

Report :

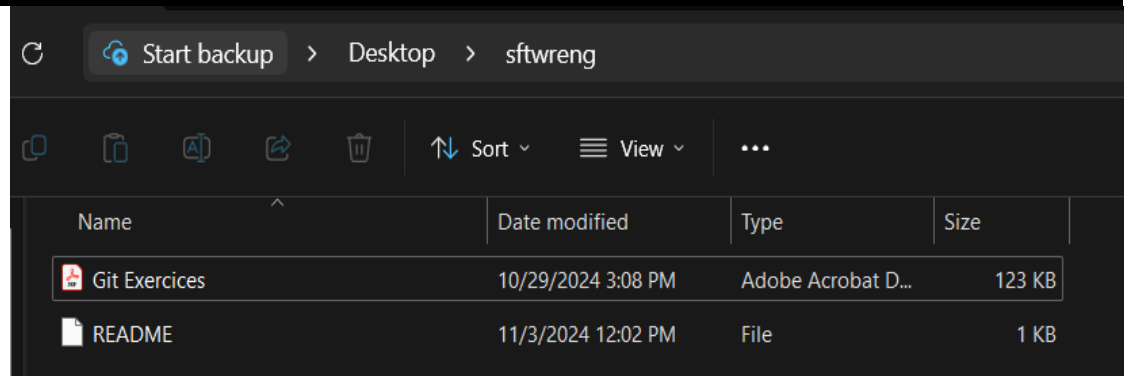
create and navigate to a new directory

- Run `git init` to initialize a Git repository, creating a `.git` folder for metadata and version history:

```
C:\Users\user\Desktop\sftwreng>git init
```

```
Initialized empty Git repository in C:/Users/user/Desktop/sftwreng/.git/
```

- `C:\Users\user\Desktop\sftwreng>echo. > README`



- Status command:

```
C:\Users\user\Desktop\sftwreng>git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
Git Exercices.pdf
```

```
README
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
C:\Users\user\Desktop\sftwreng>
```

```

C:\Users\user\Desktop\sftwreng>git add README

C:\Users\user\Desktop\sftwreng>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>

```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Add README"
```

```
[master (root-commit) 8766b96] Add README
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 README
```

- C:\Users\user\Desktop\sftwreng>mkdir src

```
C:\Users\user\Desktop\sftwreng>cd src
```

```
C:\Users\user\Desktop\sftwreng\src>echo. > file1.py
```

```
C:\Users\user\Desktop\sftwreng\src>echo. > file2.py
```

```
C:\Users\user\Desktop\sftwreng\src>cd ..
```

```

C:\Users\user\Desktop\sftwreng>git add src

C:\Users\user\Desktop\sftwreng>git status
On branch master

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/file1.py
        new file:   src/file2.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>

```

- C:\Users\user\Desktop\sftwreng>git commit -m "Add src directory with file1.py and file2.py"

```
[master 9cf1b7a] Add src directory with file1.py and file2.py
```

2 files changed, 2 insertions(+)

create mode 100644 src/file1.py

create mode 100644 src/file2.py

```
C:\Users\user\Desktop\sftwreng>notepad src\file1.py

C:\Users\user\Desktop\sftwreng>git diff src/file1.py
diff --git a/src/file1.py b/src/file1.py
index 8d1c8b6..2cc36ed 100644
--- a/src/file1.py
+++ b/src/file1.py
@@ -1,2 @@
-
+
+x=2
+print(x)
```

- C:\Users\user\Desktop\sftwreng>

```
C:\Users\user\Desktop\sftwreng>git add src/file1.py

C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/file1.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>git diff --cached
diff --git a/src/file1.py b/src/file1.py
index 8d1c8b6..2cc36ed 100644
--- a/src/file1.py
+++ b/src/file1.py
@@ -1,2 @@
-
+
+x=2
+print(x)
```

- No output when running git diff again because no changes has been made.

```
C:\Users\user\Desktop\sftwreng>git diff --cached
```

```
C:\Users\user\Desktop\sftwreng>
```

File Edit View

```
x=2
print(x)
y=3
print(y)|
```

-

```
C:\Users\user\Desktop\sftwreng>notepad src\file1.py

C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/file1.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

no changes added to commit (use "git add" and/or "git commit -a")
```

```
C:\Users\user\Desktop\sftwreng>git diff
diff --git a/src/file1.py b/src/file1.py
index 2cc36ed..dbbdd09 100644
--- a/src/file1.py
+++ b/src/file1.py
@@ -1,2+1,4 @@
x=2
-print(x)
+print(x)
+y=3
+print(y)
```

```
C:\Users\user\Desktop\sftwreng>
```

Changes in file1.py:

The line `print(x)` remains unchanged, as indicated by the `-` (it was there in the previous version).

The line `y=3` is a new addition, as shown by the `+`.

The line `print(y)` is also a new addition.

- C:\Users\user\Desktop\sftwreng>git commit -m "Add additional changes to file1.py"

```

C:\Users\user\Desktop\sftwreng>git log
commit d8afcf93c7fed4e18d9182a47b83bc472444bf69 (HEAD -> master)
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:25:26 2024 +0200

    Update file1.py with new changes

commit 9cf1b7a5f5b7a414c7b88cec2b8b263467678848
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:16:36 2024 +0200

    Add src directory with file1.py and file2.py

commit 8766b967e976bf763e0694c7b2ad5063f901bdfe
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:11:04 2024 +0200

    Add README

```

```

C:\Users\user\Desktop\sftwreng>git show 9cf1b7a
commit 9cf1b7a5f5b7a414c7b88cec2b8b263467678848
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:16:36 2024 +0200

    Add src directory with file1.py and file2.py

diff --git a/src/file1.py b/src/file1.py
new file mode 100644
index 0000000..8d1c8b6
--- /dev/null
+++ b/src/file1.py
@@ -0,0 +1 @@
+
diff --git a/src/file2.py b/src/file2.py
new file mode 100644
index 0000000..8d1c8b6
--- /dev/null
+++ b/src/file2.py
@@ -0,0 +1 @@
+

```

Then repeat the process.

