**Cloning Repositories**

In order to clone a remote repository you'll want to use the git clone command, which is typically used in this fashion.

git clone <repo-url>

**Cloning Local Repositories**:-

git clone< your repo path >

**Git File status life cycle:-**

**General workflow is as follows** −

* You clone the Git repository as a working copy.
* You modify the working copy by adding/editing files.
* If necessary, you also update the working copy by taking other developer's changes.
* You review the changes before commit.
* You commit changes. If everything is fine, then you push the changes to the repository.
* After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.

**Gitignore :-**

Ignored files are usually build artifacts and machine generated files that can be derived from your repository source, or should otherwise not be committed.

Some common examples are:

* dependency caches, such as the contents of /node\_modules or /packages
* compiled code, such as .o, .pyc, and .class files
* build output directories, such as /bin, /out, or /target
* files generated at runtime, such as .log, .lock, or .tmp
* hidden system files, such as .DS\_Store or Thumbs.db
* personal IDE config files, such as .idea/workspace.xml

**ignoring file form git directory:-**

1. create file in a working directory like below.

* touch error.log
* touch .gitignore

1. in the git ignore file mentioned the file extension as you want to ignore.

Example :- .log

**Git diff:-**

showing changes between staging area and working area, try below commands for output :-

* git diff
* git diff --staged