Caltech | Center for Technology & Management Education

Full Stack Java Developer

simplilearn

Git

Branching

Learning Objectives

By the end of this lesson, you will be able to:

- Understand basic Git commands and their syntaxes
- Describe the concepts of branches in Git
- Learn why branching is crucial and when it is used
- List the basic Git branching commands with their syntaxes

Learning Objectives

By the end of this lesson, you will be able to:

- Understand Git merge
- Learn how to use it for merging branches
- Learn what Git rebase is and write its syntaxes
- Explain the importance of Git rebase to implement a sequence of commits across branches
- Learn about Git squash and understand how it is helpful

A Day in the Life of a Full Stack Developer

Since you are working as a full-stack developer in an organization on the application development project. You being the project lead, are responsible to manage the code stack.

To do so, your organization uses Git to manage the versions of the code. Since multiple people are working on the project, you need to verify each commit that is happening to the code.

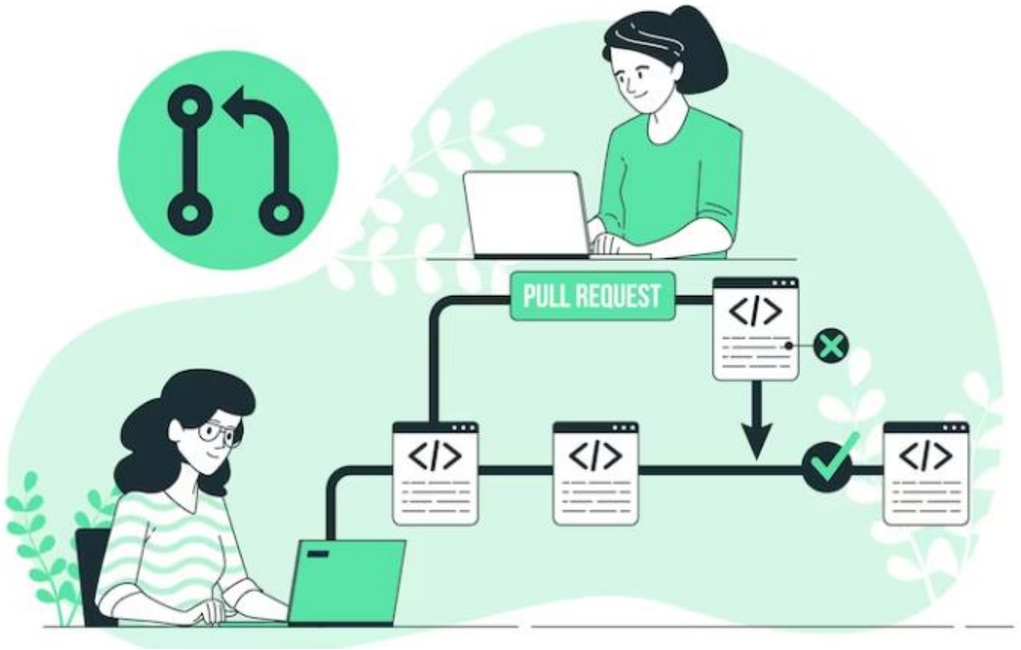
You will have to create multiple branches on Git for each feature of the application. Only once the pull request is approved by you, it will be merged in the master branch.

To understand this, explore more about branches in Git and how to create a pull request.

Visualizing Branches

Visualizing Branches

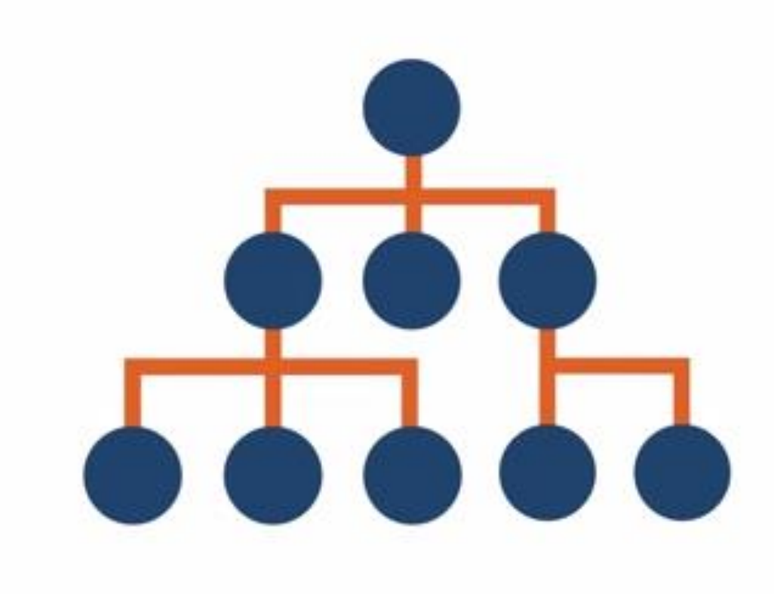
The Git branch command is employed to create, list, and delete branches locally.



It does not allow moving between the branches or putting a forked history back.

Visualizing Branches

Branches are created to fix bugs and add new features to a project.

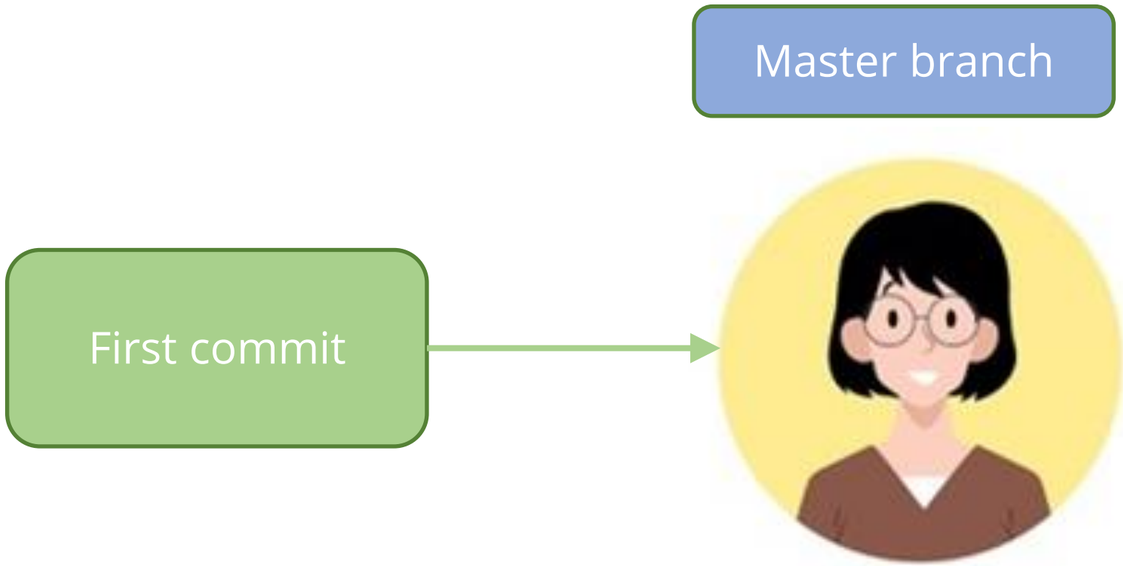


It is a process of requesting a new working directory, staging location, and project history.

