A minimal Git guide

S. Rossi Tisbeni
https://www.unibo.it/sitoweb/simone.rossitisbeni/

Programmazione per la Fisica, Università di Bologna

Anno Accademico 2020/2021

https://virtuale.unibo.it/course/view.php?id=18455

https://baltig.infn.it/giaco/pf2020

What is Git

Git is a tool that protects yourself and others from yourself and others.

What is Git

Git is a tool that protects yourself and others from yourself and others.
Git can:

- Keep record of all the (tracked) files in your directory.

What is Git

Git is a tool that protects yourself and others from yourself and others.
Git can:

- Keep record of all the (tracked) files in your directory.
- Maintain an history of all the changes.

What is Git

Git is a tool that protects yourself and others from yourself and others.
Git can:

- Keep record of all the (tracked) files in your directory.
- Maintain an history of all the changes.
- Revert changes to fix mistakes.

What is Git

Git is a tool that protects yourself and others from yourself and others.
Git can:

- Keep record of all the (tracked) files in your directory.
- Maintain an history of all the changes.
- Revert changes to fix mistakes.
- Allow for concurrent work, helping prevent conflicts.

What is Git

Git is a tool that protects yourself and others from yourself and others.
Git can:

- Keep record of all the (tracked) files in your directory.
- Maintain an history of all the changes.
- Revert changes to fix mistakes.
- Allow for concurrent work, helping prevent conflicts.

Also known as Version Control

git init

- `git init` initializes the working directory.

git init

- `git init` initializes the working directory.
- It tells git to start managing a repository for your workspace.

git init

- `git init` initializes the working directory.
- It tells git to start managing a repository for your workspace.
- The repository (or 'repo' for short) is a folder in a working directory in which Git tracks all changes made to files and builds a history of those changes.

git status

git status describes the current state of the repository.

```
~/workspace$ git init
```

git status

`git status` describes the current state of the repository.

```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$
```

git status

`git status` describes the current state of the repository.

```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$ git status
```

git status

`git status` describes the current state of the repository.

```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
~/workspace$
```

git status

`git status` describes the current state of the repository.

```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
~/workspace$ rm -r .git
~/workspace$
```

git status

`git status` describes the current state of the repository.

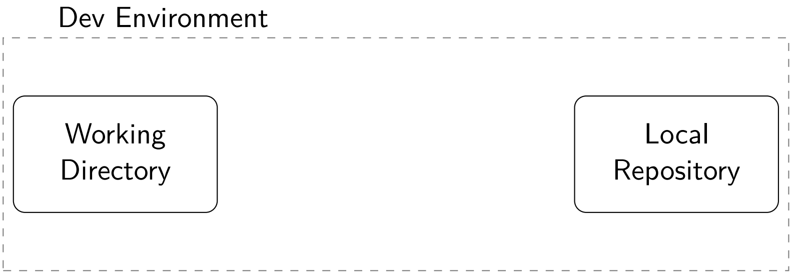
```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
~/workspace$ rm -r .git
~/workspace$ git status
```

git status

`git status` describes the current state of the repository.

```
~/workspace$ git init
Initialized empty Git repository in ~/workspace/.git/
~/workspace$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
~/workspace$ rm -r .git
~/workspace$ git status
fatal: not a git repository (or any of the parent directories): .git
```

Local Workflow



Untracked files

Files in the working directory are not automatically added to the repository.

```
~/workspace$ touch main.cpp && ls
```

Untracked files

Files in the working directory are not automatically added to the repository.

```
~/workspace$ touch main.cpp && ls
main.cpp
~/workspace$
```

Untracked files

Files in the working directory are not automatically added to the repository.

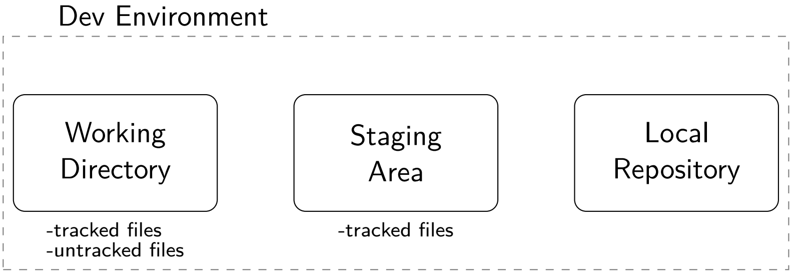
```
~/workspace$ touch main.cpp && ls
main.cpp
~/workspace$ git status
```

Untracked files

Files in the working directory are not automatically added to the repository.

```
~/workspace$ touch main.cpp && ls
main.cpp
~/workspace$ git status
On branch master

No commits yet

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

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

Local Workflow



git add

- `git add <filename>` asks Git to track changes to a file in the repository.

