**explain new branch and master branch and write a step to create a new branch and merge with master branch**

**Understanding Branches in Git**

**Master Branch (Main Branch)**

* The master branch (often named main in newer versions of Git) is the default branch that is created when you initialize a new Git repository.
* It is typically considered the main line of development, where the most stable version of the project code resides.
* All other branches are usually created off the master branch and changes are merged back into the master branch once they are stable and tested.

**New Branch**

* A new branch is a separate line of development created off an existing branch (usually the master branch).
* It allows you to work on a new feature, bug fix, or experiment without affecting the main codebase.
* Changes in the new branch can be worked on independently and later merged into the master branch.

**Steps to Create a New Branch and Merge It with the Master Branch**

**Step 1: Create a New Branch**

1. **List all branches to see the current branch:**

git branch

1. **Create a new branch:**

git branch new-branch

Here, new-branch is the name of the new branch. You can name it according to the feature or task you are working on.

1. **Switch to the new branch:**

git checkout new-branch

Alternatively, you can combine the creation and switch into a single command:

git checkout -b new-branch

**Step 2: Work on the New Branch**

* Make changes to your files in the working directory.
* Stage the changes:

git add .

* Commit the changes:

git commit -m "Describe the changes made"

**Step 3: Merge the New Branch with the Master Branch**

1. **Switch back to the master branch:**

git checkout master

1. **Merge the new branch into the master branch:**

git merge new-branch

**Step 4: Resolve Any Merge Conflicts (If Any)**

* If there are merge conflicts, Git will notify you. Open the conflicting files and resolve the conflicts.
* After resolving conflicts, stage the resolved files:

git add filename

* Complete the merge by committing the changes:

git commit

**Step 5: Clean Up (Optional)**

* Once the changes are merged and if you no longer need the new branch, you can delete it:

git branch -d new-branch

**Example Workflow**

# List all branches

git branch

# Create and switch to a new branch

git checkout -b feature-xyz

# Make changes, add them, and commit

git add .

git commit -m "Implemented feature XYZ"

# Switch back to the master branch

git checkout master

# Merge the new branch into the master branch

git merge feature-xyz

# Delete the new branch if no longer needed

git branch -d feature-xyz

This workflow allows you to isolate your work on a new branch and then integrate it back into the main branch safely.

Top of Form

Bottom of Form