

Introduction to Git

Alessandro Sciarra

Z02 – Software Development Center

December 22, 2021

Outline of the talk

- 1 What is Git?
- 2 How to use Git locally?
- 3 Summary and conclusions

What is Git?

How does Git define itself?

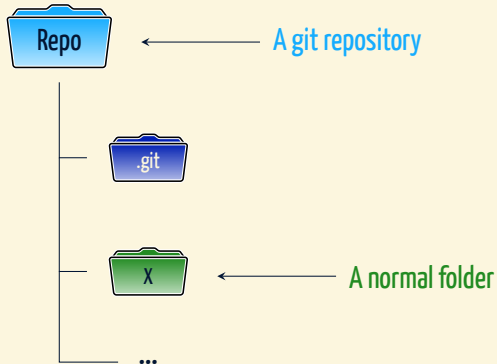
«Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance.»

 [Git homepage](https://git-scm.com/)

- 1 Free and open
- 2 **Distributed version control system**
- 3 From small to very large projects
- 4 With speed and efficiency
- 5 Easy to learn

How does it work?

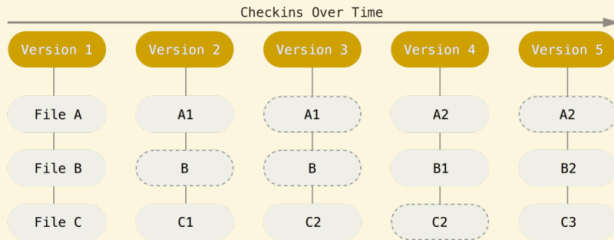
- **Repository:** a database containing all versions of the files



How does it work?

- **Repository:** a database containing all versions of the files
- **Snapshot-based system**
 - Snapshots are called commits
 - Commits are named by checksums (also used to ensure data integrity)

{ It's impossible to change the contents of any file or directory without Git knowing about it }




 From the Git-Book

How does it work?

- **Repository:** a database containing all versions of the files
- **Snapshot-based system**
 - Snapshots are called commits
 - Commits are named by checksums (also used to ensure data integrity)
{ It's impossible to change the contents of any file or directory without Git knowing about it }
- **Almost every operation is local**
 - Working without network connecting
 - Distributed system → everyone carries a backup!

Are you curious to know how Git works bottom-up? —

Refer to  this 31-pages document , well written, but not needed at start.

An example of Git history

Every commit is a snapshot of the state of the repository at that point

First
commit

HEAD



6d7253af8a986838f78adc25a801cd06897be033

54a94046ddeeb7117e77f0b6ee984165fedbc2345

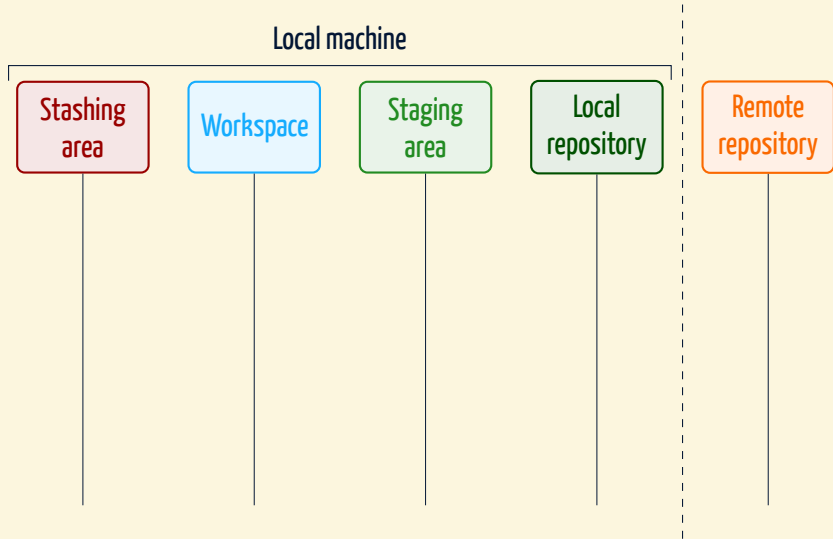
09c1b4321589b548be39063c828903901a3e14e7

