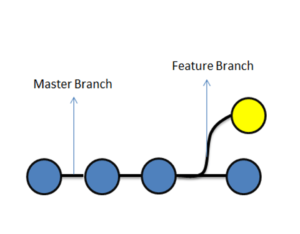
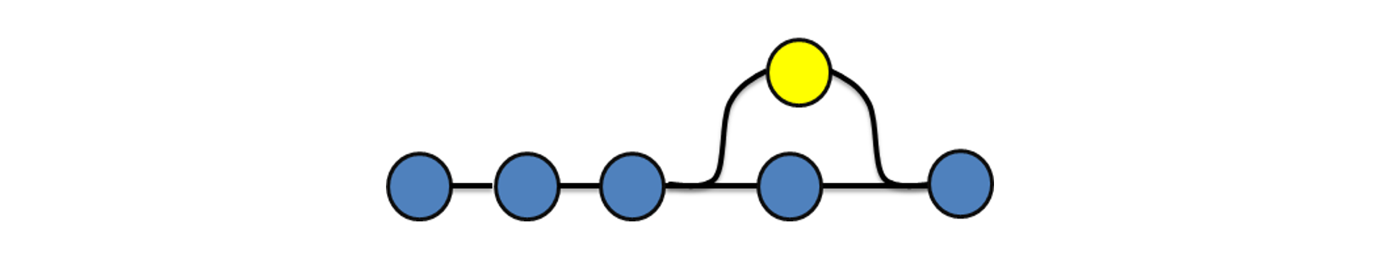
<https://www.edureka.co/blog/git-rebase-vs-merge/amp/>



**Master branch and the Feature branch**

Once you are done with the changes, we simply ‘merge’ the changes of the ‘feature’ branch to the ‘master branch’. And now, the changes which were made in the feature branch will exist on the master branch as well.



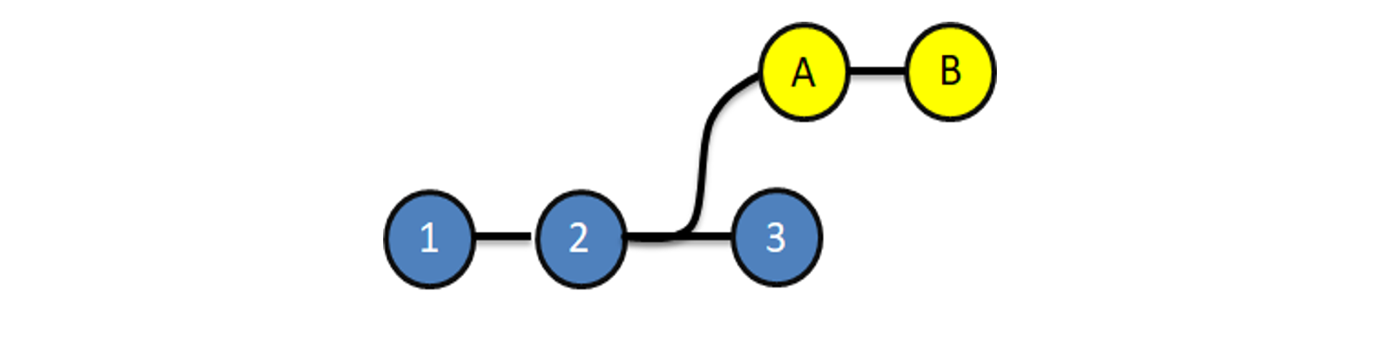
**Merging of feature branch with the master branch**

**What is Merging?**

In general, merging means combining something into a single entity.

Git Merge is a technique that is used to include the changes from one branch to the other branch.

So let us take an example from the diagram below, which shows the status before and after merging of two branches, feature and master branch. Blue commits are on the master branch, and yellow commits are on the feature branch.



