Automation Testing

FULL STACK

Introduction to Git and GitHub

A Day in the Life of an Automation Test Engineer

Joel has decided to employ Git as a version control system for his projects. So that his team can collaborate, easily manage file of the code projects.

To complete his project, he must know about Git version control and how they are used too. And also learn the importance of Git and GitHub in version control in software development, differentiate between Git and GitHub and about repository and Git commands.

To achieve the above, he will learn a few concepts in this lesson that can help him to find a solution for the scenario.

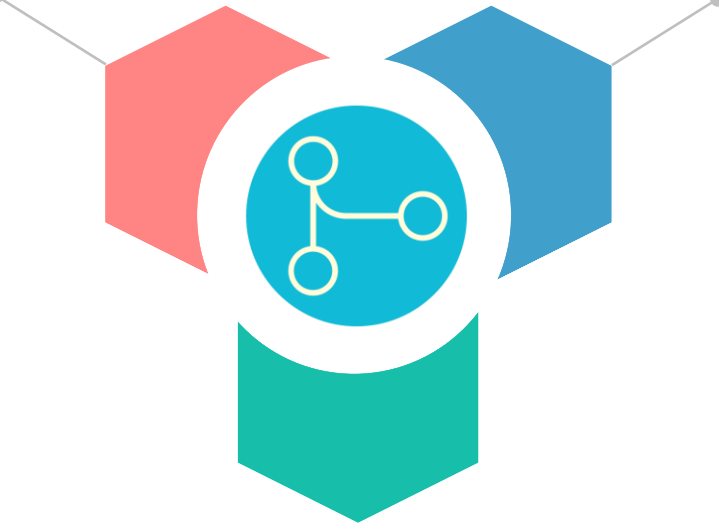
Learning Objectives

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

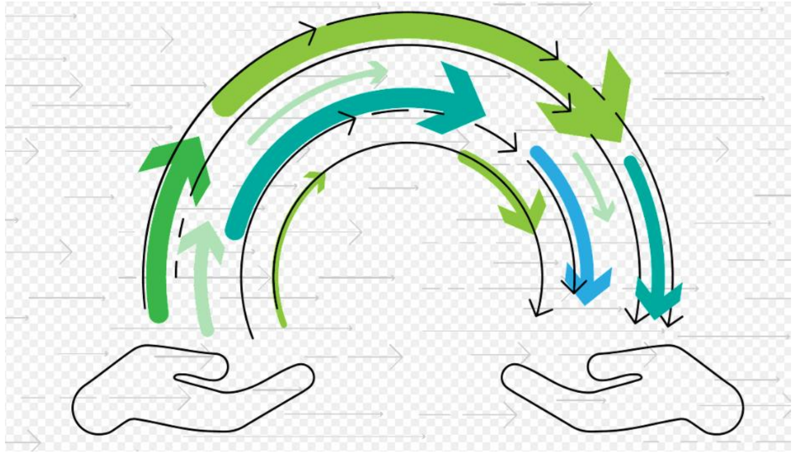
- Define Git and GitHub
- Differentiate Git and GitHub
- Create a Git repository
- Analyze reverting, deleting, renaming, deleting, and pulling commits

Introduction to Git

Introduction to Git



Free and open-source version control tool



Created in 2005

Authored by Linus Torvalds

Git

The features of Git are:

1 **Staging** and **Committing** allow users to manage and track changes.

2 To work on a local duplicate of a project, **clone** it.

3 **Repositories** are used to manage projects.

Git

The features of Git are:

4 To work on multiple parts and versions of a project, use **Branch** and **Merge**.

5 **Pull** a local copy of the most recent version of the project.

6 **Push** the parent project with local changes.

GIT

Git is used for:

- Monitoring code modifications
- Tracking who made the changes
- Collaborating coding

Introduction to GitHub

GitHub

GitHub is a rapidly growing programming resource for sharing code.

GitHub is a web-based Git repository hosting service with a graphical interface. It is the largest coding community on the planet.

GitHub

Features of GitHub:

- Affordable resource and a great open-source community
- Cloud-based platform
- Collaboration of team members from different geographical locations

Difference Between Git and GitHub

Git

- Is installed in a local system
- Was released in 2005 by Linux
- Focuses on version control and code sharing
- Lacks user management features
- Competes with Mercedes, IBM, ClearCase, and Rational Team Concert

GitHub

- Is hosted on cloud
- Was launched in 2008 and purchased by Microsoft
- Focuses on centralized source code hosting
- Includes built-in user management features
- Competes with Atlassian Bitbucket and Gitlab

Configure Git with GitHub

Problem Statement:

Configure Git account with GitHub to create a repository.