b924cdb3101220b52cd0d92c8c4e7b9eb887a83a

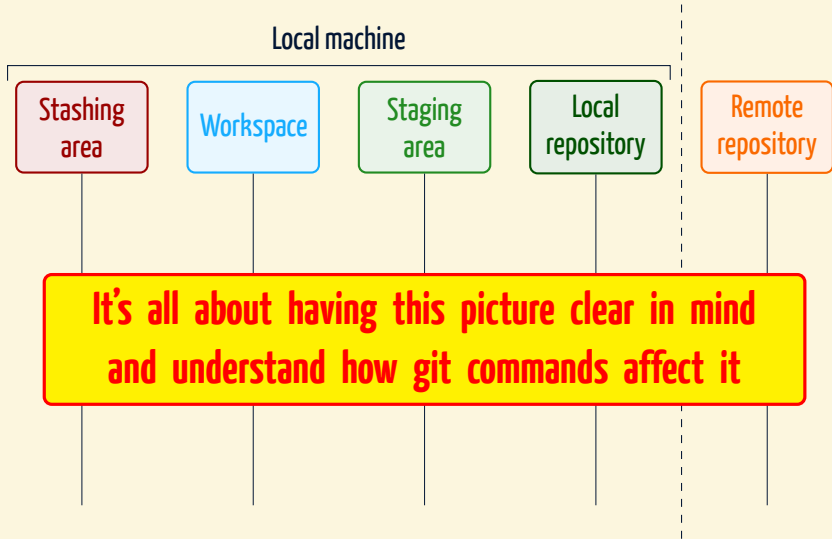
24ffea36e3b6b696cc537fb05bae8d04c63a92b

The correct abstract mental setup

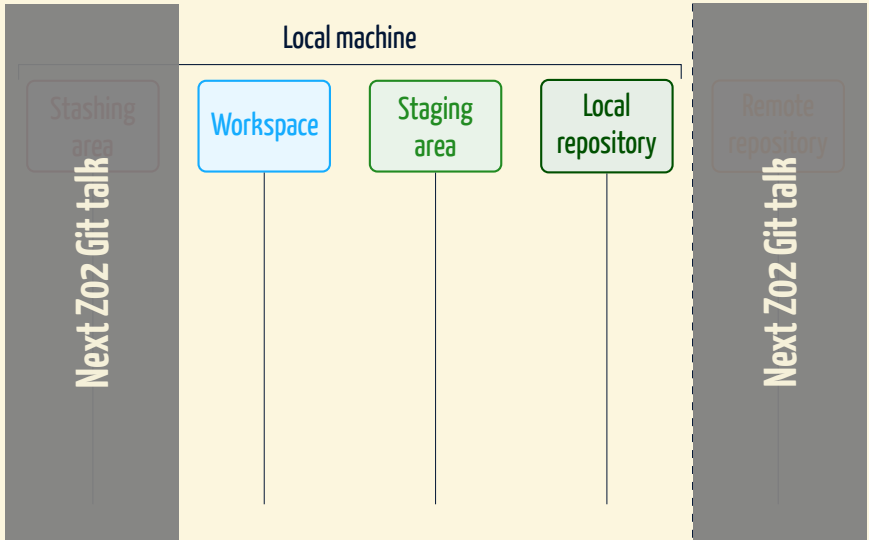
The correct abstract mental setup



The correct abstract mental setup



The correct abstract mental setup



How to use Git locally?

Preliminary steps

Be sure to introduce yourself to Git on **each** machine from which you work

- 1 It is likely that Git is installed on your machine.
 - Check it in a terminal e.g. via `git version`
 - If needed,  install it
- 2 Optionally,  get/enable autocompletion in the terminal
- 3 Tell Git who you are and your email address
 - this information will be used to sign your work in history

```
$ git config --global user.name 'Alessandro Sciarra'
$ git config --global user.email 'sciarra@itp.uni-frankfurt.de'
```

Asking for help about Git


1 There are 3 ways in terminal

- `git help <command>` e.g. `git help config`
- `git <command> --help` e.g. `git config --help`
- `man git-<command>` e.g. `man git-config`

