

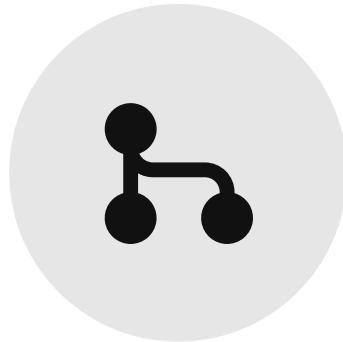
# Module 4



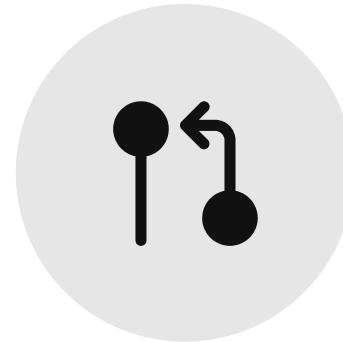
Behind The Git



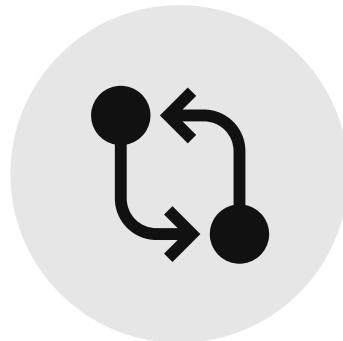
Branching



Working With Branches



Merging Branches



Merge Conflicts



Collaborators



Pull Request

# **Behind the Git**

# Behind the Git

- Internally, Git is a **<key, value>** data store.

```
// javascript object
let object = {
  id: 1,
  name: "Tom"
}
```

```
// python dictionary
dictionary = {
  "id": 1,
  "name": "Jerry"
}
```

# Key

- Key contains the hash of the data we want to store.
- Git uses a cryptographic hash function called SHA1.
- (40 digit Hexa-decimal value approx.)

a1b2c3d4e5f60718293a4b5c6d7e8f9012345678

# Value

- Value is our actual data.
- Blobs, trees, and commits are the main components of Git's data structure.