Visualizing Branches

The Git branch command is used to create, list, and delete branches in the local repository.

Master branch is the default branch in Git.



Visualizing Branches

The master branch pointer advances when a commit is started.

- There is only one master branch in a repository.
- All changes are merged back into the master branch.

Basic Git Branch Commands

Basic Git Branching Commands

The branch listing command is used to list local branches.

Syntax:

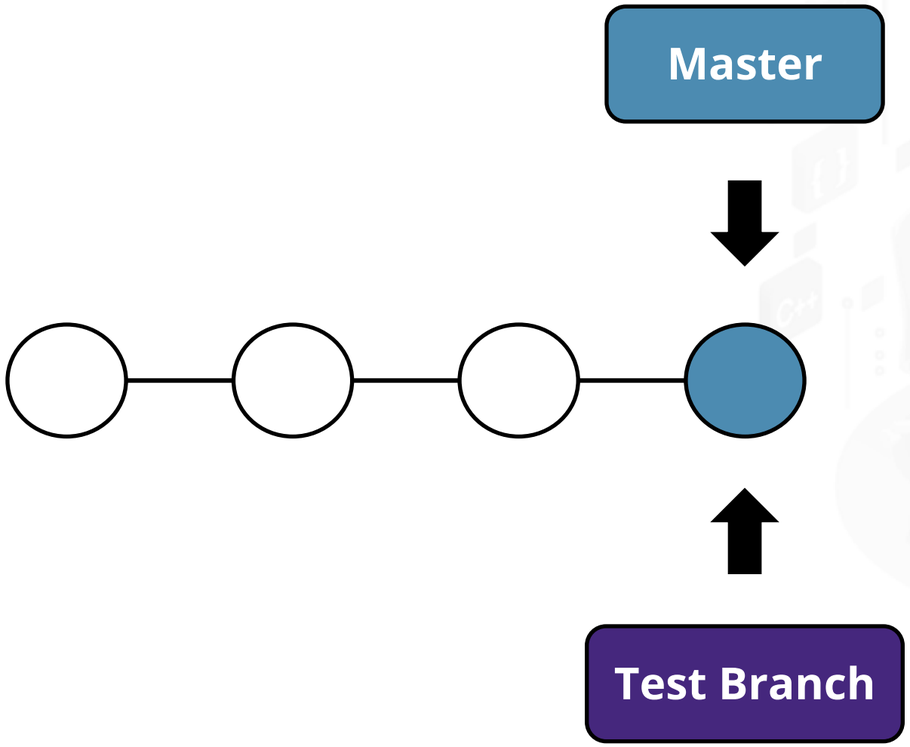
```
git branch

or

git branch -a
```

Basic Git Branching Commands

The new branch is used to create a new branch in the local repository.

Syntax

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



Basic Git Branching Commands

The delete command is used to delete a branch from the repository.

Syntax:

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

Basic Git Branching Commands

Example:

```
git branch -d test_branch
```

If the changes are not merged, Git will throw the following error while deleting:

```
error: The branch 'test_branch' is not fully merged with
'master.'

If you are sure to delete it, run 'git branch -D test_branch.'
```

Basic Git Branching Commands

To delete a branch permanently:

```
git branch -D test_branch
```

To delete a remote branch:

```
git push origin --delete test_branch

Or

git push origin:test_branch
```

Basic Git Branching Commands

Syntax of Checkout Branch:

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

Basic Git Branching Commands

Executing the git branch <new_branch> before invoking the git checkout <new_branch>:

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

The git checkout accepts an a **-b** argument that helps to create a new branch and directly switch to it.

```
git checkout -b checkout_branch
```

Basic Git Branching Commands

To point HEAD to the tip of <branch-name> execute this command.

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

Working with Branches

Problem Statement:

Work with Branches in Git

ASSISTED PRACTICE

Assisted Practice: Guidelines

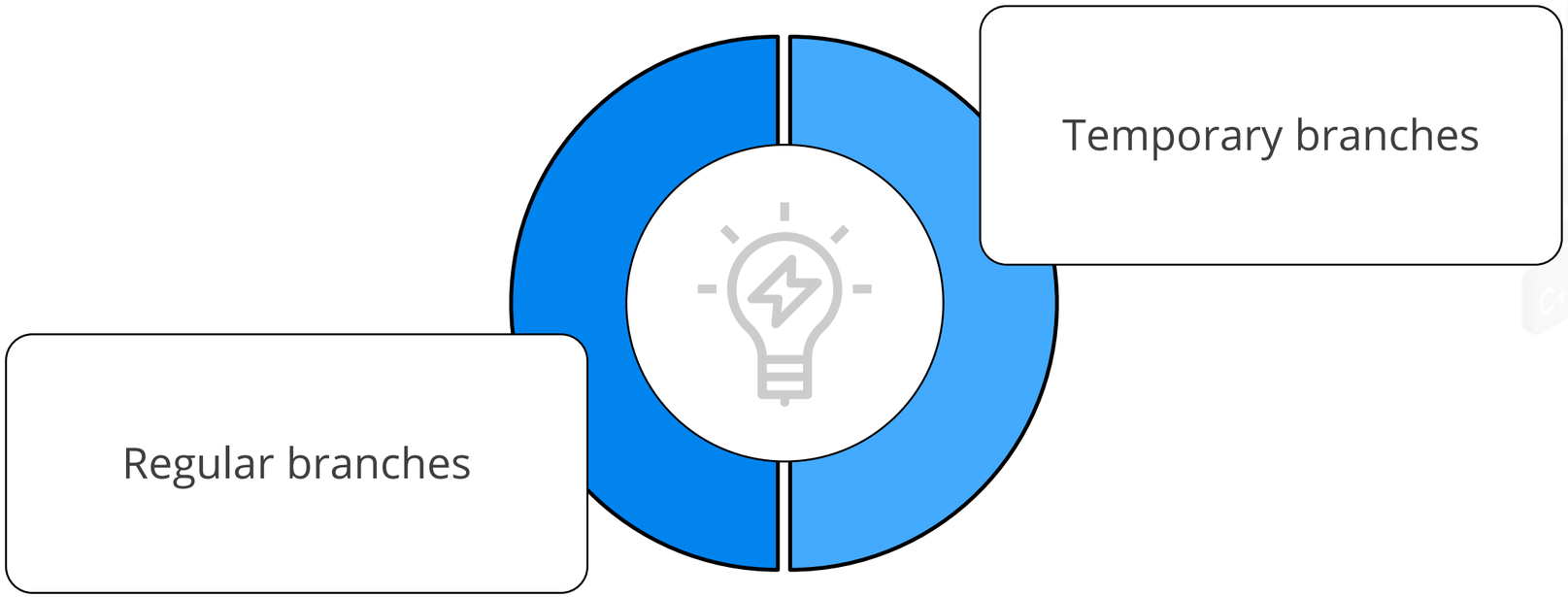
Steps to create and work with branches are:

