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.

```
mkdir git-workshop
cd git-workshop
```

2. Git configuration

- 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. If your branch name is "master", change it to "main".

```

```

Part 2 - Create and connect a remote repository

5. Create a remote repository on GitHub

- Go to your GitHub account and create a repository named "git-workshop".
    - Write a description for your repo
    - select Public
    - add a README.MD file

6. Go to terminal

- Check the connected remote repositories. The 'git remote -v' lists all currently configured remote repositories, which at this point is none.

```

```

- connect to remote repository

```

```

- Verify the new connection

```

```

7. Create a file named "file1.txt"

- check the status of the project folder

```

```

- store the change in the local repo

```

```

8. upload the changes to the remote repo

```

```

- check the files on the github repo.

Part 3 - Cloning a Remote Repo

8. Create a new remote repo named "git-workshop-1" in GitHub.
9. Clone the remote repo

- go the terminal
- clone the "git-workshop-1"

```

```

- Check the files in the "git-workshop-1" and see the README.MD and .git file.

```
ls -a
```

10. Create a file named **test1.txt**

```

```

11. Stage **test1.txt**

```
git add test1.txt
```

12. Store it to the local repository.

```

```

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

```

```

14. Stage **test2.txt**

15. Unstage **test2.txt**

16. check the status of the directory

17. Store the changes to the local repeository

17. List the commits

18. switch to the first commit

19. switch to the last commit.

20. Send the changes to the remote repository

21. Go and check the remote repository, you will see the new files

☺ **Thanks for Attending** ✍

Clarusway