# Value

Identifier

# Value

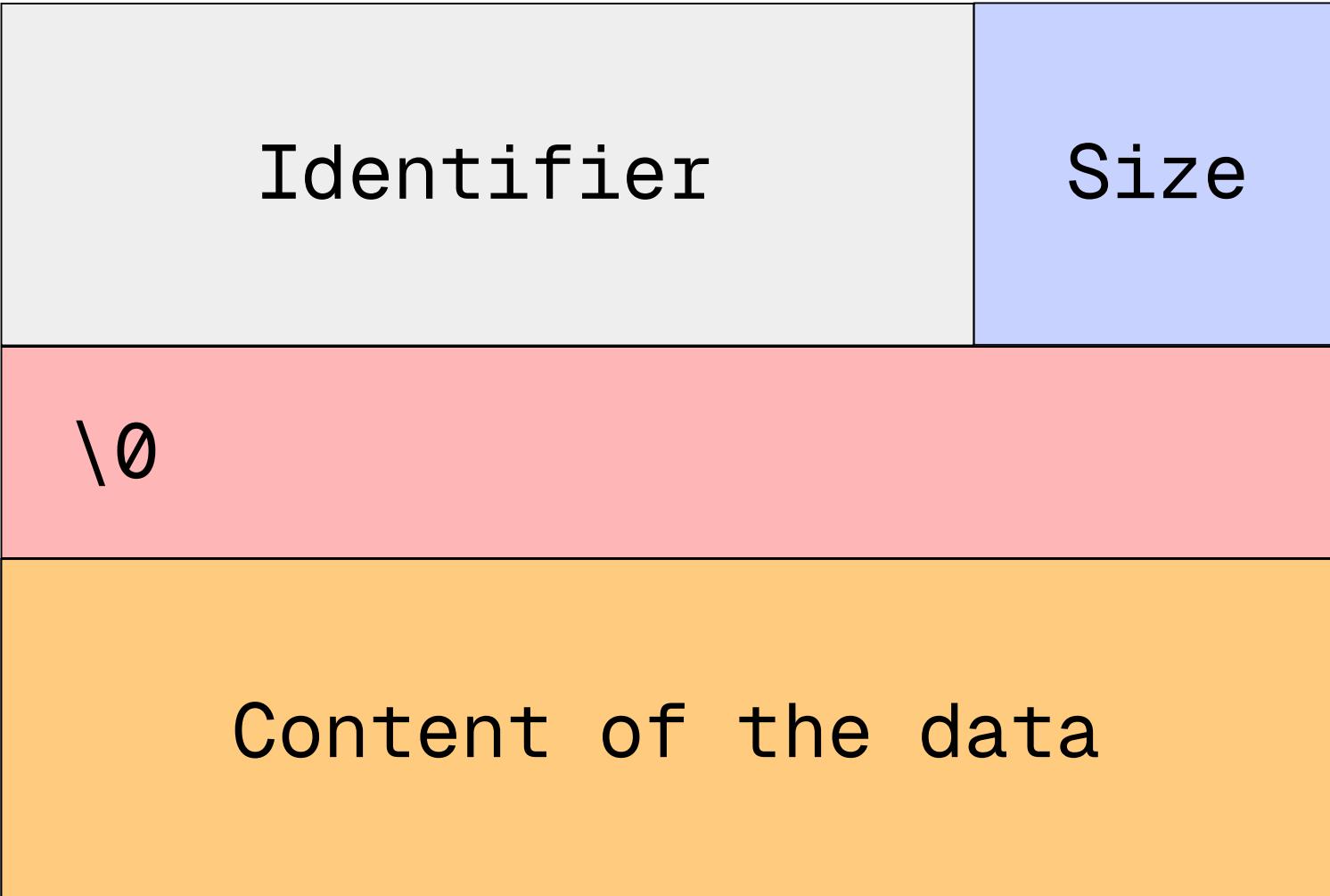
Identifier

Size

# Value

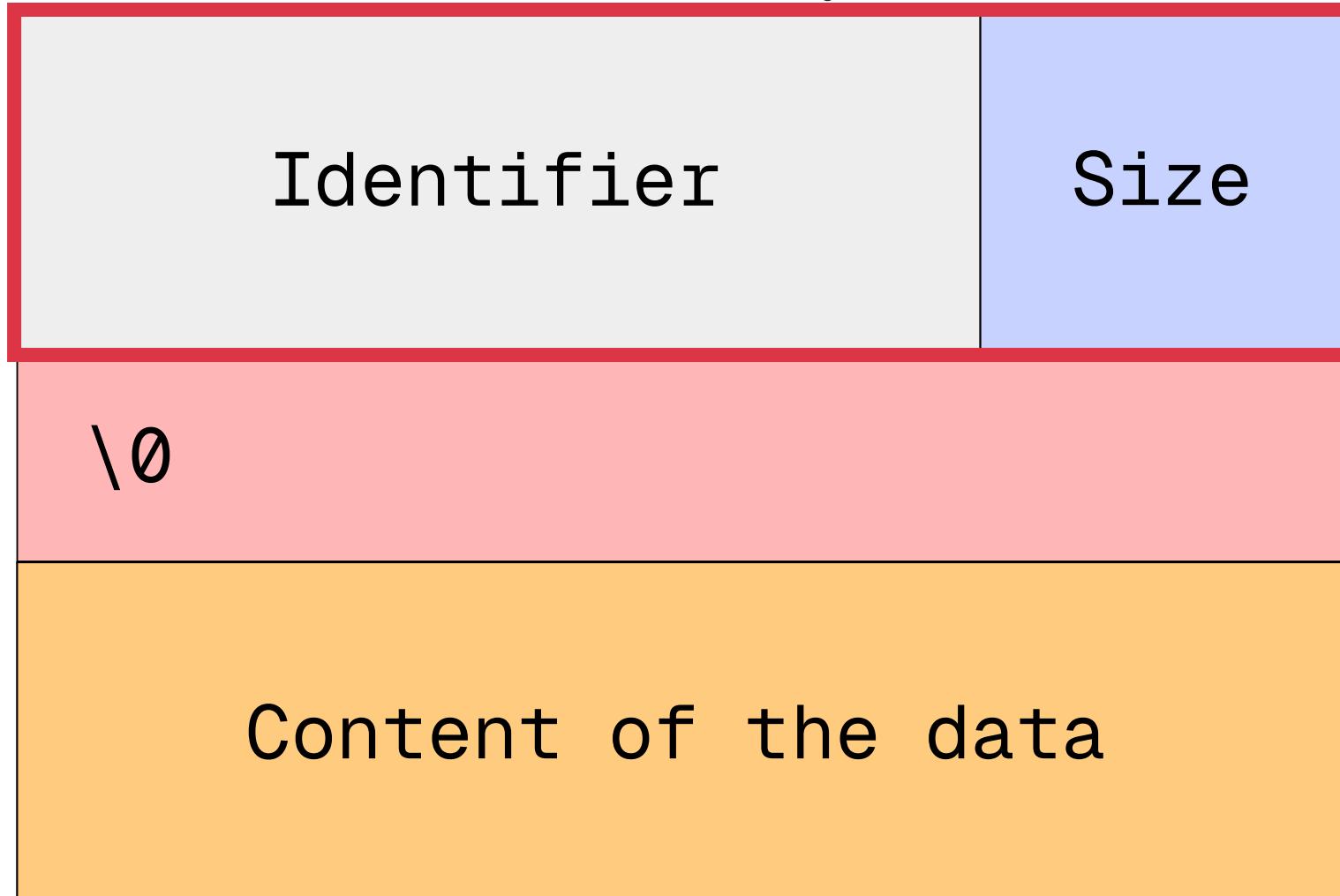


# Value



# Value

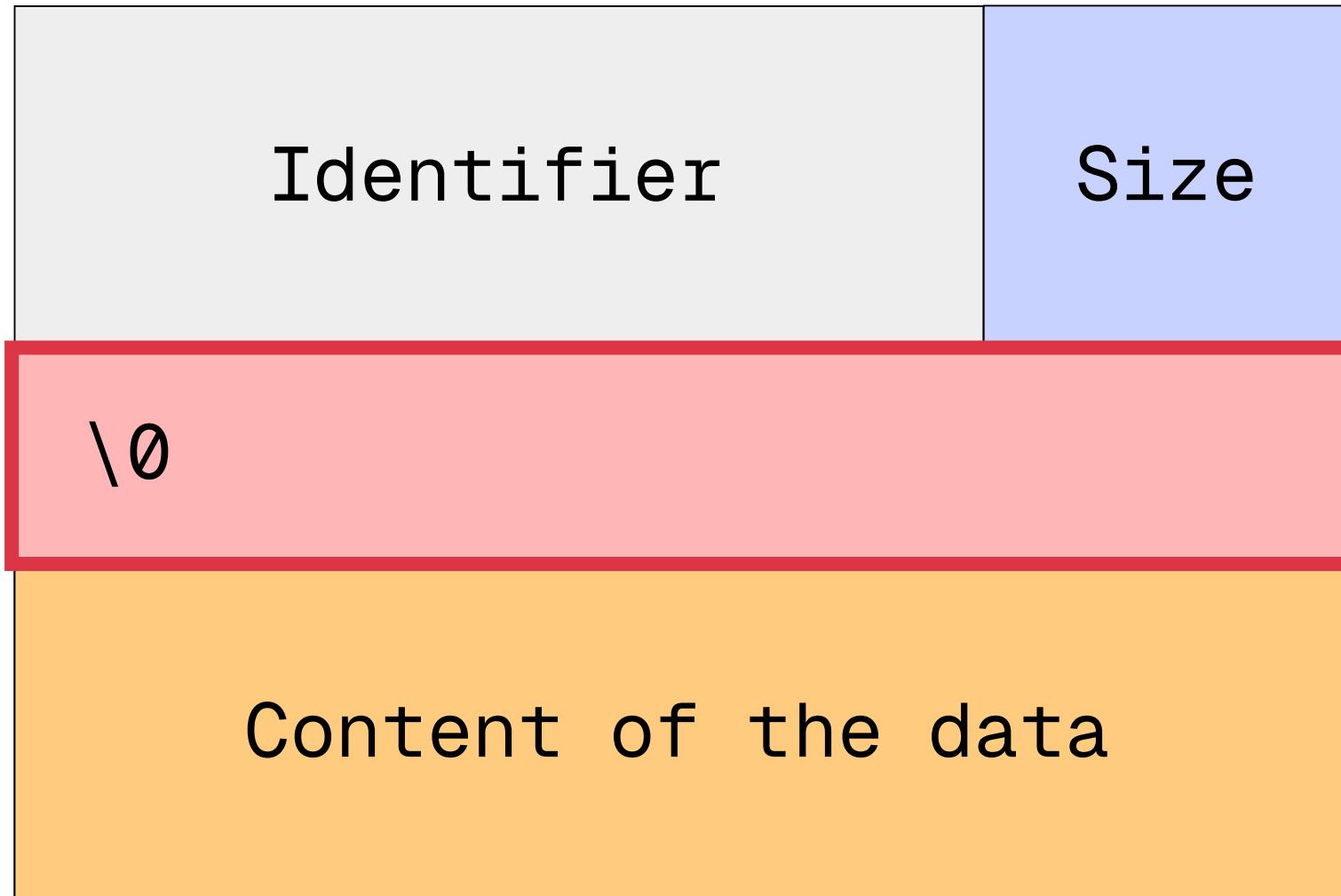
Header



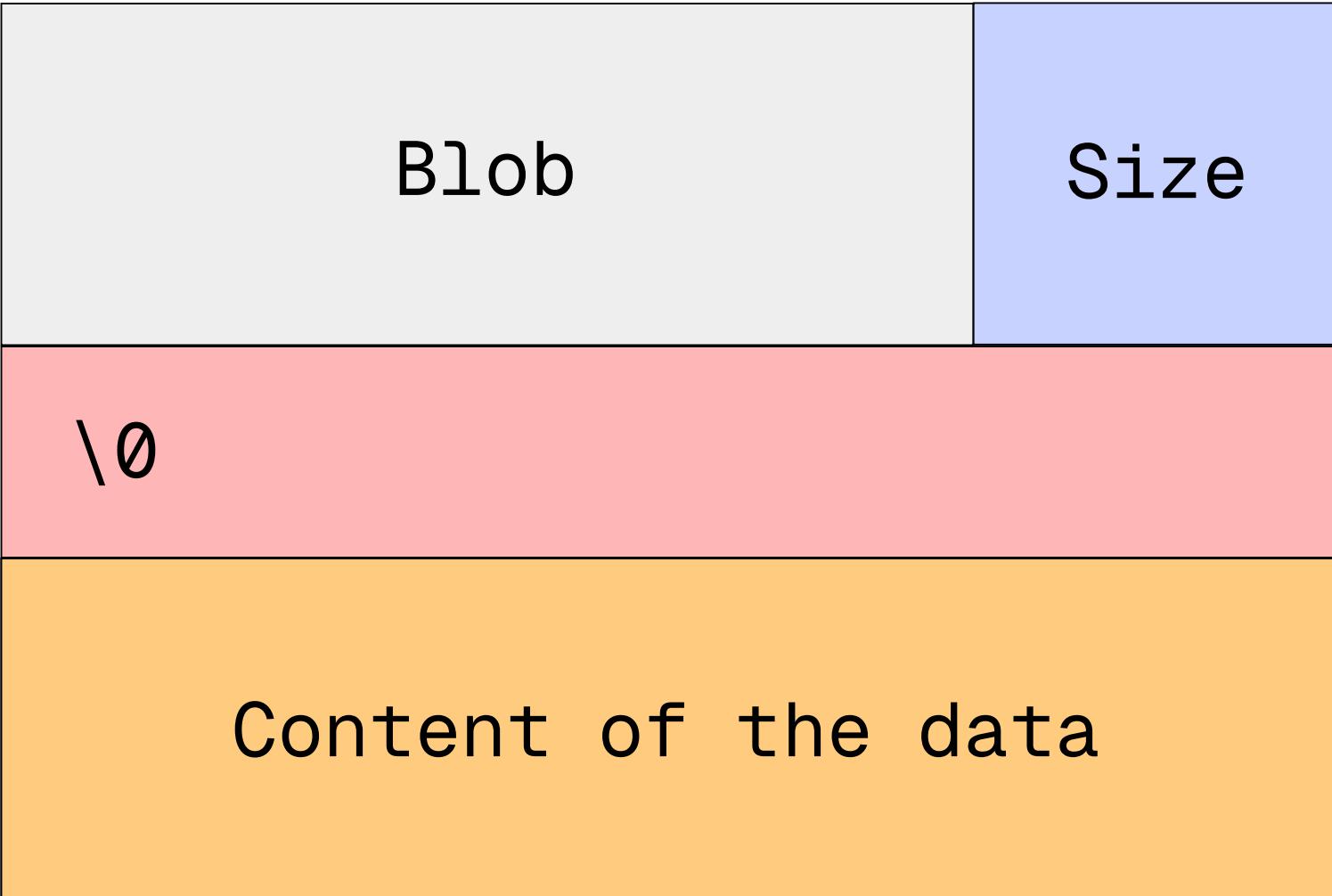
**Value**

Separator

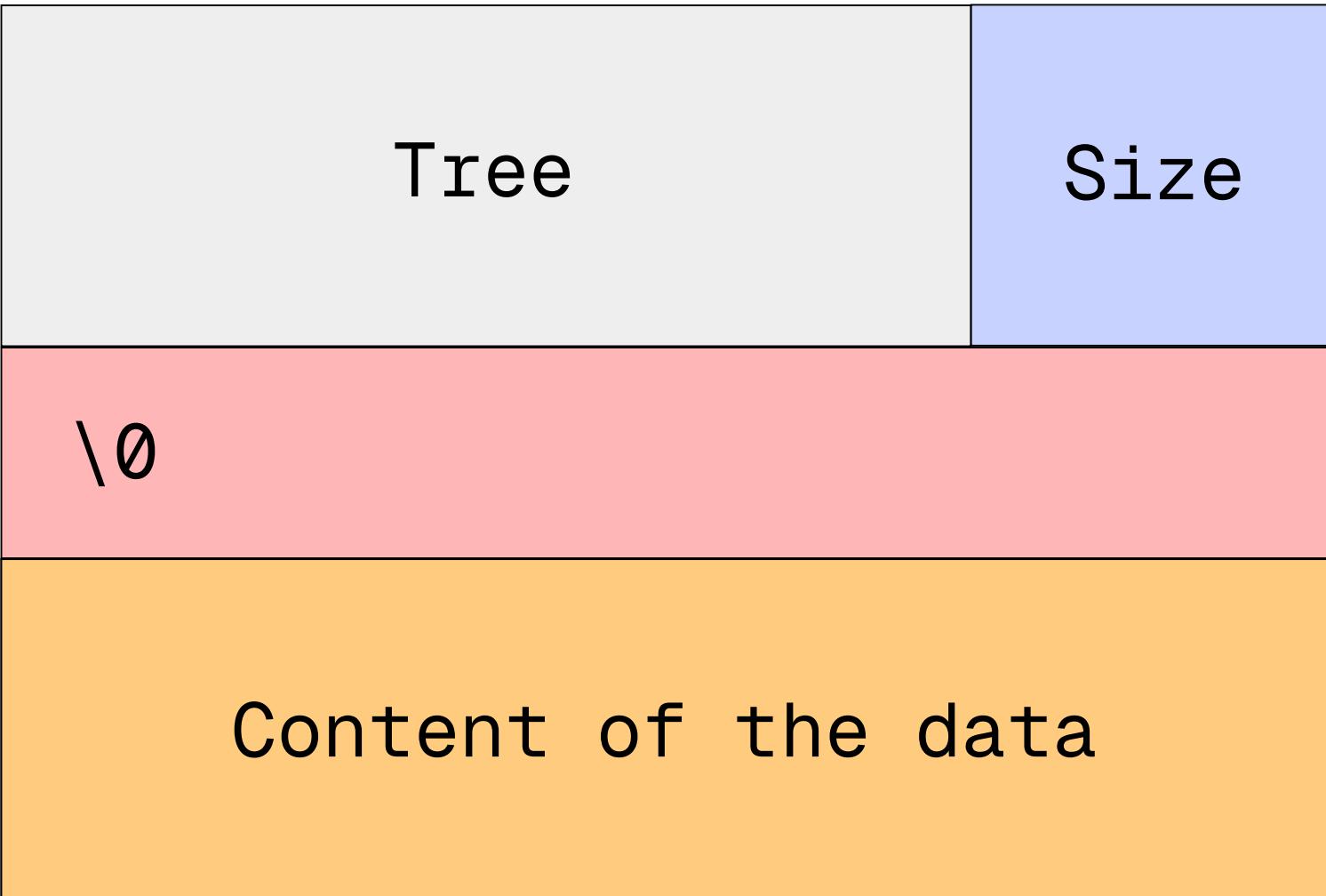
Header



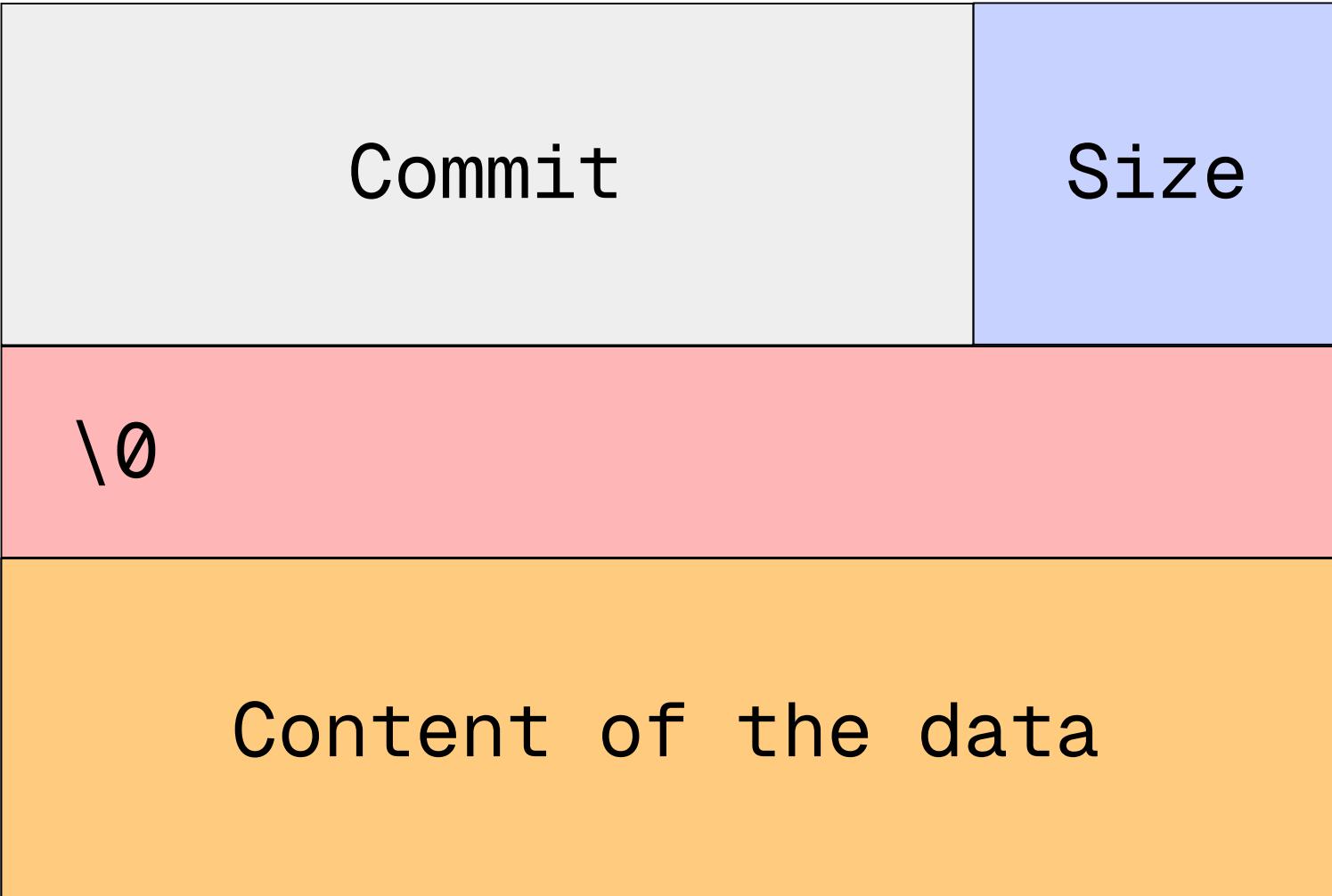
# Value



# Value



# Value

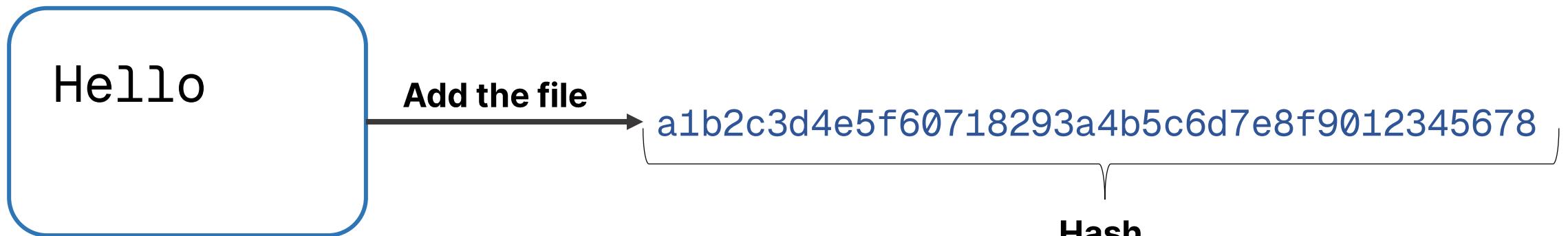


# Blob

- It generally stores the contents of a file.
- It contains only the raw content of the file (no filename, no metadata).
- Since the blob is entirely defined by its data, if two files in a directory tree have the same contents, they will share the same blob object.

