# Git and GitHub

> Git is the world's most popular VCS (Version Control System), a version control system is software that tracks and manage changes to a set of files over time. Linus Torvald is the creator of Git.

Version control systems such as Git generally allow:

- Revisiting earlier versions of the files
- Compare changes between versions (compare commits in the branch)
- Undo changes and a whole lot more

## What is Git?

Is the version control software that runs locally on your machine. You don't need to register for an account, you don't need internet to use it. You can use Git without ever touching GitHub. We can use the command line to interact with Git, as an additional option we can use graphical interfaces (GitKraken for example) that in the background just execute the same commands.

## What is GitHub?

Is a service that hosts Git repositories in the cloud and makes it easier to collaborate with other people. You do need to sign up for an account to use GitHub. It's an online place to share work that is done using Git.

## Some useful UNIX-based shell (like Git Bash) commands

- pwd : Print current directory. As the name implies, it prints the current directory or location in the folder system.

```
>> pwd
```

- cd : Change directory. It allows us to change directories (to move in the folder system).

```
>> cd ../skinet/API
```

- start : Open specified directory in the file explorer. The ".." are used to go back a level. Most of these commands receive a relative path such as in the following example (from the current directory we are at, go backwards one level, then enter "skinet", then enter "Core", and finally enter "Interfaces", that's the directory that will be shown in the file explorer):

```
>> start ../skinet/Core/Interfaces
```

- touch : Create a new file (yes, it has a weird name). You can pass multiple paths to the command in a single execution (following example creates 3 files in the previous directory because of the ".."):

```
>> touch ../appsettings.json ../BaseEntity.cs ../.gitignore
```

- mkdir : Make directory. As the name implies, it creates a single or multiple directories.

```
>> mkdir ../API/wwwroot ../Core/Entities ../Infrastructure/Data
```

- rm : Remove. Delete files or folders, to delete folders we must provide the "-rf" (recursive and force) flag.

Deleting files:

```
>> rm ../TestFile1.json ../TestFile2.json
```

Deleting folders (a single one in this case):

```
>> rm -rf ./MyFolder1
```

# Git concepts and usage

- Git repository

A git repository or "repo" is a workspace which files are being tracked and managed by Git. Every repository has its own history (series of commits).

NOTE: **Do not instantiate a new repo when you are already in an active repo** (check with "git status")

Git tracks the base folder that we initialized as a repo (where we executed "git init") and also all subsequent nested folders inside that base folder.

- Git commits

We can think of a commit in Git as a "checkpoint". Once we have made changes in the set of files of the repository, we can then group some or all of them together into a commit. We could also say that a commit is a "version" or state of the set of files in the branch (timeline).