

Day1_assignment

Task 1: Initialize a Git Repository

To start tracking your project with Git:

```
git init
```

-> This command creates a hidden .git folder to track changes in your project.

Task 2: Create a New Branch for Feature Development

To create and move to a new branch:

```
git checkout -b contact-form
```

-> This creates a branch named 'contact-form' and switches to it.

Task 3: Commit Code with Message

After editing your code:

```
git add .
```

```
git commit -m "Added contact form to website"
```

-> 'git add .' adds all changed files.

-> 'git commit -m' saves your changes with a message.

Task 4: Pull Before Push to Avoid Conflicts

Before pushing your code:

```
git pull origin main
```

-> This fetches the latest code from GitHub and merges it with your local version.

Task 5: Resolve Merge Conflicts and Push

If you get a conflict, fix it manually. Then:

```
git add style.css
```

```
git commit -m "Resolved conflict in style.css"
```

```
git push origin main
```

Task 6: Initializing and Committing a Project

Commands to start a new Git project:

```
git init
```

```
git add .
```

```
git commit -m "Initial commit"
```

Task 7: Connecting Local Repository to GitHub

Suppose your GitHub repo URL is <https://github.com/yourname/project.git>:

```
git remote add origin https://github.com/yourname/project.git
```

```
git push -u origin main
```

Task 8: Creating and Switching to a New Feature Branch

For a new login feature:

```
git checkout -b login-feature
```

Task 9: Temporarily Saving Work in Progress

To save unfinished work temporarily:

```
git stash
```

-> You can bring it back later using:

```
git stash apply
```

Task 10: Merging Feature Branch into Main Branch

First, switch to the main branch:

```
git checkout main
```

```
git merge login-feature
```

-> After merging, push it to GitHub:

```
git push origin main
```