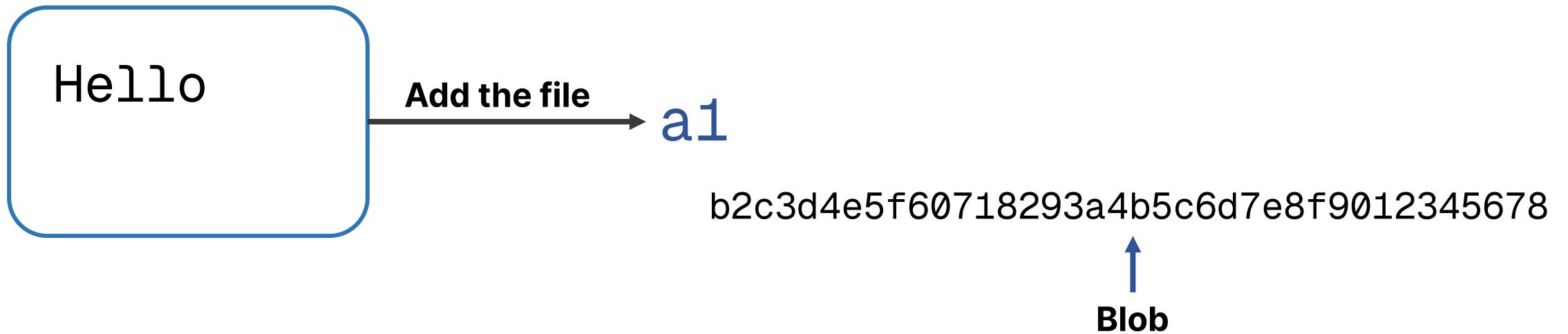
# Blob

Advanced/ Hello.txt



# Blob

Advanced/ Hello.txt



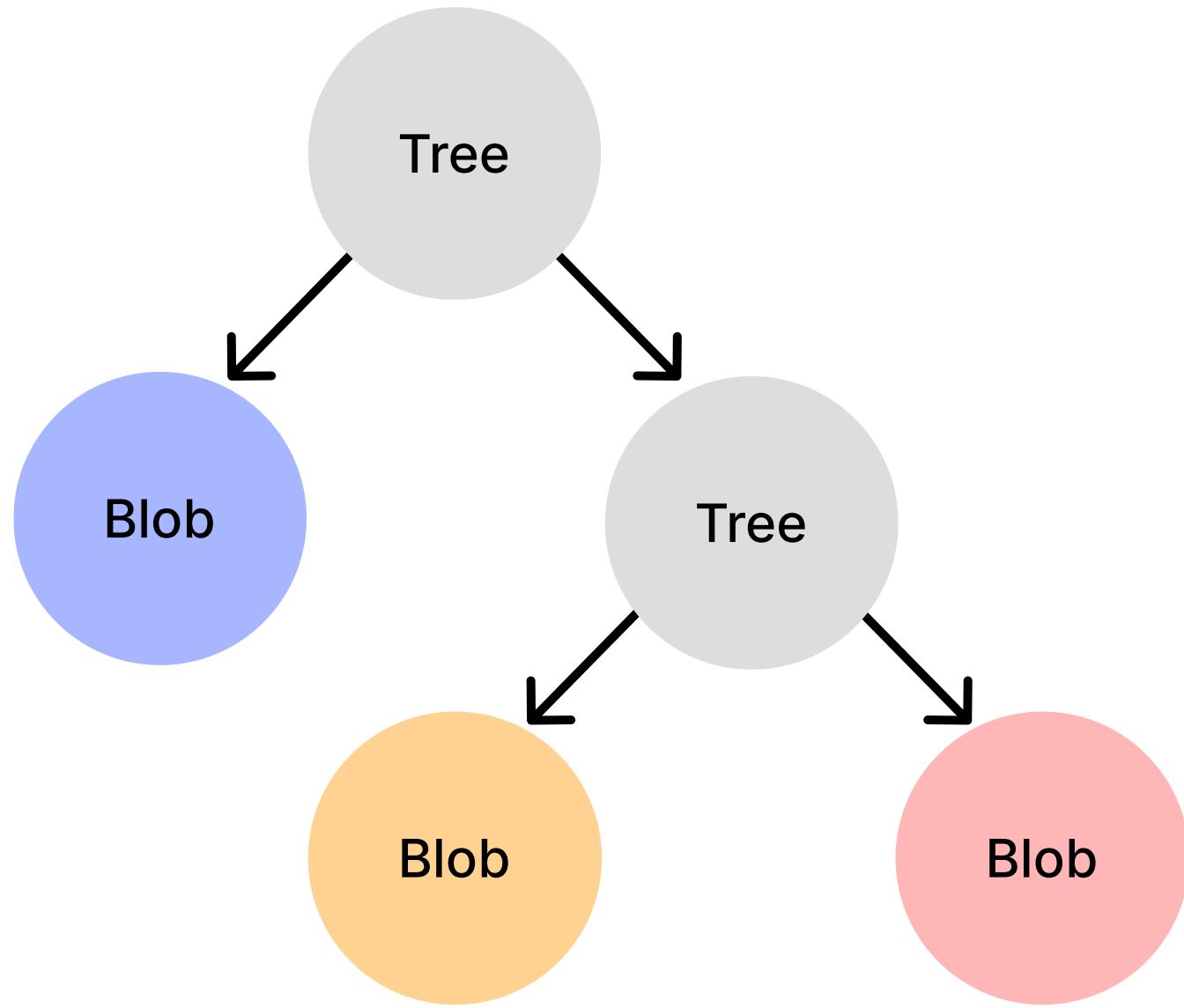
# Blob

< b2c3d4e5f60718293a4b5c6d7e8f9012345678 >

Hello

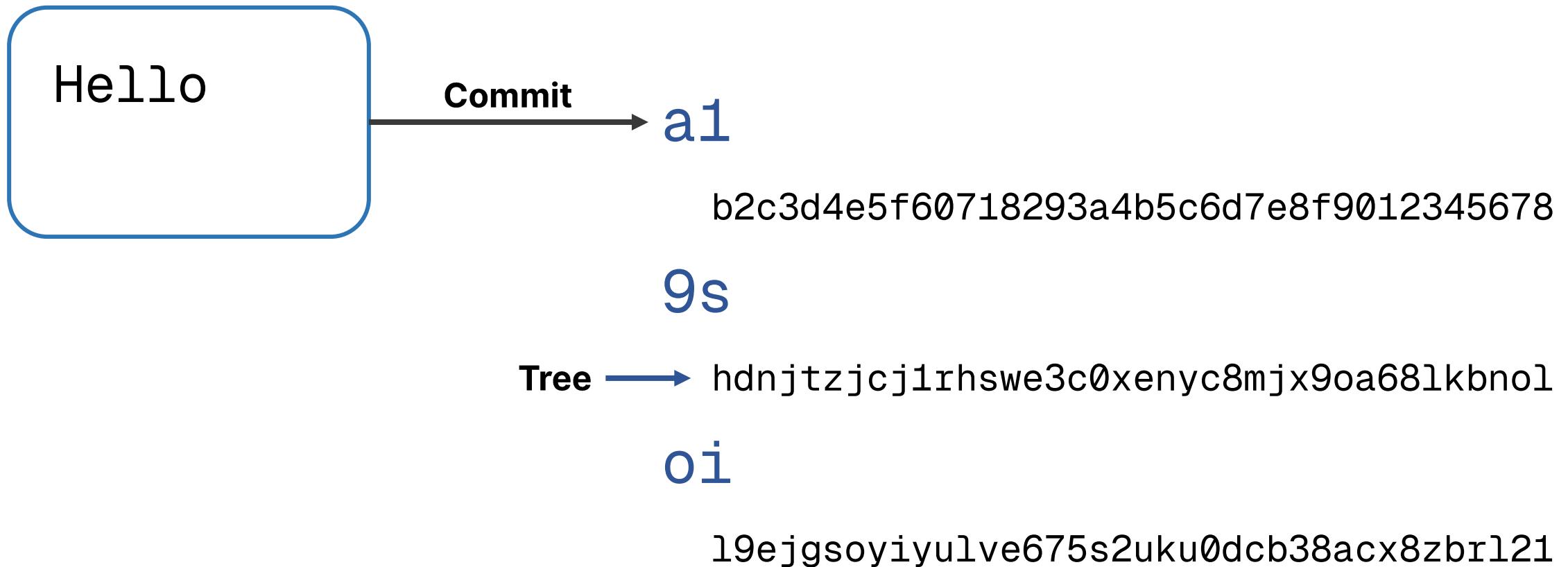
# Tree

- A tree represents the contents of a directory or subdirectory.
- It contains a list of pointers to blobs (files) and other trees (subdirectories), along with their names and permissions.



# Tree

Advanced/ Hello.txt



# Tree

<hdnjtzjcj1rhswe3c0xenyc8mjx9oa681kbnol>

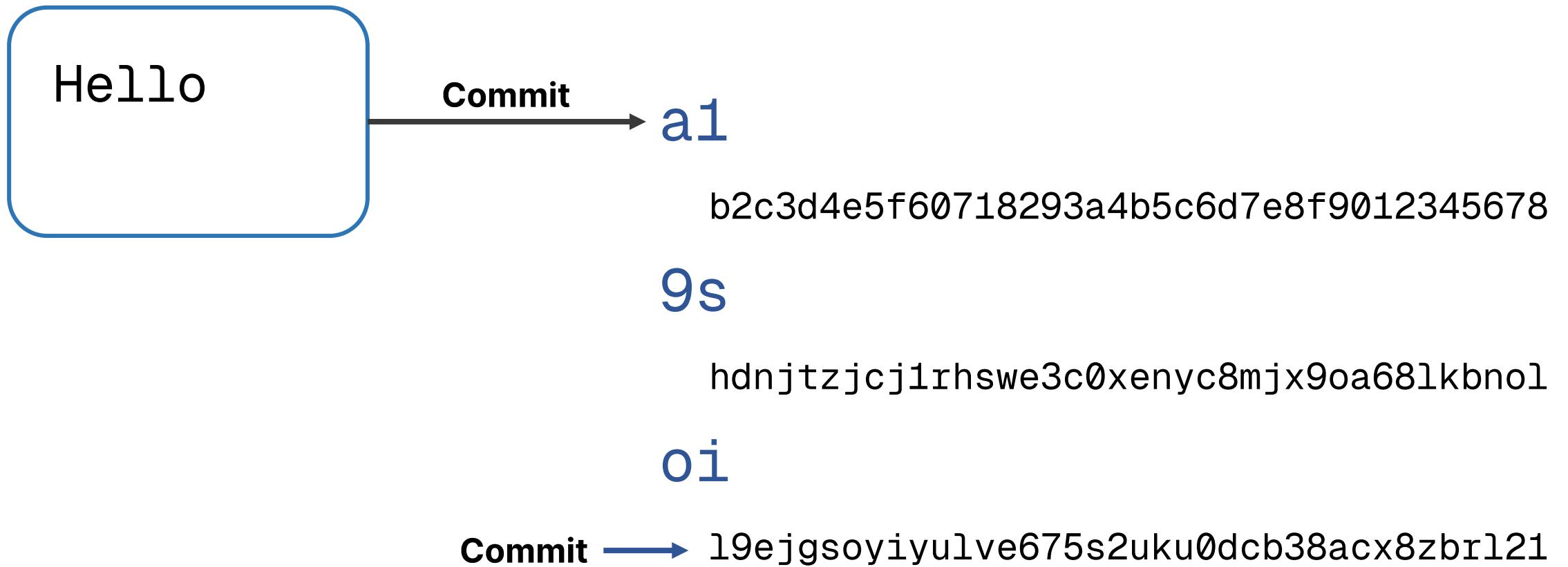
100644 blob a1b2c3d4e5f60718293a4... hello.py

# Commit

- A commit is a snapshot of your entire project at a specific point in time.
- It contains:
  - A pointer to a tree (the project structure at that point).
  - A pointer to the parent commit(s) (the previous snapshot).
  - Metadata like author, timestamp, and a commit message.

# Commit

Advanced/ Hello.txt

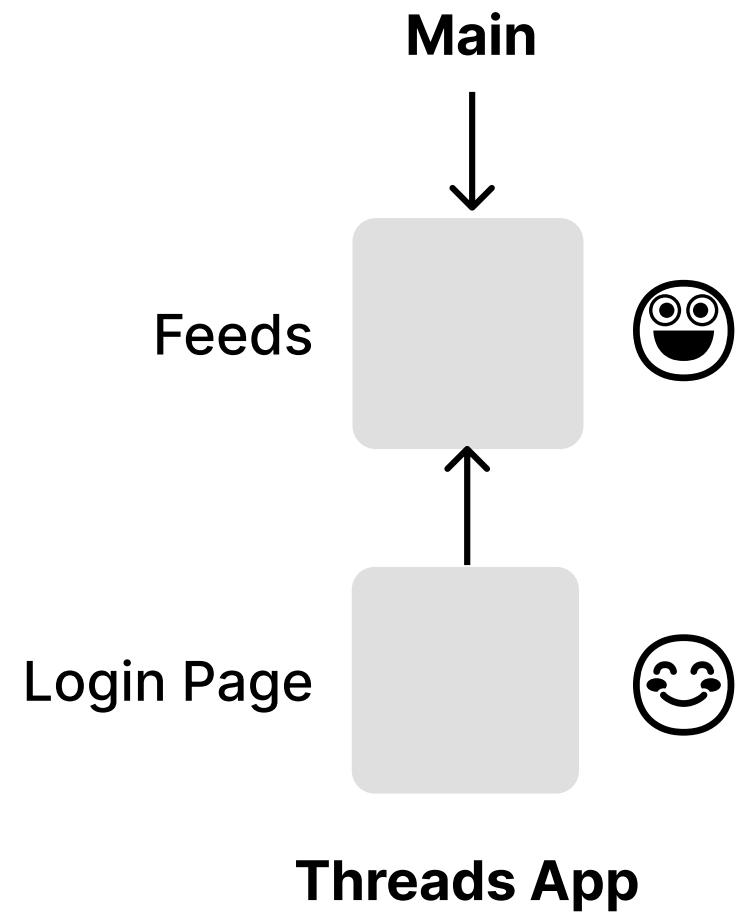


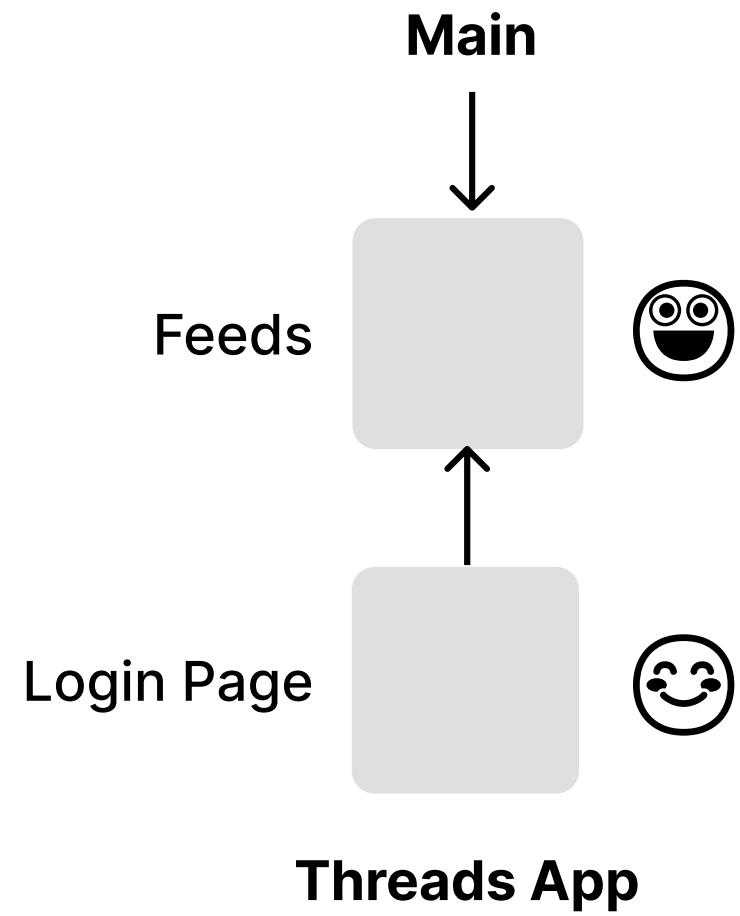
# Commit

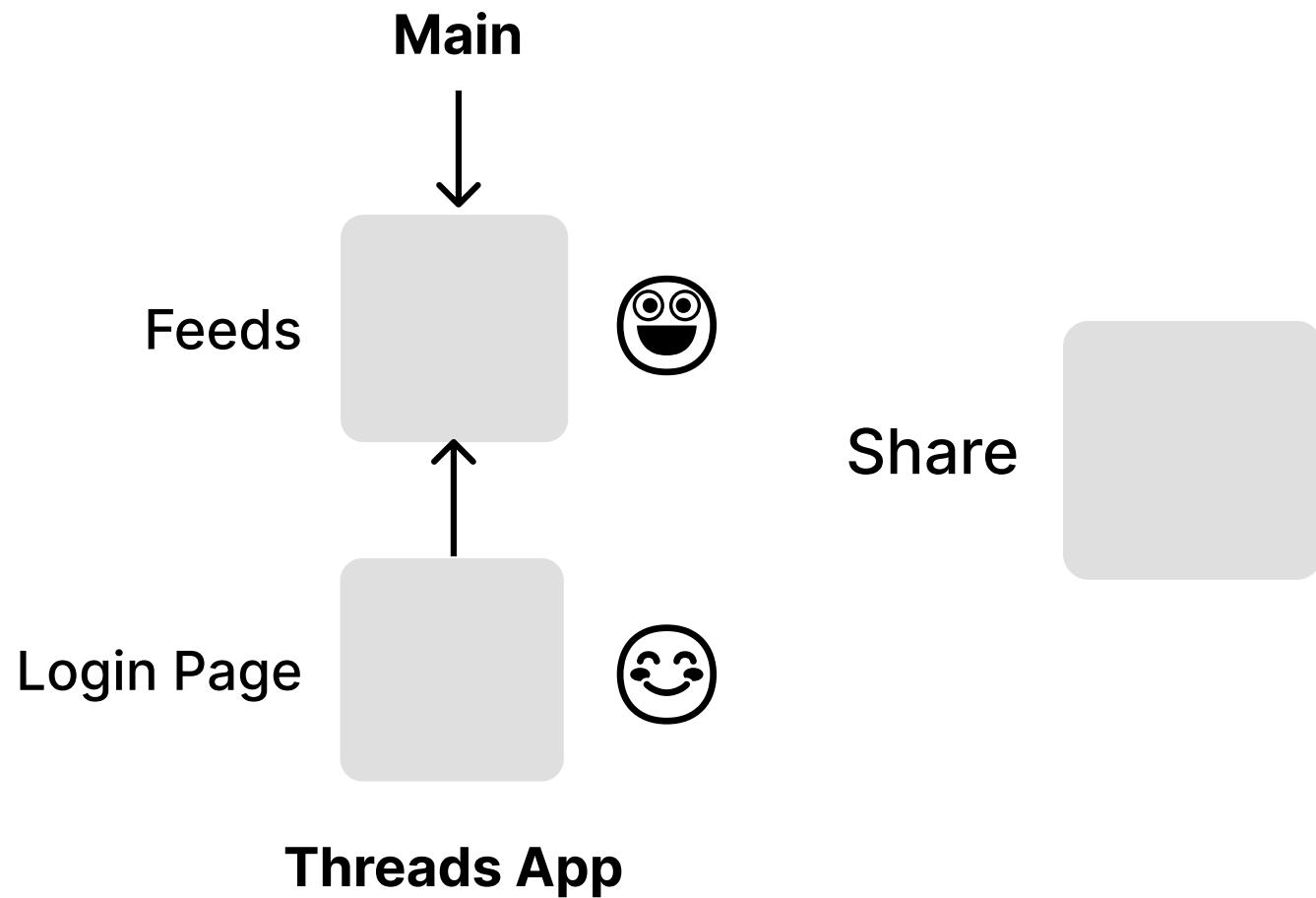
```
<19ejgsoyiyulve675s2uku0dcb38acx8zbrl21>
tree 9shdnjtzjcj1rhswe3c0xenyc8mjx9oa681kbnol
parent ...
author ... <...> 1738753650 +0545
committer ... <...> 1738753650 +0545
```