```

C:\Users\user\Desktop\sftwreng>git diff src/file2.py
diff --git a/src/file2.py b/src/file2.py
index 8d1c8b6..78e8941 100644
--- a/src/file2.py
+++ b/src/file2.py
@@ -1,1 @@
-
+print("JOELLELELELL")

C:\Users\user\Desktop\sftwreng>git commit -m "Update file2.py with new changes
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/file1.py
        modified:   src/file2.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\user\Desktop\sftwreng>

```

```

C:\Users\user\Desktop\sftwreng>git add src/file1.py src/file2.py

```

```

C:\Users\user\Desktop\sftwreng>git commit -m "Add print statements to
file1.py and file2.py"

```

```

[master a31e2de] Add print statements to file1.py and file2.py
2 files changed, 4 insertions(+), 2 deletions(-)

```

Stretch Task:

```

C:\Users\user\Desktop\sftwreng>git rm src/file2.py
rm 'src/file2.py'

C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:   src/file2.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>git commit -m "Remove file2.py"
[master 63eee7c] Remove file2.py
1 file changed, 1 deletion(-)
delete mode 100644 src/file2.py

C:\Users\user\Desktop\sftwreng>

```

 file1

11/3/2024 12:29 PM

Python file

1 KB

Only file 1 is present.

Now deleting file1 manually and checking git status

```
C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   src/file1.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>git add -u

C:\Users\user\Desktop\sftwreng>git commit -m "Manually delete file1.py"
[master 97c1d22] Manually delete file1.py
1 file changed, 4 deletions(-)
delete mode 100644 src/file1.py

C:\Users\user\Desktop\sftwreng>
```

```

C:\Users\user\Desktop\sftwreng>git mv README README.md

C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   README -> README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>git commit -m "Rename README to README.md"
[master f7356e2] Rename README to README.md
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename README => README.md (100%)

C:\Users\user\Desktop\sftwreng>ren README.md README.txt

C:\Users\user\Desktop\sftwreng>git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf
    README.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\user\Desktop\sftwreng>

```

```

C:\Users\user\Desktop\sftwreng>git add README.txt

C:\Users\user\Desktop\sftwreng>git add -u

C:\Users\user\Desktop\sftwreng>git commit -m "Rename README.md to README.txt
using OS command"
[master b8d1ac6] Rename README.md to README.txt using OS command
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename README.md => README.txt (100%)

C:\Users\user\Desktop\sftwreng>

```

Git may not have recognized the rename because the rename was done outside of its tracking. Git typically recognizes renames when you use `git mv`. In the case of renaming with the OS command, it sees the original file as deleted and the new one as untracked rather than as a rename.


```
Administrator: Command Prompt

C:\Users\user\Desktop\sftwreng>git show --stat
commit b8d1ac6f8b11ffb2543d6763c6f515764fcc149e (HEAD -> master)
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:59:51 2024 +0200

    Rename README.md to README.txt using OS command

README.md => README.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\user\Desktop\sftwreng>git log --stat
commit b8d1ac6f8b11ffb2543d6763c6f515764fcc149e (HEAD -> master)
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:59:51 2024 +0200

    Rename README.md to README.txt using OS command

README.md => README.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)

commit f7356e2a16f3783aef8f19c01cec7b82089ec372
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:57:41 2024 +0200

    Rename README to README.md

README => README.md | 0
1 file changed, 0 insertions(+), 0 deletions(-)

commit 97c1d72a69a3b936a20dbf45c73035993b48c669
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:53:10 2024 +0200

    Manually delete file1.py

src/file1.py | 4 ----
1 file changed, 4 deletions(-)

commit 63ee7cacab830612a7c7be9308409abc99a4018
Author: joelleeej <joelle.jouni@hotmail.com>
... skipping ...
commit b8d1ac6f8b11ffb2543d6763c6f515764fcc149e (HEAD -> master)
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 12:59:51 2024 +0200

    Rename README.md to README.txt using OS command

README.md => README.txt | 0
```

```
C:\Users\user\Desktop\sftwreng>git diff
b8d1ac6f8b11ffb2543d6763c6f515764fcc149e..f7356e2a16f3783aef8f19c01cec7b820
89ec372
diff --git a/README.txt b/README.md
similarity index 100%
rename from README.txt
rename to README.md

C:\Users\user\Desktop\sftwreng>
```

Exercise 2

```
C:\Users\user\Desktop\sftwreng>git branch my_first_branch

C:\Users\user\Desktop\sftwreng>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\user\Desktop\sftwreng>echo "print('my_first_branch')" > new_file.py

C:\Users\user\Desktop\sftwreng>git add new_file.py

C:\Users\user\Desktop\sftwreng>git commit -m "Add new_file.py with a greeting"
[my_first_branch 06e9bab] Add new_file.py with a greeting
1 file changed, 1 insertion(+)
create mode 100644 new_file.py
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>echo "This is an additional line." >> existing_file.py
```

```
C:\Users\user\Desktop\sftwreng>git add existing_file.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update existing_file.py with an additional line"
```

```
[my_first_branch 21ecc52] Update existing_file.py with an additional line
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 existing_file.py
```

```
C:\Users\user\Desktop\sftwreng>git log --oneline
```

```
21ecc52 (HEAD -> my_first_branch) Update existing_file.py with an additional line
```

```
06e9bab Add new_file.py with a greeting
```

```
b8d1ac6 (master) Rename README.md to README.txt using OS command
```

```
f7356e2 Rename README to README.md
```

```
97c1d22 Manually delete file1.py
```

```
63eee7c Remove file2.py
```

```
a31e2de Add print statements to file1.py and file2.py
```

```
d8afcf9 Update file1.py with new changes
```

```
9cf1b7a Add src directory with file1.py and file2.py
```

```
8766b96 Add README
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>git checkout master
```

```
Switched to branch 'master'
```

```
C:\Users\user\Desktop\sftwreng>git log --oneline
```

```
b8d1ac6 (HEAD -> master) Rename README.md to README.txt using OS command
```

```
f7356e2 Rename README to README.md
```

```
97c1d22 Manually delete file1.py
```

```
63eee7c Remove file2.py
```

```
a31e2de Add print statements to file1.py and file2.py
```

```
d8afcf9 Update file1.py with new changes
```

```
9cf1b7a Add src directory with file1.py and file2.py
```

```
8766b96 Add README
```

```
C:\Users\user\Desktop\sftwreng>git checkout my_first_branch
```

```
Switched to branch 'my_first_branch'
```

```

C:\Users\user\Desktop\sftwreng>git log --graph
* commit 21ecc5211e124d18884fe12adf391c8a1493343f (HEAD -> my_first_branch)
| Author: joelleeej <joelle.jouni@hotmail.com>
| Date: Sun Nov 3 13:14:48 2024 +0200
|
| Update existing_file.py with an additional line
|
* commit 06e9bab91d0bdfc4869de5635c13fe3f6430fc3b
| Author: joelleeej <joelle.jouni@hotmail.com>
| Date: Sun Nov 3 13:13:36 2024 +0200
|
| Add new_file.py with a greeting
|
* commit b8d1ac6f8b11ffb2543d6763c6f515764fcc149e (master)
| Author: joelleeej <joelle.jouni@hotmail.com>
| Date: Sun Nov 3 12:59:51 2024 +0200
|
| Rename README.md to README.txt using OS command
|
* commit f7356e2a16f3783aef8f19c01cec7b82089ec372
| Author: joelleeej <joelle.jouni@hotmail.com>
| Date: Sun Nov 3 12:57:41 2024 +0200
|
| Rename README to README.md
|
* commit 97c1d22a69a3b936a20dbfd5c73035993b48c669
| Author: joelleeej <joelle.jouni@hotmail.com>
| Date: Sun Nov 3 12:53:10 2024 +0200
|
| Manually delete file1.py

```

