General Workflow

Version control is a systematic approach to record changes made in a file, or set of files, over time. This allows you and your collaborators to track the history, see what changed, and recall specific versions later when needed. A typical procedure for using version control is as follows:

1. Create files - these may contain text, code or both.
2. Work on these files, by changing, deleting or adding new content.
3. Create a snapshot of the file status (also known as version) at this time.
4. Document what was changed in the version history of that file.

The snapshot process is often done manually for text or presentation documents (for instance by naming files with the suffixes `v01`, `v02` and so on). A description of the changes for each version is sometimes made via an external document like a spreadsheet. Finding the latest version can also be facilitated by putting old versions in a subfolder. This manual process is not very practical when a lot of files are changing, like when one creates code or work with data. In these cases, the use of a version control software is highly recommended.

This process of creating a snapshot is described differently in different version control software. For example, Git describes it as "a commit". Some systems call it "a time-point" or "a checkpoint"; and this is referred to as "saving your work" in other cases such as in Google docs or HackMD. The version history may be more or less informative.

As you keep saving your work by adding changes, you make more and more snapshots. You can think of these as saving versions of these files. If you need to go back to a previous version of a file because of a mistake, or if you changed your mind about a previous update, you can access the file in your preferred version, or return your entire project to a past state.

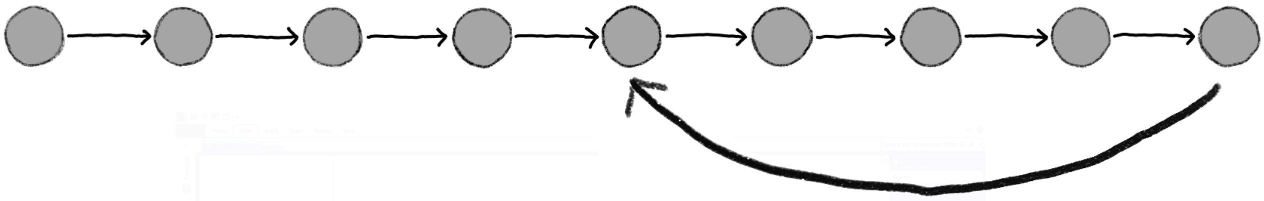

Fig. 25 Version history with a single branch.

In many version control systems (or in a special document if you do manual version control), you will be able to add a comment for each snapshot. Clear and concise comments make it easier to get an fast overview of the changes that were made in each versions. This ensures that it is easy to find what you are looking for when you need to go back to a past version. Your collaborators will thank you, but so will future versions of yourself.