How to use Git

UD Hackaton

24 April 2021

Marko D. Petrović, Abhin Suresh

Downloading and installing Git

General information

Please visit:
https://git-scm.com/download/win

Linux (Ubuntu)

```
sudo apt-get install git
```

MacOS

```
brew install git
```

Windows

You need to download Git for Windows.

Optionally, you can install GitHub Desktop.

Configuring Git

Configure user and email

```
git config --global user.name "John Bell"
```

```
git config --global user.email "jbell123@gmail.com"
```

Configure the default text editor (optional)

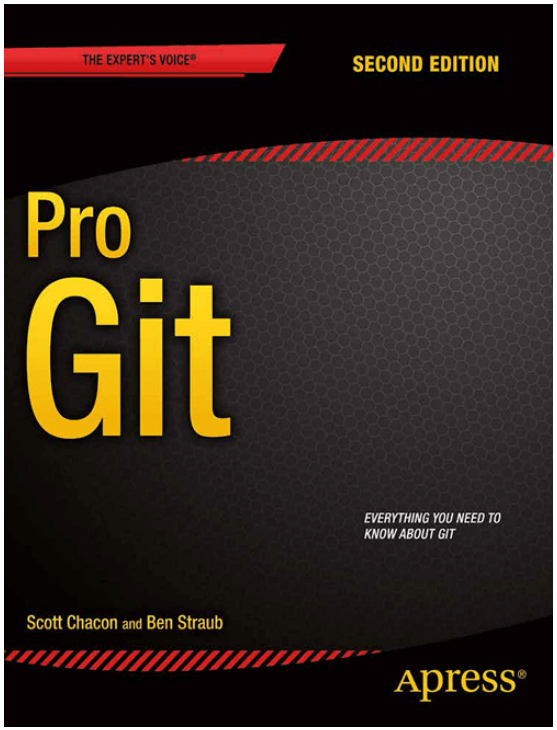
```
git config --global core.editor "nano -w"
```

or

```
export GIT_EDITOR=vim
```

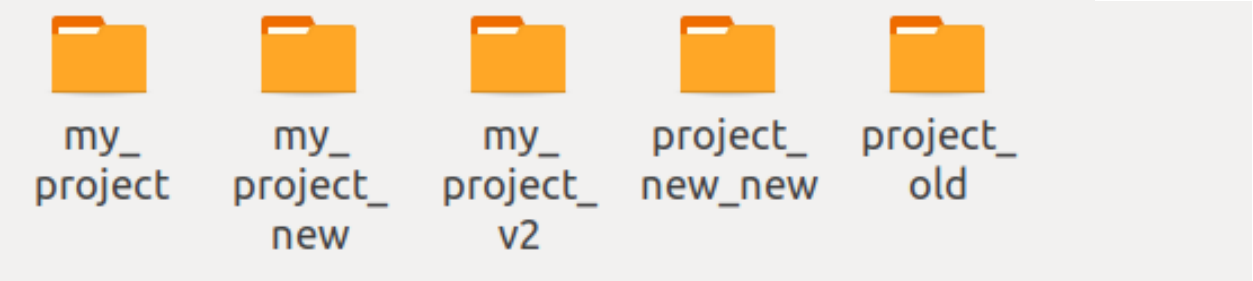
Git documentation

- More info on how to use Git can be found at
  https://git-scm.com/book/en/v2
  Most of the examples presented here are taken from this book.



What is version control?

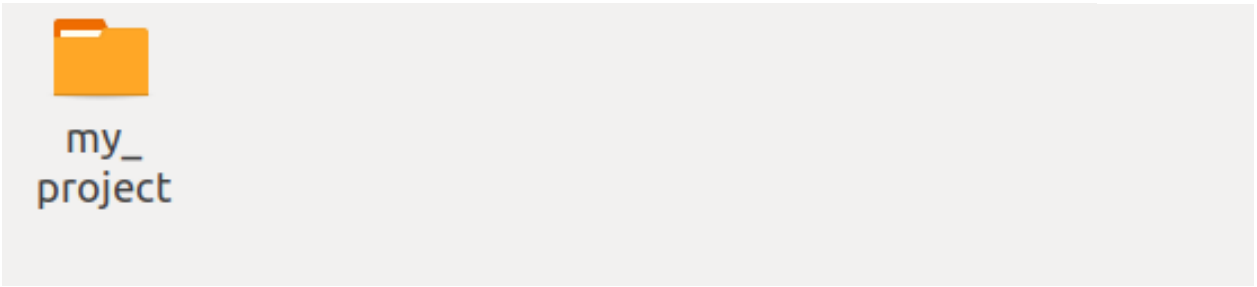
A project without version control

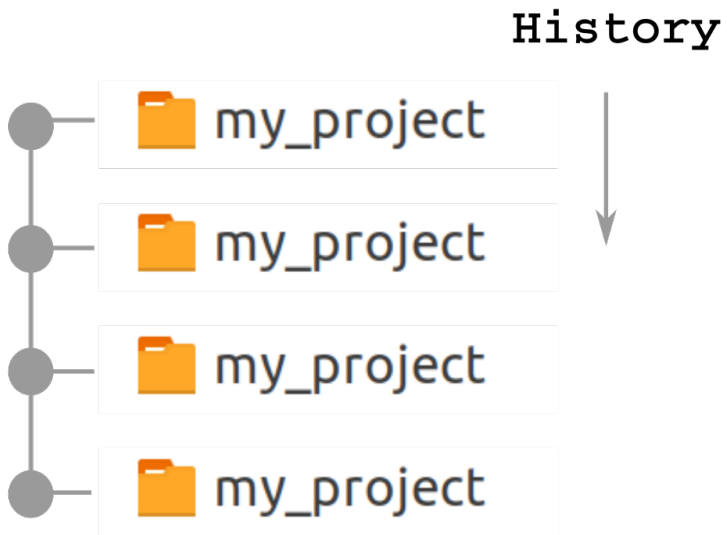


With a bad naming convention, you can easily get lost if you leave your project for a while.

What is version control?

A project with version control (Git).

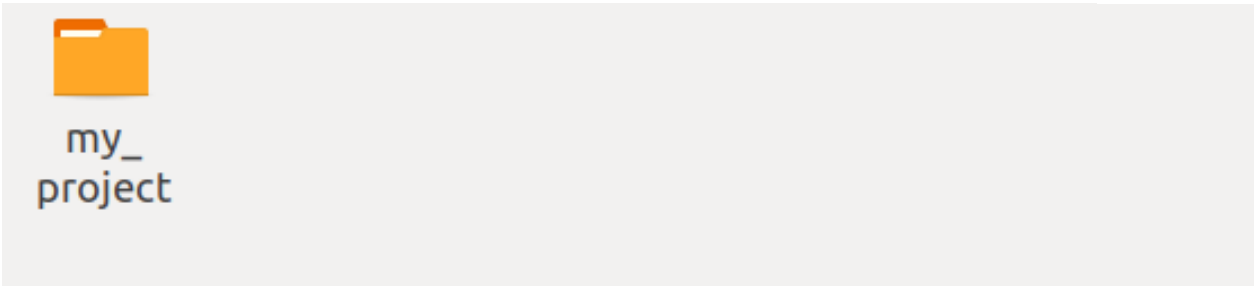


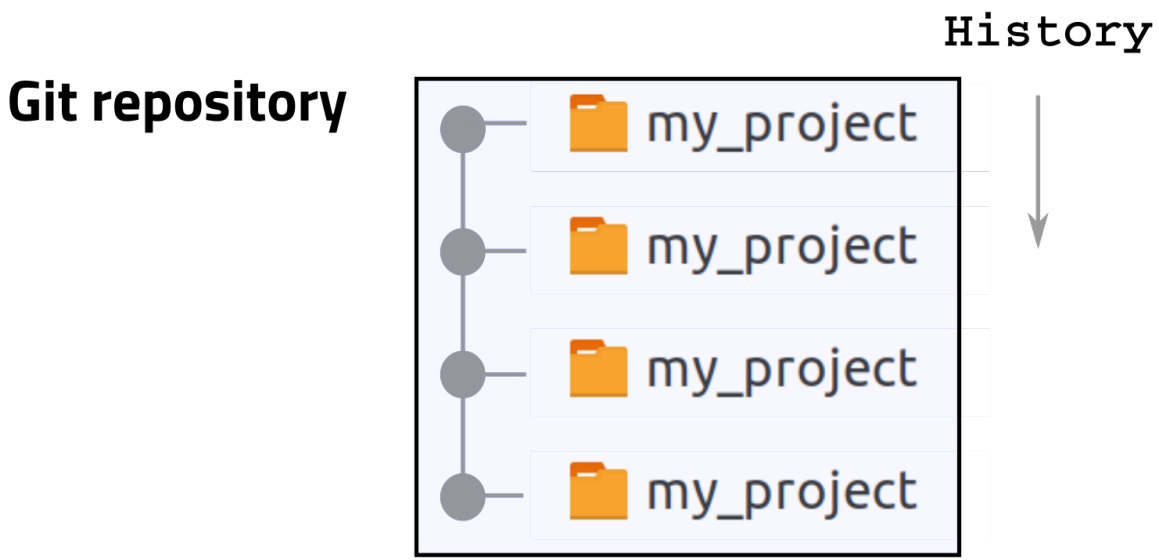


A version control tool such as Git can help you keep track of the history of your project.