```

C:\Users\user\Desktop\sftwreng>git log --graph --abbrev-commit --date=relative --branches --pretty=format:"%Cred%h%Crese
t -%C(yellow)%d%Crese %s %Cgreen(%cr) %C(bold blue)<%an>%Crese"
* 21ecc52 - (HEAD -> my_first_branch) Update existing_file.py with an additional line (5 minutes ago) <joelleeej>
* 06e9bab - Add new_file.py with a greeting (6 minutes ago) <joelleeej>
* b8d1ac6 - (master) Rename README.md to README.txt using OS command (20 minutes ago) <joelleeej>
* f7356e2 - Rename README to README.md (22 minutes ago) <joelleeej>
* 97c1d22 - Manually delete file1.py (27 minutes ago) <joelleeej>
* 63eee7c - Remove file2.py (30 minutes ago) <joelleeej>
* a31e2de - Add print statements to file1.py and file2.py (32 minutes ago) <joelleeej>
* d8afc9 - Update file1.py with new changes (54 minutes ago) <joelleeej>
* 9cf1b7a - Add src directory with file1.py and file2.py (63 minutes ago) <joelleeej>
* 8766b96 - Add README (69 minutes ago) <joelleeej>

```

```
C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'
```

```
C:\Users\user\Desktop\sftwreng>git merge my_first_branch
Updating b8d1ac6..21ecc52
Fast-forward
 existing_file.py | 1 +
 new_file.py      | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 existing_file.py
 create mode 100644 new_file.py
```

```
C:\Users\user\Desktop\sftwreng>git log --oneline
21ecc52 (HEAD -> master, my_first_branch) Update existing_file.py with an additional line
06e9bab Add new_file.py with a greeting
b8d1ac6 Rename README.md to README.txt using OS command
f7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
63eee7c Remove file2.py
a31e2de Add print statements to file1.py and file2.py
d8afc9 Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README
```

```
C:\Users\user\Desktop\sftwreng>git log --graph --oneline
* 21ecc52 (HEAD -> master, my_first_branch) Update existing_file.py with an additional line
* 06e9bab Add new_file.py with a greeting
* b8d1ac6 Rename README.md to README.txt using OS command
* f7356e2 Rename README to README.md
* 97c1d22 Manually delete file1.py
* 63eee7c Remove file2.py
* a31e2de Add print statements to file1.py and file2.py
* d8afc9 Update file1.py with new changes
* 9cf1b7a Add src directory with file1.py and file2.py
* 8766b96 Add README
```

```
C:\Users\user\Desktop\sftwreng>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\user\Desktop\sftwreng>echo "Adding another line to existing_file.py." >> existing_file.py
C:\Users\user\Desktop\sftwreng>git add existing_file.py
C:\Users\user\Desktop\sftwreng>git commit -m "Add another line to existing_file.py"
[my_first_branch 0dafcde] Add another line to existing_file.py
1 file changed, 1 insertion(+)

C:\Users\user\Desktop\sftwreng>
C:\Users\user\Desktop\sftwreng># Optionally, create a new file
'#' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\user\Desktop\sftwreng>echo "This is a new file for testing." > another_file.txt
C:\Users\user\Desktop\sftwreng>git add another_file.txt
C:\Users\user\Desktop\sftwreng>git commit -m "Add another_file.txt for testing"
[my_first_branch 0c2c7e0] Add another_file.txt for testing
1 file changed, 1 insertion(+)
 create mode 100644 another_file.txt

C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'

C:\Users\user\Desktop\sftwreng>echo "Updating README for clarity." >> README.txt
C:\Users\user\Desktop\sftwreng>git add README.txt
C:\Users\user\Desktop\sftwreng>git commit -m "Update README.txt for clarity"
[master b327295] Update README.txt for clarity
1 file changed, 1 insertion(+)

C:\Users\user\Desktop\sftwreng>git merge my_first_branch
Merge made by the 'ort' strategy.
 another_file.txt | 1 +
 existing_file.py | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 another_file.txt

C:\Users\user\Desktop\sftwreng>
```

```

C:\Users\user\Desktop\sftwreng>git merge my_first_branch
Merge made by the 'ort' strategy.
  another_file.txt | 1 +
  existing_file.py | 1 +
  2 files changed, 2 insertions(+)
  create mode 100644 another_file.txt

C:\Users\user\Desktop\sftwreng>git log --oneline
ebaf259 (HEAD -> master) Merge branch 'my_first_branch'
b327295 Update README.txt for clarity
0c2c7e0 (my_first_branch) Add another_file.txt for testing
0dafcde Add another line to existing_file.py
21ecc52 Update existing_file.py with an additional line
06e9bab Add new_file.py with a greeting
b8d1ac6 Rename README.md to README.txt using OS command
f7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
63eee7c Remove file2.py
a31e2de Add print statements to file1.py and file2.py
d8afcf9 Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README

C:\Users\user\Desktop\sftwreng>git log --graph --oneline
*   ebaf259 (HEAD -> master) Merge branch 'my_first_branch'
|
| *   0c2c7e0 (my_first_branch) Add another_file.txt for testing
| *   0dafcde Add another line to existing_file.py
| * | b327295 Update README.txt for clarity
|/
*   21ecc52 Update existing_file.py with an additional line
*   06e9bab Add new_file.py with a greeting
*   b8d1ac6 Rename README.md to README.txt using OS command
*   f7356e2 Rename README to README.md
*   97c1d22 Manually delete file1.py
*   63eee7c Remove file2.py
*   a31e2de Add print statements to file1.py and file2.py
*   d8afcf9 Update file1.py with new changes
*   9cf1b7a Add src directory with file1.py and file2.py
*   8766b96 Add README
C:\Users\user\Desktop\sftwreng>

```

Conflict :

```

C:\Users\user\Desktop\sftwreng>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\user\Desktop\sftwreng>echo "Adding a line to existing_file.py." >> existing_file.py

