**Q.    How can we see n most recent commits in GIT?**

We can use git log command to see the latest commits. To see the three most recent commits we use following command:

git log -3

**Q.   How can we know if a branch is already merged into master in GIT?**

We can use following commands for this purpose:

git branch --merged master  : This prints the branches merged into master

git branch --merged lists : This prints the branches merged into HEAD (i.e. tip of current branch)

git branch --no-merged : This prints the branches that have not been merged

By default this applies only to local branches.

We can use -a flag to show both local and remote branches.

Or we can use -r flag to show only the remote branches.

**Q.    What is the purpose of git stash drop?**

In case we do not need a specific stash, we use git stash drop command to remove it from the list of stashes.

By default, this command removes to latest added stash

To remove a specific stash we specify as argument in the git stash drop <stashname> command.

**Q.   What is the HEAD in GIT?**

A HEAD is a reference to the currently checked out commit.

It is a symbolic reference to the branch that we have checked out.

At any given time, one head is selected as the ‘current head’ This head is also known as HEAD (always in uppercase).

**What is the most popular branching strategy in GIT?**

There are many ways to do branching in GIT. One of the popular ways is to maintain two branches:

master: This branch is used for production. In this branch HEAD is always in production ready state.

develop: This branch is used for development. In this branch we store the latest code developed in project. This is work in progress code.

Once the code is ready for deployment to production, it is merged into master branch from develop branch.

[What is the most popular branching strategy in GIT?](javascript:void(0))

[Preview](javascript:void(0))

**What is SubGit?**

SubGit is software tool used for migrating SVN to Git. It is very easy to use. By using this we can create a writable Git mirror of a Subversion repository.

It creates a bi-directional mirror that can be used for pushing to Git as well as committing to Subversion.

SubGit also takes care of synchronization between Git and Subversion.

**Q.   What is the use of git instaweb?**

Git-instaweb is a script by which we can browse a git repository in a web browser.

It sets up the gitweb and a web-server that makes the working repository available online.

What is SubGit?

02:25

**What are git hooks?**

Git hooks are scripts that can run automatically on the occurrence of an event in a Git repository. These are used for automation of workflow in GIT.

Git hooks also help in customizing the internal behavior of GIT.

These are generally used for enforcing a GIT commit policy.

**Q.    What is a repository in GIT?**

A repository in GIT is the place in which we store our software work.

It contains a sub-directory called .git. There is only one .git directory in the root of the project.

In .git, GIT stores all the metadata for the repository. The contents of .git directory are of internal use to GIT.

What are git h

**What is GIT?**

GIT is a mature Distributed Version Control System (DVCS). It is used for Source Code Management (SCM).

It is open source software. It was developed by Linus Torvalds, the creator of Linux operating system.

GIT works well with a large number of IDEs (Integrated Development Environments) like- Eclipse, InteliJ etc.

GIT can be used to handle small and large projects.

**What are the disadvantages of GIT?**

GIT has very few disadvantages. These are the scenarios when GIT is difficult to use. Some of these are:

* I.         Binary Files: If we have a lot binary files (non-text) in our project, then GIT becomes very slow. E.g. Projects with a lot of images or Word documents.
* II.         Steep Learning Curve: It takes some time for a newcomer to learn GIT. Some of the GIT commands are non-intuitive to a fresher.
* III.         Slow remote speed: Sometimes the use of remote repositories in slow due to network latency. Still GIT is better than other VCS in speed.

**How will you start GIT for your project?**

We use git init command in an existing project directory to start version control for our project.

After this we can use git add and git commit commands to add files to our GIT repository.

**Q.    What is git clone in GIT?**

In GIT, we use git clone command to create a copy of an existing GIT repository in our local.

This is the most popular way to create a copy of the repository among developers.

It is similar to svn checkout. But in this case the working copy is a full-fledged repository.

How will you star

To create a new repository in GIT, first we create a directory for the project. Then we run ‘git init’ command.

Now, GIT creates .git directory in our project directory. This is how our new GIT repository is created.

**Q.    How will you create a repository in GIT?**

To create a new repository in GIT, first we create a directory for the project. Then we run ‘git init’ command.

