1.What is Git and why is it important?

Git is a distributed version control system used for tracking changes in files, typically source code, during software development.

The main use of this tool is to manage and track changes effectively, ensuring collaboration and maintaining a history of modifications.

Key Benefits of Git:

Version Control: Git keeps track of changes made to files over time, allowing you to revert to previous versions if needed. This is particularly useful when you are unsure if new changes will work as expected.

Easy Reversion: If something goes wrong after adding new code, Git allows you to easily revert back to the previous state. This can be a lifesaver, especially when dealing with complex changes.

Backup and Restore: In case a file is deleted or corrupted, Git enables you to restore it from the repository. This ensures that you have a backup of your code at all times.

Collaboration: When multiple developers are working on the same project, Git helps manage simultaneous changes without conflicts. It maintains a commit history that records who made what changes and when.

Commit History: Git maintains a detailed history of changes, which is important for understanding the evolution of the project and for debugging purposes. It also helps in code reviews and audits.

Branching and Merging: Git allows developers to create branches for new features or bug fixes, work on them independently, and then merge them back into the main codebase once they are ready and tested.

Overall, Git boosts productivity, improves code quality, and helps developers work together smoothly, making it a crucial tool in modern software development.

2.What is difference Between Main Branch and Master Branch?

In Git, the terms "main" and "master" refer to the default branches that are created when a new repository is initialized.

"Master" was the traditional name, but in recent years, there has been a shift towards using "main" as the default to be more respectful to everyone. Both branches function the same way.

3.Can you explain the difference between Git and GitHub?

Git: Git is a distributed version control system used to track changes in source code during software development. It allows developers to manage and keep a history of changes, enabling collaboration and ensuring code integrity.

GitHub: GitHub is a web-based platform that hosts Git repositories. It provides a user-friendly interface and additional features such as issue tracking, project management, and social coding (e.g., forking and pull requests).

GitHub enhances Git's capabilities by making it easier to share code and collaborate with others online.

4.How do you create a new repository on GitHub?

1.Go to the GitHub website and log in to your account.

2.Click the "+" icon in the top-right corner and select "New repository" from the dropdown menu.

3.Fill in the repository name, description, and choose visibility (public or private).

4.Optionally, initialize the repository with a README file, add a .gitignore file, and choose a license.

5.Click "Create repository."

Your new repository is now created and ready to use on GitHub!

5.What is difference between local & remote repository? How to connect local to remote?

Remote Repository is a repository which is hosted on remote servers i.e github servers.

Local repository - it is a copy of the remote repository which is present locally on your system.

A local repository is only on your computer, while a remote one is accessible over the internet, allowing multiple people to collaborate from different locations.

To connect them:

1.Set up a remote repository on a platform like GitHub.

2.In your local project directory, use git remote add origin to link your local repo to the remote one.

3.Push your changes with git push.

4.Now your local and remote repositories are connected!