22. INSIDE GIT: .GIT DIRECTORY

Goals

- To learn about Git directory structure `.git`

The `.git` directory

It is time to do some research. Starting from the project's root directory...

RUN:

```
ls -C .git
```

RESULT:

```
$ ls -C .git
COMMIT_EDITMSG  MERGE_RR        config          hooks
info            objects         rr-cache
HEAD            ORIG_HEAD       description     index
logs            refs
```

This is a special folder where all the git stuff is. Let us explore the directory.

Object Database

RUN:

```
ls -C .git/objects
```

RESULT:

```
$ ls -C .git/objects
09      24      28      45      59      6a      77      80      8c
97      af      c4      e7      info
11      27      43      56      69      6b      78      84      91
```

```
9c      b5      e4      fa      pack
```

You should see a lot of folders named with two characters. The first two letters sha1 hash of the object stored in git are the directory names.

inquire the database objects

RUN:

```
ls -C .git/objects/<dir>
```

RESULT:

```
$ ls -C .git/objects/09
6b74c56bfc6b40e754fc0725b8c70b2038b91e
9fb6f9d3a104feb32fcac22354c4d0e8a182c1
```

Let us look at one of the folders named with two characters. There should be files with names of 38 characters. These files contain objects stored in git. They are compressed and encrypted, so it's impossible to view their content directly. Let us have a better look at Git directory

Config File

RUN:

```
cat .git/config
```

RESULT:

```
$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
        ignorecase = true
[user]
        name = Marina Pushkova
        email = marina (at) githowto.com
```

This configuration file is created for each individual project. At least in this project, entries in this file will overwrite the entries in the `.gitconfig` file of your main directory.

Branches and tags

RUN:

```
ls .git/refs
ls .git/refs/heads
ls .git/refs/tags
cat .git/refs/tags/v1
```

RESULT:

```
$ ls .git/refs
heads
tags
$ ls .git/refs/heads
master
$ ls .git/refs/tags
v1
v1-beta
$ cat .git/refs/tags/v1
fa3c1411aa09441695a9e645d4371e8d749da1dc
```

Files in tags subdirectory should be familiar to you. Each file corresponds to the tag previously created using the git tag command. Its content is nothing but a hash commit attached to the tag.

The *heads* folder is almost identical and is used not for tags, but branches. At the moment we have only one branch, and everything you see in this folder is a *master* branch.

HEAD File

RUN:

```
cat .git/HEAD
```

RESULT:

```
$ cat .git/HEAD
ref: refs/heads/master
```

There is a reference to the current branch in the HEAD file. At the moment it must be the master branch.