1. Create and work with branches

Branch Naming Conventions

Branch Naming Conventions

Git Branching can be broadly classified into two categories:



Branch Naming Conventions

Regular branches are the permanent branches of the repository.

Development (dev) is the main development branch.

Changes in the dev branch undergo peer reviews, unit testing and functional testing

Gets merged with the master branch

Branch Naming Conventions

Master (master) is the default branch available in the Git repository.

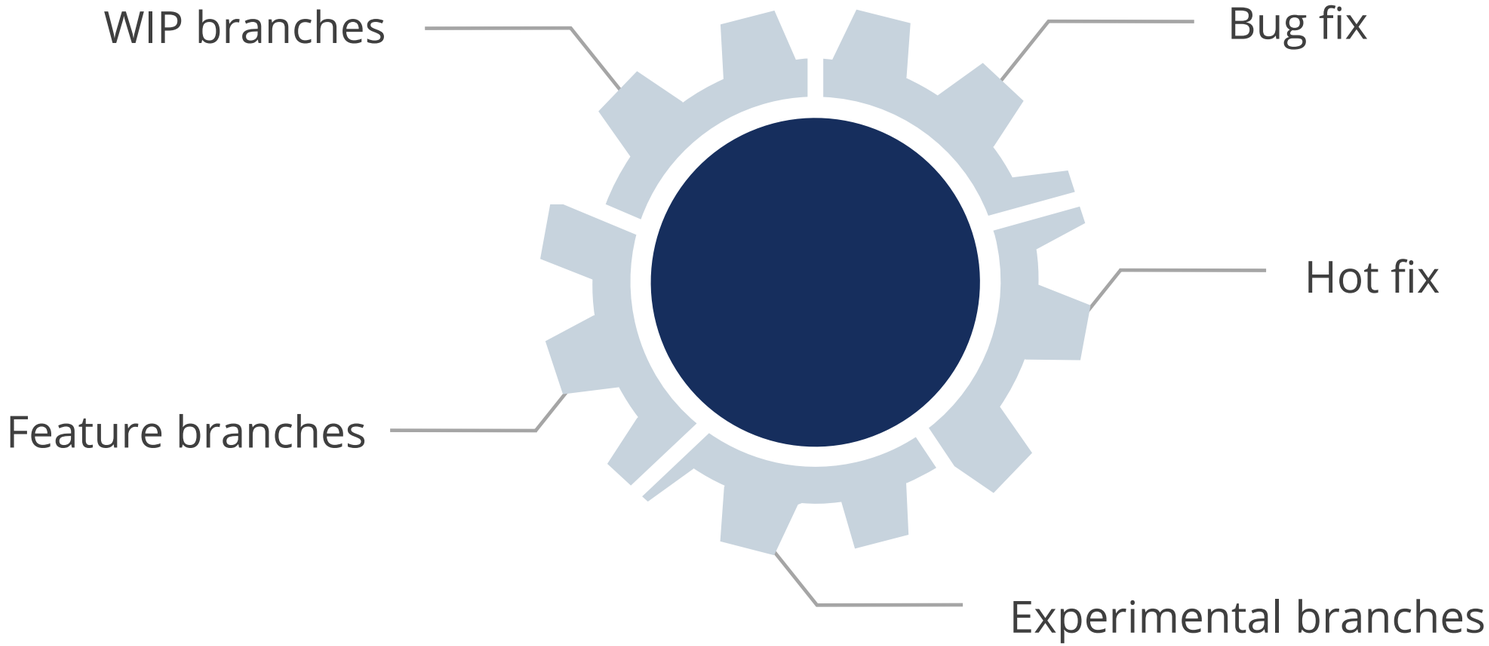
- This should be the most stable and updated branch all the time.
- All the team members are responsible for keeping the master stable and up to date.

Branch Naming Conventions

QA or test branch will be managed by the QA team and contains all the code for QA automation testing.

Branch Naming Conventions

Temporary branches can be created and deleted when needed.