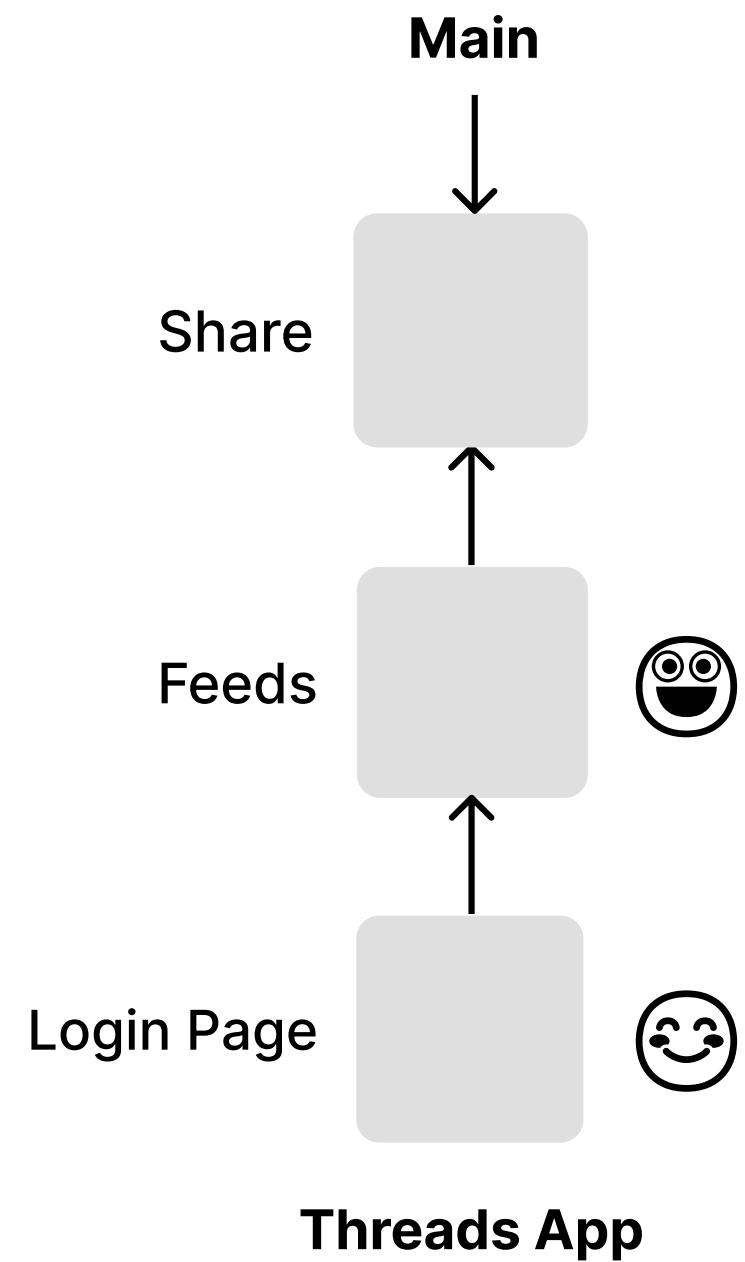
# **Branching**

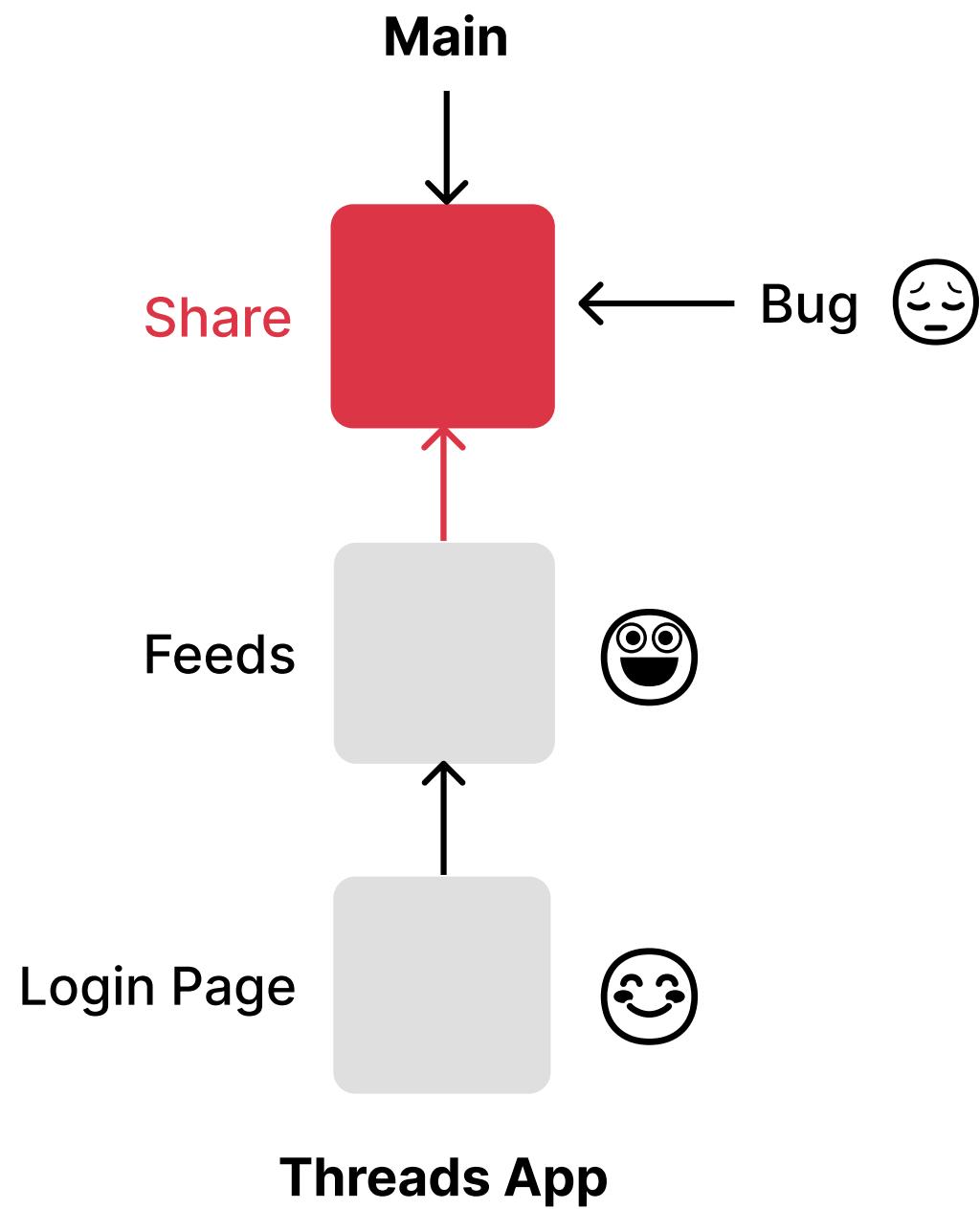
# Threads App

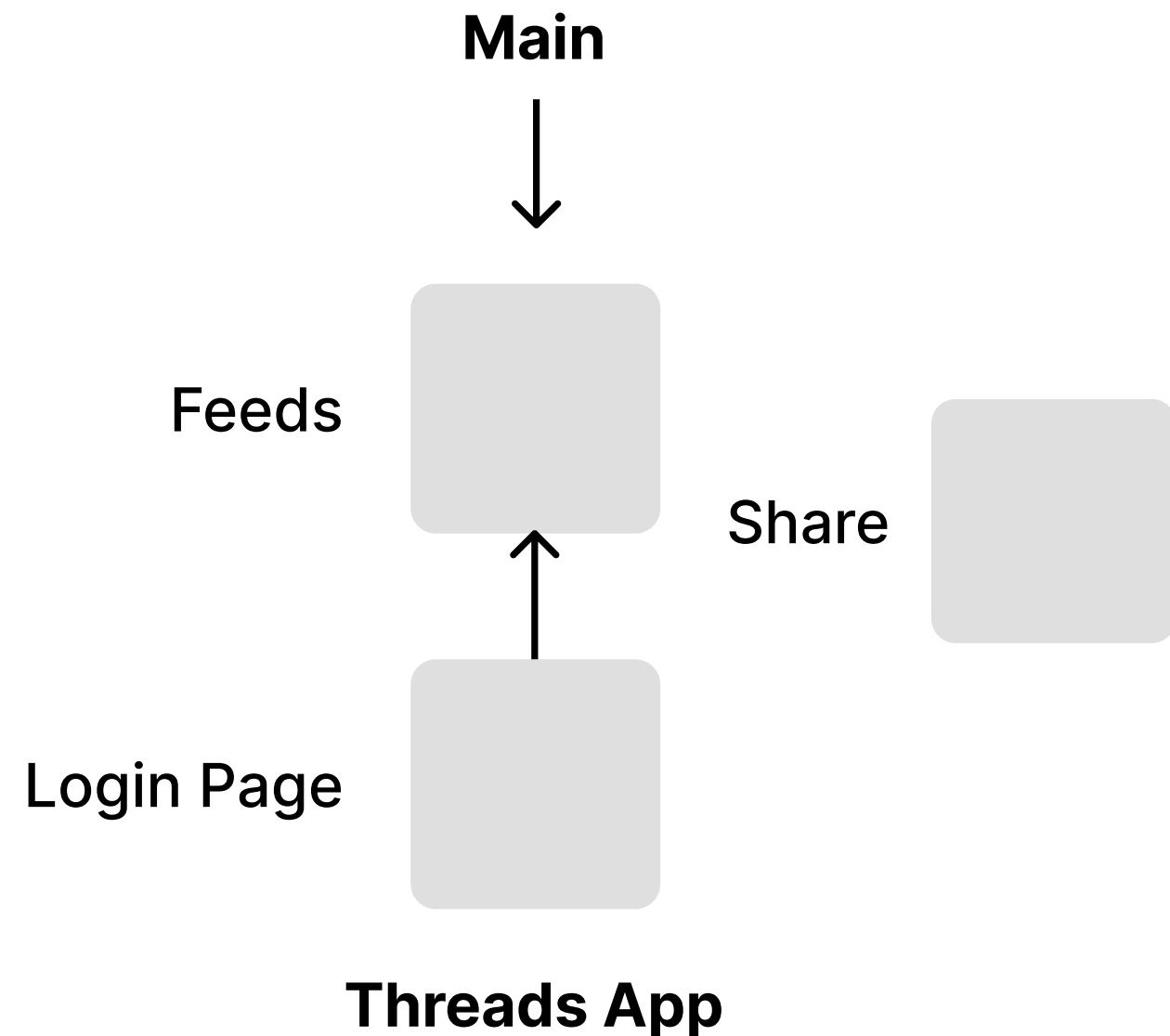


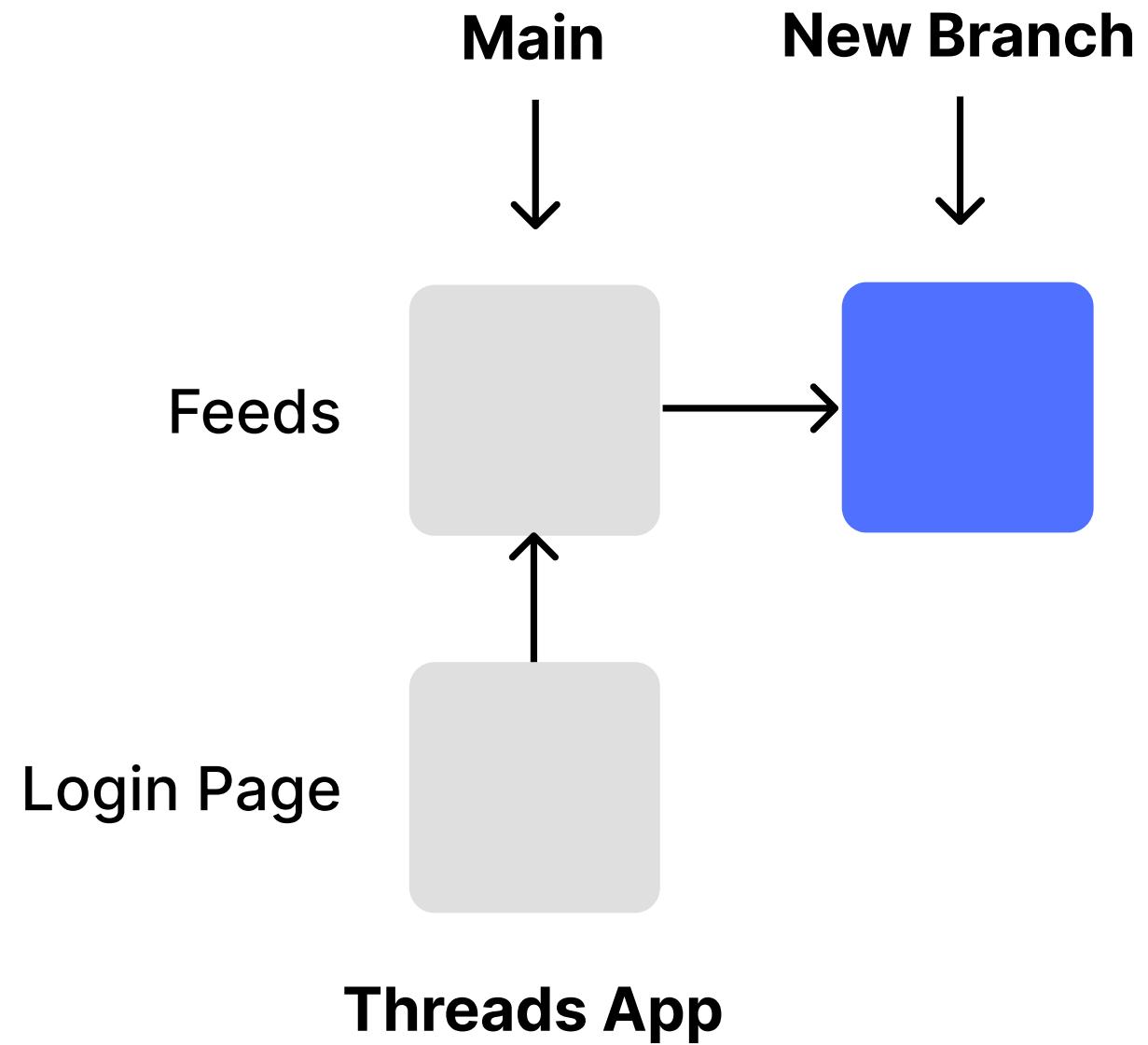


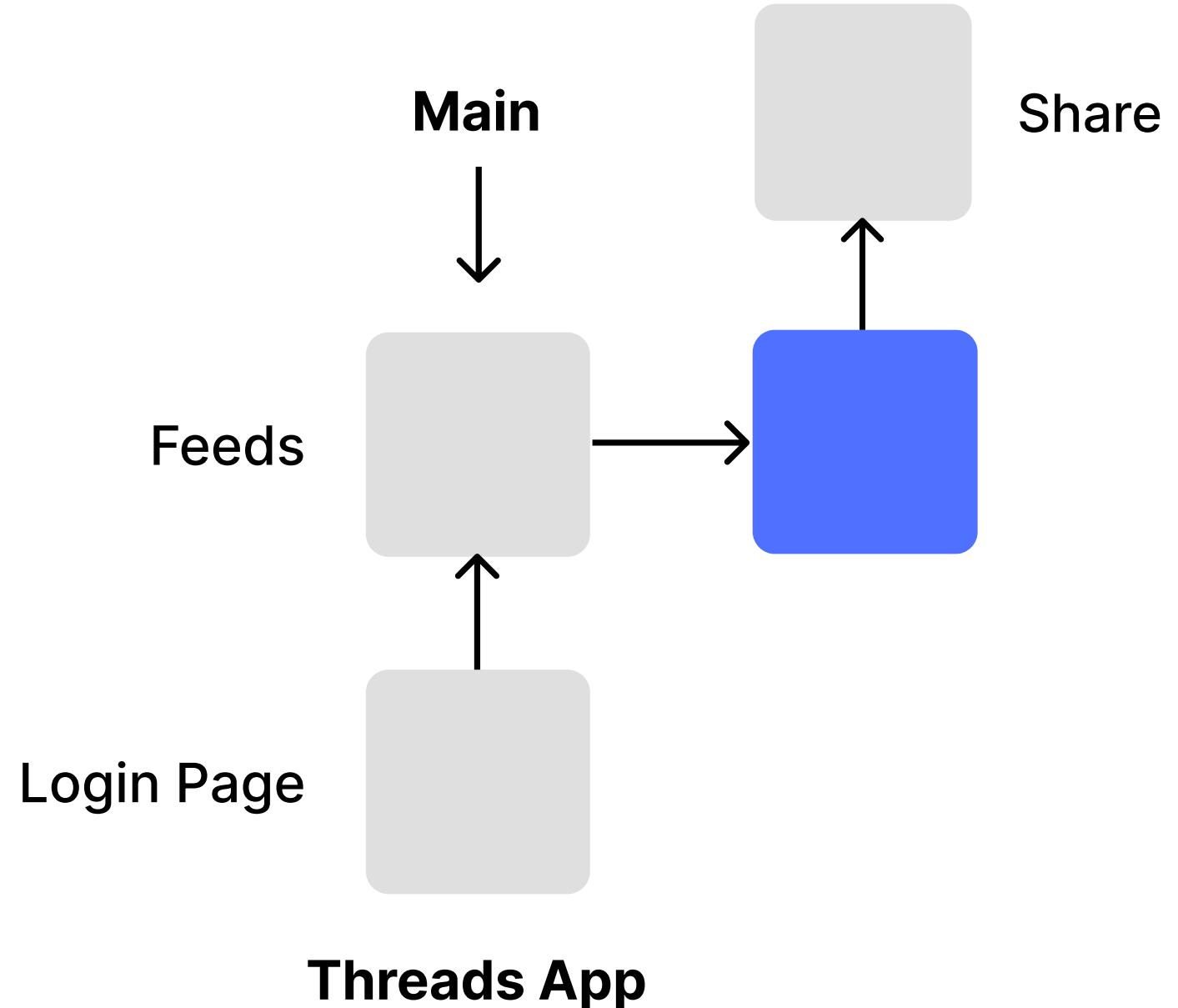


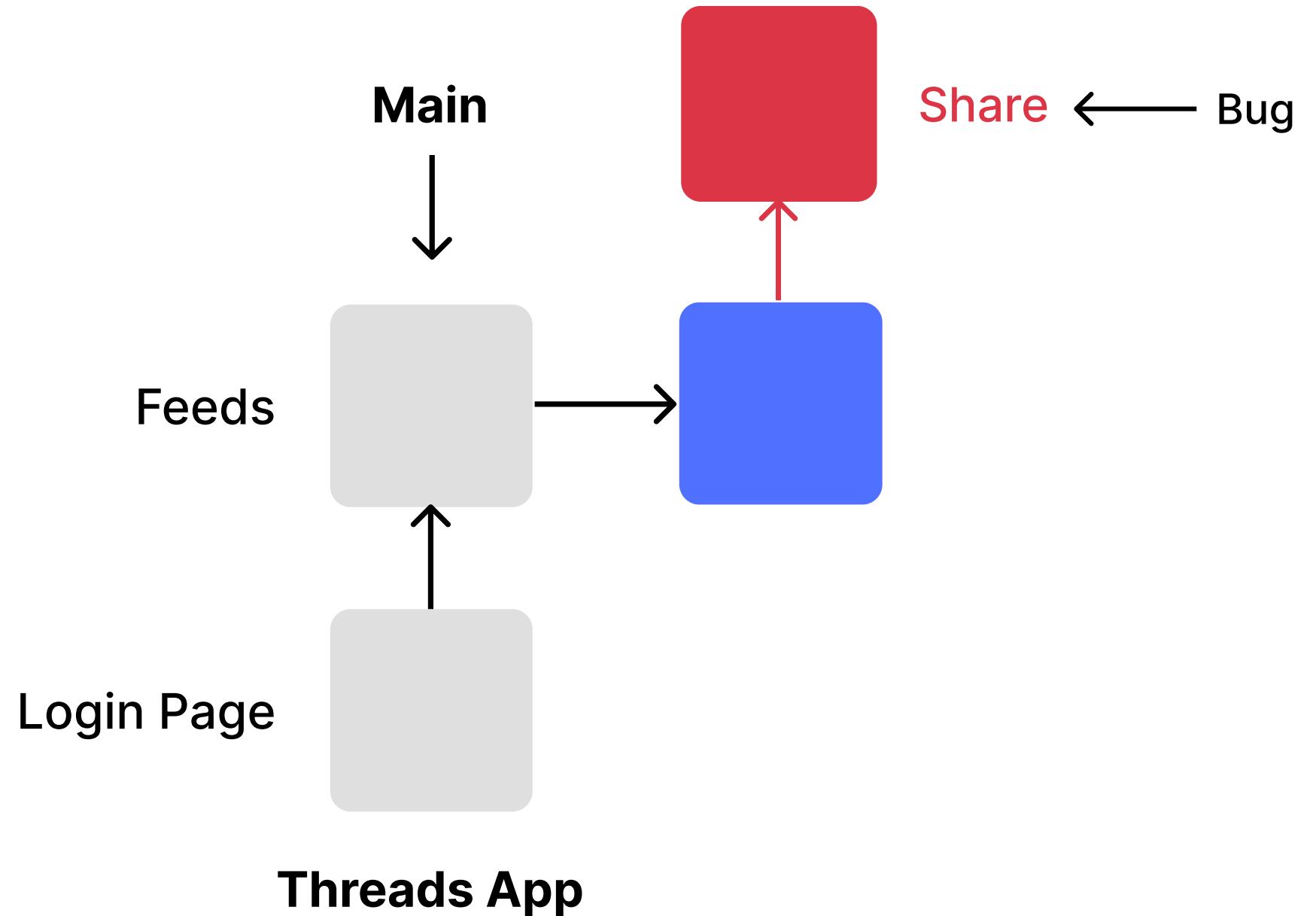












# Branching

- Branching in Git allows us to make a copy of our project to try new things without affecting the main version.
- Different people can work on different features in separate branches without affecting the main.

`git branch`

`git branch <branch-name>`