C:\Users\user\Desktop\sftwreng>git add existing_file.py

C:\Users\user\Desktop\sftwreng>git commit -m "Added a line to existing_file.py in my_first_branch"
[my_first_branch 9fc24c6] Added a line to existing_file.py in my_first_branch
1 file changed, 1 insertion(+)

C:\Users\user\Desktop\sftwreng>echo "Another line for testing." >> existing_file.py

C:\Users\user\Desktop\sftwreng>git add existing_file.py

C:\Users\user\Desktop\sftwreng>git commit -m "Another update to existing_file.py in my_first_branch"
[my_first_branch 05aec1d] Another update to existing_file.py in my_first_branch
1 file changed, 1 insertion(+)

C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'

C:\Users\user\Desktop\sftwreng>echo "Changing the same line in existing_file.py." > existing_file.py

C:\Users\user\Desktop\sftwreng>git add existing_file.py

C:\Users\user\Desktop\sftwreng>git commit -m "Updated existing_file.py in master with conflicting changes"
[master 3dae88e] Updated existing_file.py in master with conflicting changes
1 file changed, 1 insertion(+), 2 deletions(-)

C:\Users\user\Desktop\sftwreng>git merge my_first_branch
Auto-merging existing_file.py
CONFLICT (content): Merge conflict in existing_file.py
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\user\Desktop\sftwreng>_
C:\Users\user\Desktop\sftwreng>notepad existing_file.py

```

Before vs After :

File	Edit	View
CV.docx	CV.docx	0000001 wine.data
<pre> <<<<<< HEAD "Changing the same line in existing_file.py." ===== "This is an additional line." "Adding another line to existing_file.py." "Adding a line to existing_file.py." "Another line for testing." >>>>>> my_first_branch </pre>		
<pre> # existing_file.py "Changing the same line in existing_file.py." "This is an additional line." "Adding another line to existing_file.py." "Adding a line to existing_file.py." "Another line for testing." </pre>		

```

C:\Users\user\Desktop\sftwreng>git add existing_file.py

C:\Users\user\Desktop\sftwreng>git commit -m "Resolved merge conflict in existing_file.py"
[master 6dce249] Resolved merge conflict in existing_file.py

C:\Users\user\Desktop\sftwreng>

```

```

C:\Users\user\Desktop\sftwreng>git log --oneline
6dce249 (HEAD -> master) Resolved merge conflict in existing_file.py
3dae88e Updated existing_file.py in master with conflicting changes
05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
9fc24c6 Added a line to existing_file.py in my_first_branch
ebaf259 Merge branch 'my_first_branch'
b327295 Update README.txt for clarity
0c2c7e0 Add another_file.txt for testing
0dafcde Add another line to existing_file.py
21ecc52 Update existing_file.py with an additional line
06e9bab Add new_file.py with a greeting
b8d1ac6 Rename README.md to README.txt using OS command
f7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
63eee7c Remove file2.py
a31e2de Add print statements to file1.py and file2.py
d8afcf9 Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README

C:\Users\user\Desktop\sftwreng>git log --graph --oneline
*   6dce249 (HEAD -> master) Resolved merge conflict in existing_file.py
|
| * 05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
| * 9fc24c6 Added a line to existing_file.py in my_first_branch
* | 3dae88e Updated existing_file.py in master with conflicting changes
* | ebaf259 Merge branch 'my_first_branch'
|/
| * 0c2c7e0 Add another_file.txt for testing
| * 0dafcde Add another line to existing_file.py
* | b327295 Update README.txt for clarity
|/
* 21ecc52 Update existing_file.py with an additional line
* 06e9bab Add new_file.py with a greeting
* b8d1ac6 Rename README.md to README.txt using OS command
* f7356e2 Rename README to README.md
* 97c1d22 Manually delete file1.py
* 63eee7c Remove file2.py
* a31e2de Add print statements to file1.py and file2.py
* d8afcf9 Update file1.py with new changes
* 9cf1b7a Add src directory with file1.py and file2.py
* 8766b96 Add README

```

Exercise 3:

```

C:\Users\user\Desktop\sftwreng>git checkout -b undoing_changes
Switched to a new branch 'undoing_changes'

C:\Users\user\Desktop\sftwreng>echo "This is the content of file3.py." > file3.py

C:\Users\user\Desktop\sftwreng>git add file3.py

C:\Users\user\Desktop\sftwreng>git commit -m "Add file3.py with initial content"
[undoing_changes ef6b3b2] Add file3.py with initial content
1 file changed, 1 insertion(+)
create mode 100644 file3.py

C:\Users\user\Desktop\sftwreng>echo "Adding a new line to file3.py." >> file3.py

C:\Users\user\Desktop\sftwreng>git diff
diff --git a/file3.py b/file3.py
index 8c60a39..807223e 100644
--- a/file3.py
+++ b/file3.py
@@ -1,2 @@
 "This is the content of file3.py."
+ "Adding a new line to file3.py."

C:\Users\user\Desktop\sftwreng>git checkout -- file3.py

C:\Users\user\Desktop\sftwreng>git status
On branch undoing_changes
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\user\Desktop\sftwreng>git restore file3.py

C:\Users\user\Desktop\sftwreng>

```

```

C:\Users\user\Desktop\sftwreng>echo "This is the content of file4.py." > file4.py

C:\Users\user\Desktop\sftwreng>git add file4.py

C:\Users\user\Desktop\sftwreng>git commit -m "Add file4.py with initial content"
[undoing_changes e42fe87] Add file4.py with initial content
1 file changed, 1 insertion(+)
create mode 100644 file4.py

C:\Users\user\Desktop\sftwreng>git revert HEAD
[undoing_changes f3a0c5b] Revert "Add file4.py with initial content"
1 file changed, 1 deletion(-)
delete mode 100644 file4.py

C:\Users\user\Desktop\sftwreng>

```

```

C:\> Select Administrator: Command Prompt - git revert HEAD

Revert "Add file4.py with initial content"

This reverts commit e42fe874a3fb5112e249412831b273756f0debda.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch undoing_changes
# Changes to be committed:
#   deleted:   file4.py
#
# Untracked files:
#   Git Exercices.pdf
#
~
~

```

```

C:\Users\user\Desktop\sftwreng>git log --oneline
f3a0c5b (HEAD -> undoing_changes) Revert "Add file4.py with initial content"

```



