# Create a folder called destruction

mkdir destruction

# Change into the destruction folder

cd destruction

# Initialize an empty git repository

git init

# Create a file called done.txt

touch done.txt

# Remove done.txt from the working directory

git rm --cached done.txt

# Create a file called stage\_me.txt

touch stage\_me.txt

# Add stage\_me.txt to the staging area

git add stage\_me.txt

# Move stage\_me.txt from the staging area to the working directory

git reset HEAD stage\_me.txt

# Add stage\_me.txt file to the staging area again

git add stage\_me.txt

# Remove stage\_me.txt from the staging area and the working directory

git rm --cached stage\_me.txt

# Create a new file called commit\_me.txt

touch commit\_me.txt

# Add commit\_me.txt to the staging area

git add commit\_me.txt

# Commit with the message "adding commit\_me.txt"

git commit -m "adding commit\_me.txt"

# Create another file called second.txt

touch second.txt

# Add second.txt to the staging area

git add second.txt

# Commit with the message "adding second.txt"

git commit -m "adding second.txt"

# Check out previous commits using git log --oneline

git log --oneline

# Using git reset, undo the previous commit and move changes back to the working directory

git reset HEAD~1

# Add second.txt again

git add second.txt

# Commit with the message "Trying to commit again"

git commit -m "Trying to commit again"

# Using git reset, undo the previous commit and move changes back to the staging area

git reset --soft HEAD~1

# Commit with the message "Trying to commit again and again"

git commit -m "Trying to commit again and again"

# Using git reset, undo the previous commit and discard changes from the working directory

git reset --hard HEAD~1