Now, GIT creates .git directory in our project directory. This is how our new GIT repository is created.

**Q.   What are the different ways to start work in GIT?**

We can start work in GIT in following ways:

New Project: To create a new repository we use git init command.

Existing Project: To work on an existing repository we use git clone command.

**Q.   GIT is written in which language?**

Most of the GIT distributions are written in C language with Bourne shell. Some of the commands are written in Perl language.

**What is the purpose of git config command?**

We can set the configuration options for GIT installation by using git config command.

**Q.   How can we see the configuration settings of GIT installation?**

We can use ‘git config --list’ command to print all the GIT configuration settings in GIT installation.

**Q.   How will you write a message with commit command in GIT?**

We call following command for commit with a message:

$/> git commit –m <message>

**What is stored inside a commit object in GIT?**

GIT commit object contains following information:

SHA1 name: A 40 character string to identify a commit

Files: List of files that represent the state of a project at a specific point of time

Reference: Any reference to parent commit objects

**Q.   How many heads can you create in a GIT repository?**

There can be any number of heads in a repository.

By default there is one head known as HEAD in each repository in GIT.

**How will you create a new branch in GIT?**

We use following command to create a new branch in GIT:

$/> git checkout –b <branchname>

**Q.   How will you add a new feature to the main branch?**

We do the development work on a feature branch that is created from master branch. Once the development work is ready we use git merge command to merge it into master branch.

**.   What is a pull request in GIT?**

A pull request in GIT is the list of changes that have been pushed to GIT repository. Generally these changes are pushed in a feature branch or hotfix branch. After pushing these changes we create a pull request that contains the changes between master and our feature branch. This pull request is sent to reviewers for reviewing the code and then merging it into develop or release branch.

**Q.   What is merge conflict in GIT?**

A merge conflict in GIT is the result of merging two commits. Sometimes the commit to be merged and current commit have changes in same location. In this scenario, GIT is not able to decide which change is more important. Due to this GIT reports a merge conflict. It means merge is not successful. We may have to manually check and resolve the merge conflict.

What is a pull request in GIT?

**What is the alternative command to merging in GIT?**

Another alternative of merging in GIT is rebasing. It is done by git rebase command.

**Q.   What is Rebasing in GIT?**

Rebasing is the process of moving a branch to a new base commit. It is like rewriting the history of a branch.

In Rebasing, we move a branch from one commit to another. By this we can maintain linear project history.

Once the commits are pushed to a public repository, it is not a good practice to use Rebasing.

**What GUI do you use for working on GIT?**

There are many GUI for GIT that we can use. Some of these are:

* GitHub Desktop
* GITX-dev
* Gitbox
* Git-cola
* SourceTree
* Git Extensions
* SmartGit
* GitUp

**Q.   What is the use of git diff  command in GIT?**

In GIT, git diff command is used to display the differences between 2 versions, or between working directory and an index, or between index and most recent commit.

It can also display changes between two blob objects, or between two files on disk in GIT.

It helps in finding the changes that can be used for code review for a feature or bug fix.

What GUI do you use for working on GIT?

**What is git rerere?**

In GIT, rerere is a hidden feature. The full form of rerere is “reuse recorded resolution”.

By using rerere, GIT remembers how we’ve resolved a hunk conflict. The next time GIT sees the same conflict, it can automatically resolve it for us.

**Q.   What are the three most popular version of git diff command?**

Three most popular git diff commands are as follows:

git diff: It displays the differences between working directory and the index.

git diff –cached: It displays the differences between the index and the most recent commit.

git diff HEAD: It displays the differences between working directory and the most recent commit

What is git rerere?

**Why do we need git add command in GIT?**

GIT gives us a very good feature of staging our changes before commit. To stage the changes we use git add command. This adds our changes from working directory to the index.

When we are working on multiple tasks and we want to just commit the finished tasks, we first add finished changes to staging area and then commit it. At this time git add command is very helpful.

**Q.   Why do we use git reset command?**

We use git reset command to reset current HEAD to a specific state. By default it reverses the action of git add command.

So we use git reset command to undo the changes of git add command.

**What are the programming languages in which git hooks can be written?**

Git hooks are generally written in shell and PERL scripts. But these can be written in any other language as long as it has an executable.