Git Merge

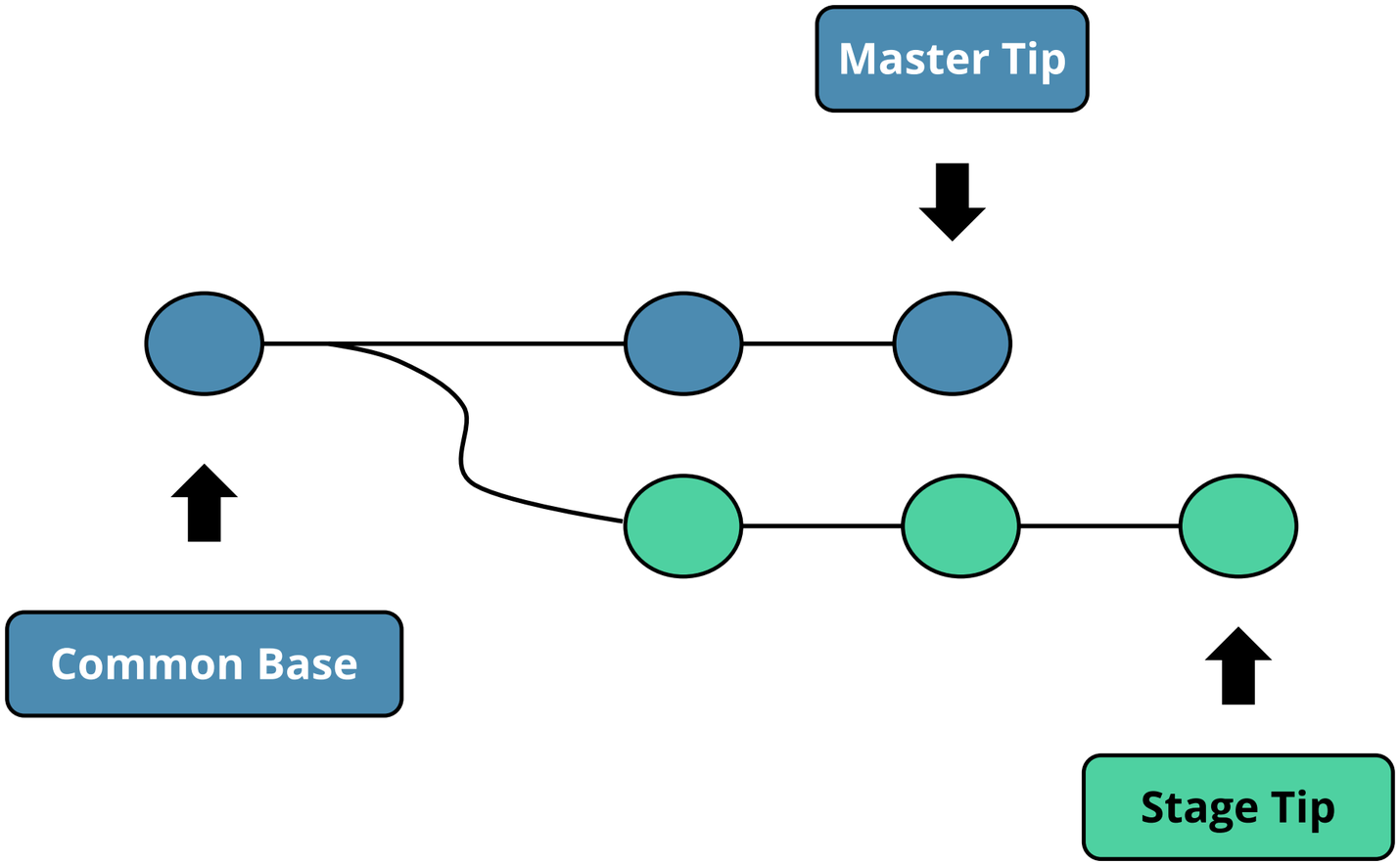
Git Merge

The git merge command works together with the git checkout command to select the current branch.

The git branch command with the -d as an argument deletes the outdated branch.

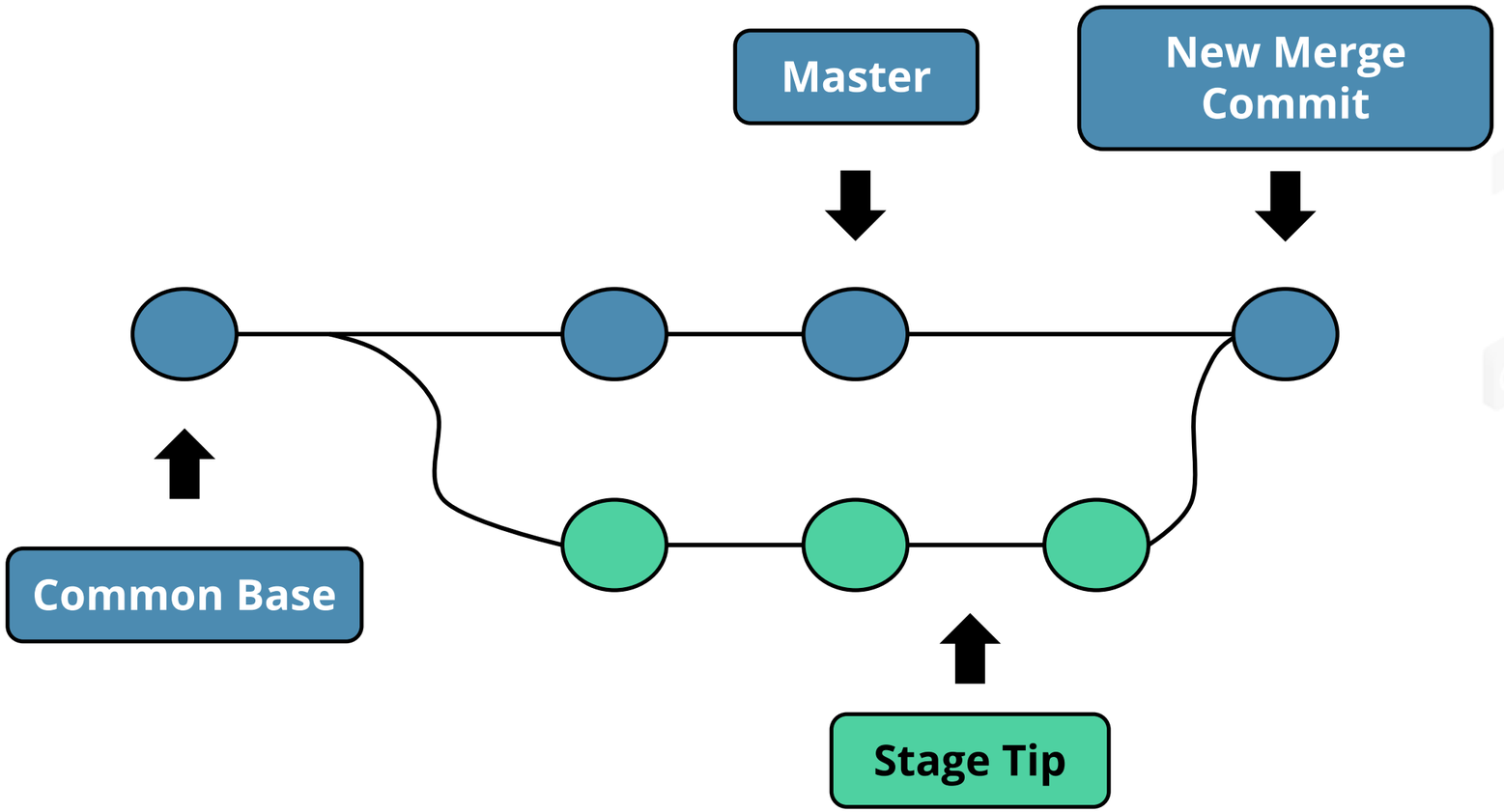
Git Merge

It is used to join two git branches and combine multiple commits into one commit.