Assisted Practice: Guidelines

Steps to configure Git account with GitHub are:

1. Create a GitHub account
2. Configure Git account with GitHub

Git Repository

Create a Git Repository

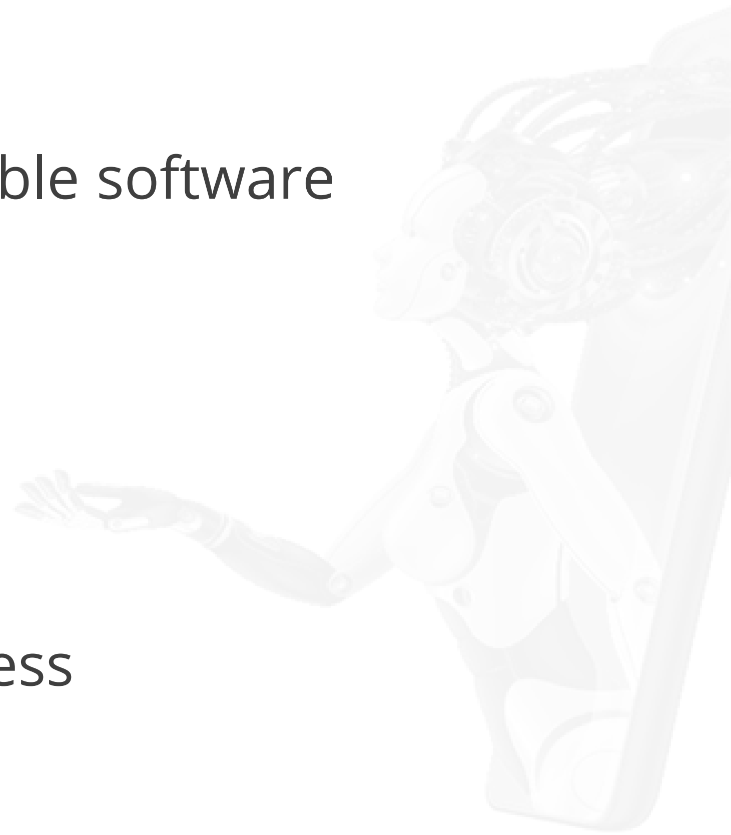
GitHub repositories hold a wide range of projects, including open-source projects.

