* + - 1. How to use GIT
      2. git remote -v <- link between local and foreign repos
      3. git remote add origin https://github.com/groman-gonzalez/git-tutorial.git
      4. git remore -v
      5. git push origin master
      6. settings in github
      7. - branches
      8. \* branch protection
      9. -> apply to master
      10. git checkout -b dev
      11. git push origin dev
      12. fork
      13. click on an account
      14. ##############################################
      15. # Now for set up from an existing work space #
      16. ##############################################
      17. git clone https://github.com/[name]/git-tutorial.git
      18. git remote -v
      19. git checkout dev
      20. git checkout -b newFeature
      21. code .
      22. # change stuff
      23. git status
      24. git add .
      25. git commit -m""
      26. git push origin newFeature
      27. go to the main repo in github
      28. compare changes from the fork to dev
      29. create pull request
      30. check the changes and approve/denied
      31. #############################################
      32. # Now keep the repo in sync #
      33. #############################################
      34. main repo
      35. grab the link to clone
      36. git remote -v
      37. git remote add upstream https://github.com/[name]/git-tutorial.git
      38. git remote -v
      39. git checkout dev
      40. git pull remote upstream dev
      41. git log
      42. git push origin dev
      43. git checkout -b message
      44. [change the title in the code]
      45. git add .
      46. git commit -m"[new message]"
      47. git push origin message
      48. [in github]
      49. accept pull request
      50. #############################################
      51. # Resolving pull request #
      52. #############################################
      53. git checkout dev
      54. git pull upstream dev
      55. git push origin dev
      56. git checkout message
      57. [notes]
      58. if two branches need to merge together there are
      59. if both are public fancing branches then merge. Will create a commit in time
      60. if one branch is a feature branch then use rebased into dev
      61. [end of notes]
      62. git rebase dev
      63. [fix issue in vscode]
      64. git status
      65. git add .
      66. git rebase --continue
      67. [it will apply the changes]
      68. git push -f origin message
      69. [in github]
      70. check the pull request, it will show the new force push and the message
      71. [in vscode]
      72. git history
      73. [update dev branch]
      74. git checkout dev
      75. git pull upstream dev
      76. git pull origin dev

cd my\_git\_repos/1810\_oct22

* + - 1. o git checkout master
      2. o git pull
      3. o git checkout Thien\_Le
      4. git add Homework1Foloer
  + git commit -m \_“Firstname Lastname a\_d\_d\_ \_R\_E\_A\_D\_M\_E\_.\_m\_d\_” \_**\*\*commit changes** 
    - 1. o git push

Git init //start the repos in local area, beginning

Git add chapter.txt //staging area

Git log

Git status

Git commit - "My first commit"

Git diff chapter3.txt

Git checkout chapter3.txt //reverse save point

Git remote add origin url

Git push -u origin master //push to remote repository

Git clone url //clone from remote

Git ignore

Git rm --catch -r

Git branch alien-plot

Git checkout alien-plot

Git merge alien-plot

:q! To exit vmi editor