What is version control?

A project with version control (Git).

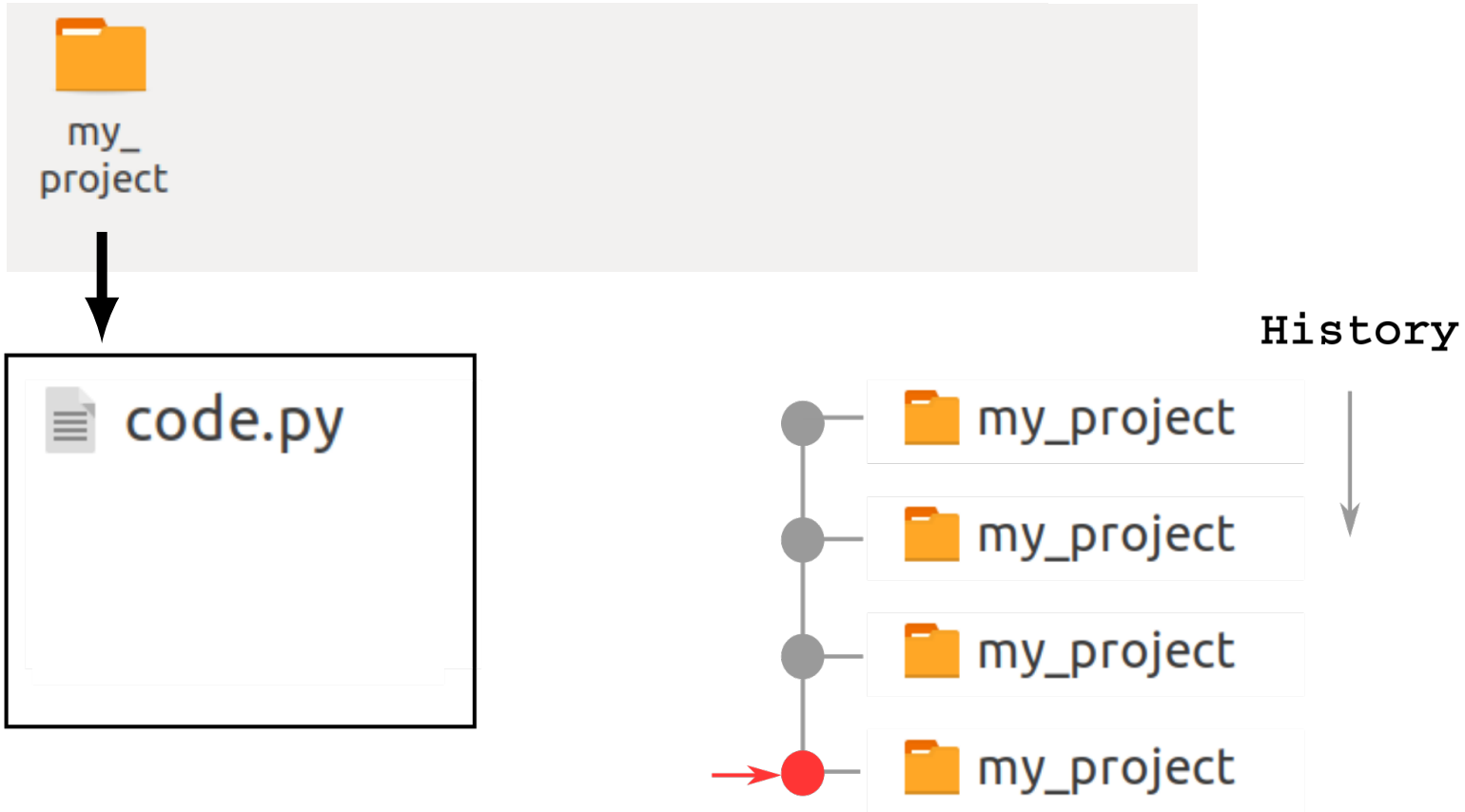




The history of your project is kept in a Git repository.

What is version control?

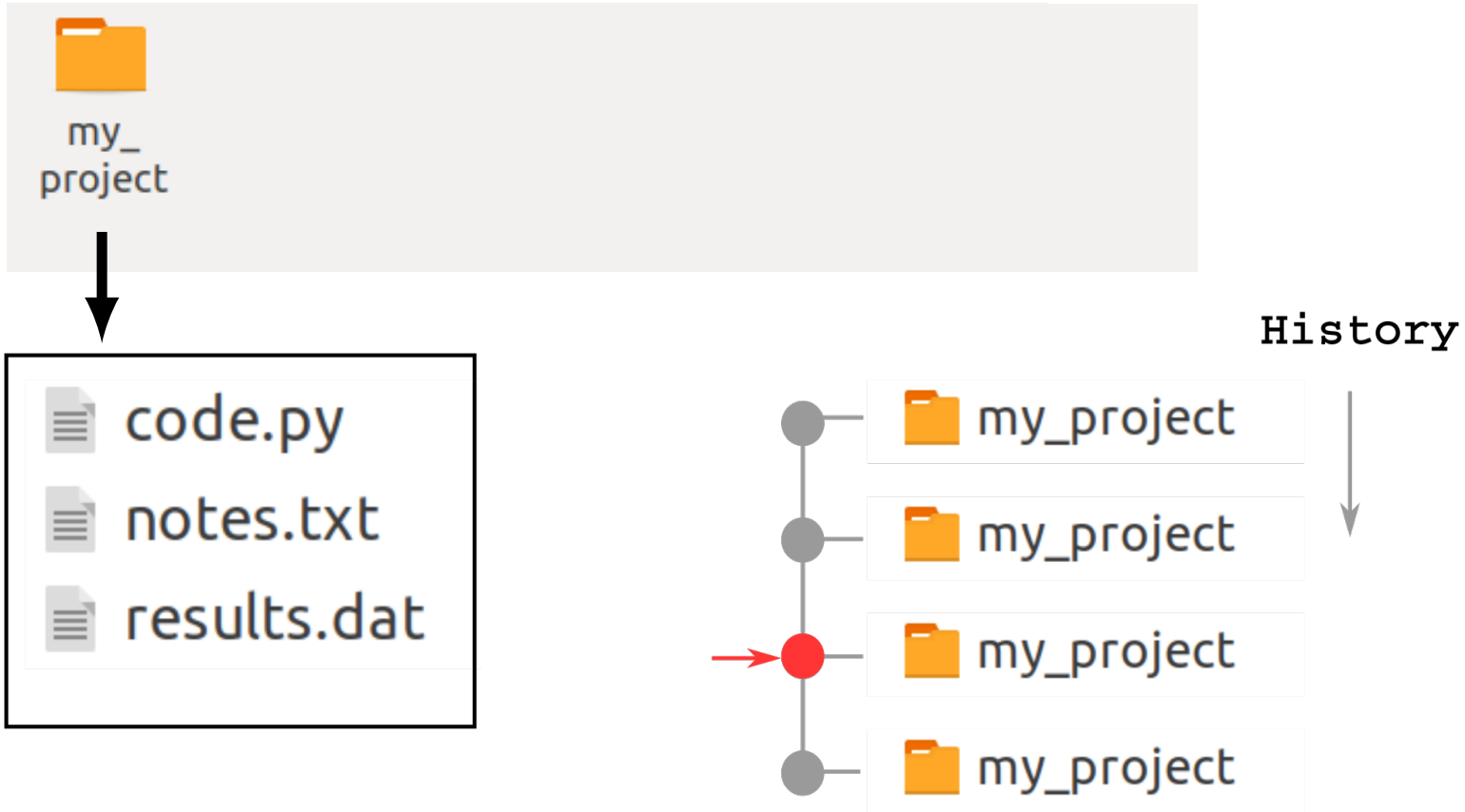
A project with version control (Git).



