

## Git Rebase — One Simple Example with Commands

What is Git Rebase?

Git rebase is used to move or replay your commits on top of another branch. It helps maintain a clean and linear commit history.

---

### Step 1: Create Repository and Initial Commits

#### Commands:

```
git init rebase-demo
cd rebase-demo
```

```
echo "Line 1" > file.txt
git add file.txt
git commit -m "Commit A"
```

```
echo "Line 2" >> file.txt
git commit -am "Commit B"
```

---

### Step 2: Create Feature Branch

```
git checkout -b feature
echo "Feature work 1" >> file.txt
git commit -am "Commit D"
```

```
echo "Feature work 2" >> file.txt
git commit -am "Commit E"
```

---

### Step 3: Switch Back to Main and Add Commit

```
git checkout main
echo "Main work" >> file.txt
git commit -am "Commit C"
```

---

#### Step 4: Rebase Feature Onto Main

```
git checkout feature  
git rebase main
```

-----

#### Step 5: View History

```
git log --oneline --graph
```

-----

#### Before Rebase:

```
A -- B -- C (main)  
  \  
   D -- E (feature)
```

#### After Rebase:

```
A -- B -- C -- D' -- E' (feature)
```

-----

#### Key Points:

- Rebase creates a clean linear history
- Commits are replayed, not merged
- Do not rebase shared commits