



# Introduction: Git Submodules

Let us go over what you will learn in this chapter.

We'll cover the following ^

- Topics covered
- Submodules

## Topics covered #

This chapter covers:

- `git submodule`
- The `^` operator
- The `~` operator

## Submodules #

Submodules are a useful concept and often seen in real projects.

Git submodules can be very confusing if you stumble into them without much preparation or experience. Following this tutorial, you should have a good understanding of a simple submodule workflow and what is going on when you run the core submodule commands.

Submodules allow you to loosely link different Git repositories together so that they are bundled together. At the same time, it ensures that changing one will not break the other.

Sometimes, you want to include one repository in another but do not want to simply copy it over and have to maintain its changes separately.

Submodules allow you to manage the separate codebase within your

Submodules allow you to manage the separate codebase within your repository without affecting the other repository.



If this wasn't clear, re-read the above slowly. If it's still not clear, follow the example in the following lesson and see if it makes more sense.

[← Back](#)[Next →](#)[Tracking Remote Branches With Diffe...](#)[A Worked Example](#)[Mark as Completed](#)[Report an Issue](#)[Ask a Question](#)

([https://discuss.educative.io/tag/introduction-git-submodules\\_\\_git-submodules\\_\\_learn-git-the-hard-way](https://discuss.educative.io/tag/introduction-git-submodules__git-submodules__learn-git-the-hard-way))