**Q1. How to check whether git is available on your system?**

**Answer:** To check if Git is available on your system, you can use the command line interface (CLI) and run the following command:

git --version

If Git is installed on your system, the command will output the version number of Git, for example:

git version 2.30.2

If Git is not installed on your system, the command will output an error message indicating that the command is not recognized. In this case, you will need to install Git on your system before you can use it.

**Q2. how to initialize new git repository**?

**Answer:** To initialize a new Git repository, you need to follow these steps:

Open the terminal and navigate to the directory where you want to create the repository.

Run the following command:

git init

**Q3. how to tell git about your name and email**?

**Answer:** To tell Git about your name and email, you need to run the following commands:

To set your name:

git config --global user.name "Your Name"

Replace "Your Name" with your actual name.

To set your email:

git config --global user.email "your.email@example.com"

Replace "your.email@example.com" with your actual email address.

These configurations will be saved globally, so you only need to set your name and email once on your computer. The --global option sets these values for your current user account. If you want to set these values for a specific repository, run the commands without the --global option in the repository directory.

**Q4. how to add a file to the staging area?**

**Answer:** To add a file to the staging area in Git, you can use the git add command followed by the file name. For example, to add a file named file.txt to the staging area, run the following command:

git add file.txt

You can also use git add. to add all modified and untracked files in the current directory and its subdirectories to the staging area.

Once the files are in the staging area, they are ready to be committed to the repository with the git commit command. This allows you to stage and commit files in small, logical chunks rather than committing everything in one go.

**Q5. how to remove a file from staging area?**

**Answer:** To remove a file from the staging area in Git, you can use the git reset command followed by the file name. For example, to remove a file named file.txt from the staging area, run the following command:

git reset file.txt

Alternatively, you can use the git rm command with the --cached option to remove a file from the staging area while keeping the file in your file system:

git rm --cached file.txt

After removing the file from the staging area, it will not be included in the next commit. Note that this does not delete the file from your file system, it just removes it from Git's tracking.

**Q6. how to make a commit?**

**Answer:** To make a commit in Git, you need to follow these steps:

Add the files you want to include in the commit to the staging area using the git add command.

Run the following command to commit the changes:

git commit -m "Your commit message"

Replace "Your commit message" with a meaningful message that describes the changes you are committing. The commit message should be brief, but descriptive, and written in the present tense. For example, "Add new feature X" or "Fix bug Y".

**Q7**. **how to send changes to remote repository?**

**Answer:** To send changes to a remote repository in Git, you need to follow these steps:

add the remote repository to your local repository by running the following command:

git remote add origin <repository-url>

To send changes to a remote repository in Git, you need to follow these steps:

add the remote repository to your local repository by running the following command:

git remote add origin <repository-url>

Replace <repository-url> with the URL of the remote repository you want to connect to.

Push the local changes to the remote repository by running the following command:

git push -u origin master

**Q8. what is difference between clone and pull?**

**Answer:** The git clone and git pull commands are used to retrieve code from a Git repository, but they serve different purposes:

git clone: This command is used to create a local copy of a remote repository. When you run git clone, Git retrieves the entire repository, including all the branches and history, and creates a new local repository on your computer. You can then use git checkout to switch between branches and make changes to the code.

git pull: This command is used to update an existing local repository with changes from a remote repository. When you run git pull, Git retrieves any changes from the remote repository and merges them into your local repository. This is useful when you want to ensure that your local repository is up-to-date with the latest changes from the remote repository.