**Merging:**

The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

Depending on the merge options enabled for your repository, you can

1. **Create a merge commit**.
2. **Squash and merge** button.
3. **Rebase and merge**

If you don't want the changes in a topic branch to be merged to the upstream branch, you can “[close the pull request](https://help.github.com/articles/closing-a-pull-request)” without merging.

**Rebasing**:

If someone merged new code while your patch is still in review, git might not be able to figure out how to apply your patch on top of the new code. and the GitHub UI will say "This branch has conflicts that must be resolved". This is when you need a rebase.

P -> Q -> R (<= remote servo/servo)

\

.--> X -> Y (<= your local branch)

P -> Q -> R ---> S

\

.--> X -> Y

P -> Q -> R -> S

\

.--> X -> Y

Commands for rebasing:

git remote –v

create a new remote called 'upstream'

git remote add upstream

download the latest code

git fetch upstream.

git rebase upstream/master.

**Squashing**:

 If your change consists of multiple commits, we want to squash them into a single commit.

E.g.  if your change consists of two commits X and Y, we want to squash them into a single commit Z.

P -> Q -> R

\

.--> X -> Y

P -> Q -> R

\

.--> Z