`git log --all --graph`

# **Switching Branches**

# Switching Branches

- We can switch between the our branches via following:

```
git checkout <branch-name>
```

**Warmup 19 ↩**

# Warmup 19 ↴

- Create a new directory called **wup19** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **hello.py**
- Add and commit with the message “**initial commit**”.
- Write the following in “**hello.py**”:

**name = “Ashish Shrestha”**

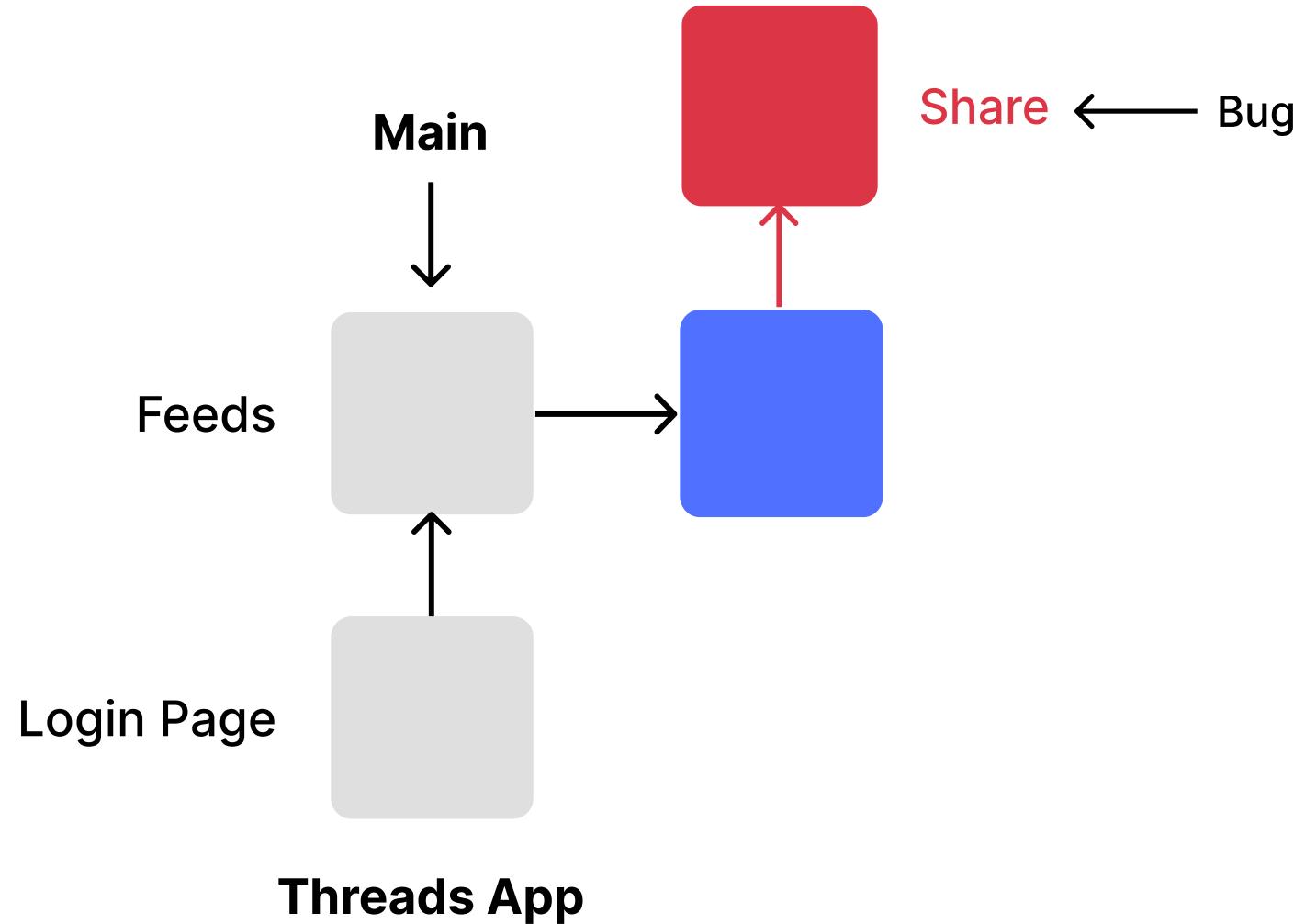
Now, add and commit with the message “**add name**”.

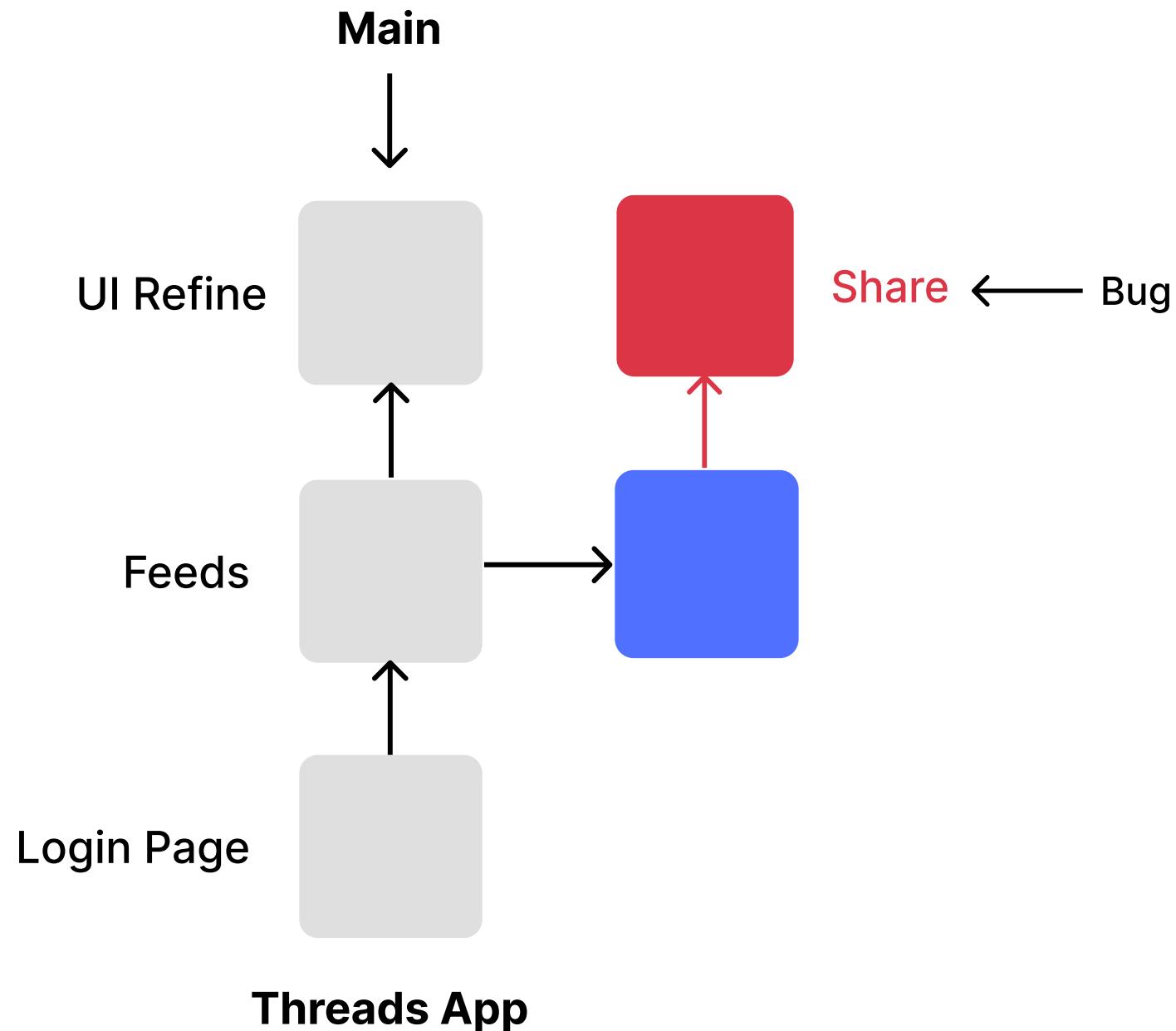
Contd..