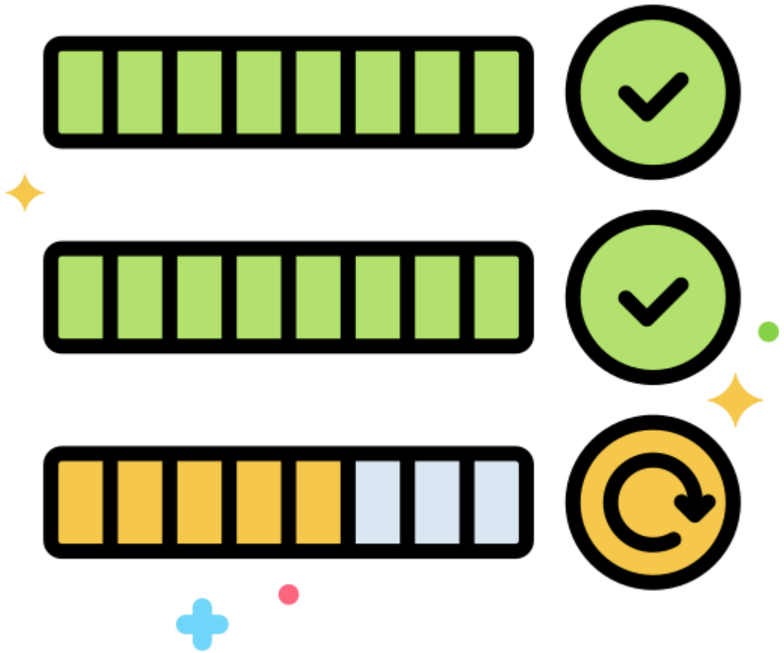


Git Merge

Git programmatically merges the two separate commits, then a new merge commit is created to tell the merge process that happened in the repository.



Git Merge



To begin:

Run git status to ensure that HEAD is pointing to the correct merge-receiving branch.

To switch to the receiving branch:

Run git checkout receiving branch

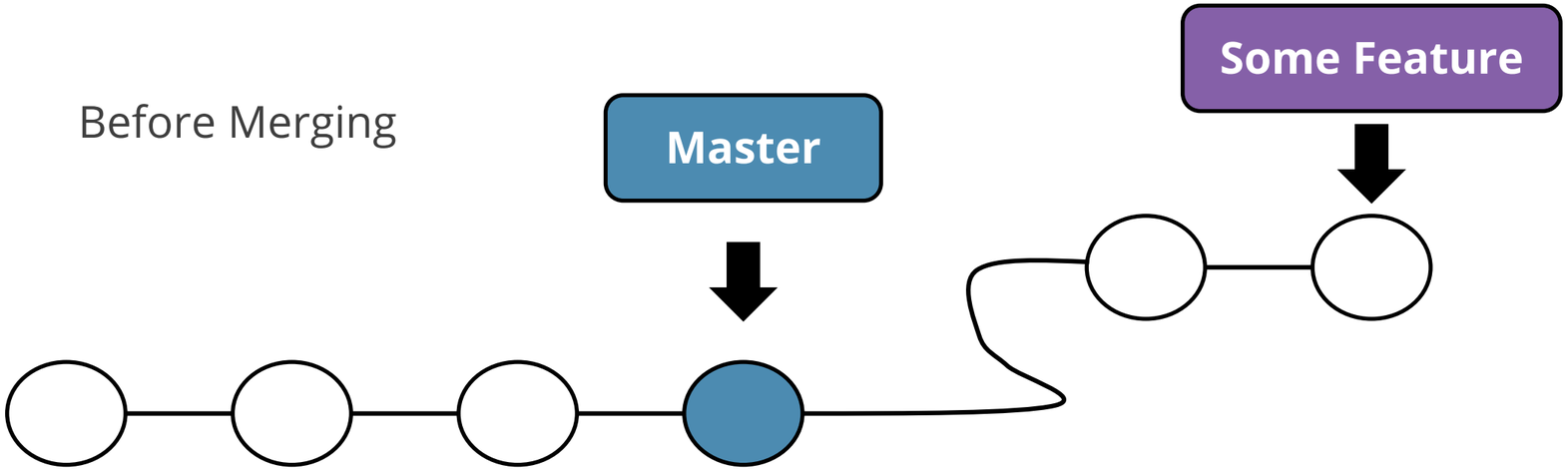
Git Merge

It uses the git pull command to update the main branch after the fetching process is finished.