You can access different versions of your project from history.

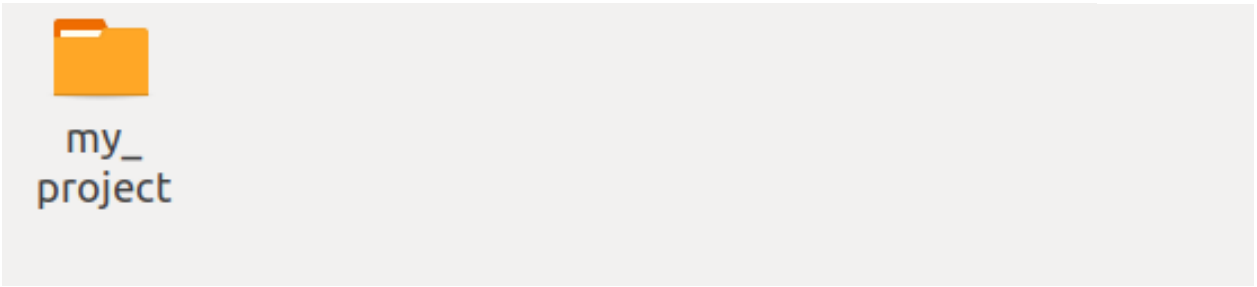
What is version control?

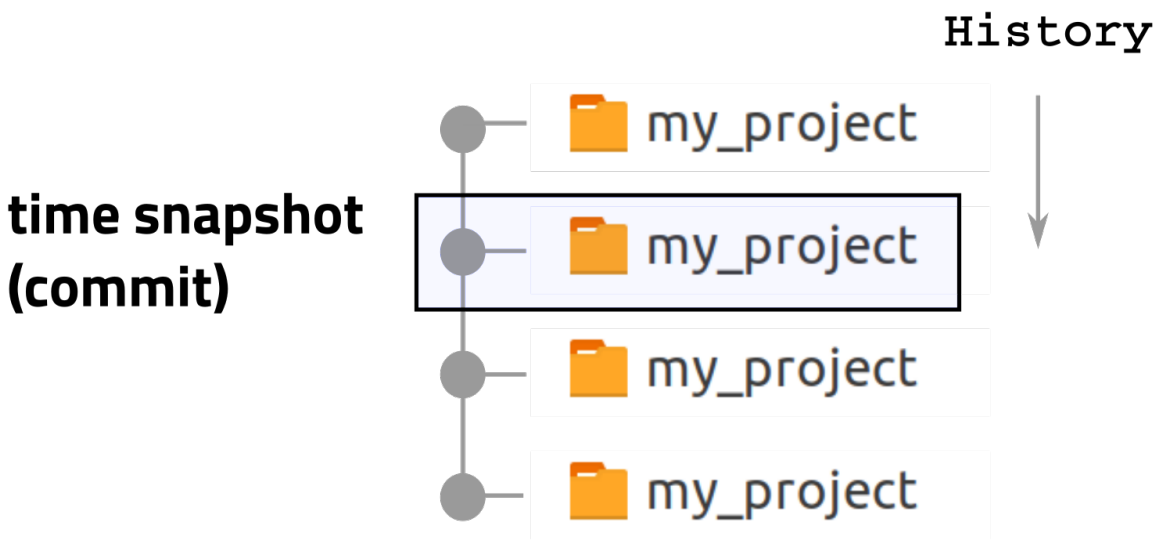
A project with version control (Git).



You can access different versions of your project from history.

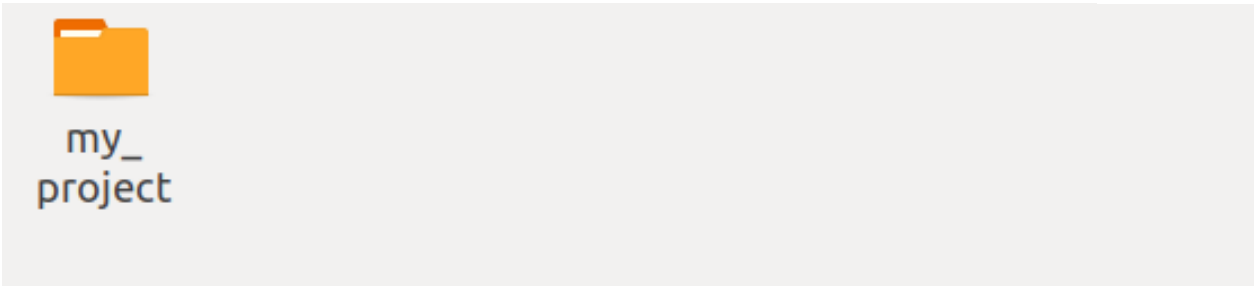
What is version control?