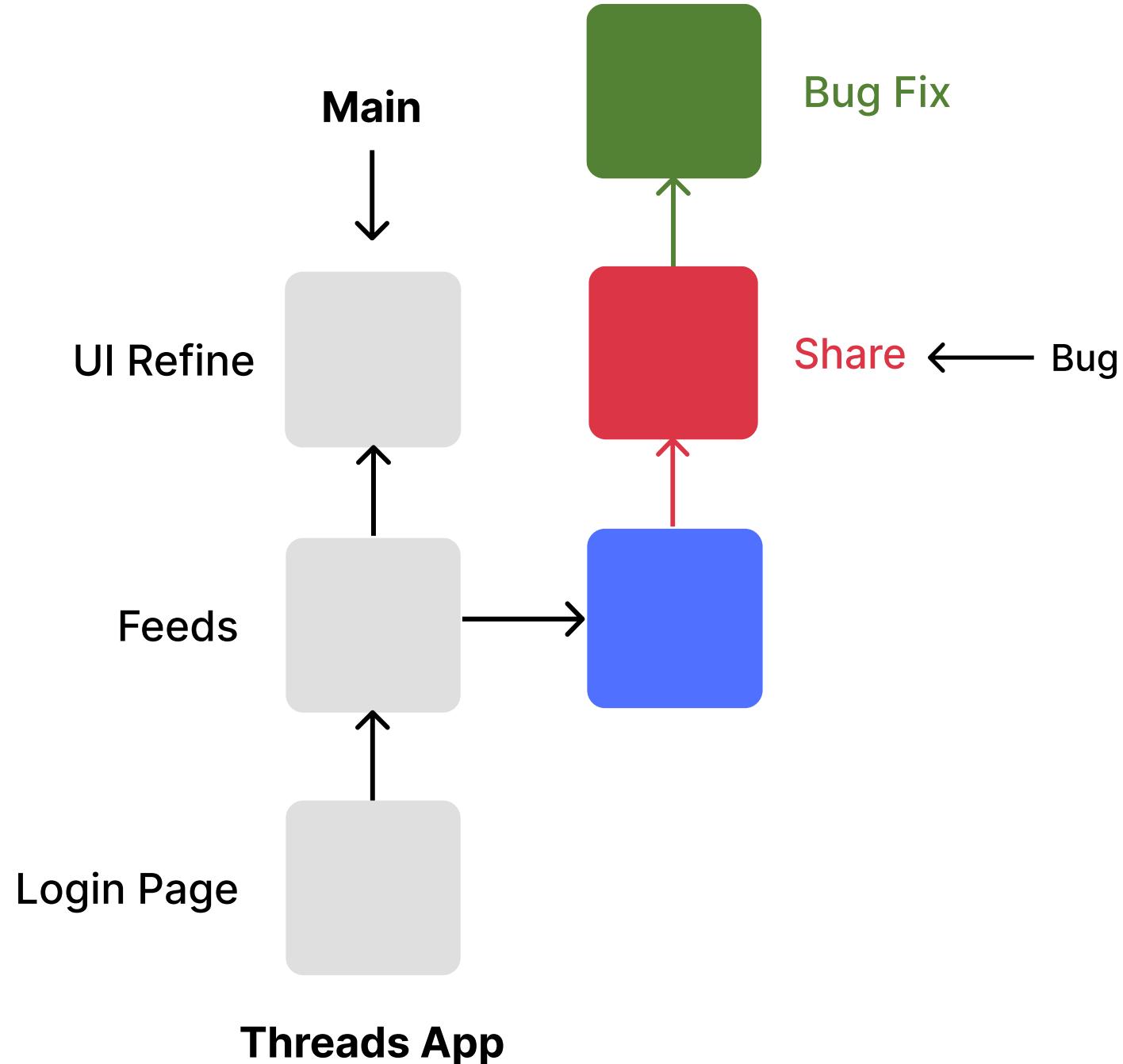
# Warmup 19 ↧

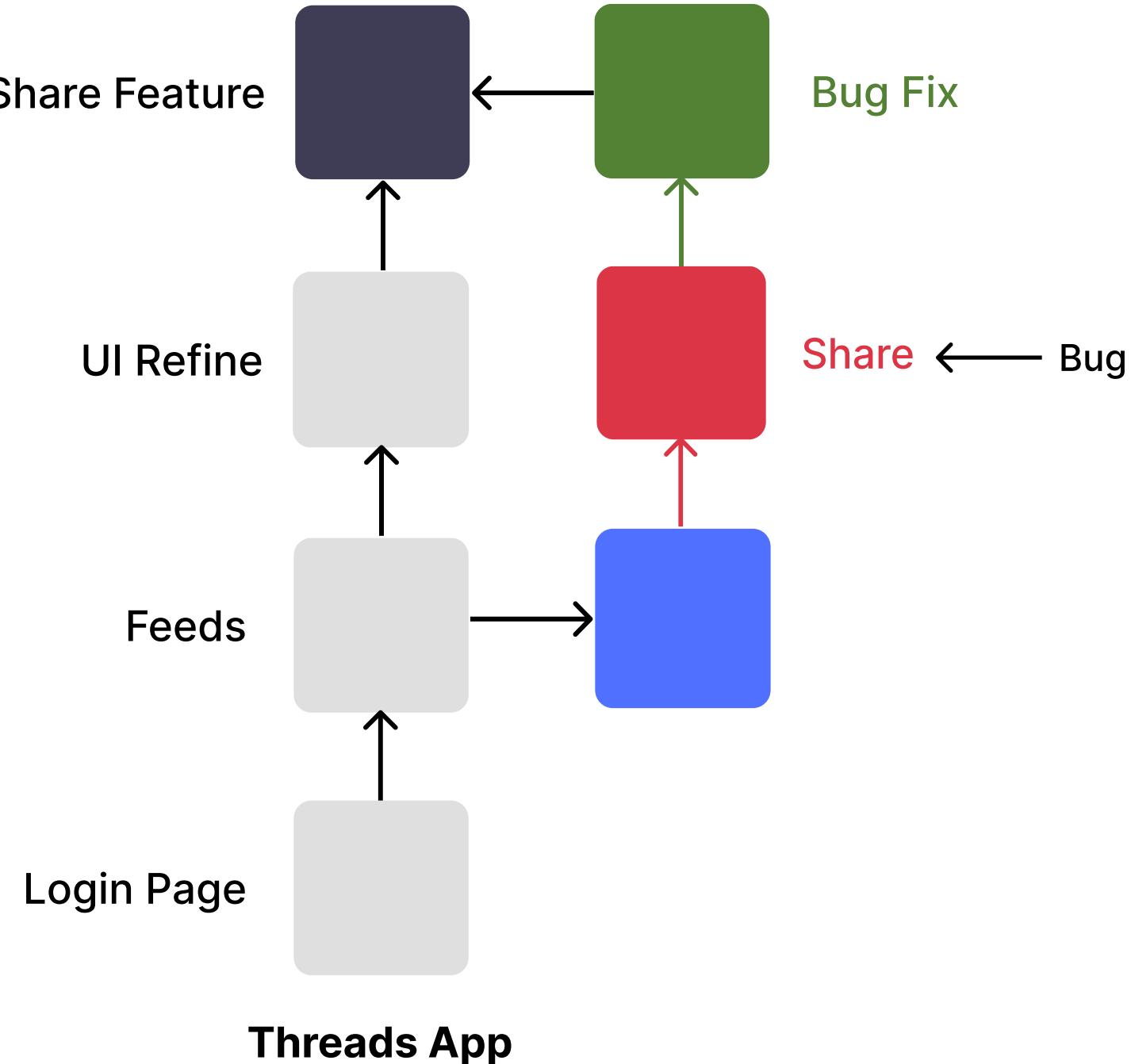
- Now, create a branch named “**feat/print**”.
- Switch to the “**feat/print**” branch and write the following in “**hello.py**”:  
`print(f"Hello, {name}")`
- Now, add and commit with the message “**add print**”.
- View the commit history. What do you observe?
- Checkout back again to your main branch. What do you observe?

# Merging Branches

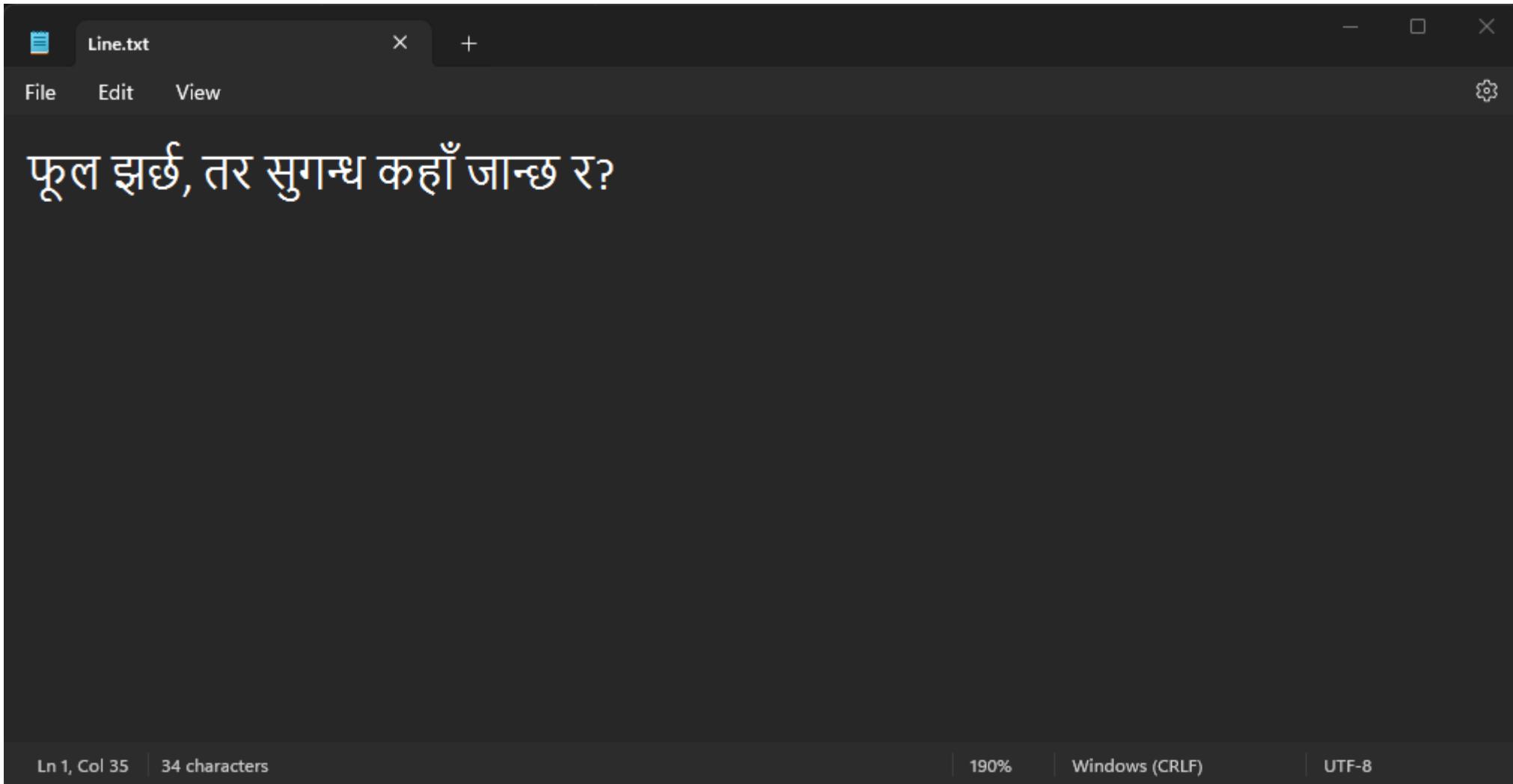






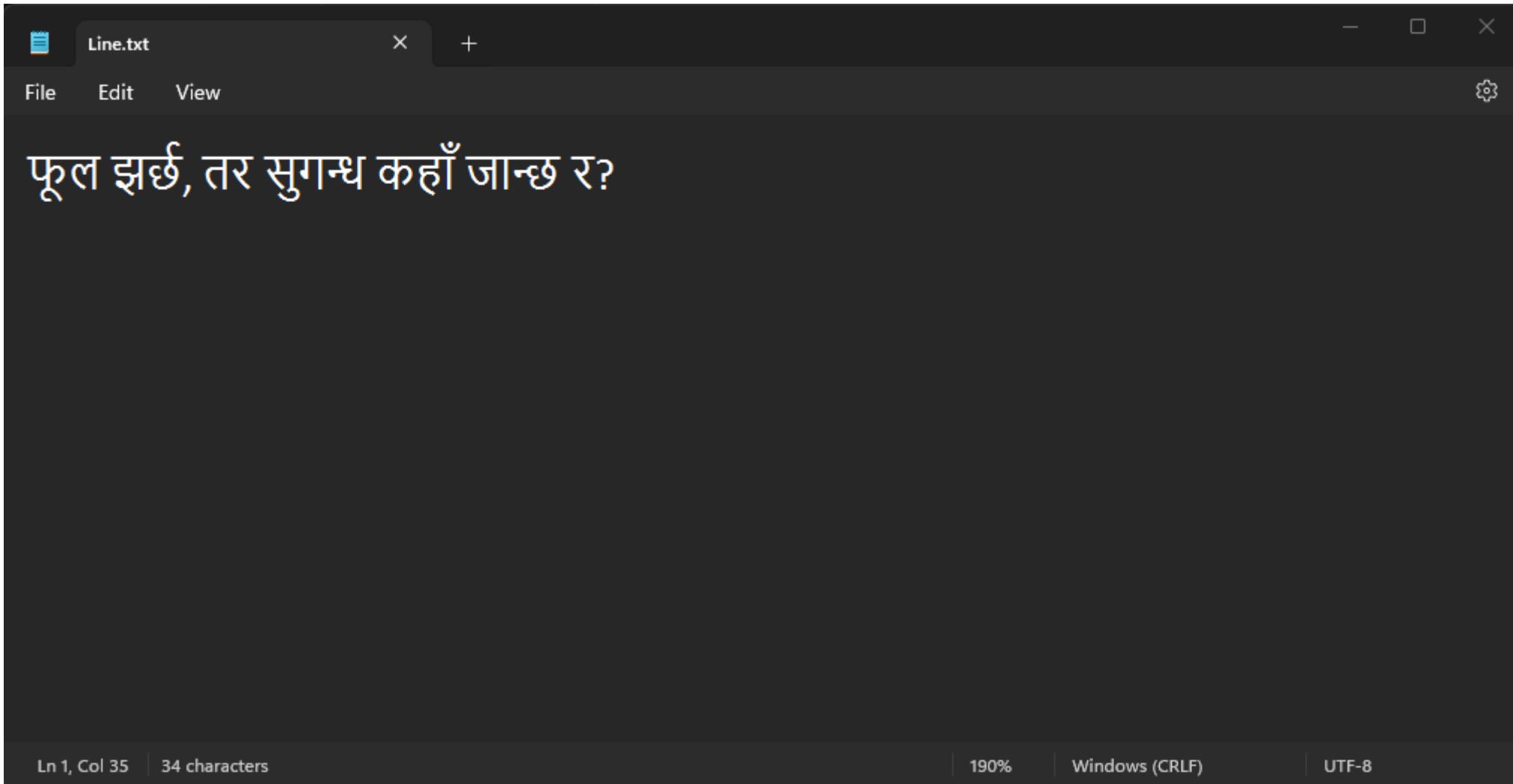


# Line @main



A screenshot of a dark-themed text editor window titled "Line.txt". The window includes standard OS X-style controls (minimize, maximize, close) and a menu bar with "File", "Edit", and "View" options. A gear icon in the top right corner indicates settings. The main text area contains the Marathi sentence "फूल झर्छ, तर सुगन्ध कहाँ जान्छ र?", which translates to "Flowers fall, but where does the fragrance go?". The status bar at the bottom shows "Ln 1, Col 35 | 34 characters" on the left, "190%" and "Windows (CRLF)" in the center, and "UTF-8" on the right.