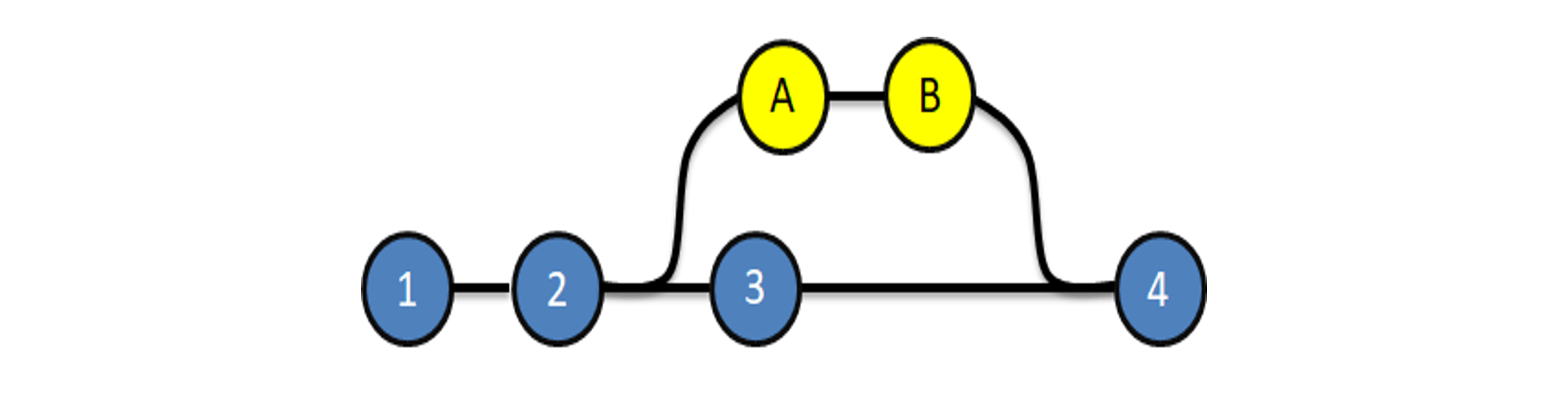
**Before merging**

before merging, we can see that there are some commits on the master branch. We created a feature branch, after commit ‘2’ on master. Later, we did some changes in the feature branch in the form of commits A and B. We also did some changes on the master branch, in the form of commit 3.

In the current scenario, the master branch has the code of Commit 1,2 and 3, but it does not have changes of Commit A and B from the feature branch.

Similarly, since the feature branch was branched out from master on commit ‘2’, it has code changes from Commit 1,2, A, and B, but it does not have changes from Commit 3 on master.

Below, we have merged the feature branch on master, let us understand what happened.



**After merging**

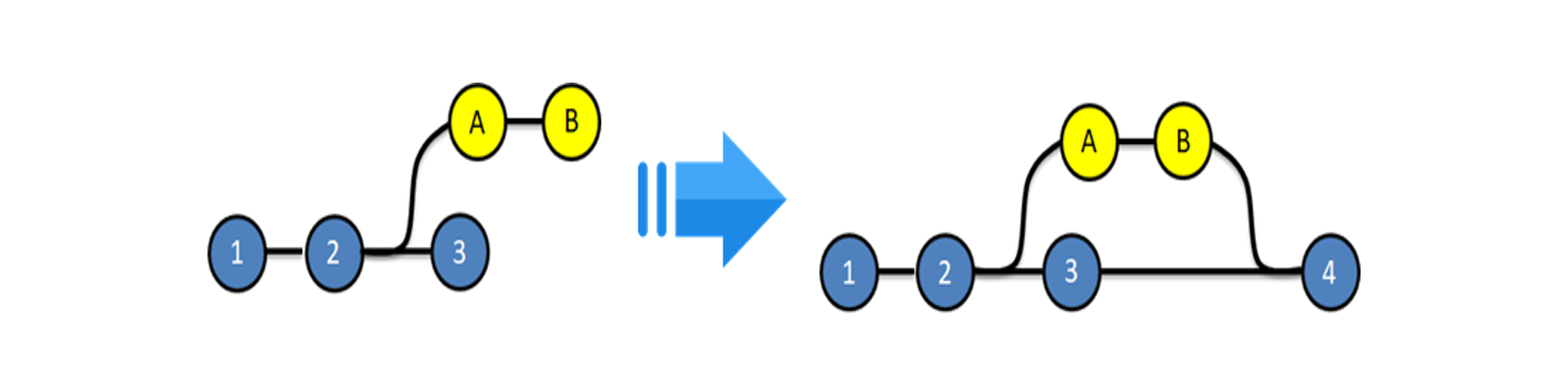
We merged the feature branch on master, resulting in commit 4. Commit 4 on the master, has all the changes of the code i.e Commit 1,2, 3, A, and B. Now, that we understand merging, let’s understand the different types of merging that we can perform in git.

