**Undoing Git**

1. **Undoing “add”**

If we accidentally added some changes (staged), and then we want to undo this, we can use

**git reset <filename>** (exclude filename if we want to undo every files)

1. **Undoing “commit”**

**git reset HEAD~1**

what HEAD means is the pointer to the last commit (that we’ve just made). Therefore, if we want to go back to the previous commit, we use HEAD~1, which is 1 commit further. By doing this, we go back to when the changes are not staged (not added) and not committed.

1. **Accessing a past version**

If there are multiple commits, and we want to go back further than 1 commit, there is no short cut for that on Git. However, we can see all the previous commits made

**git log**

The changes are arranged in reversed chronological order (most recent first). Each commit will have a specific hash (commit <hash-is-here>), we can copy the commit hash and do:

**git reset <commit-hash>**

This goes back to the state of ALL FILES (if they are merged to create the changes).

1. **Delete all new changes, go back to past version**

If we want to get rid of all the changes at some point. All the changes will not only be made unstaged, but also completely removed.

**git reset –hard <commit-hash>**

1. **Undo git init for the whole folder** 🡪 have to delete the whole .git file

**rm -rf .git**