

×


Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

This scenario has explained how you can initialise a repository and then commit files to it. In the next scenario we'll investigate how to compare and commit changes to these files. In future scenarios we'll cover how to share these changes with other people.

This scenario has been added to your scrapbook where you can review the examples and commands you executed.

 Share Your Success

 Share Your Success

 Share Your Success

DOWNLOAD SCENARIO

NEXT SCENARIO

×

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

The most important takeaways from this lab are:

- `git checkout` can be used to create branches, switch branches, and checkout remote branches
- `git branch` commands primary functions are to create, list, rename and delete branches
- `git tag` is used to create semantic version number identifier tags that correspond to software release cycles
- `git merge` is used to combine multiple sequences of commits into one unified history
- `git rebase`
- `git reset`

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

Now that you have an understanding of the projects you will use throughout this course, let's get started!



Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

The most important takeaways from this lab are:

- `git clone` is used to create a copy of a target repo
- `git remote` is used to create, view, and delete connections to other repositories
- `git push` is used to propagate changes on the local repository to remote repository
- `git fetch` is used to download objects and refs from another repository
- `git pull` is used to fetch from and integrate with another repository or a local branch