In Git, merging is of two types:

* Git Merge
* Git Rebase

**Git Merge**  
Git merge is one of the merging techniques in git, in which the logs of commits on branches are intact.

Let us take an example if we have a project with 3 commits on the master branch as commit 1,2,3 and feature branch commits as commit A and B. If we perform a git merge operation then commits A and B will be merged as commit 4 onto the master branch. This is depicted below



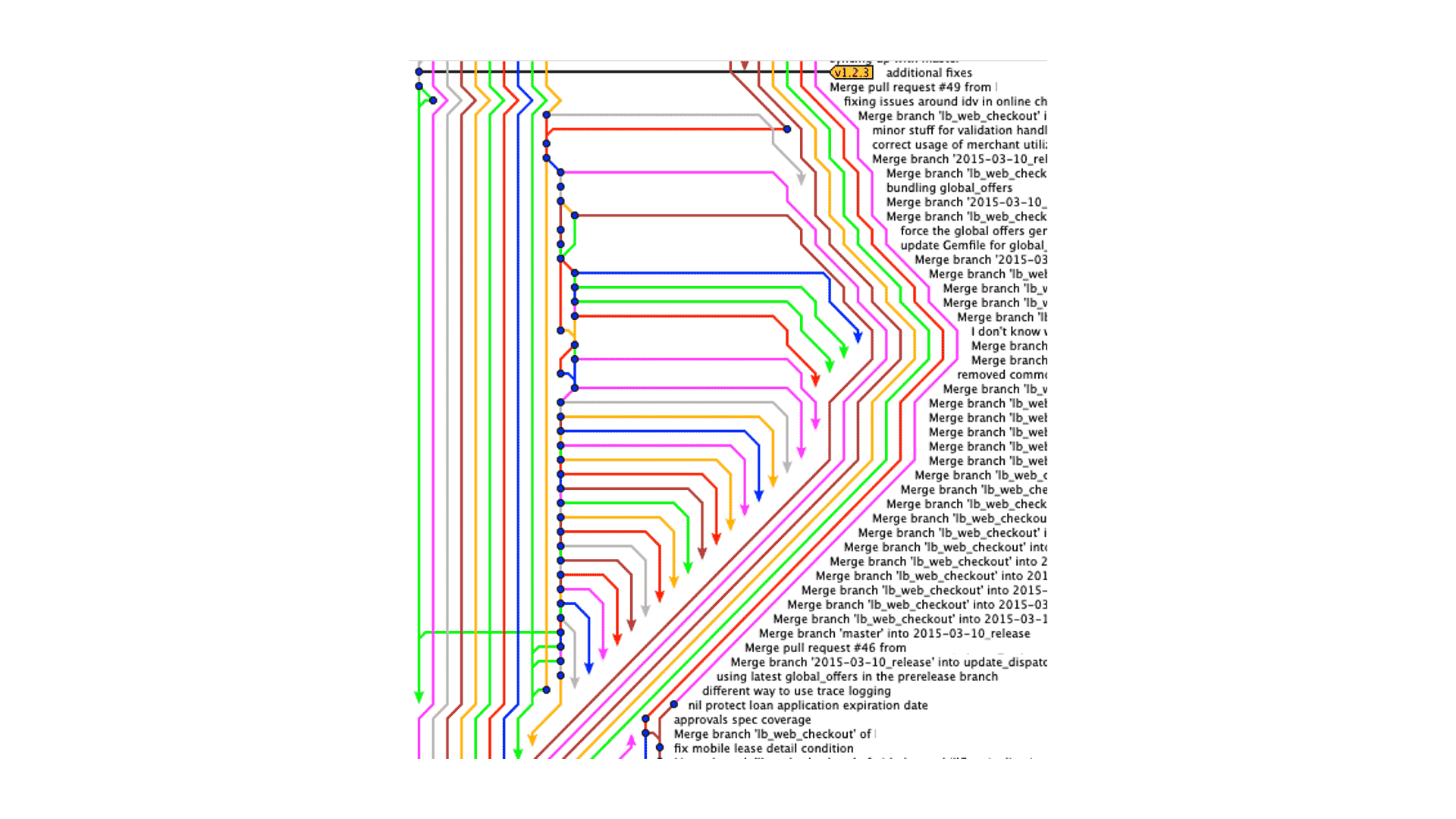
**Before and after Git Merge**

**Advantages:**

* The logs are very exhaustive and can help in understanding the complete history of how and when each merge happened
* It is easy to find mistakes and resolve them.

**Disadvantages:**

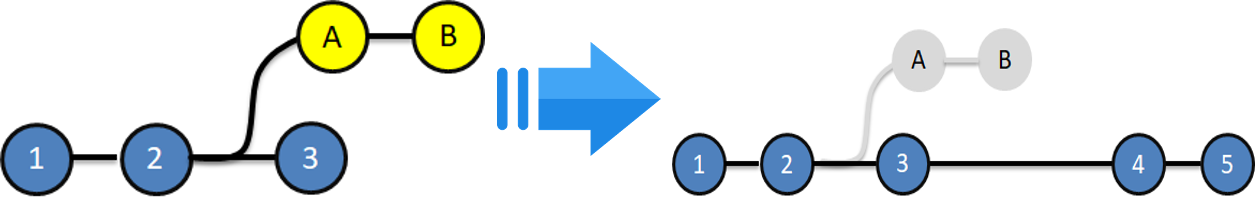
* Results in a clumsy log / history
* Not very user-friendly



**Example showing a repository with multiple branches which were merged using git-merge**

**Git-Rebase**  
Git Rebase is similar to git merge, but the logs are modified after merge in this technique. Git rebase was introduced to overcome the limitation of merging, i.e., to make logs of repository history look linear.

Let us take an example if we have a project with 3 commits on the master branch as commit 1,2,3 and feature branch commits as commit A and B. If we perform a git rebase operation then the commits A and B will be rebased on to the master branch as commit 4 and 5 and there will be no logs of the feature branches.



**Before and After git rebase**

**Advantages:**

* The logs are linear
* It’s easy to move through the project.

**Disadvantages:**

* We cannot track, when and how the commits were merged on the target branch

**Git Merge Vs Git Rebase:**

|  |  |
| --- | --- |
| Merge | Rebase |
| Git merge is a command that allows you to merge branches from Git. | Git rebase is a command that allows developers to integrate changes from one branch to another. |
| In Git Merge logs will be showing the complete history of the merging of commits. | Logs are linear in Git rebase as the commits are rebased |
| All the commits on the feature branch will be combined as a single commit in the master branch. | All the commits will be rebased and the same number of commits will be added to the master branch. |
| Git Merge is used when the target branch is shared branch | Git Rebase should be used when the target branch is private branch |