Learn Git commands

The following tasks must be done only using command lines. You can not use Github or any graphical interface that handles git commands.

I. Create a repository

Saving your work using git (or any other version control system) is essential to keep a record of all changes made, and recover a backup of your code. Today, we will learn the most important commands you need to know to get started with Git.

1. Initialise git in a new directory using "**git init**"
2. **Add** a file for staging and create your first **commit**
3. Create a new commit, and try the following commands :
   status, **reset**, **revert**

II. Working on different branches

Using different branches is especially useful when working as a team. Developers can commit their work on their own branches, thus avoiding conflicts in the code. In this chapter, we will learn how to create different branches and manage your git repository as a tree structure.

1. Create a new branch called "develop"
2. Run the **branch** command to view all your current branches
3. Edit a file in the repository, add it, then commit the changes on "develop"
   *Note: you might need to use the **checkout** command*
4. From your main branch (also called master), **merge** the changes committed on "develop"
5. Run "**git log --graph**" to view your commit history

III. Resolving conflicts

When merging branches together, conflicts can happen between different commits. In this chapter, we will learn how to handle them and resolve them

1. Edit a file in your main branch, then commit it
2. Edit the same file with *different* changes, and commit it on “develop”
3. Try to merge the develop branch into the main branch. Git should now detect **conflicts.**
4. Edit the conflicting files in your IDE
5. Commit the changes the solve the conflicts
6. Check that the merge appears in your commit history

IV. Practice !

Practice everything we learned, and discover new commands with this game : https://ohmygit.org/