

Ex 1

Chapter 4

initializing an empty repository

```
mima@DESKTOP-QRA0HQH:~$ mkdir /tmp/hello
mima@DESKTOP-QRA0HQH:~$ cd /tmp/hello/
mima@DESKTOP-QRA0HQH:/tmp/hello$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /tmp/hello/.git/
mima@DESKTOP-QRA0HQH:/tmp/hello$ find .
.
./.git
./.git/branches
./.git/info
./.git/info/exclude
./.git/objects
./.git/objects/pack
./.git/objects/info
./.git/hooks
./.git/hooks/update.sample
./.git/hooks/fsmonitor-watchman.sample
./.git/hooks/pre-merge-commit.sample
./.git/hooks/pre-commit.sample
./.git/hooks/pre-receive.sample
./.git/hooks/pre-push.sample
./.git/hooks/pre-rebase.sample
./.git/hooks/push-to-checkout.sample
./.git/hooks/prepare-commit-msg.sample
./.git/hooks/commit-msg.sample
./.git/hooks/pre-applypatch.sample
./.git/hooks/post-update.sample
./.git/hooks/applypatch-msg.sample
./.git/HEAD
./.git/config
./.git/description
./.git/refs
./.git/refs/tags
./.git/refs/heads
mima@DESKTOP-QRA0HQH:/tmp/hello$
```

git objects

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ find .git/objects
.git/objects
.git/objects/pack
.git/objects/info
mima@DESKTOP-QRA0HQB:/tmp/hello$ echo "hello world" >
hello.txt
mima@DESKTOP-QRA0HQB:/tmp/hello$ git add hello.txt
mima@DESKTOP-QRA0HQB:/tmp/hello$ find .git/objects
.git/objects
.git/objects/pack
.git/objects/info
.git/objects/3b
.git/objects/3b/18e512dba79e4c8300dd08aeb37f8e728b8dad
mima@DESKTOP-QRA0HQB:/tmp/hello$

mima@DESKTOP-QRA0HQB:/tmp/hello$ git rev-parse 3b18e51
2d
3b18e512dba79e4c8300dd08aeb37f8e728b8dad
mima@DESKTOP-QRA0HQB:/tmp/hello$ git cat-file -p 3b18e
512dba79e4c8300dd08aeb37f8e728b8dad
hello world
mima@DESKTOP-QRA0HQB:/tmp/hello$
```

Files and trees

```
hello world
mima@DESKTOP-QRA0HQB:/tmp/hello$ git ls-files -s
100644 3b18e512dba79e4c8300dd08aeb37f8e728b8dad 0
hello.txt
mima@DESKTOP-QRA0HQB:/tmp/hello$ git write-tree 68aba62e560c0ebc3396e8ae9335232cd93a3f60
68aba62e560c0ebc3396e8ae9335232cd93a3f60
mima@DESKTOP-QRA0HQB:/tmp/hello$ find .git/objects/
.git/objects/
.git/objects/pack
.git/objects/info
.git/objects/3b
.git/objects/3b/18e512dba79e4c8300dd08aeb37f8e728b8dad
.git/objects/68
.git/objects/68/aba62e560c0ebc3396e8ae9335232cd93a3f60
mima@DESKTOP-QRA0HQB:/tmp/hello$ git
cat-file -p 68aba6
100644 blob 3b18e512dba79e4c8300dd08aeb37f8e728b8dad
hello.txt
mima@DESKTOP-QRA0HQB:/tmp/hello$
```

Tree hierarchies

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ pwd
/tmp/hello
mima@DESKTOP-QRA0HQB:/tmp/hello$ mkdir subdir
mima@DESKTOP-QRA0HQB:/tmp/hello$ cp hello.txt subdir/
mima@DESKTOP-QRA0HQB:/tmp/hello$ git add subdir/hello.txt
mima@DESKTOP-QRA0HQB:/tmp/hello$ git write tree 492413269336d21fac079d4a4672e55d5d2147ac
git: 'write' is not a git command. See 'git --help'.
```

The most similar commands are

```
worktree
write-tree
```

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ git write-tree 492413269336d21fac079d4a4672e55d5d2147ac
492413269336d21fac079d4a4672e55d5d2147ac
mima@DESKTOP-QRA0HQB:/tmp/hello$ git cat-file -p 4924132693
100644 blob 3b18e512dba79e4c8300dd08aeb37f8e728b8dad    hello.txt
040000 tree 68aba62e560c0ebc3396e8ae9335232cd93a3f60    subdir
mima@DESKTOP-QRA0HQB:/tmp/hello$
```

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ find .git/objects/
.git/objects/
.git/objects/pack
.git/objects/49
.git/objects/49/2413269336d21fac079d4a4672e55d5d2147ac
.git/objects/info
.git/objects/3b
.git/objects/3b/18e512dba79e4c8300dd08aeb37f8e728b8dad
.git/objects/68
.git/objects/68/aba62e560c0ebc3396e8ae9335232cd93a3f60
mima@DESKTOP-QRA0HQB:/tmp/hello$
```

Commits

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ echo -n "Commit a file that says hello\n" | git commit-tree 492413269336d21fac079d4a4672e55d5d2147ac
4731b68f2863b8045ee7e626d26fd115015ba3c1
mima@DESKTOP-QRA0HQB:/tmp/hello$ git cat-file -p 4731b68f2
tree 492413269336d21fac079d4a4672e55d5d2147ac
author yes <yes> 1709740524 +0100
committer yes <yes> 1709740524 +0100

Commit a file that says hello\nmima@DESKTOP-QRA0HQB:/tmp/hello$
```

Tags

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ git tag -m "Tag version 1.0" V1.0 4731b68f2
```

```
mima@DESKTOP-QRA0HQB:/tmp/hello$ git cat-file -p 4731b68f2
tree 492413269336d21fac079d4a4672e55d5d2147ac
author yes <yes> 1709740524 +0100
committer yes <yes> 1709740524 +0100

Commit a file that says hello\nmima@DESKTOP-QRA0HQB:/tmp/hello$

mima@DESKTOP-QRA0HQB:/tmp/hello$ git rev-parse V1.0
0a80a24ccadb1e5557394c09d9eaddff024fceb5
mima@DESKTOP-QRA0HQB:/tmp/hello$ git cat-file -p 0a80a24c
object 4731b68f2863b8045ee7e626d26fd115015ba3c1
type commit
tag V1.0
tagger yes <yes> 1709741386 +0100

Tag version 1.0
mima@DESKTOP-QRA0HQB:/tmp/hello$
```

Textual Explanation :

When I call `git add`, Git will create a git object of type blob and stores inside the SHA1 hash corresponding to the datas of the file, and the name of this object will be an "hexadecimal representation" of the hash. This is why a Git object representing a simple file named `hello.txt` with "hello world" as content will have the same name and same hash.

Also, Git will insert a `/` after the two first digits of the file, because some filesystems can slow down when you put too many files in the same directory.