GitHub Commands

1. Installation and Setup of Git Command Line:

* Download Git from the official website (https://git-scm.com/downloads) and install it on your system.
* Open a terminal or command prompt to verify the installation:

.................................................

git --version

.................................................

2. Creation of a new local Git Repository:

* Navigate to the desired directory where you want to create the repository.
* Initialize a new Git repository:

.................................................

git init

.................................................

3. Commit Files to Git locally:

* Add the files you want to include in the commit:

.................................................

git add <file1> <file2> ...

.................................................

* Commit the changes with a descriptive message:

.................................................

git commit -m "Commit message"

.................................................

4. Branches in Git:

* - List all branches:

.................................................

git branch

.................................................

* - Create a new branch:

.................................................

git branch <branch\_name>

.................................................

* - Switch to a different branch:

.................................................

git checkout <branch\_name>

.................................................

* - Merge branches (while on the destination branch):

.................................................

git merge <source\_branch>

.................................................

5. Create Tags in Git:

* - Create a lightweight tag:

.................................................

git tag <tag\_name>

.................................................

* - Create an annotated tag with a message:

.................................................

git tag -a <tag\_name> -m "Tag message"

.................................................

6. GitHub account:

* - Go to GitHub (https://github.com) and sign up for a new account.

7. Create a new GitHub repository:

* - Log in to your GitHub account.
* - Click on the "+" sign in the top-right corner and select "New repository".
* - Provide a repository name, optional description, and choose other settings as needed.
* - Click "Create repository" to create the new repository.

8. Push Git Local Repository to Remote GitHub repository:

* - Add the remote GitHub repository URL as the origin:

.................................................

git remote add origin <repository\_url>

.................................................

* - Push the local repository to the remote repository:

.................................................

git push -u origin <branch\_name>

.................................................

9. Create a GitHub Readme file:

* - Create a new file named "README.md" in the root directory of your local Git repository.
* - Add content to the README file using Markdown syntax.
* - Save the file and commit it to your Git repository.

10. Create a Git tag and GitHub Release:

* - Create a tag locally:

.................................................

git tag -a <tag\_name> -m "Tag message"

.................................................

* - Push the tag to the remote GitHub repository:

.................................................

git push origin <tag\_name>

.................................................

* - Go to the GitHub repository page and click on "Releases".
* - Click "Create a new release" and provide the tag name, release title, and description.
* - Publish the release.

11. Creating Git submodules:

* - Navigate to the root directory of the existing Git repository where you want to add a submodule.
* - Add the Git submodule using its repository URL:

.................................................

git submodule add <submodule\_repository\_url>

.................................................

* - Commit and push the changes to the remote repository.

Troubleshooting

When working with GitHub, having troubleshooting skills can help you overcome common issues and ensure a smooth workflow. Here are some troubleshooting skills and commands you may find useful:

1. Checking Git Configuration:

* - To view your Git configuration settings:

.................................................

git config --list

.................................................

* - Verify your name and email settings:

.................................................

git config user.name

git config user.email

.................................................

2. Checking Git Status:

* - To see the current state of your local repository and any modifications:

.................................................

git status

.................................................

3. Handling Merge Conflicts: (x)

* - When encountering a merge conflict, open the conflicting file(s) and resolve the conflicts manually.
* - After resolving conflicts, add the modified files and commit the changes.

4. Undoing Changes:

* - To discard changes in a file and revert it back to the last commit:

.................................................

git checkout -- <file>

.................................................

* - To revert a specific commit (creating a new commit that undoes the changes):

.................................................

git revert <commit\_hash>

.................................................

5. Syncing with Remote Repository:

* - If your local branch is behind the remote branch, fetch and merge the changes:

.................................................

git fetch

git merge origin/<branch\_name>

.................................................

* - If your local branch is ahead of the remote branch, push your changes:

.................................................

git push origin <branch\_name>

.................................................

6. Resolving Authentication Issues:

* - If you encounter authentication problems, verify your remote URL and credentials.
* - Update the remote URL if needed:

.................................................

git remote set-url origin <repository\_url>

.................................................

7. Checking Remote Repository Information:

* - To view remote repositories:

.................................................

git remote -v

.................................................

* - Check the upstream repository for a branch:

.................................................

git branch -vv

.................................................

8. Cleaning Up: \*\*\*\*

* - Remove untracked files and directories:

.................................................

git clean -df

.................................................

* - Discard all local changes and revert to the last commit:

.................................................

git reset --hard HEAD

.................................................

9. Reviewing Git Logs:

* - To view commit history:

.................................................

git log

.................................................

* - Display a compact summary of recent commits:

.................................................

git log --oneline

.................................................

10. Git Help:

* - Access the Git documentation and command reference:

.................................................

git help

git <command> --help

.................................................