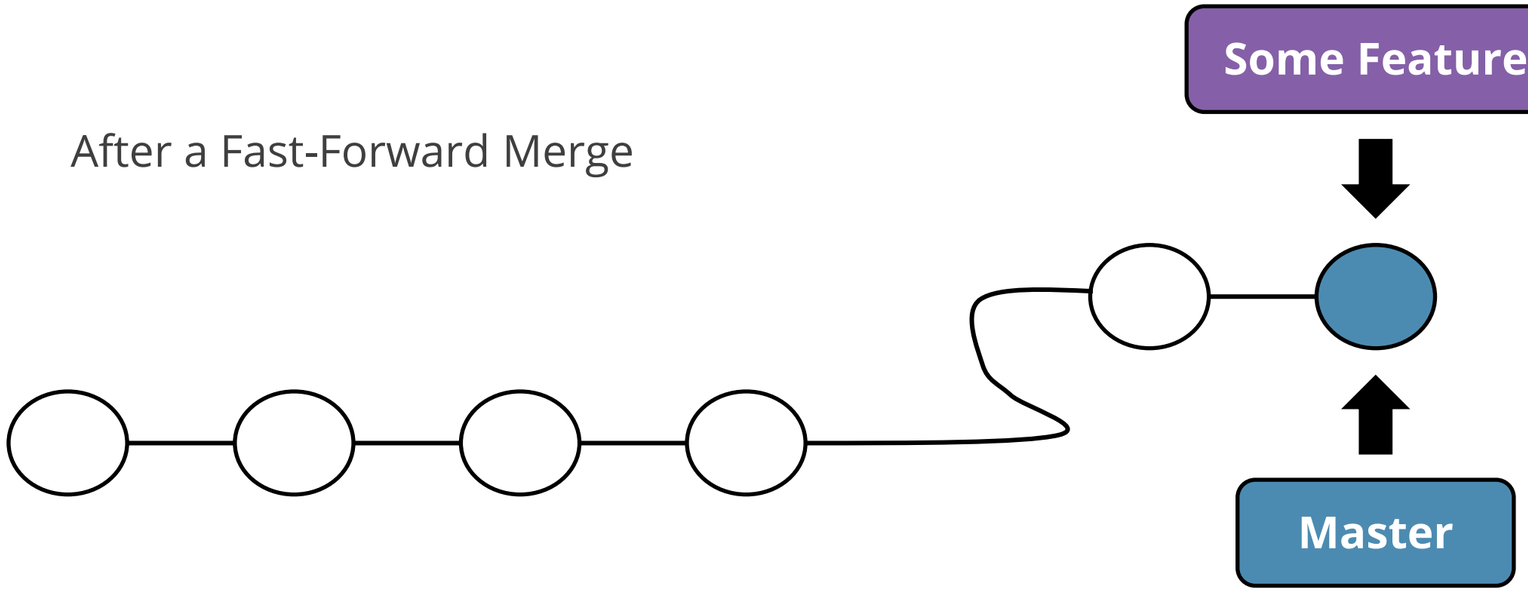
```
Running git merge <branch name>
```

Git Merge

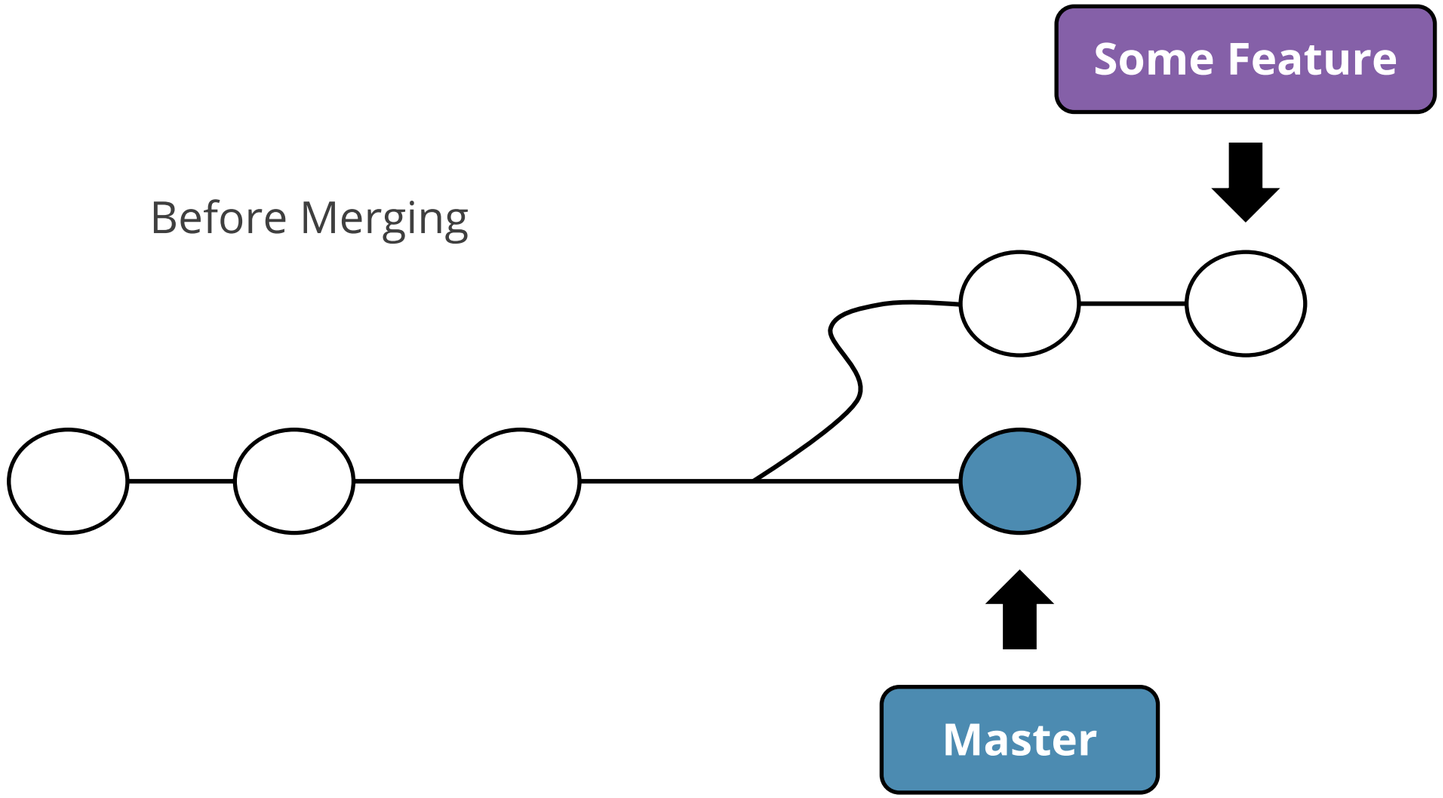
Before Merging



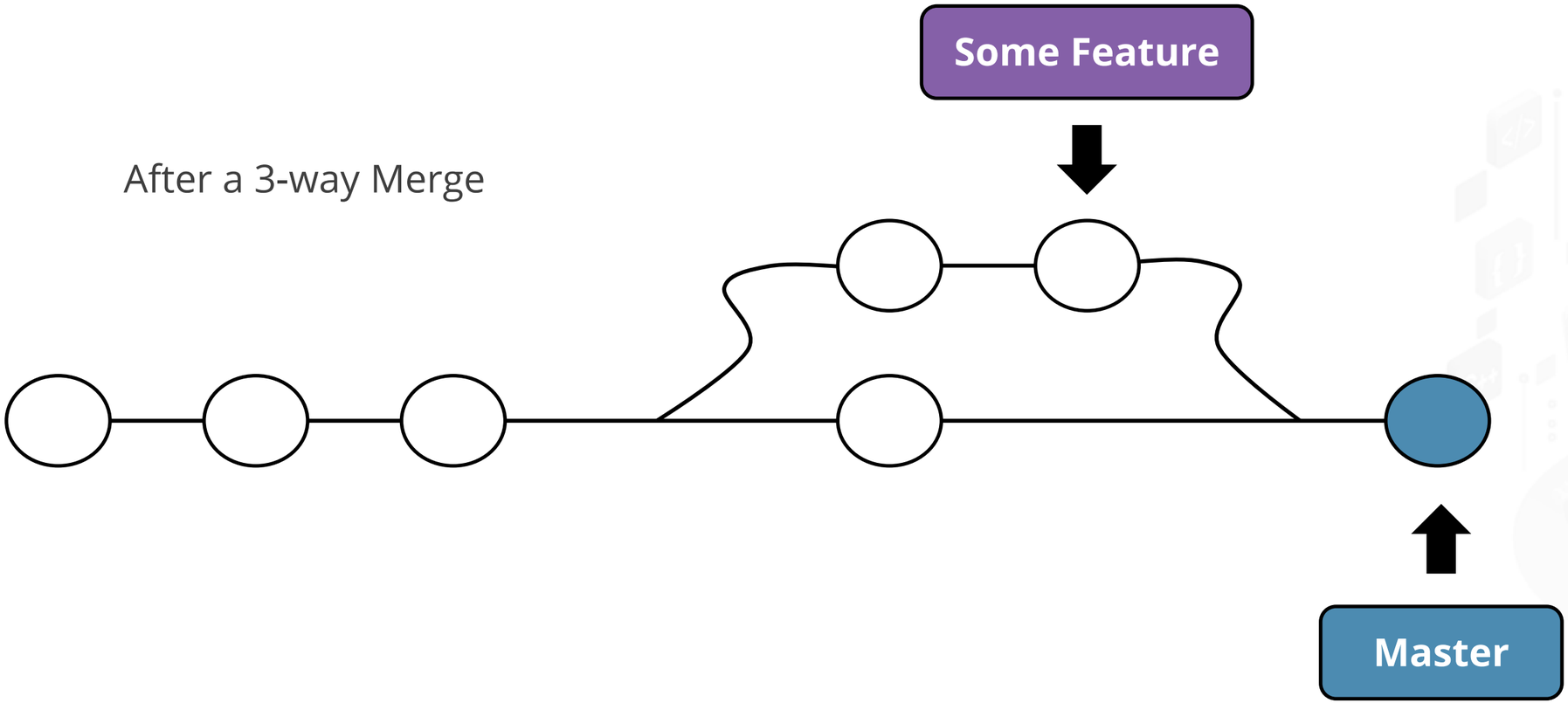
After a Fast-Forward Merge



