# Tutorial: Using Git and GitHub

## 1. Setting Up Git and GitHub

**1.1 Installing Git**

* **Windows:**
  1. Download Git from [git-scm.com](https://git-scm.com/download/win).
  2. Run the installer and follow the instructions.
* **macOS:**
  1. Install Homebrew if not already installed: /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
  2. Install Git: brew install git
* **Linux:**
  1. Use your package manager. For example, on Ubuntu: sudo apt-get install git

**1.2 Configuring Git**

Open your terminal and set your username and email:

**git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"**

**1.3 Creating a GitHub Account**

1. Go to [github.com](https://github.com/).
2. Sign up for a free account.
3. [Generating a new SSH key and adding it to the ssh-agent](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)
4. [Connect to GitHub with SSH](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account)

## 2. Creating a Repository

**2.1 Initializing a Local Repository**

1. Open your terminal.
2. Navigate to your project directory:
   1. My practice is to create a projects directory in my home directory (\Projects)
   2. I then create project directories under \Projects.

For example:

**mkdir \Projects\CaesarCipher**

**cd \Projects\CaesarCipher**

1. Initialize Git:

**git init**

**2.2 Creating a Repository on GitHub**

1. Go to your GitHub profile.
2. Click the **+** icon in the upper-right corner and select **New repository**.
3. Fill in the repository name and description.
4. Choose to make it public or private.
5. Click **Create repository**.

**2.3 Connecting Local Repository to GitHub**

1. In your terminal, add the remote repository:

**git remote add origin** [**git@github.com:<your**](mailto:git@github.com:%3cyour) **username>/<your-repo.git>**

**git branch -M main**

**git push -u origin main**

## 3. Basic Git Commands

**3.1 Checking Repository Status**

**git status**

**3.2 Adding Files to Staging Area**

**git add filename  
# or to add all changes  
git add .**

**3.3 Committing Changes**

**git commit -m "Your commit message"**

**3.4 Pushing Changes to GitHub**

**git push origin main  
# If using the master branch  
# git push origin master**

**3.5 Pulling Changes from GitHub**

**git pull origin main  
# If using the master branch  
# git pull origin master**

## 4. Branching and Merging

**4.1 Creating and Switching to a New Branch**

**git checkout -b new-branch-name**

**4.2 Switching Between Branches**

git checkout branch-name

**4.3 Merging Branches**

1. Switch to the branch you want to merge into (usually main or master):

git checkout main

1. Merge the branch:

**git merge new-branch-name**

**4.4 Deleting a Branch**

**git branch -d branch-name**

## 5. Collaborating with Others

**5.1 Forking a Repository**

1. Go to the repository on GitHub you want to fork.
2. Click the **Fork** button in the upper-right corner.

**5.2 Cloning a Repository**

**git clone https://github.com/username/repository.git**

**5.3 Submitting a Pull Request**

1. Make changes in your forked repository.
2. Push changes to your GitHub fork.
3. Go to the original repository and click **New pull request**.
4. Compare changes and create the pull request.

## 6. Using GitHub for Project Management

**6.1 Creating Issues**

1. Go to your repository on GitHub.
2. Click the **Issues** tab.
3. Click **New issue** and fill in the details.

**6.2 Using Project Boards**

1. Go to your repository on GitHub.
2. Click the **Projects** tab.
3. Click **New project** and set up your board.

**6.3 Writing Documentation**

1. Create a README.md file in your repository root.
2. Write documentation using Markdown.

## 7. Advanced Topics

**7.1 Using Tags**

1. Create a tag:

**git tag -a v1.0 -m "Version 1.0"**

1. Push tags to GitHub:

**git push origin --tags**

**7.2 Setting Up Continuous Integration (CI)**

1. Choose a CI tool (e.g., GitHub Actions, Travis CI).
2. Follow the tool's documentation to set up a .yml file in your repository for automated testing and deployment.