Exchange code in open-source projects

Create dependable software

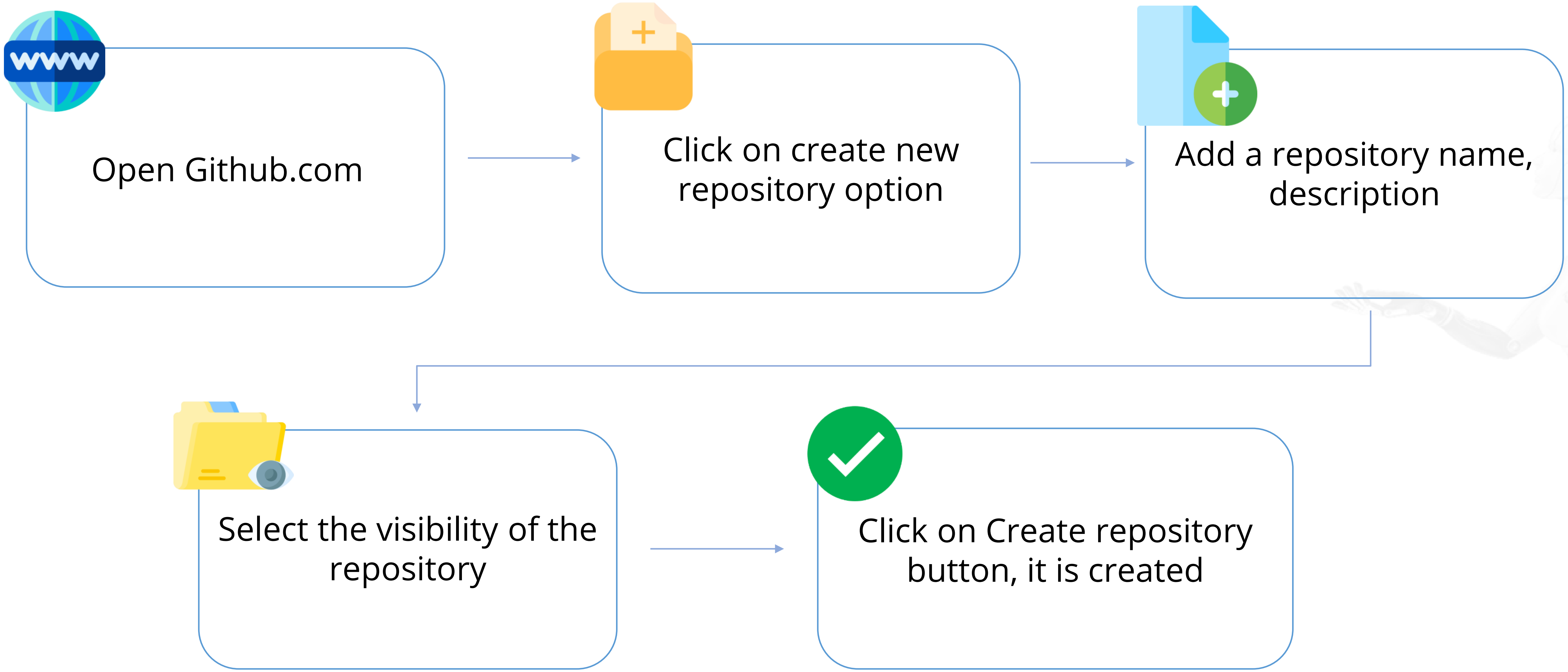
Collaborate with others

Track the progress



Create a Git Repository

A Git repository can be built from scratch or from an existing project in one of two ways.



Open Github.com

Click on create new repository option

Add a repository name, description

Select the visibility of the repository

Click on Create repository button, it is created

Various Git Commands

Problem Statement:

Use various Git commands to manage the distributed version control system with GitHub.

Assisted Practice: Guidelines

Steps to work with various Git commands are:

1. Execute Git commands

Push the Code to GitHub

Problem Statement:

Push the code to GitHub to make the commits accessible to others who you may be collaborating with.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to push the code to GitHub are:

1. Create a local repository
2. Add a remote origin for the remote repository
3. Push the code to GitHub

Revert to Older Commit

Git Commit

The **git commit** command captures a snapshot of the project's currently staged changes.

Command:

```
$ git commit
```

Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to.

Revert to Older Commit in Git

There are three situations that may happen:

- Go to the previous commit and then return to the current commit (temporary jump)
- Return to a previous commit and make some changes there but not at the expense of losing the current update history
- Return to the old commit and remove any new changes

Revert to Older Commit in Git

1. Go to the previous commit and then return to the current commit (temporary jump).

Command:

```
$ git checkout <commit-hash>
```

Revert to Older Commit in Git

2. Return to a previous commit and make some changes there, but not at the expense of losing the current update history.

Command:

```
$ git checkout -b branch-name <commit-harsh>
```

Revert to Older Commit in Git

3. Return to the old commit and remove any new changes.

Command:

```
$ git reset --hard <commit-harsh>
```

Or

Command:

```
$ git revert HEAD~3
```

FULL STACK

Delete and Ignore Files

Deleting Files in Git

Executing the **git rm** command and specifying the file to be deleted is the simplest way to delete a file in your Git repository.

Command:

```
$ git rm <file>
$ git commit -m "Delete the file repository"
$ git push
```

Ignoring Files in Git

A specific file called **.gitignore** is used to indicate the files that are ignored. To ignore any file, this file must be edited and committed manually.

Command:

```
$ echo debug.log >> .gitignore
$ git rm -cached debug.log
  rm 'debug.log'
$ git commit -m "Start ignoring debug,log"
```

Using the patterns specified in the .gitignore file, files may be excluded from the repository.

Renaming Files in Git

Renaming Files in Git

To rename and relocate files in git, use the **git mv** command.
It does not really rename or relocate the file; instead, it deletes the original file and generates a new one with a different name or in a different folder.