Git Merge

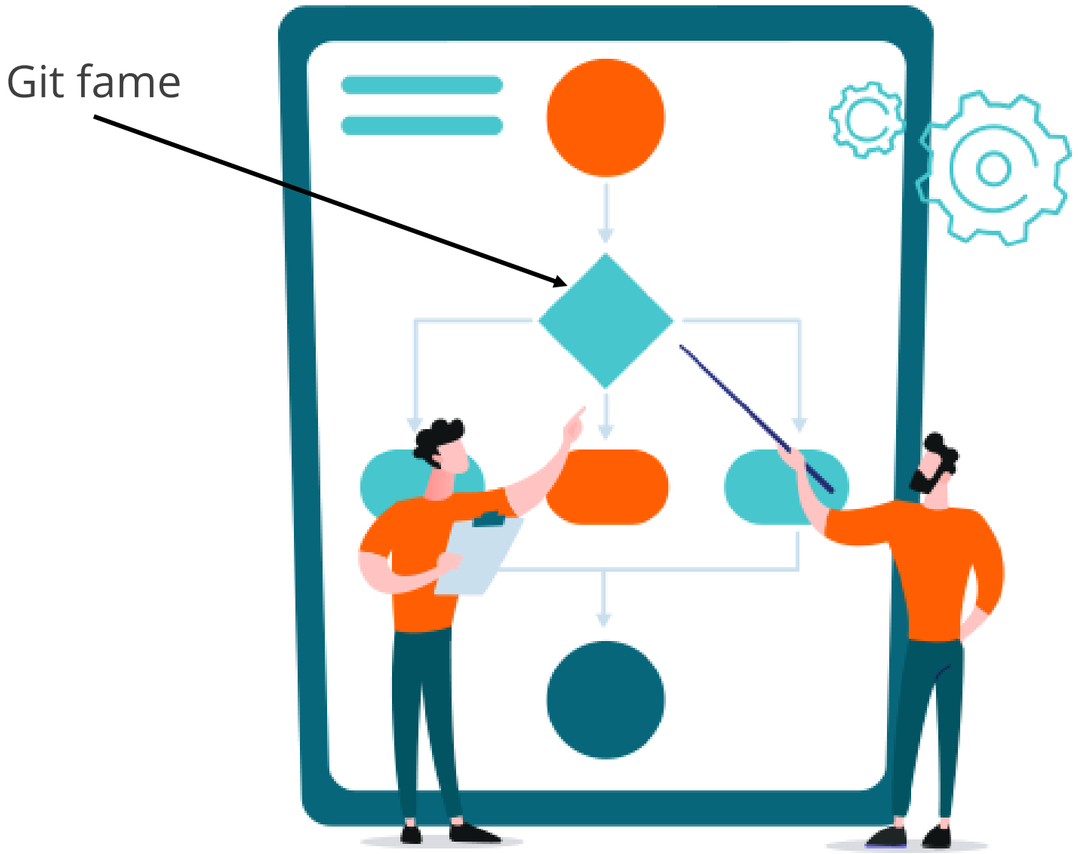


Git Merge



Git Merge

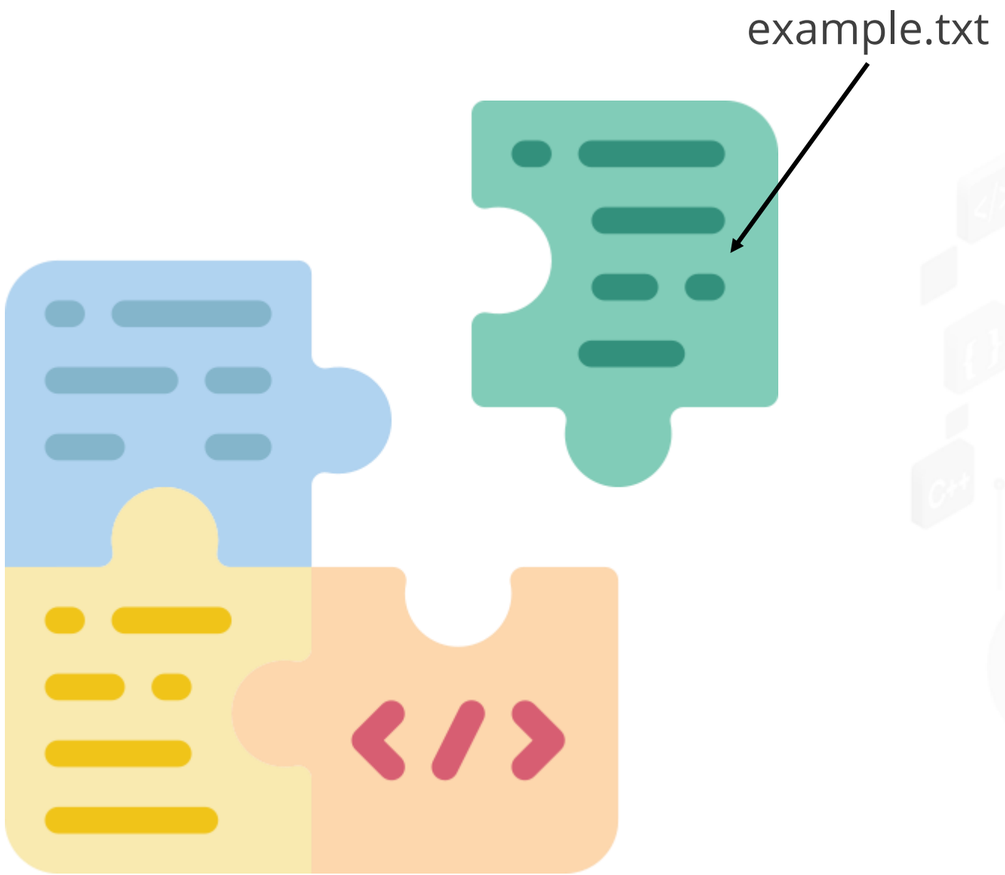
To resolve merge conflicts, Git merge uses the **edit/stage/commit** workflow.



