Git Workbook

# Basic vi Commands

In case, you are not very familiar with Linux vi editor you may find the following commands useful while editing files in vi editor.

Alternatively, you may use another very popular editor nano which is not all that command driven. However, nano usually is not shipped with most of the common Linux distributions - so, you may have to install it using the following commands:

* **For Debian systems** (Ubuntu) - sudo apt-get install nano
* **For RedHat systems** (CentOS/Fedora/RHEL/Oracle Linux) - sudo yum install nano

|  |  |  |
| --- | --- | --- |
|  | **Command** | **Action** |
| 1 | vi <file name> | Opens file in escape mode (view mode) |
| 2 | e  a  i | Edit mode - to write in the file  Append mode - to write in the file  Insert mode - to write in file |
| 3 | **Shift**+a | Append at end of line |
| 4 | **Esc** key | Bring back the editor in escape mode - once editing is done, press Esc key to go back to escape mode and proceed to save the file |
| 5 | :w | Write the changes (escape mode) |
| 6 | :wq | Write and quit - saves the changes (works in escape mode) |
| 7 | :q | Quit without saving |
| 8 | :wq! | Write and quit in case of read only files (per force) e.g. /etc/hosts |
| 9 | :q! | Quit in case of read only files (per force) e.g. /etc/hosts |
| 10 | x | Deletes a character (escape mode) |
| 11 | dd | Deletes a line (escape mode) |
| 12 | yy followed by p | yanks the line at current position of the cursor and then paste after the line when p is typed (escape mode) |
| 13 | r | Replace the character (escape mode) |
| 14 | **Shift**+r | Keep on replacing the characters (escape mode) |
| 15 | :se nu | Displays the line numbers (escape mode) |
| 16 | :se nonu | Hide the line numbers (escape mode) |
| 17 | h j k l or arrows | left down up right movement of cursor in a line (escape mode) |
| 18 | :$ | Bring cursor at the end of file (escape mode) |
| 19 | :$<line num> | Bring cursor at the line number specified (e.g. :$0 will bring cursor to the first line) - escape mode |
| 20 | /<string> | Search forward the specified string (escape mode) |
| 21 | ?<string> | Search backward the specified string (escape mode) |
| 22 | :r <source filename> | Read from the source file and insert the content after current line (escape mode) |
| 23 | :w <target filename> | Write the current content to the target file (new file) - escape mode |
| 24 | :12, 35w <target file> | write the contents of the lines numbered 12 through 35 to the target file (new file) - escape mode |

# Git Handy Commands

Refer the Git Cheat Sheet (D02-GitCheatSheet.pdf) from GitHub Education. This PDF is available in the shared drive location - <https://drive.google.com/drive/folders/1MaHnDheKjic5R8jbsNeSj2d9PN0lYMzZ?usp=sharing>

# Assignments

1. Install Git in your local machine
2. Initiate a git repository in your local machine (preferably in C:\ or D:\ drive)
3. Add username and e-mail using git config command - set them at repository level (or set them as global)
4. Create a file hello.txt and add few lines there. Stage the file and commit.
5. Add few more lines in the same file and commit again. Check the history of changes with git log
6. Create your GitHub account and create a remote repository - say ‘myrepos’.
7. Clone the remote repository ‘myrepos’ to your local system - preferably in C:\ or D:\ drive with git clone command.
8. Repeat the steps 3 to 5 in the cloned repos. Commit your changes and push to remote repos. Check in both local and remote.
9. Create a branch in remote repository - name it as dev.
10. Checkout to dev - make changes in hello.txt and then commit and push to remote dev branch.
11. Merge changes from dev to master.
12. Make a tag ‘v1.0’ on the branch and thereby freeze it.
13. Add your friend to your repository as collaborator and let him / her to push changes. Handle the pull request and merge his / her work to master.
14. Design a git workflow for a git managed project that works on multiple features in parallel and decide to release product as when things are ready. What standard design you will adopt in your SCM strategy design?
15. Practice the above exercises in a Linux VM - get yourself familiar with the vi editor.