

Versionning Class - Final Assignment

For this assignment, you must use the command line interface. The submission for each exercise should be a screenshot of your terminal. Each screenshot should be named as follow:

LASTNAME_Firstname_Git_Final_Assignment_Exercise_X

Any submission that would differ from that format won't be taken into account.

Only one screenshot per exercise!

Any "unnecessary" commands will be penalized, so I recommend doing your tests in a separate window.

Finally you'll compress your submissions into a zip or tar format with the following naming:

LASTNAME_Firstname_Git_Final_Assignment

Exercise 1: Create and version a repository using the CLI

1. Create a folder named `git_class_final_assignment`.
2. Initialize a Git repository inside it.
3. Add a file named `story.txt` and write the following note inside: "Hello, this is my submission for Git class!".
4. Stage and commit the file with a clear message.
5. Modify the file by adding a new line: "Once upon a time, in a galaxy far, far away..."
6. Check the modifications.

7. Stage and save the changes with a relevant commit message.

Exercise 2: Publish on a Git repository and read history

1. Create a repository on GitHub. Name it `versionning_course_final_assignment`.
2. Set the remote source of your local repository and call it `remote_origin`.
3. Push your code to the remote repository.
4. Share your repository link with a neighbor so they can review it.
5. Display the commit history.
6. Find the identifier of the first commit and display its content.

Exercise 3: Resolve a conflict in a shared file

1. Create a new branch named `add-crawling` and switch to it.
2. Modify `story.txt` by adding this new line: "Turmoil has engulfed the Galactic Republic."
3. Stage and commit the changes with a relevant comment.
4. Go back to the `main` branch and modify `story.txt` by adding the following line (in the same place): "It is a dark time for the Rebellion."
5. Stage and commit the changes with a relevant comment.

6. Merge the two branches and handle the conflict. Once the conflict is resolved, display the commit history of all branches in a single line, along with a graph.

Exercise 4: Resolve an advanced conflict

1. Create two branches for simultaneous changes. On the first branch, called `add-hero`, add the following text to the end of the file: "A brave hero named Aria was born under the brightest star." Stage and commit! On the second branch, called `add-villain`, add this line: "A shadowy villain emerged, spreading fear across the land." Stage and commit!
 2. Switch back to `add-hero` and perform a `rebase` onto `add-villain`. Resolve the conflict and mark it as resolved. Go back to the `main` branch and modify `story.txt` by adding the following line (in the same place): "A dark storm was brewing on the horizon."
 3. Merge the branches into `main` and handle the conflict. Once the conflict is resolved, display the commit history of all branches in a single line, along with a graph.
 4. Finally, merge into `main` and display the commit history of all branches in a single line, along with a graph.
-