Command:

```
$ git mv options oldFilename newFilename
```

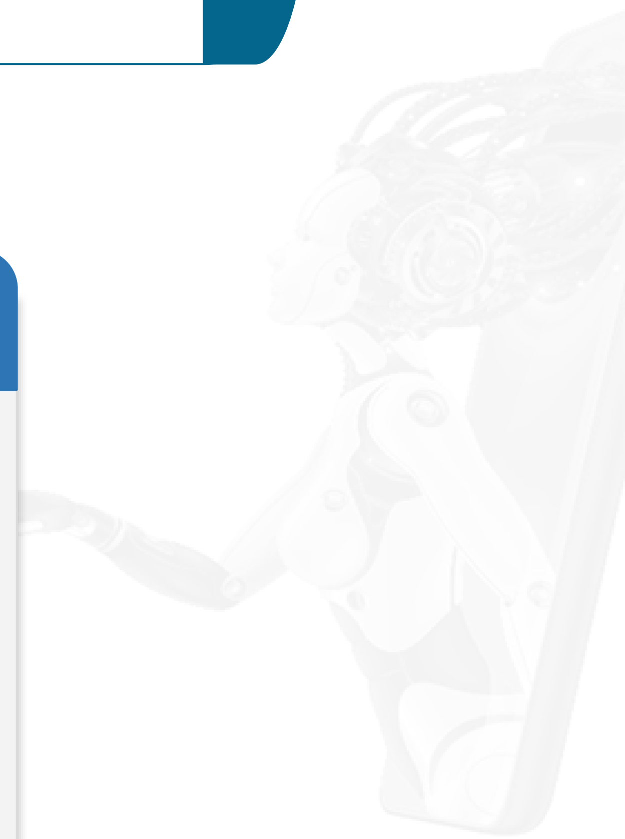
Pulling Commits

Pulling Commits in Git

If the user wants to bring only that COMMIT ID to their local branch, they can use **git-cherry-pick** to bring only that commit over, or **git-merge** to bring all changes up to that commit to their local branch.

Command:

```
$  git  fetch  $  git  checkout  -b  add-log-component  origin/add-
log-component
$ git checkout
$ git cherry-pick
$ git push
```



