Institute For Molecular Science
Department Of Photo-molecular Science
OHMORI GROUP

Git course - Lesson 1

What is Git, installation and local use

Giorgio Micaglio

Italian Association of Physics Students
Institute for Molecular Science - Ohmori Group

July 18, 2025

Outline of the course

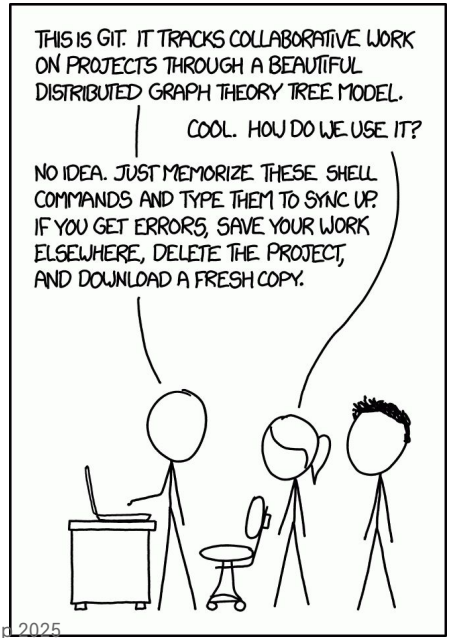
This course was initially designed by the Trento Local Committee of the Italian Association of Physics Students (AISF). You can find all material on this website: `https://ai-sf.it/trento/git/`

The main reference for the course is the great *Pro Git Book, 2nd edition, by Scott Chacon and Ben Straub*, available for free at the following link: `https://git-scm.com/book/en/v2` (both PDF and web versions)

- Lesson 1 (today, Fri 18): what is Git, installation and local use
- Lesson 2 (next week, Fri 25): branching, remote use with GitHub and collaboration

Overview

What is Git?


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Version Control Systems

- Git is a system that keeps track of all changes made to a set of files called a **repository**.[1] Such a system is called a *VCS*, or *Version Control System*.
- A correct analogy is that of a **“time machine”**: you can **easily** go back to any previous version of the project without losing data.
- Git is the *de facto standard* for software development projects of all sizes, in both industry and academia.
- **Snapshots, not differences**: Git thinks about its data as a *stream of snapshots*, other VCSs store the file changes
- **Nearly every operation is local** (browse history, see changes)
- **Git has integrity**: detects all changes thanks to checksumming

[1] Often abbreviated in **repo**

When everything began...

Linus Torvalds (Linux and Git creator) described like this the meaning of the name ”Git”:

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of ”get” may or may not be relevant.

When everything began...

Linus Torvalds (Linux and Git creator) described like this the meaning of the name ”Git”:

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of ”get” may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.

When everything began...

Linus Torvalds (Linux and Git creator) described like this the meaning of the name ”Git”:

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of ”get” may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- ”global information tracker”: you’re in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.

When everything began...

Linus Torvalds (Linux and Git creator) described like this the meaning of the name ”Git”:

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of ”get” may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- ”global information tracker”: you’re in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- ”goddamn idiotic truckload of sh*t”: when it breaks

Git in detail

Basic Idea

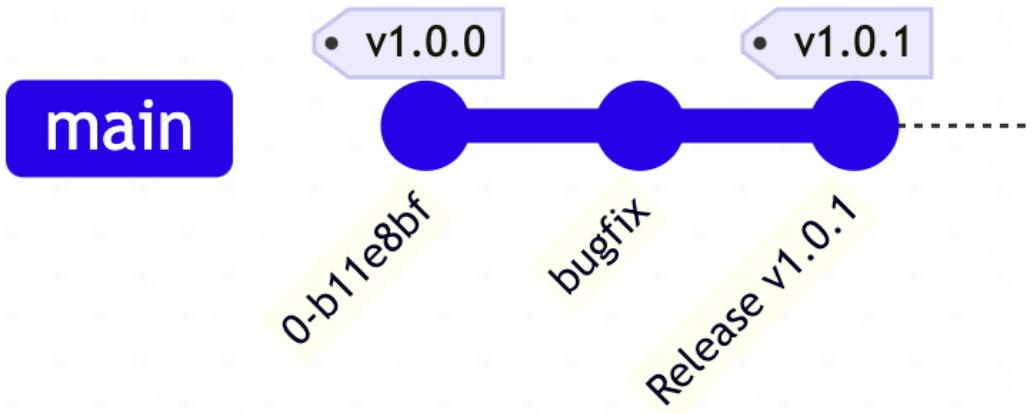
It's like taking snapshots of files in a folder, with the ability to view all the snapshots from the past at any time.

