git

basic workflow

```
# initialize an empty repository
> git init

# get the current status of every file present in working directory
> git status

# get short status
> git status -s

# note: git status -s command returns a status with two characters
# 1st character: shows the status of file with respect to the staging area
# 2nd character: shows the status of file with respect to the working area

# ??: untracked file (the repository does not recognize the file or does
not have any version of this file created yet)
# A : the changes are present in the staging area and will be added to the
repository after committing
#  M: the changes are present in the working directory and the file is
modified in the working directory
# M : the changes are moved to the staging area

# add the file/files to staging area
> git add <file name>
> git add .

# create a version of all the files present in staging area
> git commit -m <message>

# get the list of all commits
> git log

# get the list of logs in one line with color and graph
> git log --oneline --graph --color

# get the difference between the latest version and the previous version
> git diff

# get the last version from repository and replace it with current version
# note: even if the file is deleted from the working directory, you can
still bring it (last version of it) back from the repository
> git checkout <file name>

# soft reset:
# - move all the changes from staging area to the working directory
# - no changes will be lost in the process
```

```
> git reset

# hard reset:
# - remove all the changes from staging area or working directory
# - get last version of all the files and replace with current version
(irrespective of their location)
# - note: please execute this command on your own risk
> git reset --hard

# remove all the metadata or repository
# note: this command will remove all the history
> rm -rf .git
```

shared repository

```
# types
# - local
#   - present on local machine
# - shared
#   - present on the server
#   - also known as remote repository
#   - to create a shared repository
#     - create a project or repository on shared repository server
#     - connect the local repository to the remote one
#   - e.g. GitHub, GitLab, BitBucket

# get the remote repository url
> git remote -v

# add the remote repository
# > git remote add <server alias> <repository url>
> git remote add origin <repository>

# send all the local changes to the remote repository
> git push <alias> <branch name>

# pull all the changes from remote to local repository
> git pull <alias> <branch name>

# remove the remote repository from local one
> git remote remove <alias>
```