```
e42fe87 Add file4.py with initial content
ef6b3b2 Add file3.py with initial content
6dce249 (master) Resolved merge conflict in existing_file.py
3dae88e Updated existing_file.py in master with conflicting changes
05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
9fc24c6 Added a line to existing_file.py in my_first_branch
ebaf259 Merge branch 'my_first_branch'
b327295 Update README.txt for clarity
0c2c7e0 Add another_file.txt for testing
0dafcde Add another line to existing_file.py
21ecc52 Update existing_file.py with an additional line
06e9bab Add new_file.py with a greeting
b8d1ac6 Rename README.md to README.txt using OS command
f7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
63eee7c Remove file2.py
a31e2de Add print statements to file1.py and file2.py
d8afcf9 Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>git show HEAD
commit f3a0c5b7bf4904d08a74b93e791296b2afd11353 (HEAD -> undoing_changes)
Author: joelleeej <joelle.jouni@hotmail.com>
Date: Sun Nov 3 13:57:46 2024 +0200
```

```
Revert "Add file4.py with initial content"
```

```
This reverts commit e42fe874a3fb5112e249412831b273756f0debda.
```

```
diff --git a/file4.py b/file4.py
deleted file mode 100644
index e3d990c..0000000
--- a/file4.py
+++ /dev/null
@@ -1,0,0 @@
-"This is the content of file4.py."
```

```
C:\Users\user\Desktop\sftwreng>
```

```

C:\Users\user\Desktop\sftwreng>echo "Adding a new line to file3.py." >> file3.py

C:\Users\user\Desktop\sftwreng>echo "This is the second content line of file4.py." >>
file4.py

C:\Users\user\Desktop\sftwreng>
C:\Users\user\Desktop\sftwreng>git add file3.py file4.py

C:\Users\user\Desktop\sftwreng>git commit -m "Update file3.py and file4.py with new
lines"
[undoing_changes a70efb5] Update file3.py and file4.py with new lines
2 files changed, 2 insertions(+)
create mode 100644 file4.py

C:\Users\user\Desktop\sftwreng>git reset --soft HEAD^
More? git status
fatal: ambiguous argument 'HEADgit': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

C:\Users\user\Desktop\sftwreng>

```

Step 10b: Can You Describe What Happened?

- The `git reset --soft HEAD^` command moves the HEAD pointer back by one commit (to the previous commit) but keeps the changes from that commit staged in the index. This means the files changed in the latest commit are still in the staging area, ready to be committed again if desired.

Step 10c: Do You Still See Your Previous Commit?

- You will no longer see the most recent commit in the log because the HEAD pointer has been moved back. However, the changes from that commit remain staged.

Step 10d: What Could You Have Done to Avoid Losing the Commit After Doing a Reset?

- To avoid losing the commit, you could have created a branch before resetting (using `git checkout -b new-branch-name`) or used `git reset --soft`

with a specific commit reference instead of HEAD^. Alternatively, using `git stash` could save the changes temporarily.

```
C:\Users\user\Desktop\sftwreng>git reset HEAD

C:\Users\user\Desktop\sftwreng>git status
On branch undoing_changes
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\user\Desktop\sftwreng>
```

Step 11b: Notice How We Use HEAD Instead of HEAD^ Now? Why Are We Doing That?

- HEAD refers to the current commit, while HEAD^ refers to the commit before the current one. By using HEAD, you are indicating that you want to reset the index (staging area) of the current commit, not move back in history.

Step 11c: Notice Also That We Didn't Pass Any Scope to Our Reset; What Do You Think That Scope Is?

- The default scope for `git reset` is to operate on the current branch's HEAD, affecting the index and optionally the working directory based on the flags used (soft, mixed, hard)

```
C:\Users\user\Desktop\sftwreng>git reset --hard
HEAD is now at a70efb5 Update file3.py and file4.py with new lines
```

Step 12b: We Didn't Use Any Commit for This Command; What Do You Think Git Used as Default Value?

- Git defaults to using the current HEAD as the commit reference, so it resets everything back to the state of the last commit.

Step 12c: What Would Have Happened If You Had Done This Command Directly After Step 9?

- If you had executed `git reset --hard` right after the commit made in Step 9, it would have erased that commit and any changes made in the working directory, reverting everything to the last commit before Step 9.

```
C:\Users\user\Desktop\sftwreng>git log --oneline
70efb5 (HEAD -> undoing_changes) Update file3.py and file4.py with new lines
3a0c5b Revert "Add file4.py with initial content"
42fe87 Add file4.py with initial content
ef6b3b2 Add file3.py with initial content
6dce249 (master) Resolved merge conflict in existing_file.py
8dae88e Updated existing_file.py in master with conflicting changes
95aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
9fc24c6 Added a line to existing_file.py in my_first_branch
abaf259 Merge branch 'my_first_branch'
6327295 Update README.txt for clarity
9c2c7e0 Add another_file.txt for testing
9dafcde Add another line to existing_file.py
21ecc52 Update existing_file.py with an additional line
96e9bab Add new_file.py with a greeting
98d1ac6 Rename README.md to README.txt using OS command
7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
63eee7c Remove file2.py
931e2de Add print statements to file1.py and file2.py
88afc9 Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README

C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>echo "This is file5.py." > file5.py
```

```
C:\Users\user\Desktop\sftwreng>git add file5.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Add file5.py"
```

```
[undoing_changes dcca3fd] Add file5.py
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 file5.py
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng># Make another commit to update file3.py
```

```
'#' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\Users\user\Desktop\sftwreng>echo "Another update to file3.py." >> file3.py
```

```
C:\Users\user\Desktop\sftwreng>git add file3.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update file3.py again"
```

```
[undoing_changes 9c92b41] Update file3.py again
```

```
1 file changed, 1 insertion(+)
```

```
C:\Users\user\Desktop\sftwreng>
```

Stretch Task:

```
C:\Users\user\Desktop\sftwreng>echo "Adding a new line to file5.py." >> file5.py
```

```
C:\Users\user\Desktop\sftwreng>git checkout -- file5.py
```

Step 2b: What Does It Do to the Working Directory?

- The working directory for `file5.py` reflects the last committed state. Any uncommitted changes you made are lost.

Step 2c: What Happens to the HEAD Pointer?

- The HEAD pointer remains unchanged; it still points to the current commit. Only the working directory is affected.

