Step-by-Step Guide to Using Git

Introduction

This guide explains how to use Git for programming projects. We will cover the steps to clone a repository, create directories and files for specific exercises, add these files to the staging area, commit them, and finally push them to the remote repository. Each step will be accompanied by detailed explanations to help you understand the role of each command.

1. Clone the Repository

Clone the repository to your workspace:

```
git clone https://your_git_link.git
cd repository_name
```

Explanation:
`git clone` : This command creates a local copy of a remote Git repository. It allows you to work on the project on your local machine.
`cd repository_name` : Changes the current directory to enter the cloned project's directory.

2. Create Directories and Files for Exercises

You can now create directories for your exercises inside the cloned repository. For example, to create directories `ex01` and `ex02`:

```
mkdir ex01
mkdir ex02
```

Next, you can create files in these directories. For example:

```
touch ex01/main.c
touch ex02/README.md
```

If you already have files for your exercises, you can move them into the appropriate directories inside the cloned repository:

```
mv /path/to/your/files/* ex01/
mv /path/to/your/other/files/* ex02/
```

3. Add Files to the Staging Area

Add a specific file to the staging area:

```
git add ex01/main.c
```

Explanation:
`git add` : This command adds files to the staging area. The staging area is a space where you prepare files before committing them. It allows you to group the changes you want to commit together.

Add all files in a directory to the staging area:

```
git add ex02/
```

Add all modified files to the staging area:

```
git add .
```

Explanation:
`git add .` : This command adds all modified files in the current directory and its subdirectories to the staging area.

4. Check the Repository Status

Check the files added to the staging area:

```
git status
```

Explanation:
`git status` : This command is used after `git add` to check which files have been added to the staging area and are ready to be committed.

5. Commit the Changes

Commit the added files with a descriptive message:

```
git commit -m "Add exercises ex01 and ex02"
```

Explanation:
`git commit` : This command records the changes in the repository's history with a descriptive message. The text after `-m` is the commit message, which should describe the changes made.

6. Push the Changes to the Remote Repository

Push the changes to the remote repository:

```
git push origin main
```

Explanation:
`git push` : This command sends the local commits to the remote repository on GitHub. `origin` is the default name for the remote repository, and `main` is the branch to which you are pushing the changes.

Complete Example

Here is a complete example following the steps above:

```
git clone https://your_git_link.git
cd repository_name
mkdir ex01 ex02
touch ex01/main.c ex02/README.md
git add ex01/main.c ex02/
git status
git commit -m "Add exercises ex01 and ex02"
git push origin main
```