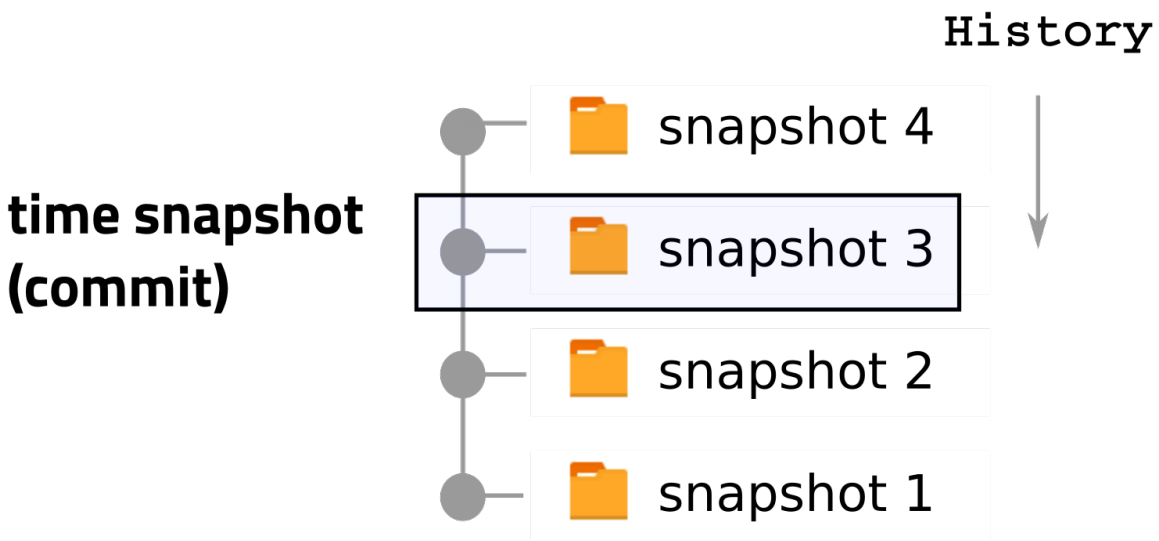




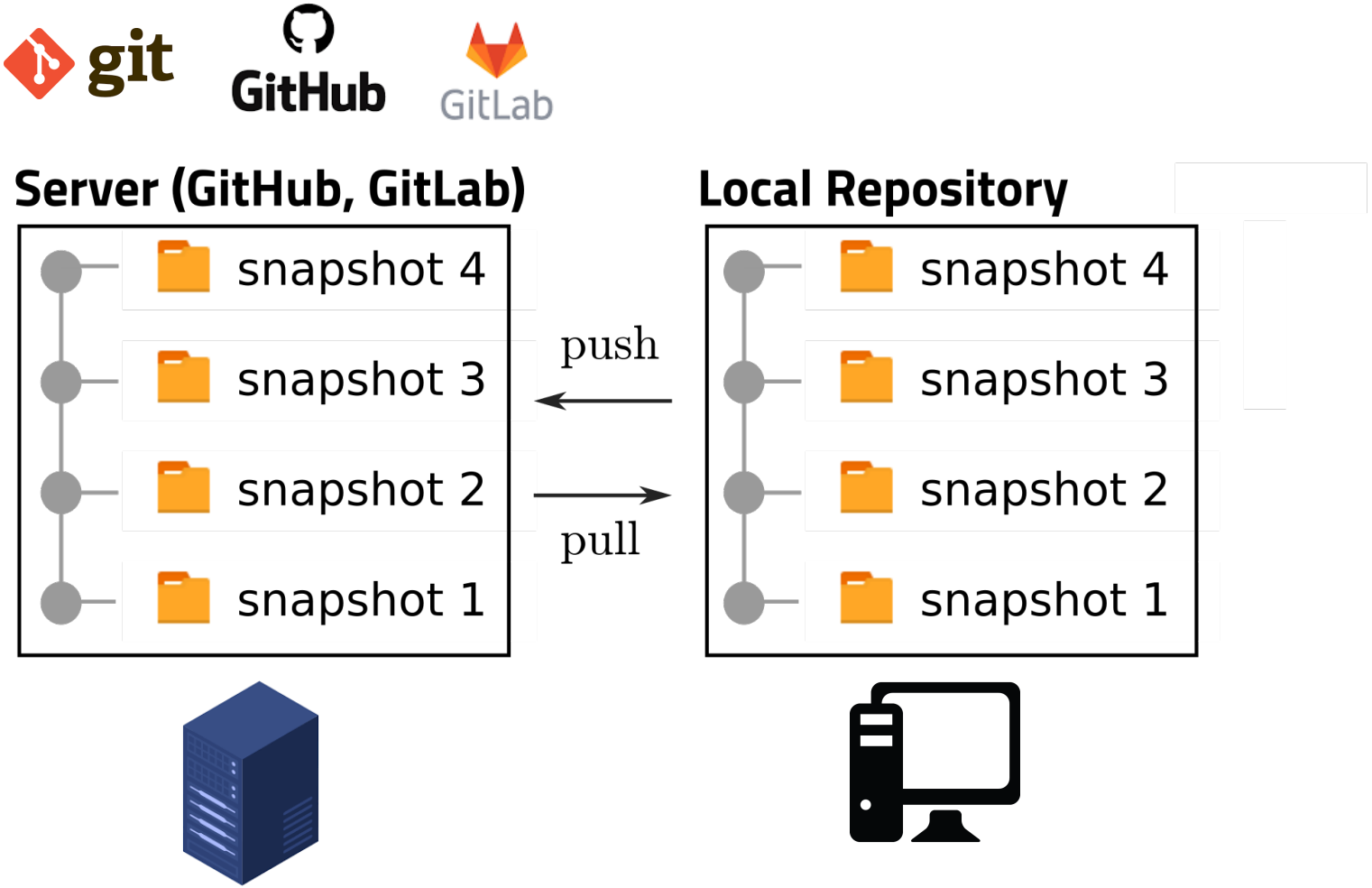
The history of your project consists of a series of time snapshots, also known as commits.

What is version control?



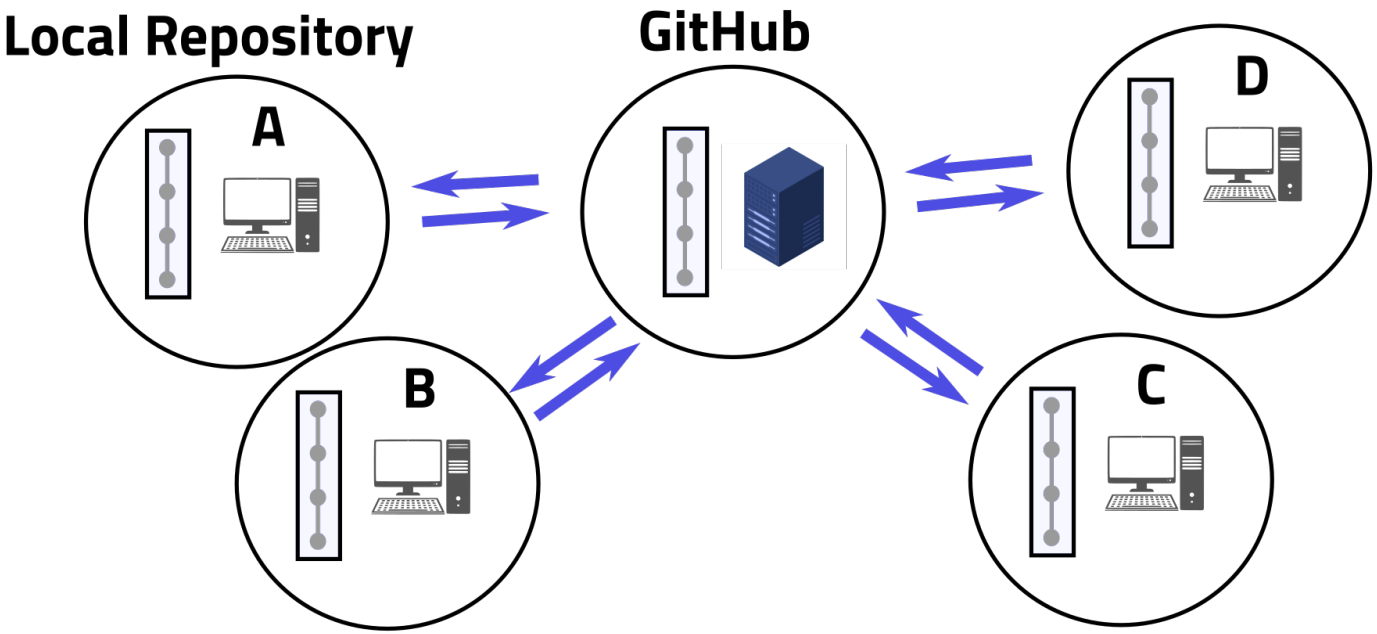


Git and GitHub



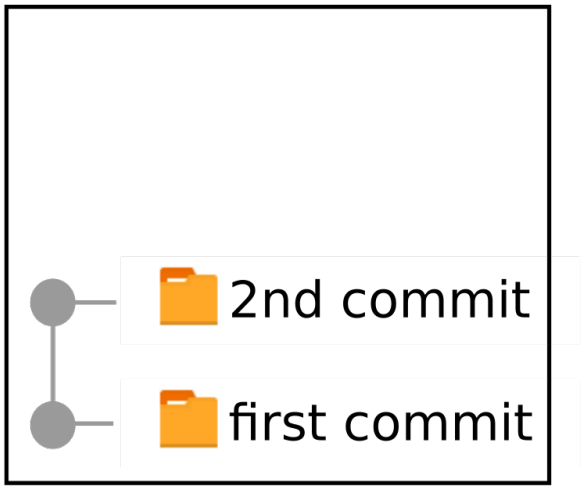
You can use hosting services provided by GitHub and GitLab to publish your repository and also collaborate with others.

Git and GitHub



What is covered in this introduction?

- Create a new repository
- Create the first commit
- Change between versions
- Cloning and pushing
- ✗ Branching, merging, rebasing



The slides provide just the basic information on branching. For more information, you should check the suggested documentation.

How to create a local git repository?

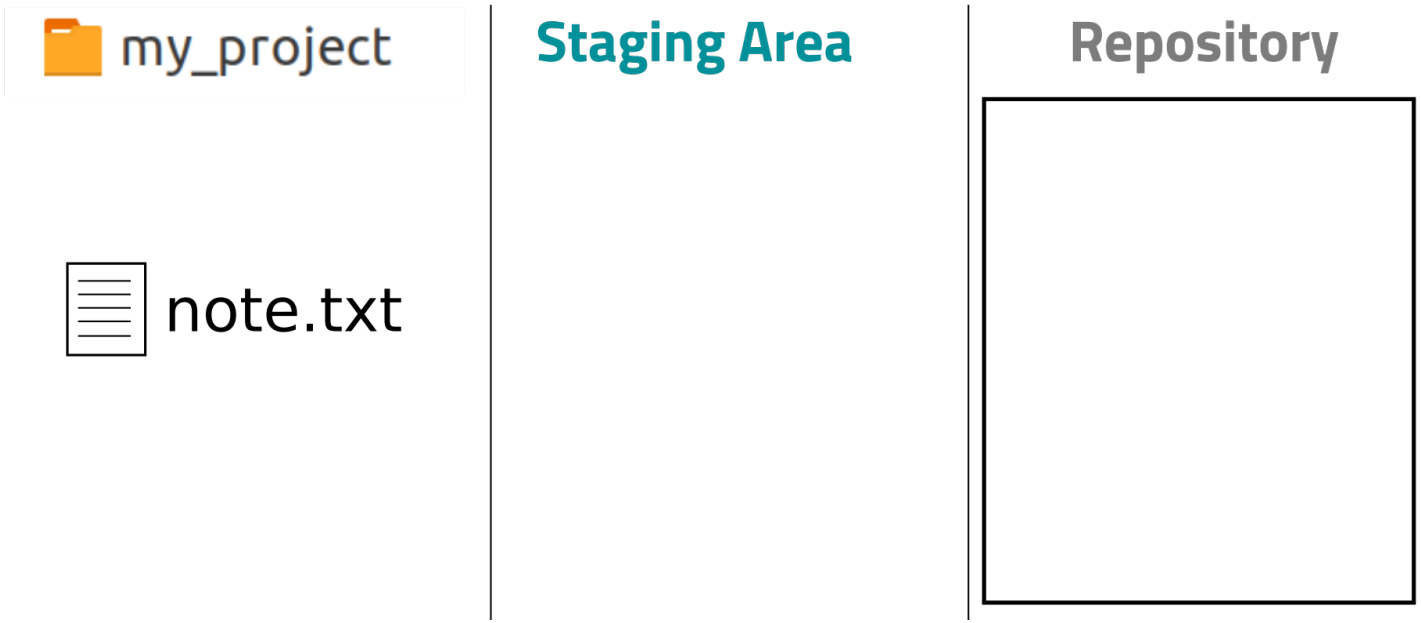
Start with an empty directory (or some existing directory)

```
mkdir my_project
cd my_project
```

