1.2 CRUD Operations on Git



This section will guide you to:

* Revert the earlier commits
* Ignore files in Git
* Pull the commits and collaborate between the repositories

This guide has three subsections, namely:

1.2.1 Reverting the previous commits

1.2.2 Ignoring specific files from the .git directory

1.2.3 Pulling the commits and collaborating between the local and remote repositories

**Step 1.2.1:** Reverting the previous commits.

* You can get the hash by running **git log**
* You can run **git log - -online** to simplify the output.
* You can revert the previous commits by running **git revert <commit hash>**
* You can fix the detached head by running **git checkout <current branch>**

**Step 1.2.2:** Ignoring specific files from the .git directory.

* Create a **.gitignore** file in the directory of the files that needs to be checked in to your GitHub account.
* Use the globbing patterns to match against file names. You can find all the globbing pattern information [here](https://linux.die.net/man/7/glob).

**Step 1.2.3:** Pulling the commits and collaborating between the local and remote repositories.

* The **git pull** command first runs **git fetch** which downloads content from the specified remote repository.
* The **git merge** is executed to merge the remote content references and heads into a new local merge commit.
* You can execute **git pull <options>** to fetch the specified remote’s copy of the current branch and immediately merge it into local copy.
* You can execute **git pull --no-commit <remote>** tofetch the remote content, but it doesn’t create a new merge commit.
* You can execute **git pull --verbose** to fetch the verbose output during a pull.
* You can also execute **git pull --rebase <remote>** instead of **git merge**