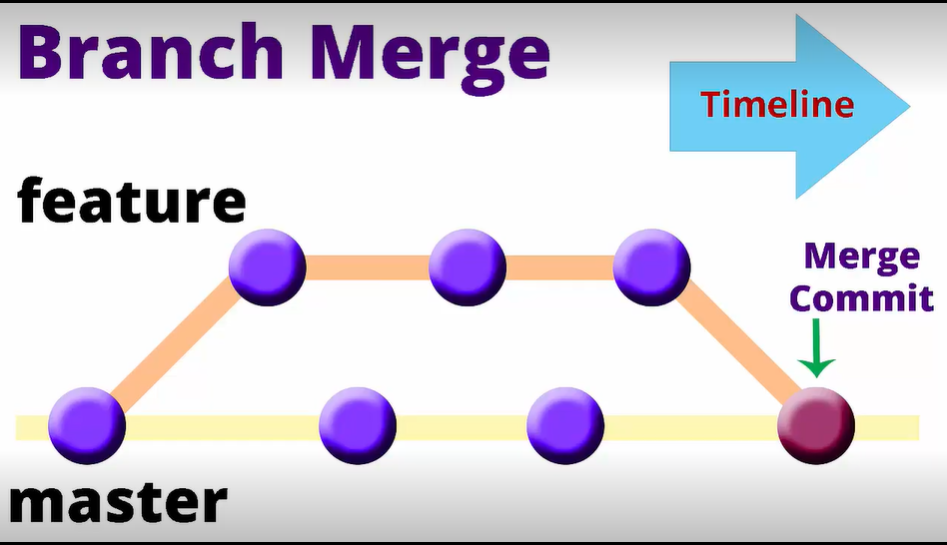
* Above the graph line is being preserved.



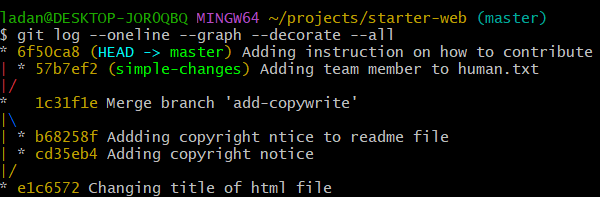


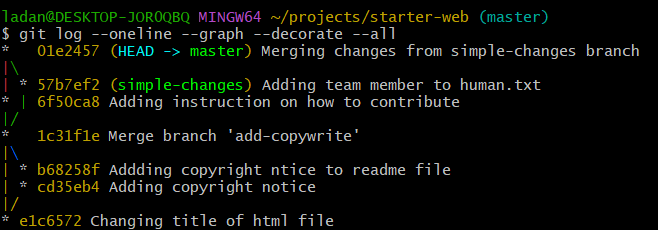


**4)Automatic Merges**



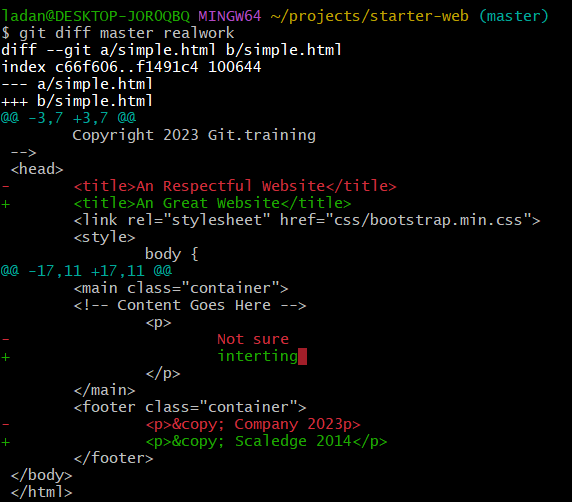
* Create a new branch simple-changes and edit readme.md file and human.txt, then commit -am both with command line and then check the oneline decorate graph.



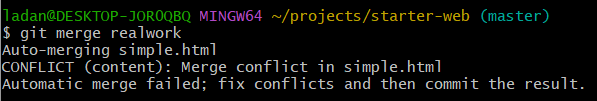
* Automatic merge command
* **$git merge simple-changes -m “Merging changes from simple-changes branch”** and then log oneline command

**5)Merge Conflict and Resolution**

* If we first change a file or edit it in branch name realwork and then we checkout of that branch and go to master branch and check the file it will not be changed**.**
* Now if we change file with different text from master branch then check for diff in both branch



* When we try to merge realwork conflict will be raised.



* After this if we open simple.html in text editor it will show where the conflict is.
* We will use visual tool for resolving the conflict **$git mergetool,** this will show all 3 conflicts and what we need to select depend on us.
* After this if we do git status then it will one Untracked file simple.html.org this is because git will automatically save a copy of original file if anything goes wrong.

**6)Section cleanup and push back to git**

* Clean up the all branches except master and then push into remote repository.