Then create a git repository

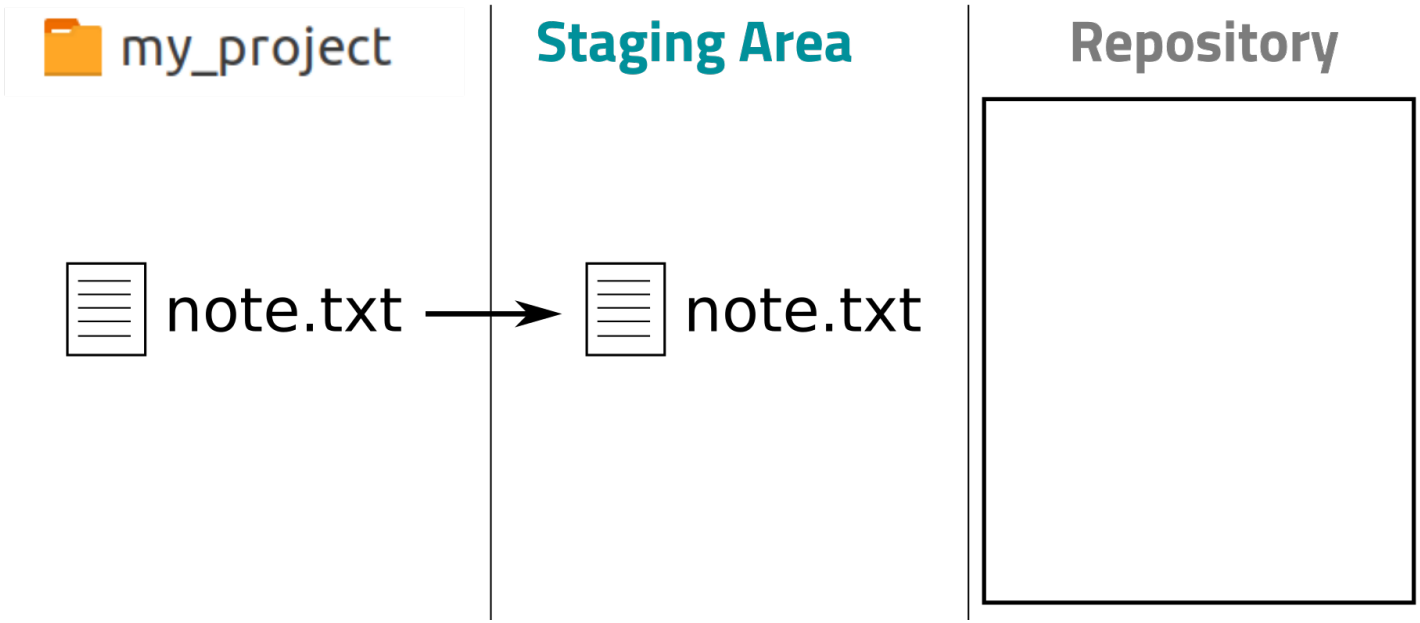
```
git init
```

How to create a snapshot (commit)?



The first step is to create a new file.

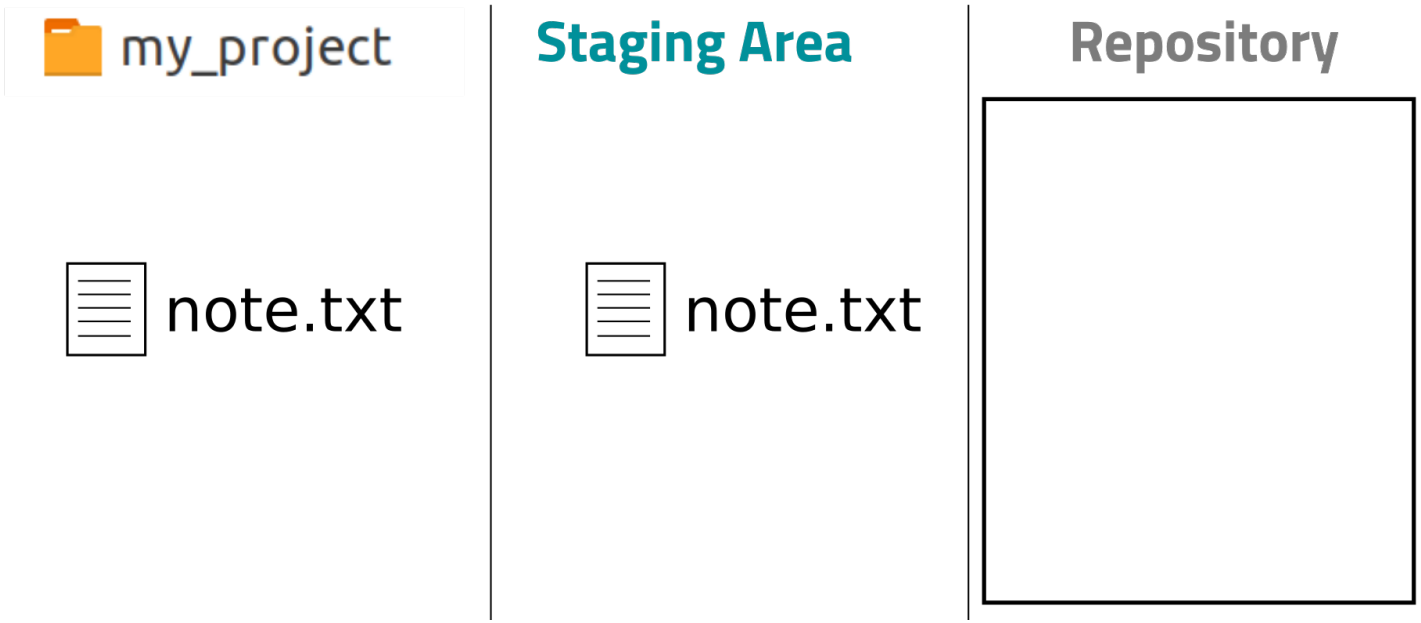
How to create a snapshot (commit)?




```
git add note.txt
```

To create a snapshot (commit), you need to tell Git which files to include in the commit. To do this, you need to add those files to the staging area.

