**Git add**

* Adds to the staging area.
* Need to add the file every time you alter the file.
* Changes not recorded until you commit them
* When you are happy with the changes made to the file, you can commit them from the staging area to the local repository.

**Git commit**

* Does not change the origin (remote repository)
* an SVN (Apache Subversion) commit consists of a diff compared to the original file added to the repository. Git, on the other hand, records the *entire contents* of each file in every commit.
  + This makes many Git operations much faster than SVN, since a particular version of a file doesn’t have to be “assembled” from its diffs—the complete revision of each file is immediately available from Git's internal database.
* lets developers work in an isolated environment, deferring integration until they’re at a convenient break point.

**Staging area**

* Lets you group related changes into highly focused snapshots (i.e. atomic commits)
* You can make all sorts of changes to unrelated files, commit only a few related files into one atomic unit and then commit others. i.e. split them into logical commits.

**Atomic commits**

* Each commit has only related files.
* So that it is easier to track bugs and revert changes with minimal impact on the rest of the project.

**Git status**

* Lists which files are staged, unstaged and untracked.

**.gitignore**

* Lists all files that shouldn’t be included in git status command’s results
* Ex: .exe, .pyc, .obj etc (compiled binaries)

**Git log**

* Operates only on the committed history.
* Displays committed snapshots
* Git log --oneline
* Git log –n <lines>
* Git log –stat
* Git log <file>
* Git log –graph --oneline

**commit 3157ee3718e180a9476bf2e5cab8e3f1e78a73b7**

* 40 character SHA-1 checksum of the commit’s contents.
* Also serves as unique ID for the commit.

**HEAD**

* Current commit