



Git / GitHub Workshop-1

Clarusway



Subject: Git Operations

Learning Goals

- Practice using the Git commands.

Introduction

- We've covered some basic Git concepts, but now it's time to put the concepts in to practice. We'll start with Git commands.

Code Along

Part 1 - Create a local repository

1. Open the terminal (Git Bash for Windows user)
- Go to Desktop and create a directory named "my-github" if you do not have already. And, go to "my-github" directory.

- Create another folder named "git-workshop" in the "my-github" folder and go to "git-workshop" directory.

2. Git configuration (if you have already done, skip this part)

- Configure git with our name and email. This is to identify who has done what on git and github.

- Check the setting

3. Create a local repository

- We can do that by running the "init" command.

- Check the if ".git" folder is created.

4. Check your default branch name and If your branch name is "master", change it to "main" then check the branch again.

5. Create a file named "file1.txt"

- check the status of the project folder

6. Stage **file1.txt** and check the status of the project folder again.

7. Store it to the local repository.

8. Using Vim editor, create a file named **test2.txt**

9. Stage **file2.txt** and check the status of the project folder.

10. Unstage **file2.txt**

11. check the status of the directory

12. Store the changes to the local repository

13. List the commits

14. switch to the first commit

15. switch to the last commit.

Part 2 - Working with branches

16. Create a file named **test.txt**

17. Create a new branch named "new-feature-1".

- See branches

- Switch to new-feature-1

- List the files and check the status of the working directory

- Make some changes in the test.txt file, and check the status

- Store the changes to the repo and check the status

- Add another line to test.txt and store it to the local repo.

- Switch the main branch and see the content of the test.txt

- Merge new-feature-1 branch to main branch.

18. Create a new branch named new-feature-2 and switch to it.

- Create a new file named test2.txt, add a line in it and store the changes to repo.

- Switch the main branch again.

- Create a new file test3.txt and send the changes to local repo.

- Open the file named test2.txt, add a line in it and store the changes to repo.

- merge main branch with new-feature-2

19. RESOLVE THE CONFLICT

- edit the file.
- then commit it.

20. Delete the branches named new-feature-1 and new-feature-2

- List the all branches

😊 **Thanks for Attending** 📝