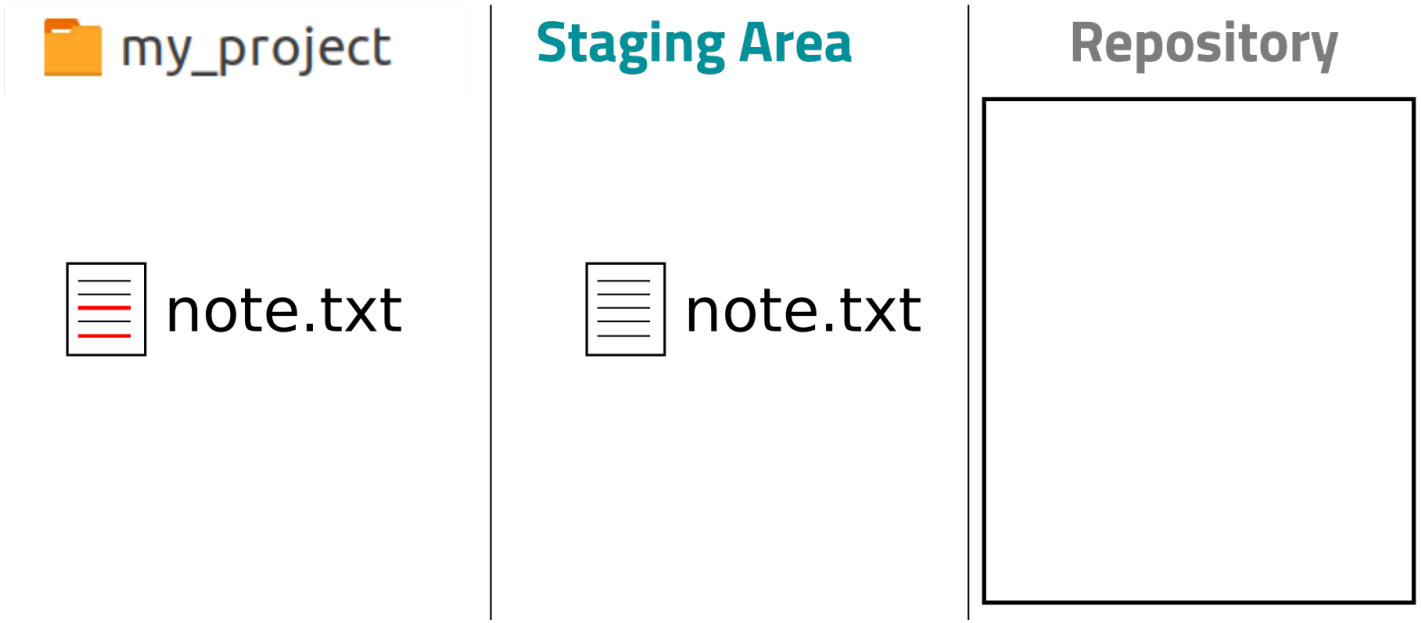
How to create a snapshot (commit)?



To check status of your staging area

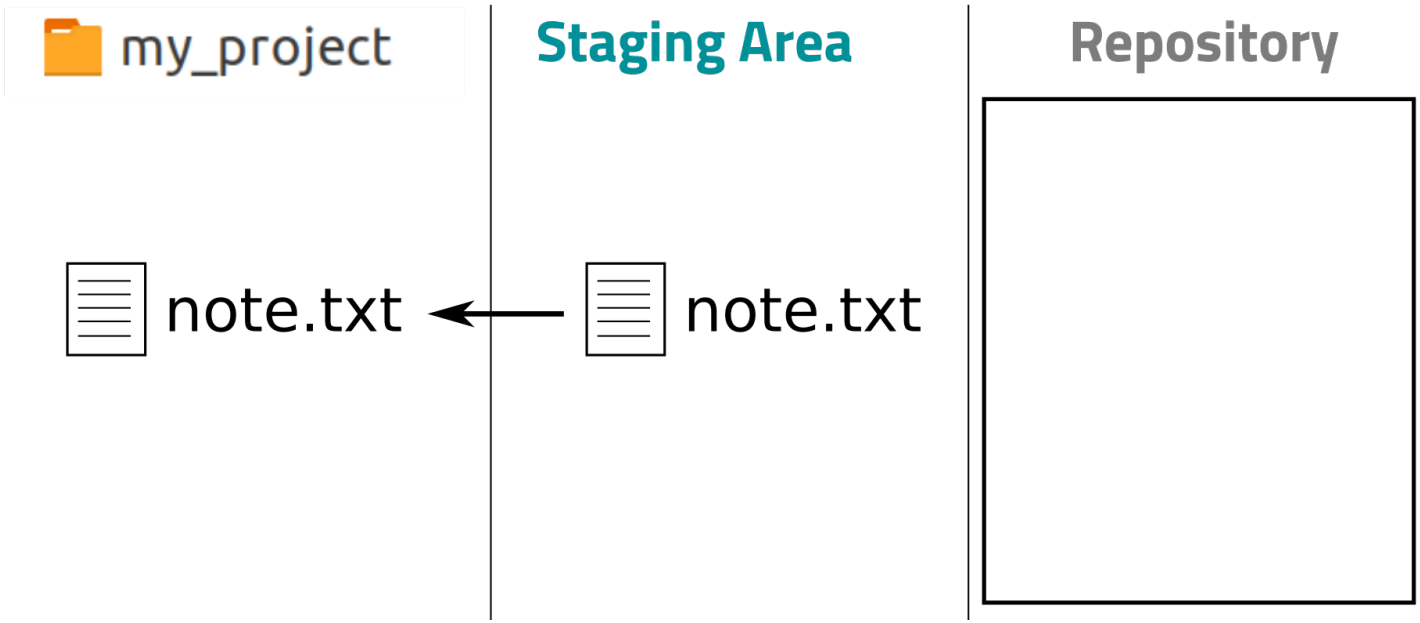
```
git status
```

How to create a snapshot (commit)?



Once you add the file to the staging area, you can still work on it and make changes.

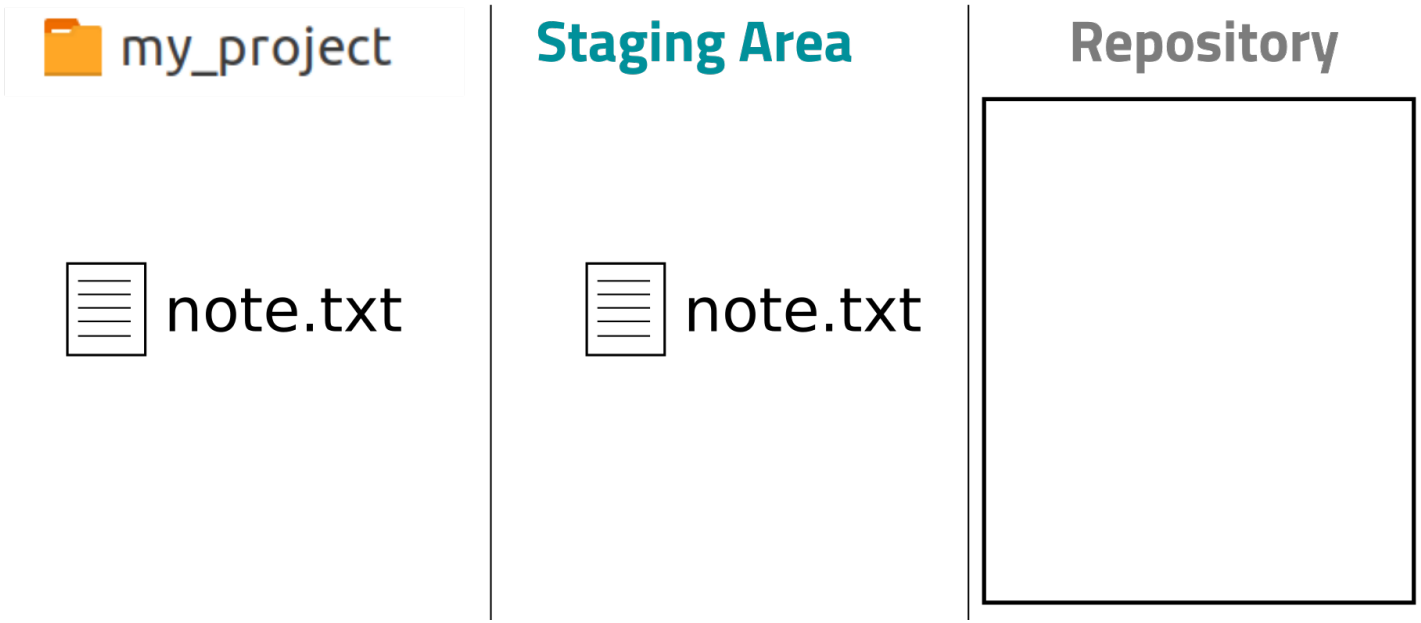
How to create a snapshot (commit)?