```
C:\Users\user\Desktop\sftwreng>echo "This is a change in file4.py." >> file4.py
```

```
C:\Users\user\Desktop\sftwreng>git add file4.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update file4.py"
```

```
[undoing_changes 18bf34b] Update file4.py
```

```
1 file changed, 1 insertion(+)
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>echo "This is a change in file3.py." >> file3.py
```

```
C:\Users\user\Desktop\sftwreng>git add file3.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update file3.py"
```

```
[undoing_changes 8ef0d62] Update file3.py
```

```
1 file changed, 1 insertion(+)
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>git revert HEAD~2..HEAD
```

```
[undoing_changes 4877e68] Revert "Update file3.py"
```

```
1 file changed, 1 deletion(-)
```

```
[undoing_changes d5eaa7f] Revert "Update file4.py"
```

1 file changed, 1 deletion(-)

C:\Users\user\Desktop\sftwreng>

```
Administrator: Command Prompt - git revert HEAD~2..HEAD
Revert "Update file3.py"

This reverts commit 8ef0d6205d206f81e1e2bda8ea253d9015801199.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch undoing_changes
# Revert currently in progress.
#
# Changes to be committed:
#   modified:   file3.py
#
# Untracked files:
#   Git Exercices.pdf
#
~
~
~
```

Step 4b: How Are the Changes Handled?

- The changes from the specified commits are undone, but the original commits remain in the history. New commits are created to reflect the reverted changes.

Step 4c: Can You Figure Out What This Syntax Means?

- The syntax HEAD~2..HEAD specifies a range of commits. HEAD~2 refers to the commit two steps back from the current HEAD, and HEAD refers to the latest commit.

Step 4d: Do You Think We Can Use This Same Syntax with `git reset`?

- Yes, you can use a similar range with `git reset`, but it would reset the commit history instead of creating new commits.

```

Administrator: Command Prompt
C:\Users\user\Desktop\sftwreng>git log --oneline
d5eaa7f (HEAD -> undoing_changes) Revert "Update file4.py"
4877e68 Revert "Update file3.py"
8ef0d62 Update file3.py
18bf34b Update file4.py
9c92b41 Update file3.py again
dcca3fd Add file5.py
a70efb5 Update file3.py and file4.py with new lines
f3a0c5b Revert "Add file4.py with initial content"
e42fe87 Add file4.py with initial content
ef6b3b2 Add file3.py with initial content
5dce249 (master) Resolved merge conflict in existing_file.py
3dae88e Updated existing_file.py in master with conflicting changes
95aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
9fc24c6 Added a line to existing_file.py in my_first_branch
ebaf259 Merge branch 'my_first_branch'
b327295 Update README.txt for clarity
9c2c7e0 Add another_file.txt for testing
0dafcde Add another line to existing_file.py
21ecc52 Update existing_file.py with an additional line
06e9bab Add new_file.py with a greeting
b8d1ac6 Rename README.md to README.txt using OS command
f7356e2 Rename README to README.md
97c1d22 Manually delete file1.py
53eee7c Remove file2.py
a31e2de Add print statements to file1.py and file2.py
d8afc9f Update file1.py with new changes
9cf1b7a Add src directory with file1.py and file2.py
8766b96 Add README

```

```

C:\Users\user\Desktop\sftwreng>git checkout -b anchor
Switched to a new branch 'anchor'

```

```

C:\Users\user\Desktop\sftwreng>git reset HEAD~3
Unstaged changes after reset:
M   file4.py

```

```

C:\Users\user\Desktop\sftwreng>git checkout anchor
M   file4.py
Already on 'anchor'

```

```

C:\Users\user\Desktop\sftwreng>git reset HEAD
Unstaged changes after reset:
M   file4.py

```

```

C:\Users\user\Desktop\sftwreng>git reset --soft HEAD~1

```

```

C:\Users\user\Desktop\sftwreng>git reset HEAD~1
Unstaged changes after reset:
M   file3.py

```

```

C:\Users\user\Desktop\sftwreng>git reset --hard HEAD~1
HEAD is now at a70efb5 Update file3.py and file4.py with new lines

```

```

C:\Users\user\Desktop\sftwreng>

```

Exercise 4:

```
C:\Users\user\Desktop\sftwreng>git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

```
C:\Users\user\Desktop\sftwreng>echo "This is a feature file." > feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git add feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Add feature1.py with initial content"
[feature-branch 51c823b] Add feature1.py with initial content
1 file changed, 1 insertion(+)
create mode 100644 feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'
```

```
C:\Users\user\Desktop\sftwreng>echo "This is the main file." > main.py
```

```
C:\Users\user\Desktop\sftwreng>git add main.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Add main.py with initial content"
[master 23346a4] Add main.py with initial content
1 file changed, 1 insertion(+)
create mode 100644 main.py
```

```
Administrator: Command Prompt
C:\Users\user\Desktop\sftwreng>git log --oneline --graph --branches
* 23346a4 (HEAD -> master) Add main.py with initial content
| * 51c823b (feature-branch) Add feature1.py with initial content
| | * d5eaa7f (undoing_changes) Revert "Update file4.py"
| | * 4877e68 Revert "Update file3.py"
| | * 8ef0d62 Update file3.py
| | * 18bf34b Update file4.py
| | * 9c92b41 Update file3.py again
| | * dcca3fd Add file5.py
| | /
| * a70efb5 (anchor) Update file3.py and file4.py with new lines
| * f3a0c5b Revert "Add file4.py with initial content"
| * e42fe87 Add file4.py with initial content
| * ef6b3b2 Add file3.py with initial content
| /
* 6dce249 Resolved merge conflict in existing_file.py
|
| * 05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
| * 9fc24c6 Added a line to existing_file.py in my_first_branch
| * 3dae88e Updated existing_file.py in master with conflicting changes
| * ebafe259 Merge branch 'my_first_branch'
| /
| * 0c2c7e0 Add another_file.txt for testing
| * 0dafcde Add another line to existing_file.py
| * b327295 Update README.txt for clarity
| /
* 21ecc52 Update existing_file.py with an additional line
* 06e9bab Add new_file.py with a greeting
* b8d1ac6 Rename README.md to README.txt using OS command
* f7356e2 Rename README to README.md
```



```
C:\Users\user\Desktop\sftwreng>git checkout feature-branch
Switched to branch 'feature-branch'
```

```
C:\Users\user\Desktop\sftwreng>git rebase master
Successfully rebased and updated refs/heads/feature-branch.
```

```
C:\Users\user\Desktop\sftwreng>git status
On branch feature-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git Exercices.pdf
```