Commit

"**commit**" is both the act of "taking a snapshot" of files (verb) and the snapshot itself (noun).
Each commit is identified by a unique hexadecimal number, for example:

```
3c552345a5613a94e5f4704a8311919071fc410d
```

Basic Idea

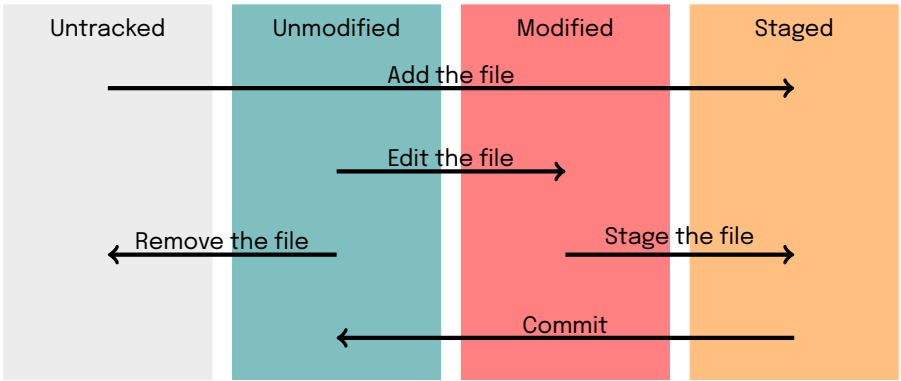


Possible States of a File

A file in a repo can be in one of 4 different states:

- **untracked**: the initial state of any newly created file. An untracked file is completely ignored by Git
- **staged**: means that the file is marked by Git as “ready to be snapshotted” with a *commit*
- **committed**: the file has been “snapshotted” (hence tracked by Git), and hasn’t been changed since the last commit
- **modified**: the file has been “snapshotted” but has been modified since the last commit

Possible States of a File



Git Repository

A Git project consists of two components:

1. **working directory**: the version of your work you're currently working on; it's the set of visible files in your folder.
2. **.git directory**: a *hidden* folder where Git operates and saves all project versions. It contains all your repository's information. DO NOT MODIFY FILES IN THIS FOLDER.

Note

Git is typically used with tools like *GitHub* to synchronize the repository on a **remote server** and enable **collaborative work**.

GitHub

- GitHub is a platform that **uses** Git and allows you to host your repositories on remote servers.
- It facilitates collaboration between users.
- It has a user-friendly and easy-to-use interface.

Note

We will use GitHub to configure remote repositories in the next lessons.

Examples

Git + GitHub are widely used in scientific collaborations and companies, many of which share open-source code:

- Ohmori Group: `https://github.com/OhmoriG`
- CERN: `https://github.com/CERN`
- Python: `https://github.com/python`
- Sony: `https://github.com/sony`
- LINE: `https://github.com/line`
- Rakuten: `https://github.com/rakutentech`

Fun Fact

Git is tracked by Git: `https://github.com/git`

Installation and shell

Giorgio Micaglio, Git @ Ohmori Group 2025

Installing Git

WSL

On **Windows 10 / 11**, you can install the **WSL - Windows Subsystem for Linux** using this guide: `https://learn.microsoft.com/en-us/windows/wsl/install`

- On Windows 10/11: Download git from this page: `https://git-scm.com/downloads/win`
- On WSL or Ubuntu:
  - `sudo apt update`
  - `sudo apt install git-all -y`
- On Mac: `git --version`. If Git is not installed, it will prompt installation.

