Notes:

* In Git, ***commit*** is (almost) synonymous with ***version***
* Commit saves changes to files – not the files themselves
* Your Project should not be on both GitHub and OneDrive/Google Drive

**Recovering an old file commit**

Go to Git tab 🡪 History (the clock icon):

* top box shows all your project commits
  + the SHA is a unique identifier for the project commit
  + for this example, assume SHA is 9a45f121
* bottom box shows:
  + files changed in this commit
  + changes to the files (green lines: stuff added, red lines: stuff deleted)

To recover file:

* Go to the file changes section to the file you want to recover
* Click ***View file @ 9a45f121*** (top-right corner of file changes)
* File is opened in editor in new windows: click ***Save As***

The changed file is now in your current Project and reflected in the Git tab and can be saved to the next Commit.

**Recovering an old Project commit** (the safest way – assume SHA is 9a45f121)

In ***Terminal*** Tab type:

**git revert --no-commit 9a45f121..HEAD**

This command will revert all the files back to the 9a45f121 commit of your project

* ***--no-commit*** means a new version of your project is not automatically created
* **9a45f121..HEAD** means reverse all changes between 9a45f121 and HEAD, which is always the last commit

This command reverse edits the files that changed between the 9a45f121 commit and the current commit (HEAD). After you execute the command, all the files that changed will be put in the ***Git*** tab – just like when you edit files. You can now create a new commit with the reverted files.

If there are files ***you do not want to revert***, you can right-click on the file and choose ***Revert.*** This will revert the revert – and put the file back to the original state.