```
.git
|-objects/
|-info/
|-logs/
|-hooks/
|-refs/
|-HEAD
|-config
```

Git uses #2 technics in storing/managing the sourcecode in the repository as below:

1. Git uses hashing technic in storing & organizing the sourcecode and commit log/history information in the repository.
2. The sourcecode and the commit history of the repository is stored in Blob file format within the repository

Let us explore how does git uses these #2 above technics in maintaining the repository:
1. We have #3 files to be commited in the repository
 - vegetables.txt
 - fruits.txt
 - groceries.txt

```
git add .
git commit -m “added 3 files to todo list”
```

These 3 files are stored inside the .git/objects directory as 3 blob files in encrypted format. While storing these files it uses the naming convention as below

Filename = hash(filecontents) = generates an hashvalue
the first 2 letters of the hashvalue will be used for creating an directory under .git/objects and remaining hashvalue is used as filename, into which the contents of the file are stored in binary/encrypted format

For eg.. we have fruits.txt file with below contents

```
fruits.txt
----------
apple
grapes
```

```
git hash-object fruits.txt = 8cc36febaef8c8eec10f5685c2589e60cd0ae5c8
```

when we issue git commit, the GIT internally uses the above command for computing the hashvalue for each file based on its contents and stores the contents of the file as described above

```
.git
|-objects
  |-8c
    |-c36febaef8c8eec10f5685c2589e60cd0ae5c8 (file contents stored in binary/encrypted format)
```

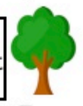
we can access the contents of the file that is stored by the git in binary encrypted format by using the below command
git cat-file -p hashvalue of the file.
for eg.. the above contents can be see using git cat-file -p 8cc36febaef8c8eec10f5685c2589e60cd0ae5c8

```
vegetables.txt. —> hash(vegetables.txt#contents) = f2fbc2e23da2e1afb0e1a2f952e50609e6df7020 —> .git/objects
--------------                                                                                  |-f2
cucumber                                                                                          |-fbc2e23da2e1afb0e1a2f952e50609e6df7020 (encrypts & stores)
peppers

groceries.txt -> hash(groceries.txt#contents) = fd703b4eb11bf383556bcbc79819044f2efd56a5 —> .git/objects
                                                                                              |-fd
                                                                                                |-703b4eb11bf383556bcbc79819044f2efd56a5 (encrypts & stores)
```

#2. Tree (object)
To identify within a commit, how many files are included, the GIT per each commit creates an Tree file and stores the hashvalue -> filename inside it as shown below

```
edb0d5683cbad5f97738862539ee59b0bbcb8924 fruits.txt
f2fbc2e23da2e1afb0e1a2f952e50609e6df7020 vegetables.txt
fd703b4eb11bf383556bcbc79819044f2efd56a5 groceries.txt
```



Tree File

```
—> hash(treefile#contents) -> b70588632590668910 1c088389b3988a89d48a29 —> .git/objects
                                                                             |-b7
                                                                             |-0588632590668910 1c088389b3988a89d48a29
```

#3. Commit Info
An Tree File holds the details of the Files that are being part of an commit, but we dont know this Tree file is representing the Files information of which commit, who is the author of the commit etc. To keep track of each commit and their details the GIT creates on more object called Commit Info object as below

```
Tree b70588632590668910 1c088389b3988a89d48a29  -> hash(contents) -> 3dffb54735f75f6f0347a33c82d05cffdbcd5e1f —> .git/objects
author tech.sriman@gmail.com                                                                                  |-3d
comitter tech.sriman@gmail.com                                                                                  |-ffb54735f75f6f0347a33c82d05cffdbcd5e1f (encrypts & stores)
```

```
.git
 |-refs
   |-main -> Last Commit Hash (C3)

 |-logs
   |-refs
     |-heads
       |-main
```

.git/logs/refs/heads/main
it stores each commit as an entry inside this file with
hierarchial entries as parent commit - latest commit

```
00000   3dffb5
3dffb5  C2
C2      C3
```