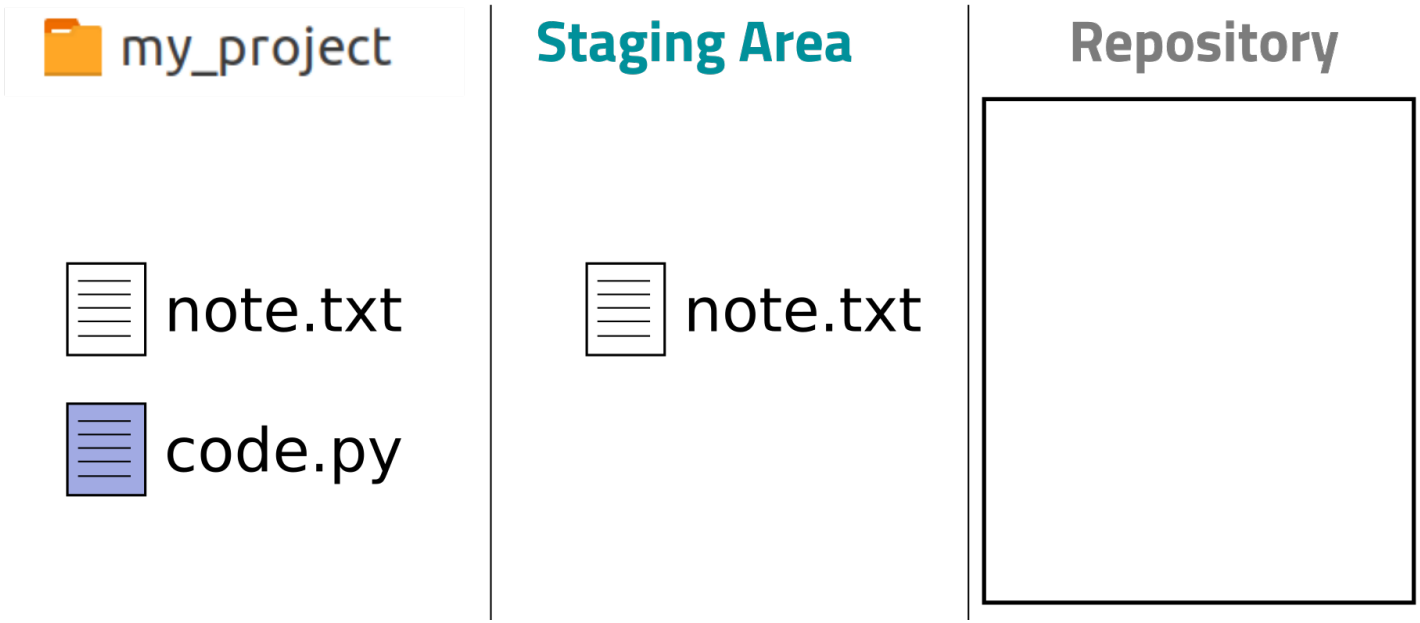

```
git checkout note.txt
```

In case you are not satisfied with the new version, you can always overwrite it with the one in the staging area.

How to create a snapshot (commit)?

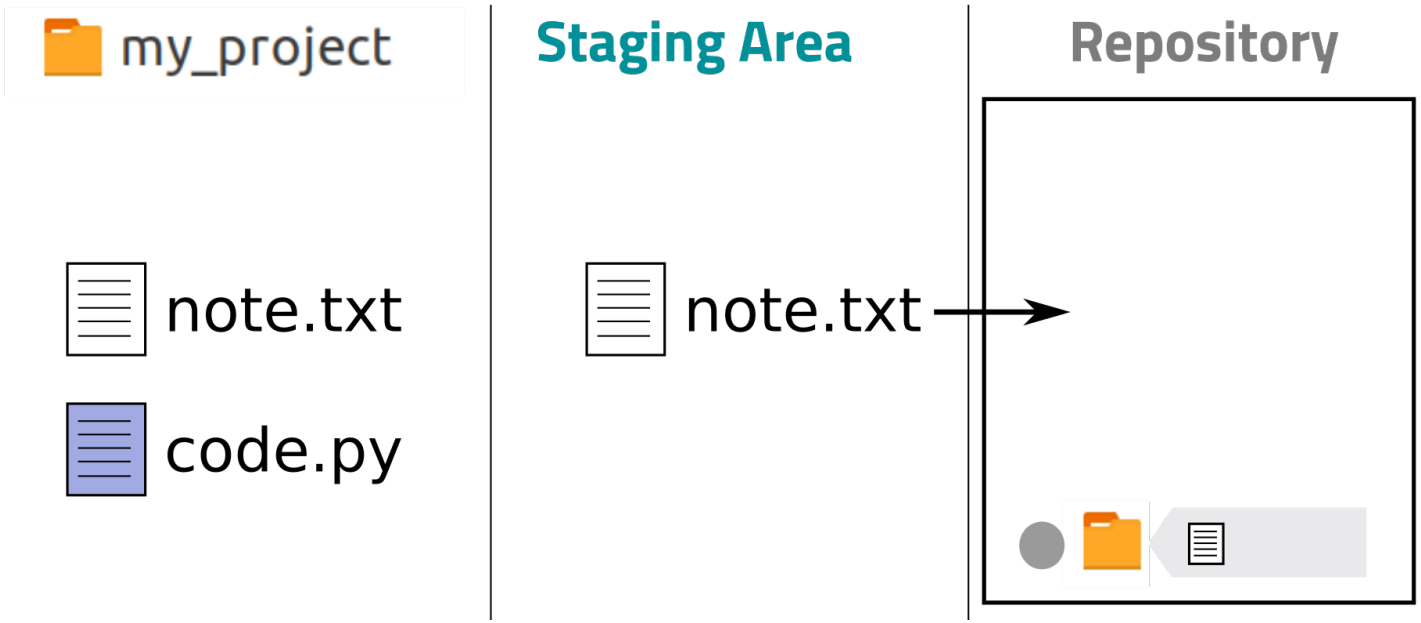


How to create a snapshot (commit)?



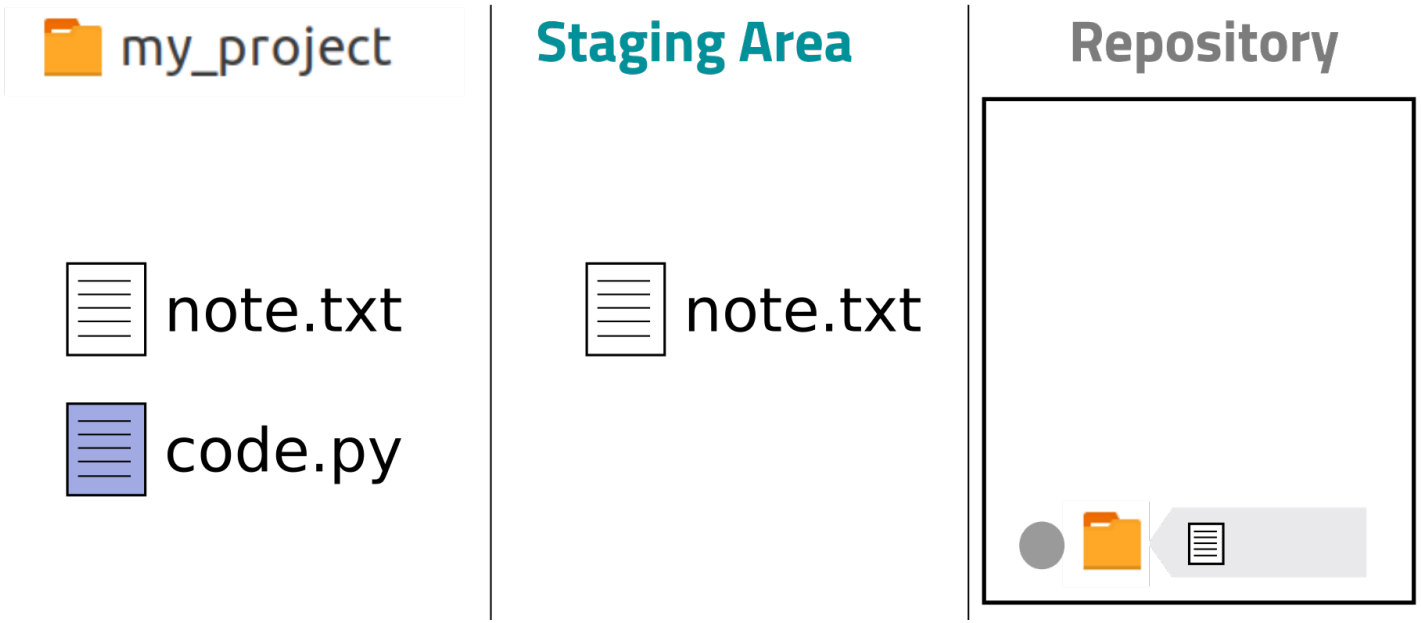
You can create additional files. If you don't add them to the staging area, they won't be included in the commit.

How to create a snapshot (commit)?

