**# merge and rebase:**

# Both are used for moving changes from one branch to other branch

# merge: it will create extra commit for us

**1.Merge with "ort" strategy**: The "ort" (Ostensibly Recursive Traversal)

    merge strategy is a recursive strategy that attempts to identify the best common ancestor

    between the branches being merged and creates a new merge commit that combines the changes from both branches.

    This strategy is used when a simple fast-forward merge is not possible due to divergent changes on the branches.

**2.fast forword strategy**

    "Fast-forward" is a term used in Git to describe a simple merge scenario where the branch being merged has all commits

     that are reachable from the current branch. In this case,

     Git can directly move the pointer of the current branch forward to the tip of the other branch,

    incorporating all the commits in a linear history.

        git merge <branch-name>

**# Rebase**: it won't create any extra commit

        git rebase <branch-name>

i didnt find any difference in git merge and git rebase when i tried first merge and then rebase extra commit is not found

the commit that made in first hotfix master and after using merge cmd in master is same

this is because there no new changes in the master after creating new branch

**impact on Collaboration:**

    Merging is safe for shared branches as it preserves history.

    Rebasing rewrites history, so it should be used cautiously with branches shared with others to avoid complications.

-------------------------------------------------

# Git can handle most merges on its own with automatic merging features.

        A conflict arises when two separate branches have made edits to the same line in a file,

        or when a file has been deleted in one branch but edited in the other.

        Conflicts will most likely happen when working in a team environment

**1. Git Fetch**

**git fetch is used to download changes from a remote repository into your local repository, without integrating them into your current working branch.**

**How it Works:**

* **git fetch retrieves the latest commits, branches, and tags from the remote repository.**
* **These updates are stored in your local repository but are not merged into your current branch.**
* **It’s a safe way to check for changes without affecting your working directory.**

**Common git fetch Usage:**

**bash**

**Copy code**

**git fetch origin**

* **This command fetches changes from the origin remote repository (typically the main remote) without merging them.**

**When to Use git fetch:**

* **Before reviewing incoming changes: Allows you to inspect what’s new in the remote repository before merging.**
* **To update remote-tracking branches: Useful for seeing remote changes without changing your working branch.**

**2. Git Pull**

**git pull is used to update your local branch with changes from the remote repository and directly integrate them with your current branch.**

**How it Works:**

* **git pull is essentially a combination of two commands: git fetch + git merge.**
* **It downloads changes from the remote repository and then merges them into your current branch.**

**Common git pull Usage:**

**bash**

**Copy code**

**git pull origin main**

* **This command fetches changes from the main branch in the origin remote repository and merges them into your current branch.**

**git pull with Rebase:**

**You can also use git pull with the --rebase option to apply your local changes on top of the remote changes, which maintains a linear commit history:**

**bash**

**Copy code**

**git pull --rebase origin main**

**When to Use git pull:**

* **When you want to update and merge in one step: git pull is faster if you’re ready to integrate changes immediately.**
* **To keep your branch up-to-date: It’s commonly used when you want your branch to stay in sync with the remote branch.**

**Key Differences**

| **Feature** | **git fetch** | **git pull** |
| --- | --- | --- |
| **Action** | **Downloads changes without merging** | **Downloads changes and merges them** |
| **Risk Level** | **Safe (doesn’t affect working directory)** | **May cause conflicts in the working directory** |
| **Usage Scenario** | **Reviewing updates without merging** | **Updating and merging changes automatically** |