Git Commands 101

Getting Started

First, we need to install Git in our system. To install Git you can go to the official Git website and download the installer from there. Once installed, go to your command prompt in your system and check if Git is installed perfectly using:

```
git –version
```

Now we need to configure our username and email address globally. We can do using the following command in cmd:

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

These are used to identify who makes the commit.

Initializing a Repo

To start using Git in a project, you need to **initialize a Git repository**. This creates a hidden `.git` directory in your project folder, which Git uses to track changes.

- `.git init` : Initializes a new Git repository in our current directory.

```
cd my-project
git init
```

Add, Commit, Push

This is the core of the Git workflow. You make changes to your files, stage them, commit them, and then push them to a remote repository.

- `git status`: Shows the status of your working directory and staging area. It tells you which files are untracked, modified, or staged.

  ```
  git status
  ```

- `git add`: Adds new or modified files to the **staging area**. The staging area acts as a temporary holding space before you commit.

```
git add . # Adds all changes in the current directory
git add file.txt # Adds a specific file
```

- `git commit`: Records the staged changes to the repository's history. Each commit is a snapshot of your project at a specific point in time. A good commit message is crucial.

  ```
  git commit -m "Your descriptive commit message here"
  ```

- `git push`: Uploads your local commits to a **remote repository** (like GitHub, GitLab, or Bitbucket). You usually push to the **`main`** or **`master`** branch.

  ```
  git push origin main
  ```

Remote Repos

Remote repositories are versions of your project hosted on the internet or a network.

- `git clone <repository-url>`: Creates a local copy of an existing remote repository.

  ```
  git clone https://github.com/user/repo.git
  ```

- `git remote -v`: Lists the remote repositories you're connected to.
- `git pull`: Fetches and downloads content from a remote repository and immediately updates your local working copy to match that content. It's essentially a combination of `git fetch` and `git merge`.

  ```
  git pull origin main
  ```

Viewing History and Undoing

Git maintains a detailed history of your project.

- `git log`: Shows the commit history. You can see who made changes, when, and the commit message.
- `git revert <commit-hash>`: Creates a new commit that undoes the changes of a specific commit. This is a safe way to undo changes as it doesn't rewrite history.
- `git reset --hard HEAD`: Discards all changes in your working directory and staging area, and reverts to the last commit. **Use with caution**, as it is a destructive command.