1. Working with Branches in GIT is pretty easy and **it is integral part of it**.
2. A Branch in Git is actually **a simple file that contains the 40 character SHA-1 checksum of the commit it points to**.
3. **Branches are cheap to create and destroy**
4. Creating a new branch is as quick and simple as writing 41 bytes to a file (40 characters and a newline).
5. This is in sharp contrast to the way most older VCS tools branch, which involves copying all of the project’s files into a second directory. This can take several seconds or even minutes, depending on the size of the project, whereas in Git the process is always instantaneous.
6. Also, because we’re recording the parents when we commit, finding a proper merge base for merging is automatically done for us and is generally very easy to do.
7. **Hence developers are encourage to create and use branches often**
8. Many git repository managers like GitLab and GitHub allow for branches to be protected, which means that not everyone is allowed to just push changes there. There the master is usually protected by default.
9. Following are major features of Branching apart from Merging
10. **Transplant a branch**: Commits in a branch can be moved to another repository.

2. **Transfer certain changes only**: Individual commits of a branch can be copied to another

branch. This is called cherry-picking.

3. **Clean up history:** A branch's history can be transformed, sorted and deleted. This would

make the history better documentation for the project. This is called interactive

rebasing.

1. Short lived branches called **Topic branches**

Note:

1. origin is default name provided by GIT for remote
2. master is the default name provided by GIT for branch
3. There is no special significance for origin and master except of as mentioned in above points