nothing added to commit but untracked files present (use "git add" to track)

```

C:\Users\user\Desktop\sftwreng>git log --oneline --graph --branches
cbeab7d (HEAD -> feature-branch) Add feature1.py with initial content
7b3e830 Update file3.py and file4.py with new lines
c87199f Revert "Add file4.py with initial content"
e2c9715 Add file4.py with initial content
8ee92e8 Add file3.py with initial content
23346a4 (master) Add main.py with initial content
* d5eaa7f (undoing_changes) Revert "Update file4.py"
* 4877e68 Revert "Update file3.py"
* 8ef0d62 Update file3.py
* 18bf34b Update file4.py
* 9c92b41 Update file3.py again
* dcca3fd Add file5.py
* a70efb5 (anchor) Update file3.py and file4.py with new lines
* f3a0c5b Revert "Add file4.py with initial content"
* e42fe87 Add file4.py with initial content
* ef6b3b2 Add file3.py with initial content

6dce249 Resolved merge conflict in existing_file.py

* 05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
* 9fc24c6 Added a line to existing_file.py in my_first_branch
| 3dae88e Updated existing_file.py in master with conflicting changes
| ebaf259 Merge branch 'my_first_branch'
|
* 0c2c7e0 Add another_file.txt for testing
* 0dafcde Add another line to existing_file.py
| b327295 Update README.txt for clarity
|
```

```
C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'
```

```
C:\Users\user\Desktop\sftwreng>echo "Modified in master branch." >> feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git add feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update feature1.py in master"
[master c3ac813] Update feature1.py in master
1 file changed, 1 insertion(+)
```

```
create mode 100644 feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git checkout feature-branch
```

```
Switched to branch 'feature-branch'
```

```
C:\Users\user\Desktop\sftwreng>echo "Conflicting change in feature branch." >>
feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git add feature1.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update feature1.py in feature-branch
with conflicting change"
```

```
[feature-branch 4004cbb] Update feature1.py in feature-branch with conflicting change
1 file changed, 1 insertion(+)
```

```
C:\Users\user\Desktop\sftwreng>git rebase master
```

```
Auto-merging feature1.py
```

```
CONFLICT (add/add): Merge conflict in feature1.py
```

```
error: could not apply cbeab7d... Add feature1.py with initial content
```

```
hint: Resolve all conflicts manually, mark them as resolved with
```

```
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
```

```
hint: You can instead skip this commit: run "git rebase --skip".
```

```
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
```

```
Could not apply cbeab7d... Add feature1.py with initial content
```

```
C:\Users\user\Desktop\sftwreng>
```

```
C:\Users\user\Desktop\sftwreng>git status
```

```
interactive rebase in progress; onto c3ac813
```

```
Last commands done (5 commands done):
```

```
  pick 7b3e830 Update file3.py and file4.py with new lines
```

```
  pick cbeab7d Add feature1.py with initial content
```

```
(see more in file .git/rebase-merge/done)
```

```
Next command to do (1 remaining command):
```

```
  pick 4004cbb Update feature1.py in feature-branch with conflicting change
```

```
(use "git rebase --edit-todo" to view and edit)
```

```
You are currently rebasing branch 'feature-branch' on 'c3ac813'.
```

```
(fix conflicts and then run "git rebase --continue")
```

```
(use "git rebase --skip" to skip this patch)
```

```
(use "git rebase --abort" to check out the original branch)
```

Unmerged paths:

(use "git restore --staged <file>..." to unstage)

(use "git add <file>..." to mark resolution)

both added: feature1.py

Untracked files:

(use "git add <file>..." to include in what will be committed)

Git Exercices.pdf

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\user\Desktop\sftwreng>

C:\Users\user\Desktop\sftwreng>notepad feature1.py

```
|<<<<<< HEAD
"Modified in master branch."
=====
"This is a feature file."
>>>>>> cbeab7d (Add feature1.py with initial content)
```

```
# feature1.py
```

```
# Feature 1 description
```

```
modified_message = "Modified in master branch."
```

```
conflicting_message = "Conflicting change in feature branch."
```

C:\Users\user\Desktop\sftwreng>git add feature1.py

C:\Users\user\Desktop\sftwreng>git rebase --continue

[detached HEAD 098c095] Update feature1.py in feature-branch with conflicting change

1 file changed, 4 insertions(+), 1 deletion(-)

Successfully rebased and updated refs/heads/feature-branch.

Select Administrator: Command Prompt - git rebase --continue

Update feature1.py in feature-branch with conflicting change

Please enter the commit message for your changes. Lines starting with '#' will be ignored, and an empty message aborts the commit.

interactive rebase in progress; onto c3ac813

Last commands done (6 commands done):

pick cbeab7d Add feature1.py with initial content

pick 4004cbb Update feature1.py in feature-branch with conflicting change

No commands remaining.

You are currently rebasing branch 'feature-branch' on 'c3ac813'.

Changes to be committed:

modified: feature1.py

Untracked files:

Git Exercices.pdf

C:\Users\user\Desktop\sftwreng>git log --online --graph --branches

* 098c095 (HEAD -> feature-branch) Update feature1.py in feature-branch with conflicting change

* c60cbfa Add feature1.py with initial content

* b49f6af Update file3.py and file4.py with new lines

* 8af78f5 Revert "Add file4.py with initial content"

* a7216ec Add file4.py with initial content

* c2af573 Add file3.py with initial content

* c3ac813 (master) Update feature1.py in master

* 23346a4 Add main.py with initial content

| * d5eaa7f (undoing_changes) Revert "Update file4.py"

| * 4877e68 Revert "Update file3.py"

| * 8ef0d62 Update file3.py

| * 18bf34b Update file4.py

| * 9c92b41 Update file3.py again

| * dcca3fd Add file5.py

| * a70efb5 (anchor) Update file3.py and file4.py with new lines

| * f3a0c5b Revert "Add file4.py with initial content"

