

TD1/3: Git Branches

These exercises must be done by teams of 3-4 students.

Exercise 1: Clone a Git repository

1. Choose the repository created on GitHub or GitLab by one of your teammates, share its web URL.
2. **For the repository owner...** Ensure there is at least a README.md file, it appears on the project frontpage in the web UI.
3. Using only command-line in your Linux shell, clone it to a local repository.
4. **For the repository owner...** Give push rights to your teammates :
 - on *GitHub* got to "Settings", "Manage access", "Invite a collaborator" see <https://docs.github.com/en/github/setting-up-and-managing-your-github-user-account/inviting-collaborators-to-a-personal-repository>
 - on *GitLab* got to "Members", "Invite member" see <https://docs.gitlab.com/ee/user/project/members/>

hint : All Git commands have a -h flag to display the corresponding help.

hint : Unless you setup a SSH private key, you want to clone using the HTTPS address, not the SSH or CLI one.

Exercise 2: Push files to common repository

Using only the shell in your local repository :

1. Create a branch named after you.
2. Create a new text file named after you (with the content you want).
3. Commit this new file.
4. Push your branch to the remote repository.

Check in the web UI you see your branch (there is a button with 'main' as default). Ensure it contains your file with the same content you entered locally.

hint : Lost in your commits ? use `git log` or `git log --graph --oneline` to print the commit tree.

Exercise 3: Merge simple changes

Using only the shell :

1. Merge your branch into the 'main' branch.
2. Push your changes in the 'main' branch to the remote repository.

hint : You may have to merge or rebase on the changes from your teammates.

Check in the web UI you see your own file in the 'main' branch. Ensure it contains the same content you entered locally.

Exercise 4: Resolve merge conflicts

For the repository owner... In the web UI, make sure lines 2 to 6 of the README.md are not empty.

One person at a time, using only the shell :

1. Switch back to your own branch (not including the latest changes from the main branch).
2. Edit the lines 2 to 6 of the README.md file with a text you like (a poem, a quote, some clever code...). It can be any readable text, it may be incomplete, it must just take about 5 lines and be different from your teammates. It must start on line 2 to trigger conflicts between team members.
3. Commit this change.
4. Pull latest status from the remote repository 'main' branch into your local 'main' branch.
5. Merge your branch into the local 'main' branch.
6. If there are conflicts, *we want the paragraph to appear in alphabetical order in the final README.md file.*
7. Push your changes in the 'main' branch to the remote repository.

hint : You may edit the README.md file using nano or vim in shell, nano may be easier as it displays available commands at the bottom.

Check the README.md content in the web UI. After everyone in the group made their change and resolved conflicts, the file in 'main' branch should contain one paragraph per team member.

Exercise 5: Take latest changes from main in local branch

For the repository owner... In the web UI, add a line of text at the beginning of the README.md with the team members' names or aliases.

Using only the shell in your local repository :

1. Pull the latest changes in the 'main' branch, check the README.md is up-to-date (contains all the paragraphs and the new line).
2. Switch back to your own branch (not including the latest changes from the main branch).
3. Merge the changes from 'main' to your own branch.
4. Commit this change.

Check the README.md content in the web UI. After everyone in the group made their change and resolved conflicts, the file in 'main' branch should contain one paragraph per team member.

Exercise 6: Delete a branch

Using only the shell in your local repository :

1. Delete your branch on local repository.
2. Delete your branch on distant repository.

Using the web UI, ensure only the 'main' branch remains.

Exercise 7: Rebase interactively to have a clean history

Using only the shell in your local repository :

1. Pull the latest changes in the 'main' branch.
2. Create a new local branch named after you and switch to it.
3. Then **with a separate commit for each change** :
 - (a) Clear the whole file, removing all text.
 - (b) Add a title line "Git interactive rebase".
 - (c) Copy the first paragraph from <https://git-scm.com/book/en/v2/Git-Tools-Rewriting-History>.
 - (d) Add the second paragraph from the same page.
 - (e) Add the first and second paragraphs from the "*Changing Multiple Commit Messages*" section in the same page.
 - (f) Remove the second paragraph from your file.
 - (g) Add the missing title "*Changing Multiple Commit Messages*" on a line just before the two paragraphs your copied (before *To modify a commit that is farther back in your history...*).
 - (h) Add a final line with your name or alias.
4. Use interactive rebase to have a single commit with message "*Explain git interactive rebase.*".
5. Push your branch on the remote repository.

*hint : You can change Git default editor with command **git config --global core.editor "path to editor"**. By default it uses **nano**.*

Using the web UI, check the README.md content in your branch on the remote repository. Check the your branch history or graph, it should only contain the aggregated commit you pushed, not all the local commits. The 'main' branch should not have changed.

Exercise 8: Create and approve a Merge/Pull Request

In the web UI open your branch then

- In **GitLab** click on '*Create merge request*' to create a Merge Request to merge your branch into the 'main' branch.

- In **GitHub** click on '*pull request*' to create a Pull Request to merge your branch into the 'main' branch.
- Ask another team member to check there is a single commit and merge the Merge/Pull Request.

hint : As multiple people will try to change the same file, you may well have conflicts. In that case you have to rebase your branch on the latest state of the 'main' branch, resolving potential conflicts, and push it again. Reloading the merge/pull request webpage will update it.