Git Merge

Unmerged paths:

(use "git add/rm ..." as appropriate to mark resolution)
both modified: example.txt



Git Rebase

Git Rebase

Git Rebase is a process to reapply commits on top of another base trip.

It is a linear process of merging

It implements a sequence of commits across branches

It is an alternative to the Git merge command

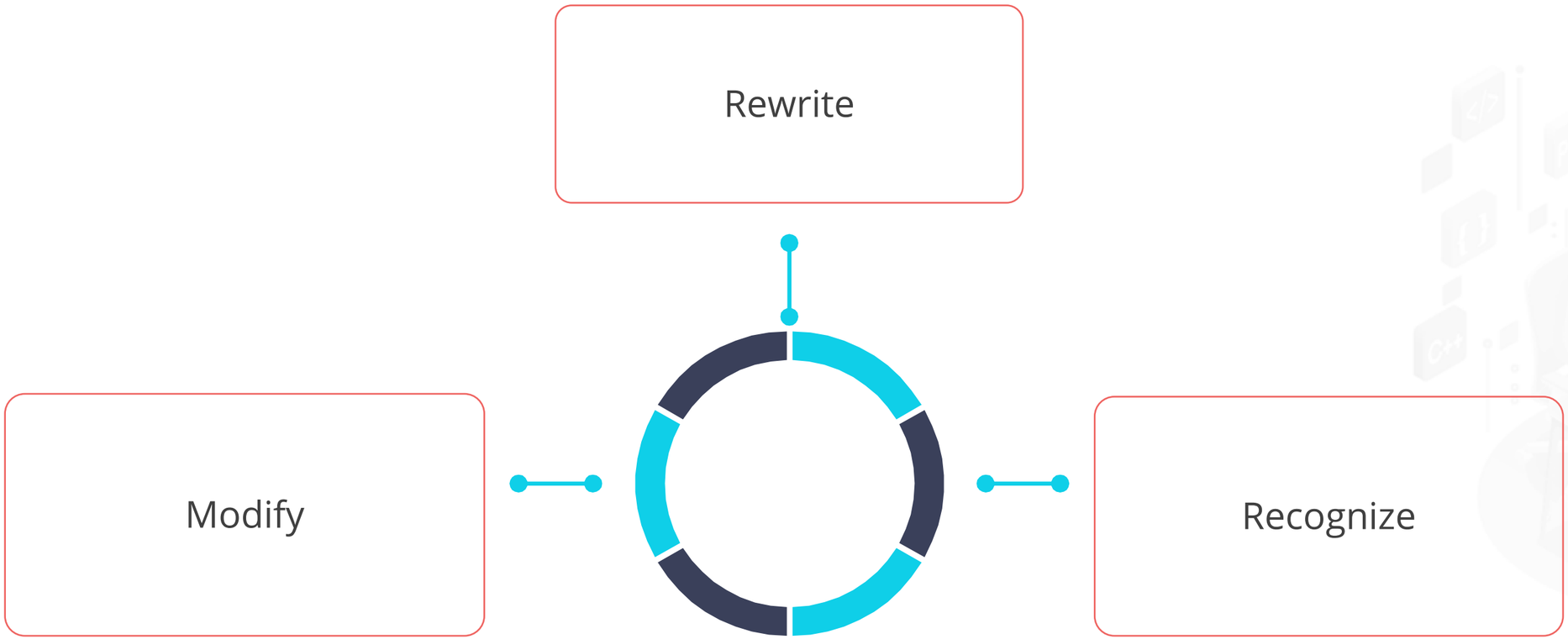
Git Rebase

It helps to visualize the process in the feature branching workflow.

```
On branch master
Unmerged paths:
(use "git add/rm …" as appropriate to mark resolution)
Both modified: example.txt
```

Git Rebase

Git promotes the use of Interactive Rebase.



It can be used on the branch that is checked out.

Git Rebase

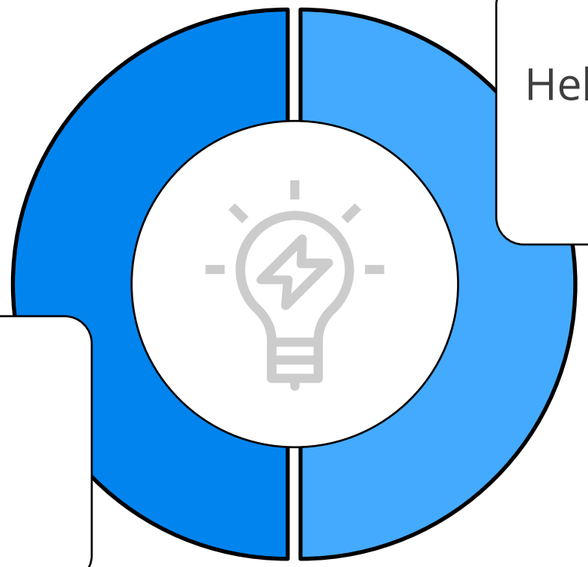
It can be requested using the rebase command; simply type –I after rebasing.

```
$ git rebase -i
```

Git Squash

Git Squash

Squash is used in Git to compress preceding comments into one.



Helps to combine multiple commits

Helps to test group-specific changes

Key Takeaways

- The Git branch command is employed to create, list, and delete branches locally.
- The new branch is used to create a new branch in the local repository.
- Master (master) is the default branch available in the Git repository.
- Git Rebase is a process to reapply commits on top of another base trip.

TECHNOLOGY

Thank You

simplilearn