2 List of commands on the official reference

3 Ask Google

There is plenty of cheat-sheets online:

 GitHub education

 GitLab

 Bitbucket

Creating a repository

It is as simple as running one command

```
$ git config --get user.name
Alessandro Sciarra
$ git config --get user.email
sciarra@itp.uni-frankfurt.de
# Suppose to be in a folder you want to turn into a repository
$ pwd
/home/asciarra/Documents/first-repo
$ ls -a
.  ..  Paper.aux  Paper.log  Paper.out  Paper.pdf  Paper.tex
```


Creating a repository

It is as simple as running one command

```
$ git config --get user.name
Alessandro Sciarra
$ git config --get user.email
sciarra@itp.uni-frankfurt.de
# Suppose to be in a folder you want to turn into a repository
$ pwd
/home/asciarra/Documents/first-repo
$ ls -a
.  ..  Paper.aux  Paper.log  Paper.out  Paper.pdf  Paper.tex
```

```
$ git init # <--- Here you go!
Initialised empty Git repository in ~/Documents/first-repo/.git/
$ ls -a
.  ..  .git  Paper.aux  Paper.log  Paper.out  Paper.pdf  Paper.tex
```

Creating a repository

It is as simple as running one command

```
$ git config --get user.name
Alessandro Sciarra
$ git config --get user.email
sciarra@itp.uni-frankfurt.de
# Suppose to be in a folder you want to turn into a repository
$ pwd
/home/asciarra/Documents/first-repo
$ ls -a
.  ..  Paper.aux  Paper.log  Paper.out  Paper.pdf  Paper.tex
```

```
$ git init # <--- Here you go!
Initialised empty Git repository in ~/Documents/first-repo/.git/
$ ls -a
.  ..  .git  Paper.aux  Paper.log  Paper.out  Paper.pdf  Paper.tex
```

Do not shoot yourself!

Never ever touch by hand the content of the `.git` folder.

What comes next?

What comes next?

On your local machine

Workspace

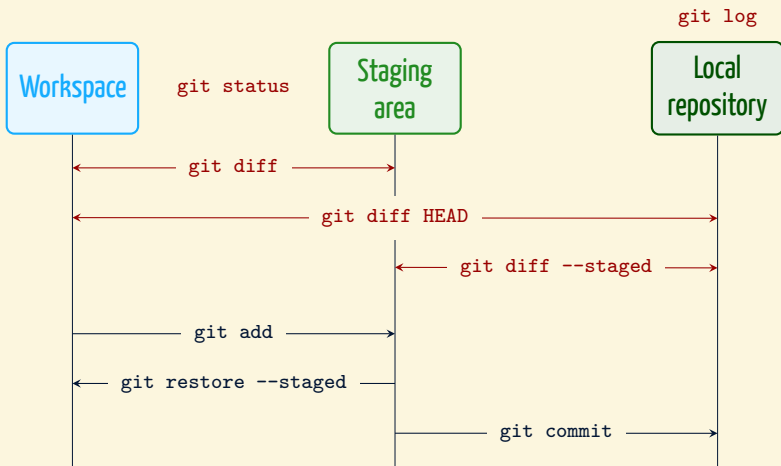
Staging
area

Local
repository

Commands marked in **dark red** do not change anything in the repository!

What comes next?

On your local machine



Commands marked in **dark red** do not change anything in the repository!

Git status

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Paper.aux
    Paper.log
    Paper.out
    Paper.pdf
    Paper.tex

nothing added to commit but untracked files present
(use "git add" to track)
```

You do not want to put everything in a repository!

It is possible to tell git to ignore some files, like temporary ones

Letting Git ignore some files

```
$ printf '*.%.s\n' {aux,log,out,pdf} > .gitignore
$ cat .gitignore
*.aux
*.log
*.out
*.pdf
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    Paper.tex

nothing added to commit but untracked files present
(use "git add" to track)
```



Our first commit

Our second commit

Working and displaying changes

Staging partial modification in a file

Summary and conclusions

That's **NOT** all folks