Git hooks can also be written in Python script.

**Q.   What is a commit message in GIT?**

A commit message is a comment that we add to a commit. We can provide meaningful information about the reason for commit by using a commit message.

In most of the organizations, it is mandatory to put a commit message along with each commit.

Often, commit messages contain JIRA ticket, bug id, defect id etc. for a project.

**How GIT protects the code in a repository?**

GIT is made very secure since it contains the source code of an organization. All the objects in a GIT repository are encrypted with a hashing algorithm called SHA1.

This algorithm is quite strong and fast. It protects source code and other contents of repository against the possible malicious attacks.

This algorithm also maintains the integrity of GIT repository by protecting the change history against accidental changes.

**Q.   How GIT provides flexibility in version control?**

GIT is very flexible version control system. It supports non-linear development workflows. It supports flows that are compatible with external protocols and existing systems.

GIT also supports both branching and tagging that promotes multiple kinds of workflows in version control.

How GIT protects the code in a repository?

**How can we set up a Git repository to run code sanity checks and UAT tests just before a commit?**

We can use git hooks for this kind of purpose. We can write the code sanity checks in script. This script can be called by pre-commit hook of the repository.

If this hook passes, then only commit will be successful.

**Q.   How can we revert a commit that was pushed earlier and is public now?**

We can use git revert command for this purpose.

Internally, git revert command creates a new commit with patches that reverse the changes done in previous commits.

The other option is to checkout a previous commit version and then commit it as a new commit.

**In GIT, how will you compress last n commits into a single commit?**

To compress last n commits a single commit, we use git rebase command. This command compresses multiple commits and creates a new commit. It overwrites the history of commits.

It should be done carefully, since it can lead to unexpected results.

**Q.   How will you switch from one branch to a new branch in GIT?**

In GIT, we can use git checkout <new branchname> command to switch to a new branch.

**Q.   How can we clean unwanted files from our working directory in GIT?**

GIT provides git clean command to recursively clean the working tree. It removes the files that are not under version control in GIT.

If we use git clean –x, then ignored files are also removed.

**What is the purpose of git tag command?**

We use git tag command to add, delete, list or verify a  tag object in GIT.

Tag objects created with options –a, -s, -u are also known as annotated tags.

Annotated tags are generally used for release.

**Q.   What is cherry-pick in GIT?**

A git cherry-pick is a very useful feature in GIT. By using this command we can selectively apply the changes done by existing commits.

In case we want to selectively release a feature, we can remove the unwanted files and apply only selected commits.

**Q.   What is shortlog in GIT?**

A shortlog in GIT is a command that summarizes the git log output.

The output of git shortlog is in a format suitable for release announceme

**What is git grep?**

GIT is shipped along with a grep command that allows us to search for a string or regular expression in any committed tree or the working directory.

By default, it works on the files in your current working directory.

**Q.   How can your reorder commits in GIT?**

We can use git rebase command to reorder commits in GIT. It can work interactively and you can also select the ordering of commits.

**Q.   How will you split a commit into multiple commits?**

To split a commit, we have to use git rebase command in interactive mode. Once we reach the commit that needs to be split, we reset that commit and take the changes that have been reset. Now we can create multiple commits out of that.

**What are ours and theirs merge options in GIT?**

In  GIT, we get two simple options for resolving merge conflicts: ours and theirs

These options tell the GIT which side to favor in merge conflicts.

In ours, we run a command like git merge -Xours branchA

As the name suggests, in ours, the changes in our branch are favored over the other branch during a merge conflict.

**Q.   How can we ignore merge conflicts due to Whitespace?**

GIT provides an option ignore-space-change in git merge command to ignore the conflicts related to whitespaces.

The command to do so is as follows:

git merge -Xignore-space-change whitespace

**What is git blame?**

In GIT, git blame is a very good option to find the person who changed a specific line. When we call git blame on a file, it displays the commit and name of a person responsible for making change in that line.

**Q.   What is a submodule in GIT?**

In GIT, we can create sub modules inside a repository by using git submodule command.

By using submodule command, we can keep a Git repository as a subdirectory of another Git repository.

It allows us to keep our commits to submodule separate from the commits to main Git repository.