```
~/workspace$ git add main.cpp
```

git add

- `git add <filename>` asks Git to track changes to a file in the repository.

```
~/workspace$ git add main.cpp
~/workspace$ git status
```

git add

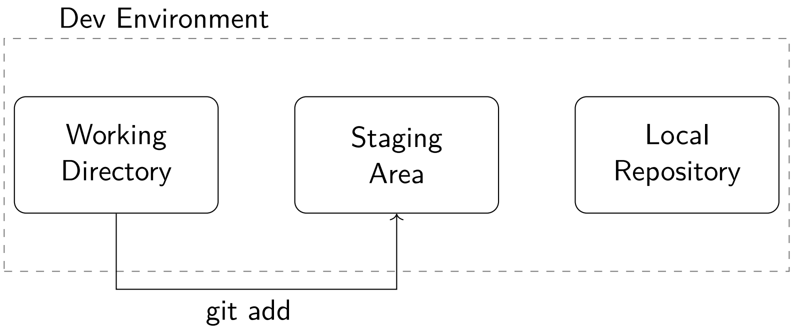
- `git add <filename>` asks Git to track changes to a file in the repository.

```
~/workspace$ git add main.cpp
~/workspace$ git status
On branch master

No commits yet

Changes to be committed:
    (use "git rm --cached <file>..." to unstage)
        new file:   main.cpp
```

git add (cont.)

- You can add multiple files to the staging area with `git add <file1> <file2> <...>`
- There is never a good reason to use `git add *`, `git add .`, `git add -A` or `git add -u`

Local Workflow



git commit -m

- Files in the staging area are tracked by Git, but their changes have not been saved on the repository.
- With `git commit` a collection of changes in the staging area is checked to the local repository and saved to the repo history.
- Each commit should be labelled with a message that clearly describes the changes made.
- The commit message can be spficied between quotation marks:

```
git commit -m "Commit message"
```

git commit -m (cont.)

- `git commit -m "Commit message"` asks Git to save changes in the local repository's history.

```
~/workspace$ git commit -m "Initial commit"
```

git commit -m (cont.)

- `git commit -m "Commit message"` asks Git to save changes in the local repository's history.

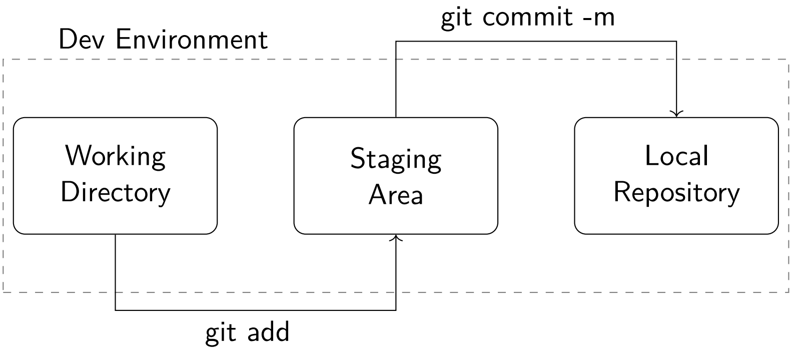
```
~/workspace$ git commit -m "Initial commit"
[master (root-commit) 0367896] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.cpp
/workspace$ git status
```

git commit -m (cont.)

- `git commit -m "Commit message"` asks Git to save changes in the local repository's history.

```
~/workspace$ git commit -m "Initial commit"
[master (root-commit) 0367896] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.cpp
/workspace$ git status
On branch master
nothing to commit, working tree clean
```

Local Workflow



git log

- `git log` show the local repository's history.

```
/workspace$ git log
commit 03678964f710e1ef1e9d6c0704e3f02f014109a0 (HEAD -> master)
Author: Simone Rossi Tisbeni <simone.rossitisbeni@unibo.it>
Date:   Fri Jan 29 14:39:12 2021 +0000

    Initial commit
```

git diff

- `git diff` is used to show the changes in the files.
- When no argument is passed, it shows the difference between the current state of the repository and the working directory.
- When `--staged` is passed as argument, it shows the difference between the current state of the repository and the staged changes.
- When a commit number is passed as argument, it shows the difference between the current state of the repository and the selected commit.