Scenario 1: Creating and Committing

- Participants create a new directory on their computers.
- Inside the directory, they create a text file and add some content.
- They initialize a new Git repository in the directory.
- Participants add the text file to the staging area and commit the changes.
- They modify the content of the file, stage it, and commit again.
- Participants use "git log" to view their commit history.

Scenario 2: Basic Branching and Merging

- Participants clone a sample repository from a remote source.
- They create a new branch called "feature/add-about-page."
- Inside this branch, they modify an existing HTML file to add an "About" page.
- Participants commit their changes on the feature branch.
- They switch back to the main branch and create a new branch called "bugfix/fix-typos."
- Participants correct some typos in the existing content and commit.
- They merge the "bugfix/fix-typos" branch into the main branch.
- Finally, participants switch to the "feature/add-about-page" branch and merge the main branch into it to integrate any changes made.

Scenario 3: Resolving Conflicts

- Participants are provided with a repository and are asked to clone it.
- They are given a file to modify and told to commit the change.
- Meanwhile, an instructor makes changes to the same file on the remote repository.
- Participants try to pull the changes from the remote repository and encounter a conflict.
- They open the conflicting file, resolve the conflict manually, and commit the resolved version.
- Participants push their changes to the remote repository.

Scenario 4: Collaborating with Remote Repositories

- Participants clone a repository from a remote source.
- They create a new branch named after their username.
- Inside this branch, they modify a text file by adding their name.
- Participants commit the changes and push the branch to the remote repository.
- They open a pull request to merge their branch into the main branch.
- Instructors simulate a review process, providing feedback and requesting changes.
- Participants update their pull request based on the feedback and push the changes.
- The pull request is eventually merged into the main branch.

Scenario 5: Reverting Changes

- Participants clone a repository with multiple commits.
- They identify a commit where an unwanted change was introduced.
- Using the commit hash, they use "git revert" to create a new commit that undoes the changes introduced by the identified commit.
- Participants verify that the unwanted change is indeed reverted.
- These beginner-level scenarios provide hands-on experience with essential Git concepts such as creating repositories, making commits, branching, merging, resolving conflicts, collaborating on remote repositories, and reverting changes. They are designed to build a solid foundation in Git for newcomers to version control.

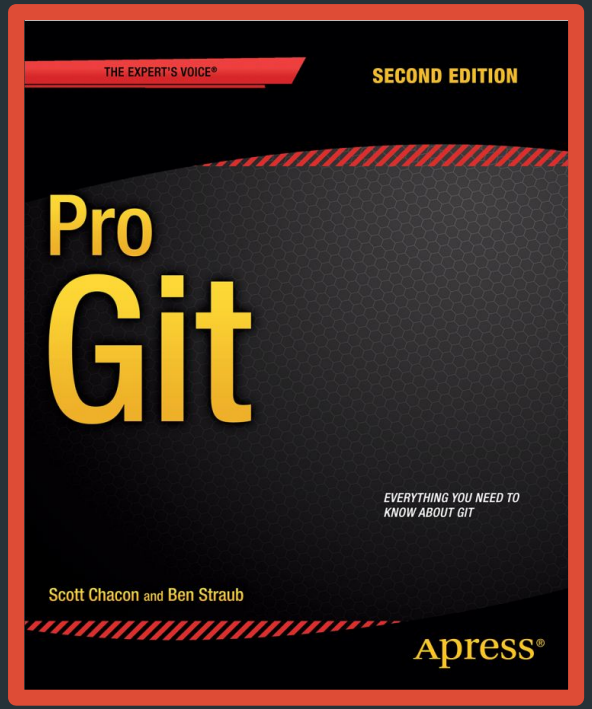
Lab 1 - Bouns

- Create a new project on your local machine, then push it to your remote repo.
- Create two branches (dev & test) then create one file on each branch, and push this changes to the remote repo.
- Merge this changes on Master branch and then push it to your remote master branch.
- Tell me how to remove them locally and remotely.
- Send an invitation to me (mahmoudhelmy31@gmail.com).

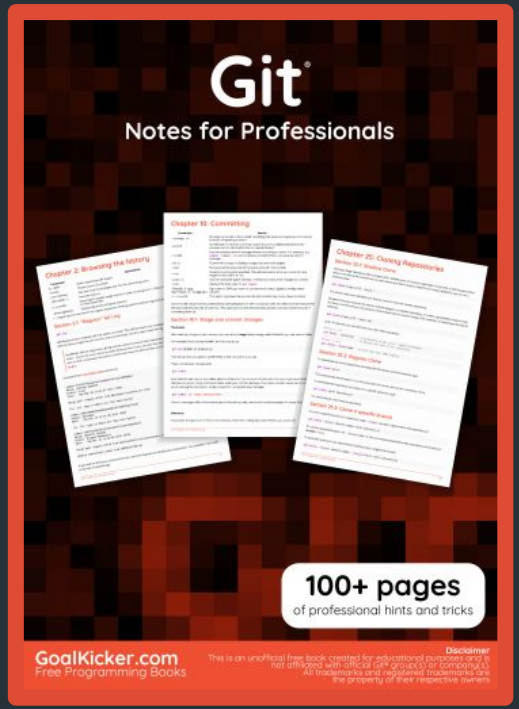
Lab 2 - Bouns

- Create an annotated tag with tagname (v1.7) .
- Push it to the remote repository.
- Tell me how to list tags.
- Tell me how to delete tag locally and remotely.
- Add an image in the README.md file.

RESOURCES



[Pro Git - Second Edition]



[Git - Notes for Professionals]

Thanks!

Do you have any questions?