https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later

very important if you're a graphic or web designer and want to keep every version of an image or layout (which you most certainly want to); a Version Control System(VCS) is a very wise thing to use; it allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more. It also means if you lose files, you can recover.

Many people’s version-control method of choice is to copy files into another directory (perhaps a time-stamped directory, if they’re clever). This approach is very common because it is so simple, but it is also incredibly error prone. It is easy to forget which directory you’re in and accidentally write to the wrong file or copy over files you don’t mean to.

To deal with this issue, programmers long ago developed local VCSs that had a simple database that kept all the changes to files under revision control.



Distributed Version Control Systems

This is where Distributed Version Control Systems (DVCSs) step in. In a DVCS (such as Git, Mercurial or Darcs), clients don’t just check out the latest snapshot of the files; rather, they fully mirror the repository, including its full history. Thus, if any server dies, and these systems were collaborating via that server, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.



https://www.freecodecamp.org/news/what-is-git-learn-git-version-control/

**What is GitHub?**

**GitHub** is a product that allows you to host your Git projects on a remote server somewhere (or in other words, in the cloud).

It's important to remember that GitHub is not Git. GitHub is just a hosting service. There are other companies who offer hosting services that do the same thing as GitHub, such as Bitbucket and GitLab.



You can either use Git by typing commands in the terminal or you can use a graphical user interface (GUI) such as Sourcetree or GitKraken.

If you choose the terminal, you will have to look up which Git commands you will need.

Luckily you don’t have to learn these by heart. Other than a handful of commands that you will use most often, the rest you can look up whenever you need them (this is what most developers do, even those with decades of experience). Git offers in-depth [documentation on their website](https://git-scm.com/docs).