

## **Task 1: Repository Setup**

### **1. GitHub Repository Creation:**

#### **- Log in to your GitHub account.**

Logging into GitHub is a simple process if you already have an account. Here are the steps to login into GitHub:

##### **1. Open GitHub Website:**

- Open your web browser and navigate to the GitHub website: [github.com](https://github.com).

##### **2. Enter Your Username:**

- On the GitHub homepage, you will see a "Sign in" button at the top right corner of the page.
- Click on "Sign in".

##### **3. Enter Your Password:**

- You will be directed to the GitHub login page where you need to enter your username (or email address associated with your GitHub account) and password.

##### **4. Authenticate:**

- After entering your credentials, click on the "Sign in" button.

##### **5. Two-Factor Authentication (if enabled):**

- If you have two-factor authentication (2FA) enabled for your GitHub account, you will be prompted to enter a verification code that you receive on your mobile device or through another method you've set up.

##### **6. Access Your Account:**

- Once authenticated, GitHub will redirect you to your account dashboard, where you can see your repositories, organizations, and other settings.

#### **- Create a new repository on GitHub (let's call it "PLPBasicGitAssignment").**

#### **- Initialize it with a README file.**

### **1. Navigate to Your Repositories:**

- Once logged in, click on your profile icon at the top right corner of the page.
- Select **'Your repositories'** from the dropdown menu. This will take you to a page listing all your repositories.

## 2. Create a New Repository:

- On your repositories page, you'll find a green button labeled **'New'**. Click on it.

## 3. Set up the New Repository:

- You will now see a page where you can set up your new repository.
- Enter "PLPBasicGitAssignment" as the 'Repository name'.
- Optionally, you can add a **'Description'** to briefly explain the purpose of this repository (e.g., "Assignment repository for learning basic Git operations").
- Choose the Visibility of the repository. For example, you can make it **'Public'** or **'Private'** depending on your needs.

## 4. Initialize with a README (Optional):

- You have the option to initialize the repository with a README file. This is useful if you want to add some initial documentation or information about your project.
- Select the checkbox labeled **'Initialize this repository with a README'** if you want to include a README file.

## 5. Create the Repository:

- After filling out the necessary details, click on the green button **'Create repository'**. GitHub will then create your new repository with the specified settings.

## 6. Repository Created:

- Once the repository is created, you will be redirected to the main page of your new repository "PLPBasicGitAssignment" on GitHub.

First change where the local repository will be pushed to from the master branch to the main branch using the following code:

```
git config --global init.defaultBranch main
```

## Task 2: Local Setup

### 2. Local Folder Setup:

- Create a new folder on your local machine (e.g., "PLPBasicGitAssignment").

### Using Command Line:

You can create the folder using the command line:

- **Windows Command Prompt:**

```
mkdir PLPBasicGitAssignment
```

- Open a terminal or command prompt and navigate to the created folder.

### Navigating to the "PLPBasicGitAssignment" Folder:

- Use the `cd` (change directory) command to navigate to the folder you created. Replace `<path>` with the actual path to your "PLPBasicGitAssignment" folder.

```
cd PLPBasicGitAssignment
```

### 3. Git Initialization:

- Initialize a new Git repository in your local folder.

- Once navigate to the "PLPBasicGitAssignment" folder, initialize a new Git repository using the following command:

```
git init
```

This command initializes an empty Git repository in the current directory.

### 4. Connecting to GitHub:

- Link your local repository to the GitHub repository you created in Task 1.

Link your local repository to the GitHub repository as a remote using the following code:

```
Git remote add origin  
https://github.com/exampleuser/PLPBasicGitAssignment.git
```

```
git remote add origin  
https://github.com/Dorisombongi/PLPBasicGitAssignment
```

You can verify the remote repository by using the following code:

```
git remote -v
```

### Task 3: Making Changes

#### 5. Create a File:

- Inside your local folder, create a new text file (e.g., `hello.txt`).
- Add a simple text message (e.g., "Hello, Git!").
  - Use Git terminal to create `hello.txt` and add content `Hello, Git`. Here's how you can do it via command line:

```
echo "Hello, Git!" > hello.txt
```

#### 6. Committing Changes:

- Stage the changes.

Use the following command to stage `hello.txt`:

```
git add hello.txt
```

This command adds `hello.txt` to the staging area in Git. The staging area is where you prepare a snapshot of your changes before committing them to the repository.

Verify Staging

You can verify which files are staged using:

```
git status
```

- Commit the changes.

Use the following command to commit the staged changes with a commit message:

```
git commit -m "Add hello.txt with a greeting"
```

Use `git log` to verify.

## **Task 4: Pushing to GitHub**

### **7. Pushing to GitHub:**

- **Push the committed changes to your GitHub repository.**

Push the committed changes to GitHub using the following code:

```
git push origin main
```

## **Task 5: Verification**

### **8. Verify on GitHub:**

- Go to your GitHub repository page "PLPBasicGitAssignment".
- Refresh the page, and you should see hello.txt with the content "Hello, Git!" that you added and committed locally.