**Version Control System**

* **There are 2 tools for version control**
  + **SVN 🡪 Sub version**
  + **Git**
* **SVN**
  + Works only offline
  + Make a copy of source code on each modification
  + Need more space

**Git**

* + Version Controlling Tool
  + Works on both offline and on line
  + Have one source and maintains only the modification done
  + Version control, also known as source control, is the practice of tracking and managing changes to software code
  + Version control software keeps track of every modification to the code in a special kind of database.
  + If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.
  + Git is free and open source

**1. Repository Creation**

**init 🡪**This command is used to create a local repository.

**Syntax**

**$ git init Demo**

**2. config**

* This command configures the user.
* The Git config command is the first and necessary command used on the Git command line.
* This command sets the author name and email address to be used with your commits.
* Git config is also used in other scenarios.

**Syntax**

**$ git config --global user.name "username”**

**or**

**$ git config --global user.email "mailid"**

**3. add**

This command is used to add one or more files to staging (Index) area.

**Syntax**

To add one file

### **$ git add Filename**

### To add more than one file

### **$ git add\***

### **4. commit**

Commit command is used in two scenarios. They are as follows.

**Git commit -m**

* This command changes the head.
* It records or snapshots the file permanently in the version history with a message.

**Syntax**

**$ git commit -m " Commit Message"**

**Git commit -a**

* This command commits any files added in the repository with git add and also commits any files you've changed since then.

**Syntax**

**$ git commit -a**

### **5. status**

* The status command is used to display the state of the working directory and the staging area.
* It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git.
* It does not show you any information about the committed project history.
* For this, you need to use the git log.
* It also lists the files that you've changed and those you still need to add or commit.

**Syntax**

**$ git status**

### **6. push Command**

* It is used to upload local repository content to a remote repository.
* Pushing is an act of transfer commits from your local repository to a remote repo.
* It's the complement to git fetch, but whereas fetching imports commits to local branches on comparatively pushing exports commits to remote branches.
* Remote branches are configured by using the git remote command.
* Pushing is capable of overwriting changes, and caution should be taken when pushing.

**Syntax**

**$ git push [variable name] master**

**Git push -all**

This command pushes all the branches to the server repository.

**Syntax**

**$ git push --all**

### **7. pull**

* Pull command is used to receive data from GitHub.
* It fetches and merges changes on the remote server to your working directory.

**Syntax**

**$ git pull URL**

### **8. Branch Command**

This command lists all the branches available in the repository.

**Syntax**

**$ git branch**

### **9. Merge Command**

This command is used to merge the specified branchs history into the current branch.

**Syntax**

**$ git merge BranchName**

### **10. log Command**

This command is used to check the commit history.

**Syntax**

**$ git log**

### **11. Git clone command**

This command is used to make a copy of a repository from an existing URL.

If I want a local copy of my repository from GitHub, this command allows creating a local copy of that repository on your local directory from the repository URL.

**Syntax**

**$ git clone URL**