//In order to gain further understanding of Git, focus on commands.

|  |
| --- |
| “Porcelain” Commands |

* [git add](https://git-scm.com/docs/git-add)
* Provides the content or type of an object in the repository.
* [git commit](https://git-scm.com/docs/git-commit)
* Creates a new commit containing the current contents of the index and the given log message describing the changes.
* git push
* git pull
* git branch
* git checkout
* git merge
* git rebase

|  |
| --- |
| “Plumbing” Commands |

* [git cat-file](https://git-scm.com/docs/git-cat-file)
* Provides the content or type of an object in the repository.
* [git hash-object](https://git-scm.com/docs/git-hash-object)
* Computers the object ID value for an object with specified type with the contents of the names file.
* [git count-objects](https://git-scm.com/docs/git-count-objects)
* Counts the number of unpacked object files and disk space consumed by them, to help you decide when it is a good time to repack.

|  |
| --- |
| Git Is An Onion |

* A Distributed Revision Control System.
* [A Revision Control System](file:///\\Naeast.ad.jpmorganchase.com\home\amerxbus\XBUS\nacdc1vdihome12\R666050\jpmDesk\Desktop\My%20Stuff\Proficiency;%20Git\Git3\2\2.PNG)

When you recommit a file on git it is given a parent (the previous commit) and a new SHA1. This new commit should be tied to a different tree as well. If the content within a file has not changed a new object will not be created and the SHA1 will be the same as its parent object.

* [A Stupid Content Tracker](file:///\\Naeast.ad.jpmorganchase.com\home\amerxbus\XBUS\nacdc1vdihome12\R666050\jpmDesk\Desktop\My%20Stuff\Proficiency;%20Git\Git3\1\1.PNG) (Control System)

Analogy:

Git is a library (repository). Your projects become the books (objects) stored within the library and they are stored in the same area of the library based on the date they are added (git add .) and committed (git commit –m “”).

When using the commit command on git, Git creates a file with the commit message and stores it the same way it does any other file, in the object’s folder of its repository. This file contains the author, committer, date of the commit, message of the commit, and SHA1 of the tree (directory, root of the project). The tree contains a list of the contents inside the directory, their names and SHA1s.

* A Persistent Map

Git generates a unique **Secure Hash Algorithm 1** for every directory, file and character in a file. The first two characters of a file’s SHA1 are used to name a directory and a blank document is stored within that directory, named with the rest of the hash code. These files are stored in the objects folder under different directories so we will call them objects. When files have the same content Git will reference the same object. It then takes the SHA1s and maps them in a repository.

|  |
| --- |
| Git Objects |

* Blobs (Files)
* Trees (Tracker for Files)
* Commits (Miscellaneous Info.)
* Annotated Tags (Shorthand Reference)

|  |
| --- |
| Three Rules |

1. The current branch tracks new commits
2. When you move to another commit, Git updates your working directory
3. Unreachable objects are garbage collected

|  |
| --- |
|  |

|  |
| --- |
|  |