# Line @new-branch

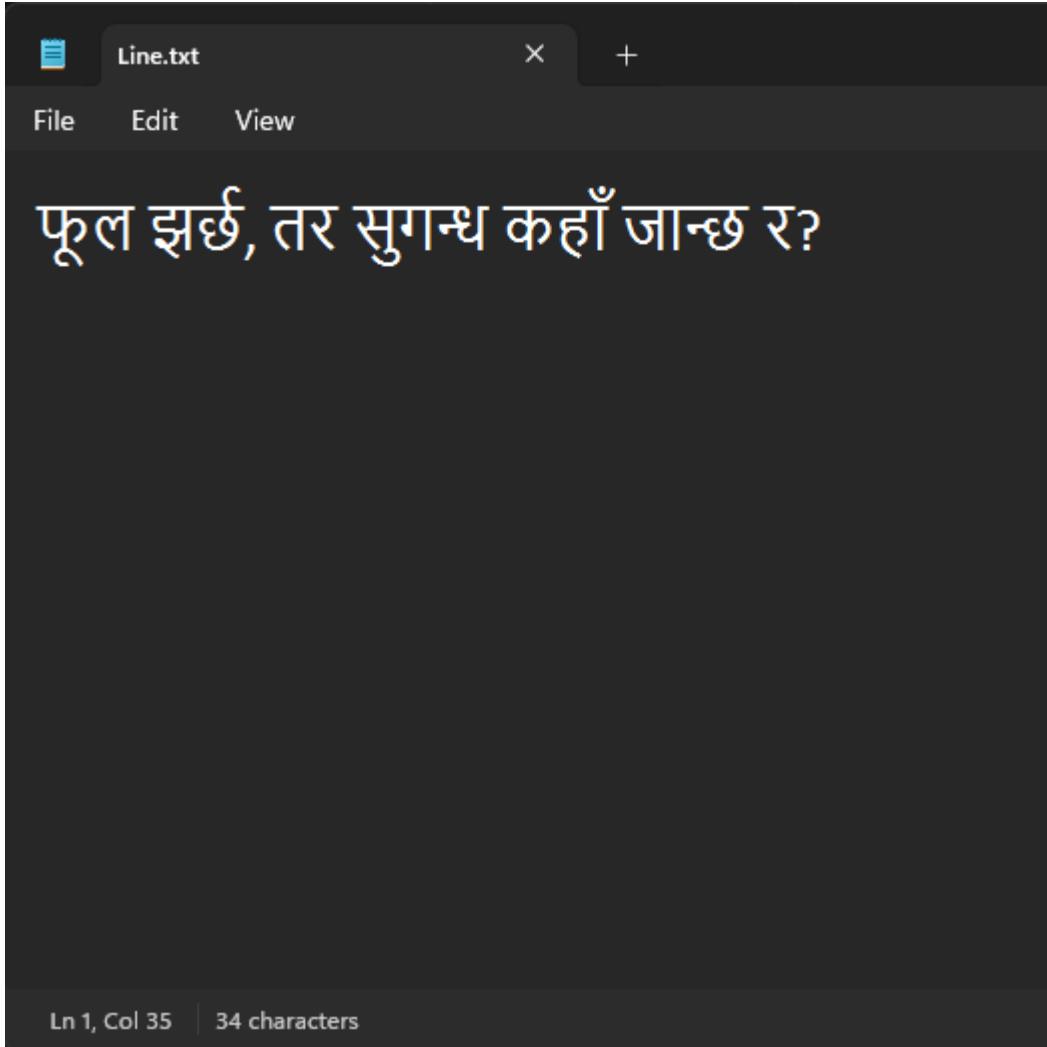


A screenshot of a dark-themed code editor window titled "Line.txt". The window includes standard OS X-style controls (minimize, maximize, close) and a menu bar with "File", "Edit", and "View" options. A gear icon in the top right corner indicates settings. The main text area contains the following Marathi text:

```
फूल झर्छ, तर सुगन्ध कहाँ जान्छ र?
```

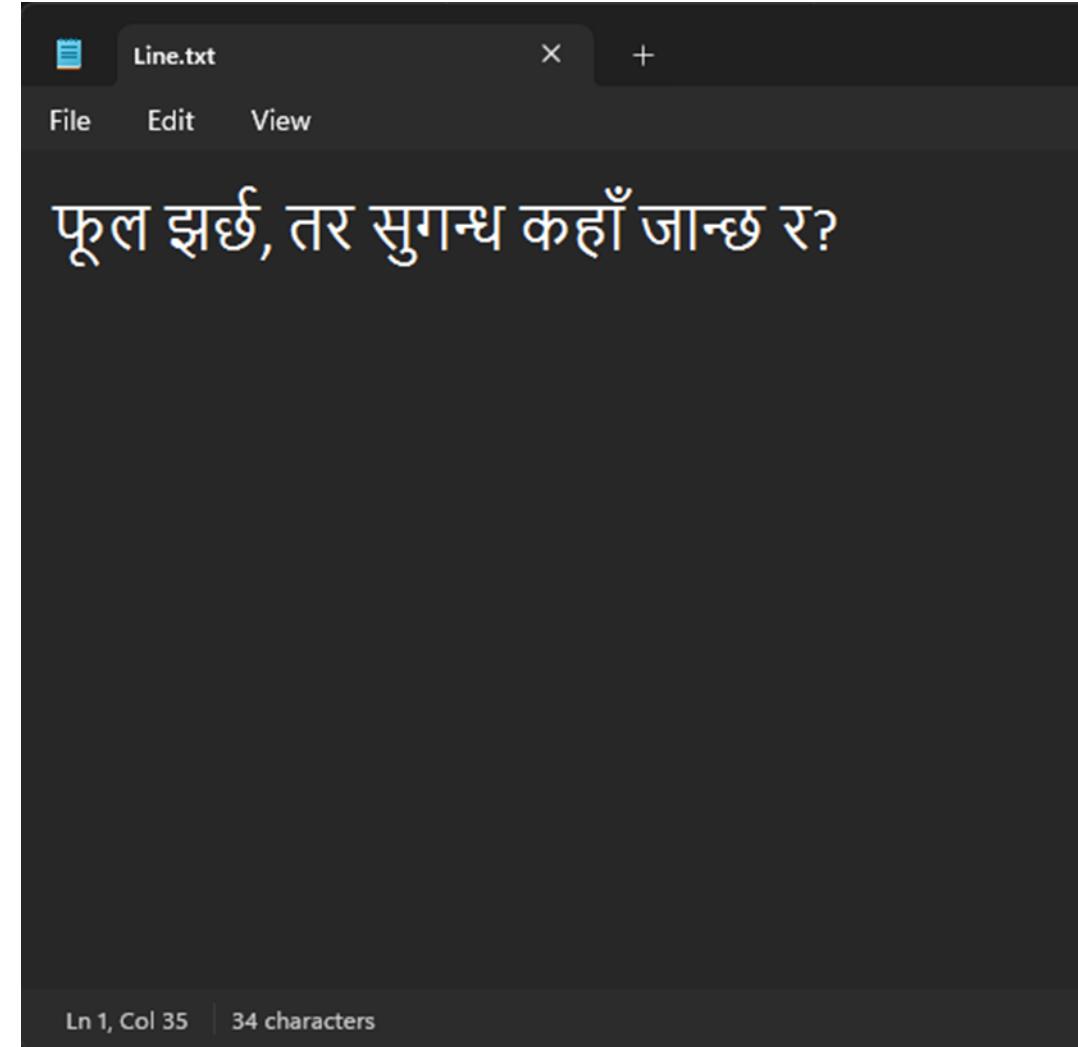
The status bar at the bottom shows "Ln 1, Col 35 | 34 characters" on the left, "190%" and "Windows (CRLF)" in the center, and "UTF-8" on the right.

## Line @main



A screenshot of a dark-themed text editor window titled "Line.txt". The window includes standard menu options: File, Edit, and View. The main content area displays the text "फूल झर्छ, तर सुगन्ध कहाँ जान्छ र?". The status bar at the bottom indicates "Ln 1, Col 35" and "34 characters".

## Line @new-branch



A screenshot of a dark-themed text editor window titled "Line.txt" on a new branch. The window includes standard menu options: File, Edit, and View. The main content area displays the text "फूल झर्छ, तर सुगन्ध कहाँ जान्छ र?". The status bar at the bottom indicates "Ln 1, Col 35" and "34 characters".

## Line @main

A screenshot of a dark-themed code editor window titled "Line.txt". The menu bar includes "File", "Edit", and "View". The main text area contains the following line of text in Gurmukhi script:

ਪੂਲ ਝਾਰਦ, ਤਰ ਸੁਗਨਧ ਕਹਾਁ ਜਾਨਛ ਰ?

In the bottom left corner, the status bar displays "Ln 1, Col 35 | 34 characters".

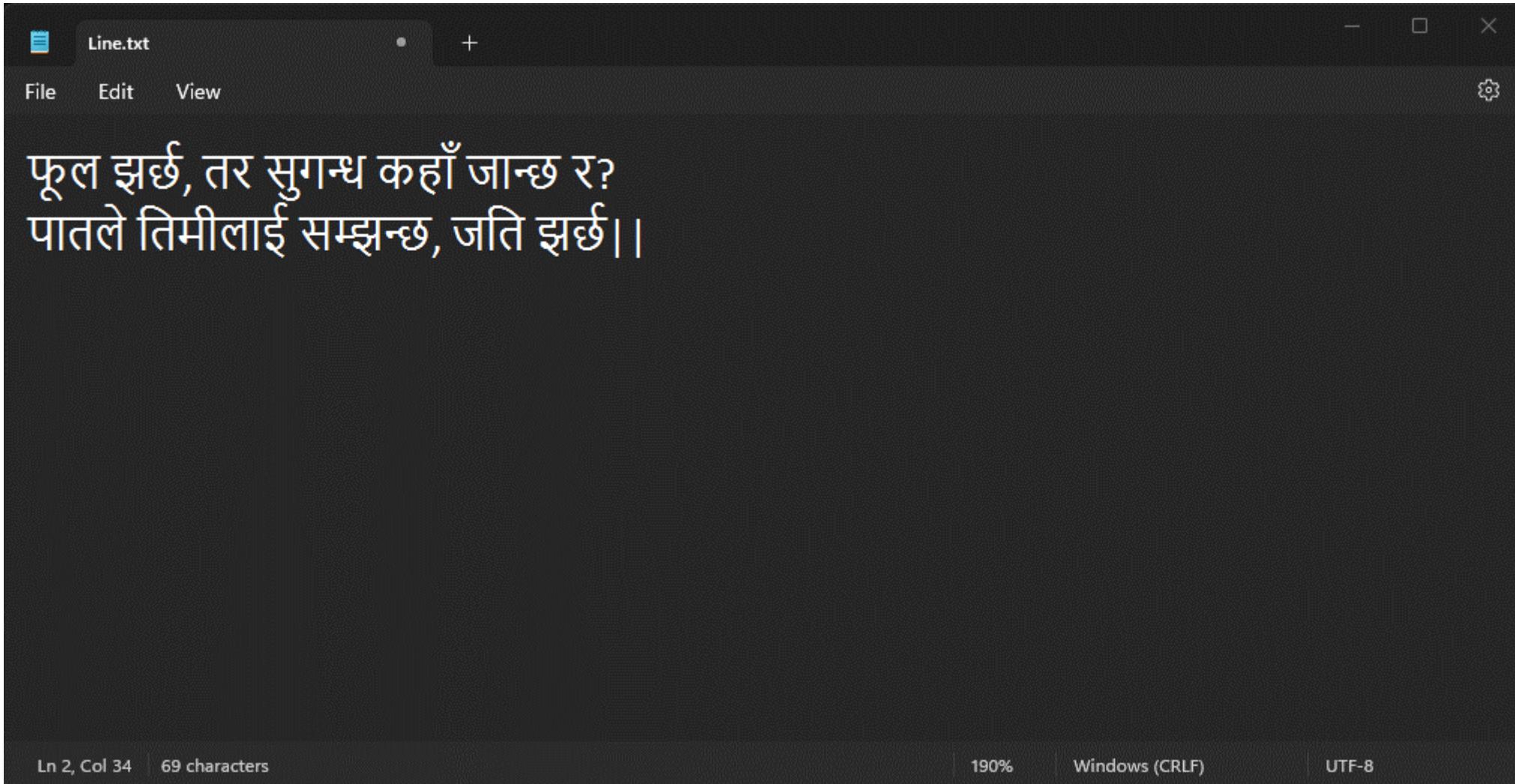
## Line @new-branch

A screenshot of a dark-themed code editor window titled "Line.txt". The menu bar includes "File", "Edit", and "View". The main text area contains the following two lines of text in Gurmukhi script:

ਪੂਲ ਝਾਰਦ, ਤਰ ਸੁਗਨਧ ਕਹਾਁ ਜਾਨਛ ਰ?  
ਪਾਤਲੇ ਤਿਮੀਲਾਈ ਸਮਝਾਨਛ, ਜਤਿ ਝਾਰਦ||

In the bottom left corner, the status bar displays "Ln 2, Col 34 | 69 characters".

# Line @main <- Line @new-branch



A screenshot of a dark-themed text editor window titled "Line.txt". The window includes standard menu options: File, Edit, View, and a settings gear icon. The text area contains the following Marathi lyrics:

फूल झार्छ, तर सुगन्ध कहाँ जान्छ र?  
पातले तिमीलाई सम्झान्छ, जति झार्छ||

The status bar at the bottom provides information: Ln 2, Col 34 | 69 characters | 190% | Windows (CRLF) | UTF-8.

# Merging Branches

- Merging branches in Git means combining changes from one branch into another.

```
git checkout <main-branch>
```

```
git merge <branch-name>
```

**Warmup 20 ♪**

# Warmup 20 ↩

- Create a new directory called **wup20** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **greet.py**
- Add and commit with the message “**initial commit**”.
- Write the following in “**greet.py**”:

***print("Good afternoon")***

Now, add and commit with the message “**add afternoon**”.