Getting comfortable with shell commands

Institute for Molecular Science
Department Of Photomolecular Science
OHMORI GROUP

Essential Linux Shell Command Cheat Sheet

Terminal Text Editors

All bash terminals come with built-in text editors. These tools are often convenient for quick file edits. Some common ones are:

- `vim` – Hard to use
- `nano` – Pre-installed with Ubuntu
- `gedit` – GUI-based, install with `sudo apt install gedit`

Note

You can always use a graphical text editor like VSCode

Git + GitHub Configuration

Giorgio Micaglio, Git @ Ohmori Group 2025

Creating an Account

From now on, we will use GitHub.

1. If you don't have one, create an account: `www.github.com/signup`
2. Use your @ims.ac.jp address
3. Join the OhmoriG organization

The account is required to upload your repositories to the server. We'll cover this in more detail next week.

Configuring Git

Let's configure Git locally. Since Git will communicate with GitHub, we must tell Git who we are:

```
git config --global user.name "your_github_username"
git config --global user.email "your@email.com"
```

Without the `--global` flag, these settings only apply to the current repo.

Note

Password authentication is not allowed. GitHub uses **SSH keys** for authentication.

Generating an SSH Key Pair

To generate an SSH key pair:

- `cd .ssh` (or `cd ~/.ssh`)
- `ssh-keygen -t ed25519`

Leave the passphrase empty when prompted.
Use `ls` to see the generated keys:

- `keyname.pub` ← public key
- `keyname` ← private key

Uploading the Public Key to GitHub

Now upload your generated public key to GitHub:

1. Open the public key with a text editor or with `cat` and copy the entire content.
2. On GitHub, go to: **Settings** → **SSH and GPG keys** → **New SSH key**.
3. Paste the full key content and give it a name.

You're now ready to use Git with remote repositories! We will continue with this in Lesson 2.

Essential Git Commands

Giorgio Micaglio, Git @ Ohmori Group 2025

Repository Initialization

At the beginning of our project, we have a folder. Whether it's empty or full, this folder will not be tracked by *git* unless it is inside another folder already tracked by *git*.

Initializing *git*

To initialize *git* in a folder and allow it to track changes, type:

```
git init repository_name
```

```
git init repository_name
```

Repository Initialization

IMPORTANT

If a `repository_name` is specified, git will initialize the repository in a folder named `repository_name` inside the folder where the command is run.
To initialize the repository in the current folder, omit `repository_name`. This can be useful when starting to track an already existing project.

File Creation and Staging

Once the repository is initialized, we create a file, e.g.: `data_analysis.py`

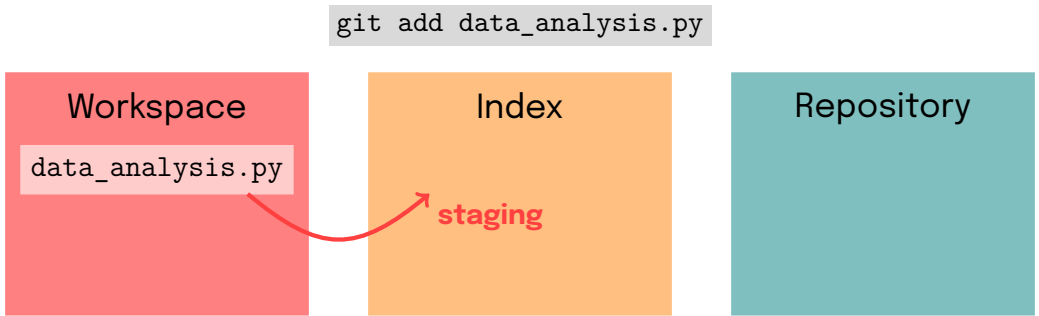
As you remember from before, this file is still untracked by *git*.
We then proceed with staging with

Staging

Move a file to the staging area with
`git add data_analysis.py`

Note:

It is possible to stage files that were already present in the folder before the repository was initialized.


git add data_analysis.py
Workspace
data_analysis.py
Index
staging
Repository

Commit

Once the files we want to track are in the *Staging Area*, we can commit. This way, *git* saves the state of those files.

