1. Difference between git reverts and reset

Git revert it undoes commit from last commit and adds as commit record in history.

Git reset is resets or makes it look as it was before by removing all your latest changes and also no commit history recorded.

For example

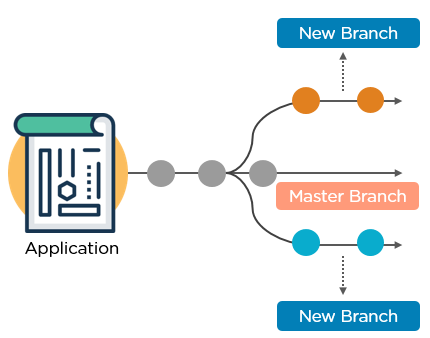
1)Assume my repo has 2 files X and Y, now I have committed file name called Z to repository.

2)Now I will do revert in this case it will change repo to have only 2 files X n Y, but in commit history you can see you added file Z and then removed.

3)Now will do reset in this case repo will again have only 2 files X n Y , but nothing will be mentioned in commit history nobody knows what happened as it came back original status .

1. Git conflict in the demo

* Git is a Version Control tool that is used to maintain each version of the source code on the GitHub repository. Git has a lot of features which are really helpful in building a large scale project. Git merge is one of them.
* Now, let’s understand what Git merge is and how it works.
* If we are working on a web application or mobile application, we often prefer to build the features of the application separately. Git allows us to create a new branch and work on it till the feature is complete. This will not disturb the feature branch and the changes us make to the application will remain in the new branch itself.
* We always work on the master branch (main branch). Once We are done with all the changes, We can merge our newly created branch on to the master branch.
* Here is a simple example of how we can create a new branch and work on it,



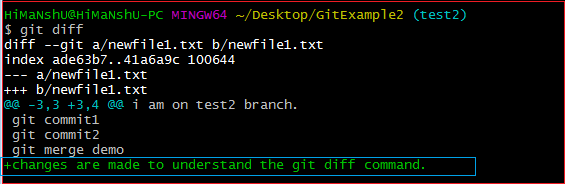
* The straight line the middle is the master branch, while the one above and below it are the new feature branches. The circles on the branch represent various commits made to the respective branches while building the application.
* Once We complete building the new features of the application, We can merge it to the master.

1. Git diff command with an demo

* Git diff is a command-line utility. It's a multiuse Git command. When it is executed, it runs a diff function on Git data sources. These data sources can be files, branches, commits, and more. It is used to show changes between commits, commit, and working tree, etc.
* It compares the different versions of data sources. The version control system stands for working with a modified version of files. So, the diff command is a useful tool for working with Git.
* However, we can also track the changes with the help of git log command with option -p. The git log command will also work as a git diff command.

**Scenerio1: Track the changes that have not been staged.**

* The usual use of git diff command that we can track the changes that have not been staged.
* Suppose we have edited the newfile1.txt file. Now, we want to track what changes are not staged yet. Then we can do so from the git diff command. Consider the below output:



* From the above output, we can see that the changes made on newfile1.txt are displayed by git diff command. As we have edited it as "changes are made to understand the git diff command." So, the output is displaying the changes with its content. The highlighted section of the above output is the changes in the updated file. Now, we can decide whether we want to stage this file like this or not by previewing the changes.

1. Tags and its purpose

* Meta tag <Meta> describes about Our entire website and they are not visible to users but specify the required description to search engine. Meta tag can be created by many of the SEO tools. But the best to create is to do it yourself. Go to your HTML file>head tag> Create a meta tag(<meta>Write what you think your website is about</meta>)
* Meta description: Meta description is visible to viewers and is created to tell the viewers what the particular website contains.meta description can be created by using <meta name=”description” content=”Precise description of your webpage in 160 characters”> inside the head tag. Our Meta description must specify exactly what our website contains.

5. Profiles in maven for different environment

* Profile in maven is nothing but subset of elements which allows customizing builds for particular environment. Profiles are also portable for different build environments.
* Build environment basically means a specific environment set for production and development instances. When developers work on development phase, they are intending to use database from the production instance and for the production phase, the live database will be used.
* So, in order to configure these instances maven provides the feature of build profiles. Any no. of build profiles can be configured and also can override any other settings.
* These defined profiles have the ability to modify the pom.xml during the build time. I.e. to configure separate environments for development and production instances. Based on the parameters passed, the corresponding profiles are activated accordingly. **E.g.** profiles can be set for dev., test and production phases.

Below are ways in which build profiles of maven can be activated or triggered:

* Explicitly using commands
* Maven settings
* Based on environment variables
* Operating system settings
* Present/missing files