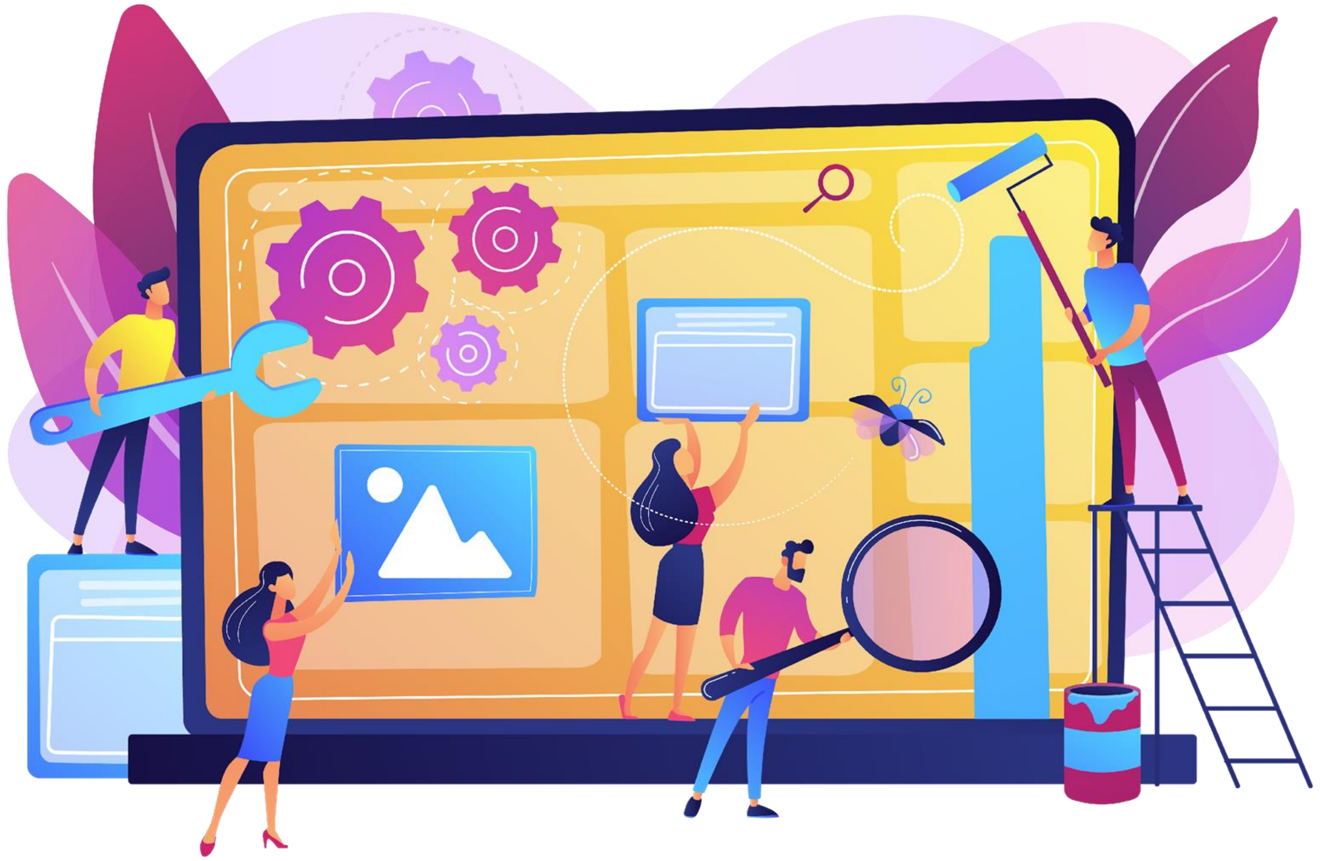
FULL STACK

Working with Branches in Git

simplilearn

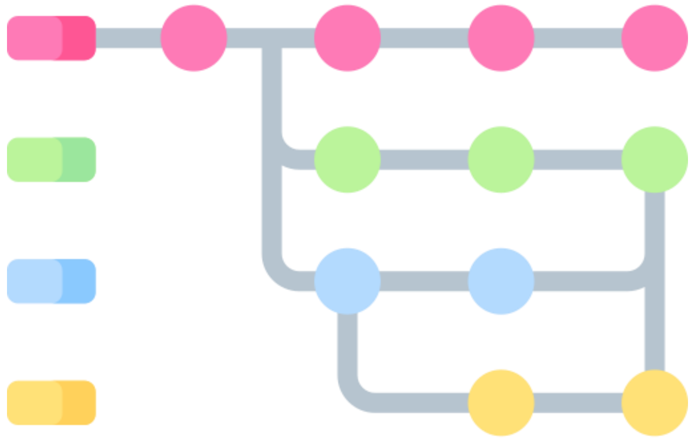
Working with Branches in Git

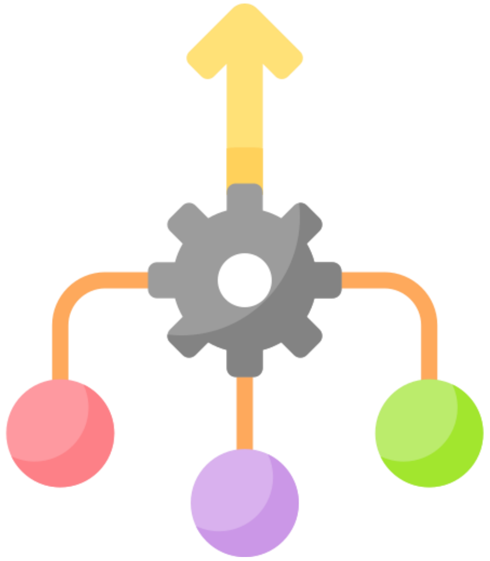
A branch is a distinct set of code modifications with a distinct name. There can be one or more branches in each repository.



Working with Branches in Git

Master is the main branch where all modifications are finally merged back into.





Branching Command

The Git branch command creates a new branch by default. It does not switch to the other branch.

Command:

```
$ git branch
```

Working with Branches

ASSISTED PRACTICE

Problem Statement:

You are required to work with Branches in Git.

Assisted Practice: Guidelines

Steps to create and work with branches are:

1. Create and work with branches

Local and Remote repository

Difference Between Local and Remote Repository

Local repository

- Stored on local or own computer
- Used by an individual
- Once changes are made, the user can add them to the staging area
- Nobody can see the changes done

Remote repository:

- Store on a remote computer
- Used by teams
- Once changes are made, the user has to push code the local repository to the remote repository
- The changes are visible to everyone in the remote repository

Key Takeaways

- An open-source version control tool created in 2005 by Linux developers.
- GitHub is a web-based Git repository hosting service with a graphical interface. It is the largest coding community on the planet.
- GitHub repositories house a wide range of projects, including open-source projects.
- Repositories may be used to collaborate with others and track their progress.