Committing

To commit, use the command

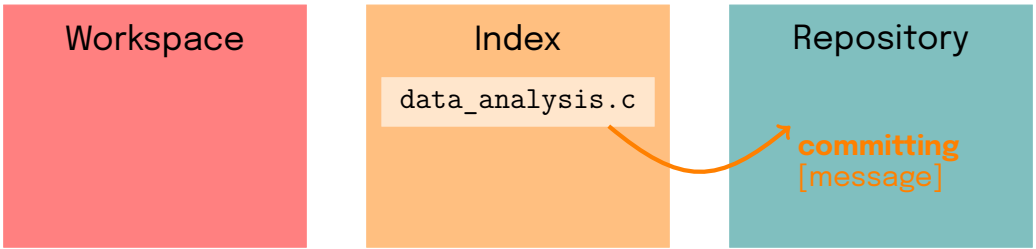
```
git commit -m 'Commit message'
```

Question

Which commit message do you think is better?

- ”Implemented list sorting method using quicksort algorithm.”
- ”Updated sorting files.”

```
git commit -m 'Commit message'
```



Notes on Committing and Staging

- Staging is easily reversible. Committing is not. To remove a file from the staging area: `git restore --staged file_name`
- git commit adds files to your repository and is HARD to undo. (`git rm` doesn't remove from past commits). Commandment: don't commit large files. (e.g. big datasets)
- Without `-m 'message'`, a text editor opens to write a more detailed description after the first line.
- `git commit --amend` edits the last commit without creating a new one and keeps the message. Useful for small fixes (like forgetting a file).

Essential Commands - git status

The `git status` command gives us an overview of the current state of our repository.
It shows which files have been modified, which are staged for the next commit, and whether there are pending operations.

Essential Commands - git log

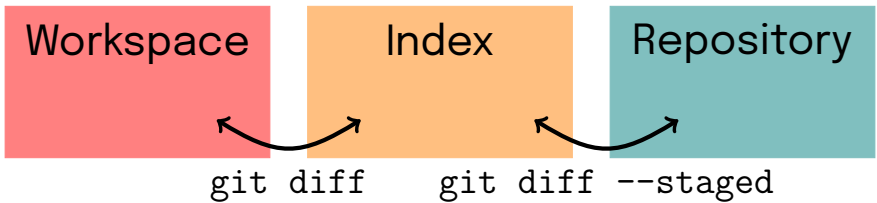
The `git log` command lets us view the commit history in our repository.
We can see who made what change, when it was made, and the associated commit messages.

Useful options:

- `--pretty=oneline`
- `-- file_name` shows only commits that modified a specific file
- `--grep blabla` shows only commits with *blabla* in the commit message

For more options: page 45 of the Pro Git book

Essential Commands - git diff



- `git diff`: shows the differences between the current state and the staging area.
- `git diff --staged`: shows differences between the staging area and the last commit.
- `git diff branch1 branch2`: shows differences between branches (we will return to this next week).

Other Notes and Useful Commands

You can create a text file named `.gitignore` listing file names you want git to IGNORE. You can use wildcards in this file.

Example:

- `*.pdf` will ignore all PDFs
- `data_*` will ignore all files starting with `data_`
- `data_*.c` will ignore all files starting with `data_` and ending with `.c`

Wildcards are much more powerful than just * and can be very helpful in large projects!

Other Notes and Useful Commands

- To restore a *modified* file to its state in the last commit: `git restore file` ← WARNING: this PERMANENTLY deletes current changes!
- Commit too often or too little? There is a balance.
- `git rm` removes files from both staging area and working directory. The `--cached` option keeps them on your computer but makes them *untracked*.
- `git mv` is used to rename/move files, tracking this with git.

Next week: branching, Github and collaboration

Giorgio Micaglio

Italian Association of Physics Students
Institute for Molecular Science - Ohmori Group

July 18, 2025