Contd..

# Warmup 20 ↩

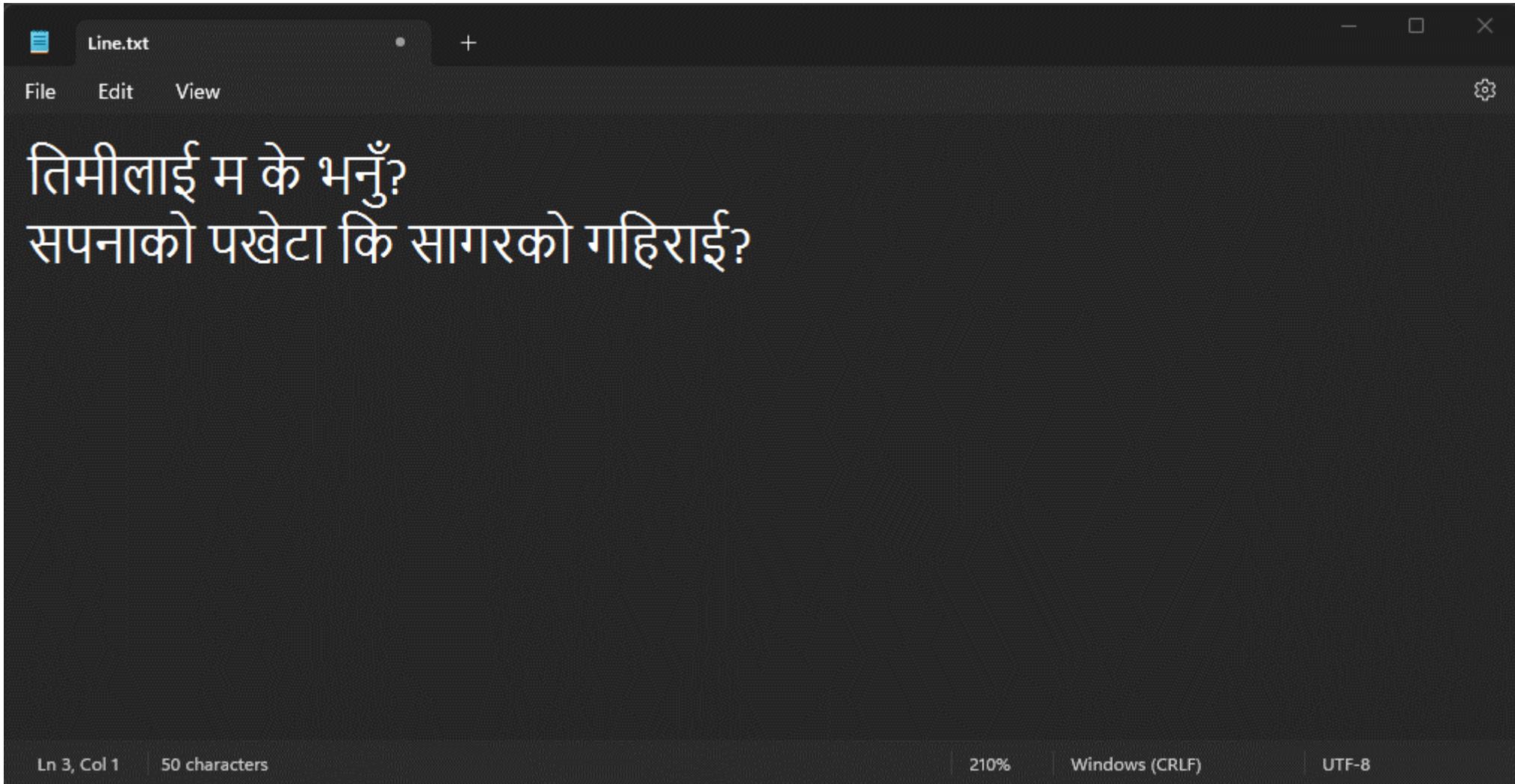
- Now, create a branch named “***evening***”.
- Switch to the “***evening***” branch and write the following in “***greet.py***”:  
`print("Good evening")`
- Now, add and commit with the message “***add evening***”.
- Checkout back again to your main branch.
- Merge the branch “***evening***” to your main branch. Observe the changes.

# Merge Conflicts

# Merge Conflicts

- A merge conflict happens in Git when two branches have changed the same part of a file differently, and Git doesn't know which change to keep.

# Line @main

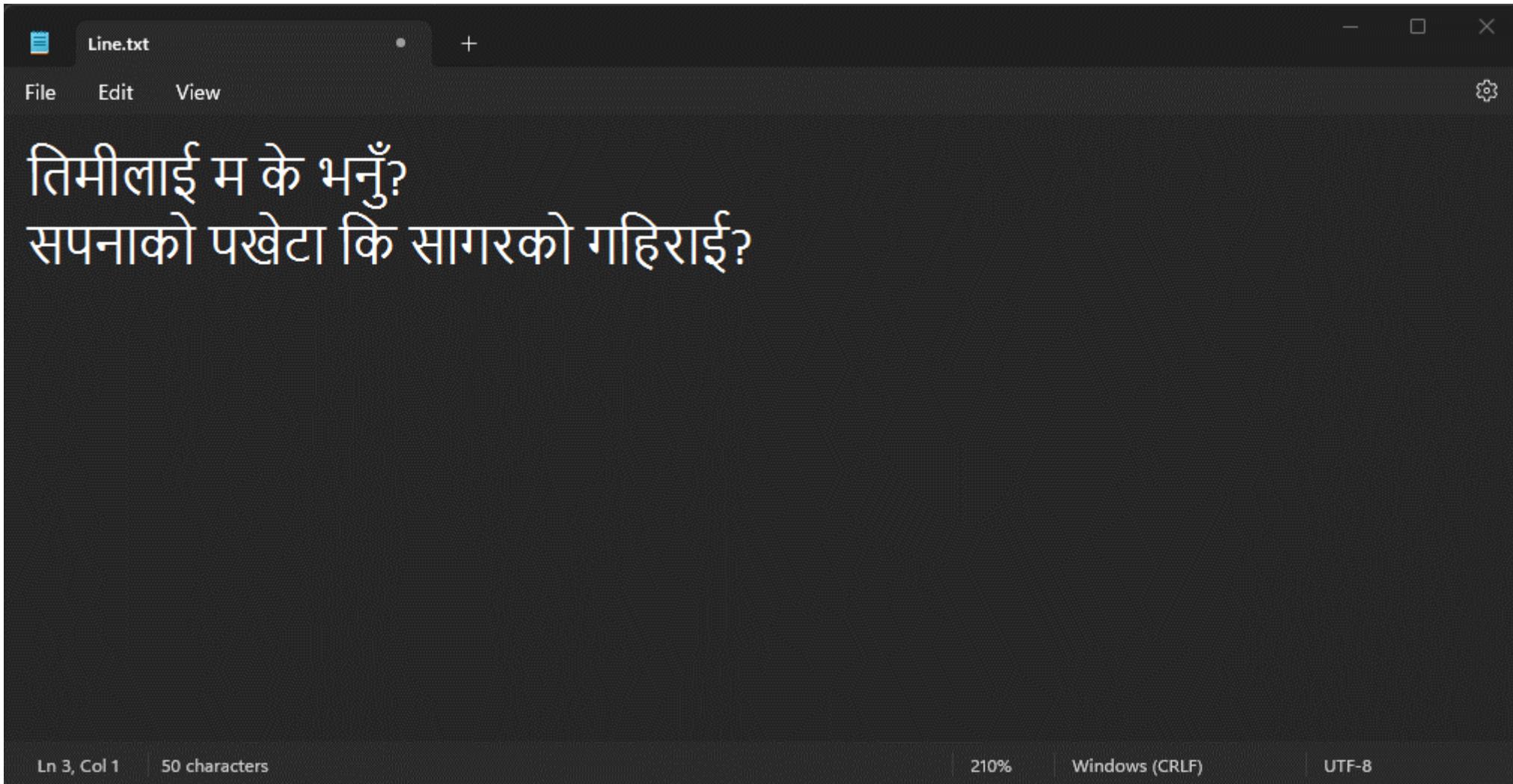


A screenshot of a dark-themed code editor window titled "Line.txt". The window includes standard OS X-style controls (minimize, maximize, close) and a menu bar with "File", "Edit", and "View" options. A gear icon in the top right corner indicates settings. The main text area contains two lines of Nepali text:

```
तिमीलाई म के भनुँ?  
सपनाको पखेटा कि सागरको गहिराई?
```

The status bar at the bottom shows "Ln 3, Col 1" and "50 characters" on the left, "210%" and "Windows (CRLF)" in the center, and "UTF-8" on the right.

# Line @new-branch



A screenshot of a dark-themed code editor window titled "Line.txt". The window includes standard OS X-style controls (minimize, maximize, close) and a menu bar with "File", "Edit", and "View" options. A gear icon in the top right corner indicates settings. The main text area contains two lines of Nepali text:

```
तिमीलाई म के भनुँ?  
सपनाको पखेटा कि सागरको गहिराई?
```

The status bar at the bottom shows "Ln 3, Col 1" and "50 characters" on the left, "210%" and "Windows (CRLF)" in the center, and "UTF-8" on the right.

## Line @main

A screenshot of a dark-themed text editor window titled 'Line.txt'. The window includes a menu bar with 'File', 'Edit', and 'View' options. The main area contains the following Nepali text:

तिमीलाई म के भनुँ?  
सपनाको पखेटा कि सागरको गहिराई?

The status bar at the bottom indicates 'Ln 3, Col 1' and '50 characters'.

## Line @new-branch

A screenshot of a dark-themed text editor window titled 'Line.txt', similar to the one above. It shows the same two lines of Nepali text:

तिमीलाई म के भनुँ?  
सपनाको पखेटा कि सागरको गहिराई?

The status bar at the bottom indicates 'Ln 3, Col 1' and '50 characters'.

## Line @main

A screenshot of a dark-themed text editor window titled 'Line.txt'. The window includes a menu bar with 'File', 'Edit', and 'View' options. The text area contains the following Nepali text:

तिमीलाई म के भनुँ?  
सपनाको पखेटा कि सागरको गहिराई?

The status bar at the bottom indicates 'Ln 3, Col 1' and '50 characters'.

## Line @new-branch

A screenshot of a dark-themed text editor window titled 'Line.txt' on a new branch. The window includes a menu bar with 'File', 'Edit', and 'View' options. The text area contains the following Nepali text:

तिमीलाई म के भनुँ?  
सपना कि यथार्थ, जिउँदो कि कल्पना?

The status bar at the bottom indicates 'Ln 2, Col 34' and '52 characters'.

## Line @main

A screenshot of a dark-themed text editor window titled 'Line.txt'. The window includes a menu bar with 'File', 'Edit', and 'View' options. The main area contains the following text:

तिमीलाई म के भनुँ?  
फूलझैं मखमली कि जूनझैं उज्यालो?

In the bottom left corner, status information reads 'Ln 2, Col 32 | 50 characters'.

## Line @new-branch

A screenshot of a dark-themed text editor window titled 'Line.txt' on a new branch. The window includes a menu bar with 'File', 'Edit', and 'View' options. The main area contains the following text:

तिमीलाई म के भनुँ?  
सपना कि यथार्थ, जिउँदो कि कल्पना?

In the bottom left corner, status information reads 'Ln 2, Col 34 | 52 characters'.

# Solving Merge Conflicts

- Accept Current Change
- Accept Incoming Change
- Accept Both Changes

## Line @main

```
Line.txt
```

File Edit View

तिमीलाई म के भन्नुँ?  
फूलझैं मखमली कि जूनझैं उज्यालो?

Ln 2, Col 32 | 50 characters

**Current Change**

## Line @new-branch

```
Line.txt
```

File Edit View

तिमीलाई म के भन्नुँ?  
सपना कि यथार्थ, जिउँदो कि कल्पना?

Ln 2, Col 34 | 52 characters

**Incoming Change**

**Warmup 21** ↗

# Warmup 21 ↪

- Create a new directory called **wup21** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **greet.py**
- Add and commit with the message “**initial commit**”.
- Write the following in “**greet.py**”:  
**`print("Good morning")`**  
Now, add and commit with the message “**add morning**”.
- Now, create a branch named “**afternoon**”.

Contd..

# Warmup 21 ↧

- Switch to the “**afternoon**” branch and rewrite “**greet.py**” with following:

**`print("Good afternoon")`**

Now, add and commit with the message “**add afternoon**”.

- Checkout back again to your main branch.
- Rewrite “**greet.py**” with following:

**`print("Good evening")`**

Now, add and commit with the message “**add evening**”

Contd..

# Warmup 21 🌶

- View the commit history.
- Merge the branch “**afternoon**” to your main branch. Observe the conflict.
- Resolve the conflict.

# **Removing Branch**

# Removing Branch

- We can remove our branch by following:

```
git branch -d <branch-name>
```

```
git push remote-name -delete <branch-name>
```

# **Collaborators**

# Collaborators

- Adding collaborators in GitHub lets multiple people work together on the same project.
- It makes development faster, improves teamwork, and enables easy feedback through code reviews and suggestions.

**Warmup 22 ↩**

# Warmup 22 ↪

- Create a new remote directory called ***remote-wup22***.
- Add at least 2 collaborators on your GitHub repo.
- Clone the repo to your device, and create a file named “bio.txt”. Write one line bio to the file:  
**[Ashish Shrestha]**  
- ***Hi, I'm Ashish Shrestha currently from 5th semester. Nice to meet you all!***
- Add and commit the file. Push the changes to your GitHub.

Contd..

# Warmup 22 ↪

- Now the repo will be cloned by the two collaborators and make their own branches as bio/Diwakar.
- Then write the bio in similar format:  
**[Diwakar Phuyal]**  
- ***Hello, I'm Diwakar Phuyal currently from 5th semester. Good Evening everyone.***
- Add and commit the file.
- Now, push your respective branches to the GitHub.

Contd..

# Warmup 22 ↩

- Now the GitHub repo owner will fetch the changes to their local directory, and merge the branches after reviewing them.
- Make sure to resolve the conflicts if necessary to keep both changes.
- Now, push the updated changes back to your GitHub repo.
- Now the updated changes will be pull by the two collaborators.
- Observe the updated changes.

# Pull Request

# Pull Request

- A Pull Request (PR) in GitHub is used to propose changes before merging them into the main branch.
- It allows code review, so others can check for mistakes or suggest improvements.

**Warmup 23 ♪**

# Warmup 23 ↪

- Fork the following repository to your GitHub from the link below:  
**<https://github.com/MMCITCLUB/git-github-session-2081>**
- Clone the repo to your device, and make sure to create your own branch as ***bio/Diwakar.***

Contd..

# Warmup 23 ↪

- Create a file named “**diwakar-bio.txt**”. Mention your GitHub profile and Linkedin as below:  
**[Diwakar Phuyal]**
  - **GitHub** [<https://www.github.com/diwacreation3>]
  - **LinkedIn** [<https://www.linkedin.com/in/diwakarphuyal>]
- Add and commit the file with the message “**add Diwakar bio**”. Push the changes to your GitHub.
- Now, create a pull request.



**Module 4 Completed**



**Module 1 Completed**



**Module 2 Completed**



**Module 3 Completed**



**Module 4 Completed**

**Thank to all of you for the  
amazing moments 😊**