### **1. cd C:**

* **Purpose**: This command is used to change the current directory to the C: drive in Windows.
* **Explanation**: cd stands for "change directory". The command cd C: navigates to the root of the C: drive.

### **2. cd github**

* **Purpose**: Changes the current directory to the github folder within the C: drive.
* **Explanation**: cd github navigates to the github directory that is located on the root of the C: drive.

### **3. git init**

* **Purpose**: Initializes a new Git repository in the current directory (/c/github).
* **Explanation**: This command creates a .git directory in the current folder, turning it into a Git repository. The message Reinitialized existing Git repository means the current folder already had an initialized repository.

### **4. git add gitnew1.txt**

* **Purpose**: Stages the file gitnew1.txt to be added to the next commit.
* **Explanation**: The git add command tells Git to start tracking the file, preparing it to be included in the next commit. If the file was not yet created or wasn't found, Git would show an error like fatal: pathspec 'gitnew1.txt' did not match any files.

### **5. git commit -m "first commit"**

* **Purpose**: Commits the staged changes (the gitnew1.txt file) with the commit message "first commit".
* **Explanation**: This command takes all the staged changes and saves them in the Git history with a message. The -m flag is used to specify the commit message. The message "first commit" is simply a label for the changes made at this point in time.

### **6. git branch -M main**

* **Purpose**: Renames the current branch to main.
* **Explanation**: The -M option is used to rename a branch. In the earlier versions of Git, the default branch was often named master, but many repositories now use main as the default branch name. This command renames the current branch (which is main in this case) to main. If the branch was already called main, it has no effect.

### **7. git remote add origin**

### **Purpose**: Adds a new remote named origin and associates it with the provided GitHub repository URL.

* **Explanation**: The git remote add command is used to associate your local Git repository with a remote repository, in this case, on GitHub. This allows you to push or pull changes between your local machine and the remote repository.

### **8. git push -u origin main**

* **Purpose**: Pushes your local commits to the remote repository (origin), specifically to the main branch.
* **Explanation**: The git push command sends your local commits to the remote repository. The -u flag tells Git to set up a tracking relationship between the local main branch and the remote main branch. After this, you can use git push or git pull without specifying the remote or branch name, as Git will know which remote branch to use.

**9. git config --global user.name "Your Name"**

* **Purpose**: This sets the global username for all Git repositories on your local machine.
* **Explanation**: The user.name value is used to identify the author of commits. It will appear in the commit history (log) whenever you make a commit. The --global flag means that this setting will be applied across all repositories on your system, unless overridden in a specific repository.

10. **git config --global user.email "email”**

* **Purpose**: This sets the global email address associated with your commits in all repositories.
* **Explanation**: The user.email is tied to your identity in the Git version control system. This email address is shown in commit logs and is important when pushing commits to remote repositories (like GitHub, GitLab, etc.), as it links your commits to your online profile. Again, the --global flag applies this setting across all repositories, unless overridden in a specific repository.