| * e42fe87 Add file4.py with initial content

| * ef6b3b2 Add file3.py with initial content

|/

* 6dce249 Resolved merge conflict in existing_file.py

|\

| * 05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch

| * 9fc24c6 Added a line to existing_file.py in my_first_branch

* | 3dae88e Updated existing_file.py in master with conflicting changes

* | eba7259 Merge branch 'my_first_branch'

```
C:\Users\user\Desktop\sftwreng>git rebase --continue
fatal: No rebase in progress?
```

```
C:\Users\user\Desktop\sftwreng>git checkout master
Switched to branch 'master'
```

```
C:\Users\user\Desktop\sftwreng>echo "Another change in master." >> main.py
```

```
C:\Users\user\Desktop\sftwreng>git add main.py
```

```
C:\Users\user\Desktop\sftwreng>git commit -m "Update main.py in master with
another change"
```

```
[master 7d7c0bc] Update main.py in master with another change
1 file changed, 1 insertion(+)
```

```
C:\Users\user\Desktop\sftwreng>git checkout feature-branch
Switched to branch 'feature-branch'
```

```
C:\Users\user\Desktop\sftwreng>echo "Another change in feature branch." >>
feature1.py
```

```
C:\> Administrator: Command Prompt - git rebase -i master

ick c2af573 Add file3.py with initial content
ick a7216ec Add file4.py with initial content
quash 8af78f5 Revert "Add file4.py with initial content"
ick b49f6af Update file3.py and file4.py with new lines
ick c60cbfa Add feature1.py with initial content
dit_098c095 Update feature1.py in feature-branch with conflicting change
ick 6d2565d Add more content to main.py
ick ac9ef66 Update feature1.py in feature-branch with another change

Rebase 7d7c0bc..ac9ef66 onto 7d7c0bc (8 commands)

Commands:
p, pick <commit> = use commit
r, reword <commit> = use commit, but edit the commit message
e, edit <commit> = use commit, but stop for amending
s, squash <commit> = use commit, but meld into previous commit
f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
    commit's log message, unless -C is used, in which case
    keep only this commit's message; -c is same as -C but
    opens the editor
x, exec <command> = run command (the rest of the line) using shell
b, break = stop here (continue rebase later with 'git rebase --continue')
d, drop <commit> = remove commit
l, label <label> = label current HEAD with a name
t, reset <label> = reset HEAD to a label
m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
    create a merge commit using the original merge commit's
    message (or the merge commit's message if its available)
git/rebase-merge/git-rebase-todo[+] [unix] (14:42 03/11/2024)
- INSERT --
```

Update feature1.py in feature-branch with conflicting change), change pick to edit for that commit to combine Update file3.py and file4.py with new lines with the previous commit Revert "Add file4.py with initial content"), change the word pick to squash (or just s for short) next to the commit you want to squash.

```
C:\Users\user\Desktop\sftwreng>git add feature1.py
interactive rebase in progress; onto 7d7c0bc
Last commands done (3 commands done):mmit -m "Update feature1.py in feature-branch with another change"
    pick a7216ec Add file4.py with initial contentature-branch with another change
    squash 8af78f5 Revert "Add file4.py with initial content"
(see more in file .git/rebase-merge/done)
Next commands to do (5 remaining commands):i master
    pick b49f6af Update file3.py and file4.py with new lines
    pick c60cbfa Add feature1.py with initial content
(use "git rebase --edit-todo" to view and edit)
You are currently rebasing branch 'feature-branch' on '7d7c0bc'.
(all conflicts fixed: run "git rebase --continue")
```

Untracked files:

(use "git add <file>..." to include in what will be committed)

Git Exercices.pdf

e --abort

No changes

You asked to amend the most recent commit, but doing so would make it empty. You can repeat your command with --allow-empty, or you can remove the commit entirely with "git reset HEAD^".

Could not apply 8af78f5... Revert "Add file4.py with initial content"

C:\Users\user\Desktop\sftwreng>git commit --amend

interactive rebase in progress; onto 7d7c0bc

Last commands done (3 commands done):

pick a7216ec Add file4.py with initial content

squash 8af78f5 Revert "Add file4.py with initial content"

(see more in file .git/rebase-merge/done)

Next commands to do (5 remaining commands):

pick b49f6af Update file3.py and file4.py with new lines

pick c60cbfa Add feature1.py with initial content

(use "git rebase --edit-todo" to view and edit)

You are currently rebasing branch 'feature-branch' on '7d7c0bc'.

(all conflicts fixed: run "git rebase --continue")

Untracked files:

(use "git add <file>..." to include in what will be committed)

Git Exercices.pdf

e --abort

No changes

You asked to amend the most recent commit, but doing so would make it empty. You can repeat your command with --allow-empty, or you can remove the commit entirely with "git reset HEAD^".

C:\Users\user\Desktop\sftwreng>git rebase --continue

[detached HEAD 271ddfc] Add file4.py with initial content

Date: Sun Nov 3 13:57:39 2024 +0200

Stopped at 098c095... Update feature1.py in feature-branch with conflicting change

You can amend the commit now, with

git commit --amend

Once you are satisfied with your changes, run

```
git rebase --continue
```

C:\Users\user\Desktop\sftwreng>

```
C:\Users\user\Desktop\sftwreng>git rebase --continue
This is a combination of 2 commits.
This is the 1st commit message:

dd file4.py with initial content

This is the commit message #2:

Revert "Add file4.py with initial content"

This reverts commit e42fe874a3fb5112e249412831b273756f0debda.

Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.

Date:      Sun Nov 3 13:57:39 2024 +0200

interactive rebase in progress; onto 7d7c0bc
Last commands done (3 commands done):
  pick a7216ec Add file4.py with initial content
  squash 8af78f5 Revert "Add file4.py with initial content"
Next commands to do (5 remaining commands):
  pick b49f6af Update file3.py and file4.py with new lines
  pick c60cbfa Add feature1.py with initial content
You are currently rebasing branch 'feature-branch' on '7d7c0bc'.

Untracked files:
  (use "git add" to track)
  Git Exercices.pdf
  e --abort

No changes
```

C:\Users\user\Desktop\sftwreng>git log --oneline --graph --branches

```
* ac9ef66 (feature-branch) Update feature1.py in feature-branch with another change
* 6d2565d Add more content to main.py
* 098c095 Update feature1.py in feature-branch with conflicting change
* c60cbfa Add feature1.py with initial content
* b49f6af Update file3.py and file4.py with new lines
* 8af78f5 Revert "Add file4.py with initial content"
* a7216ec Add file4.py with initial content
* c2af573 Add file3.py with initial content
| * 7d7c0bc (master) Update main.py in master with another change
|/
* c3ac813 Update feature1.py in master
* 23346a4 Add main.py with initial content
| * d5eaa7f (undoing_changes) Revert "Update file4.py"
| * 4877e68 Revert "Update file3.py"
```



```
| * 8ef0d62 Update file3.py
| * 18bf34b Update file4.py
| * 9c92b41 Update file3.py again
| * dcca3fd Add file5.py
| * a70efb5 (anchor) Update file3.py and file4.py with new lines
| * f3a0c5b Revert "Add file4.py with initial content"
| * e42fe87 Add file4.py with initial content
| * ef6b3b2 Add file3.py with initial content
|/
* 6dce249 Resolved merge conflict in existing_file.py
|\
| * 05aec1d (my_first_branch) Another update to existing_file.py in my_first_branch
| * 9fc24c6 Added a line to existing_file.py in my_first_branch
* | 3dae88e Updated existing_file.py in master with conflicting changes
* | ebafe259 Merge branch 'my_first_branch'
|\
| * 0c2c7e0 Add another_file.txt for testing
| * 0dafcde Add another line to existing_file.py
* | b327295 Update README.txt for clarity
|/
* 21ecc52 Update existing_file.py with an additional line
* 06e9bab Add new_file.py with a greeting

C:\Users\user\Desktop\sftwreng>
```