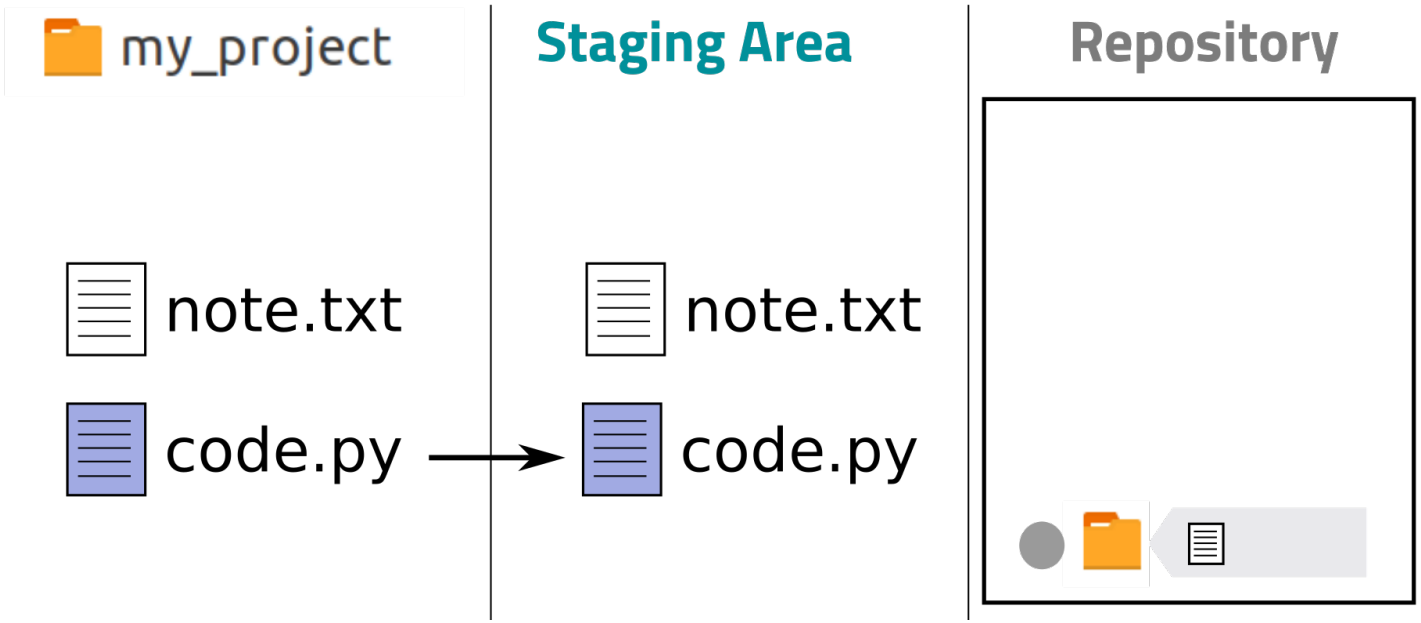



```
git commit -m "First commit"
```

How to create a snapshot (commit)?

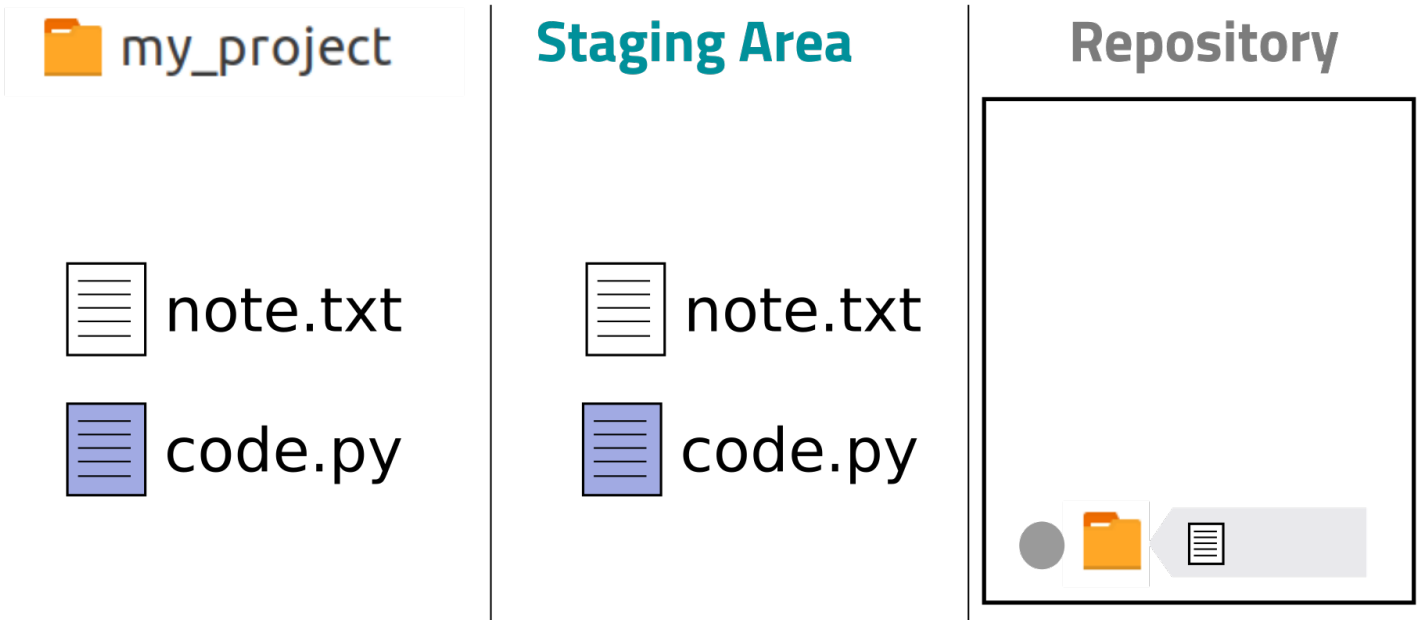


How to create a snapshot (commit)?

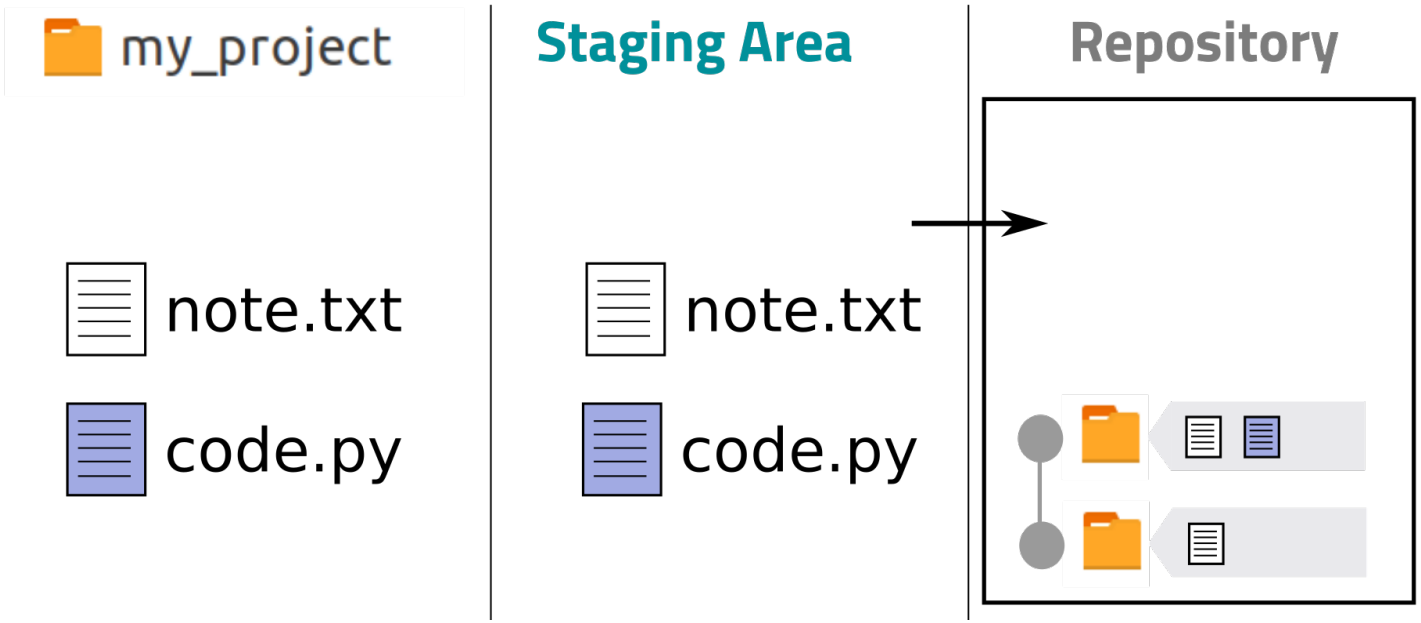



```
git add code.py
```

How to create a snapshot (commit)?

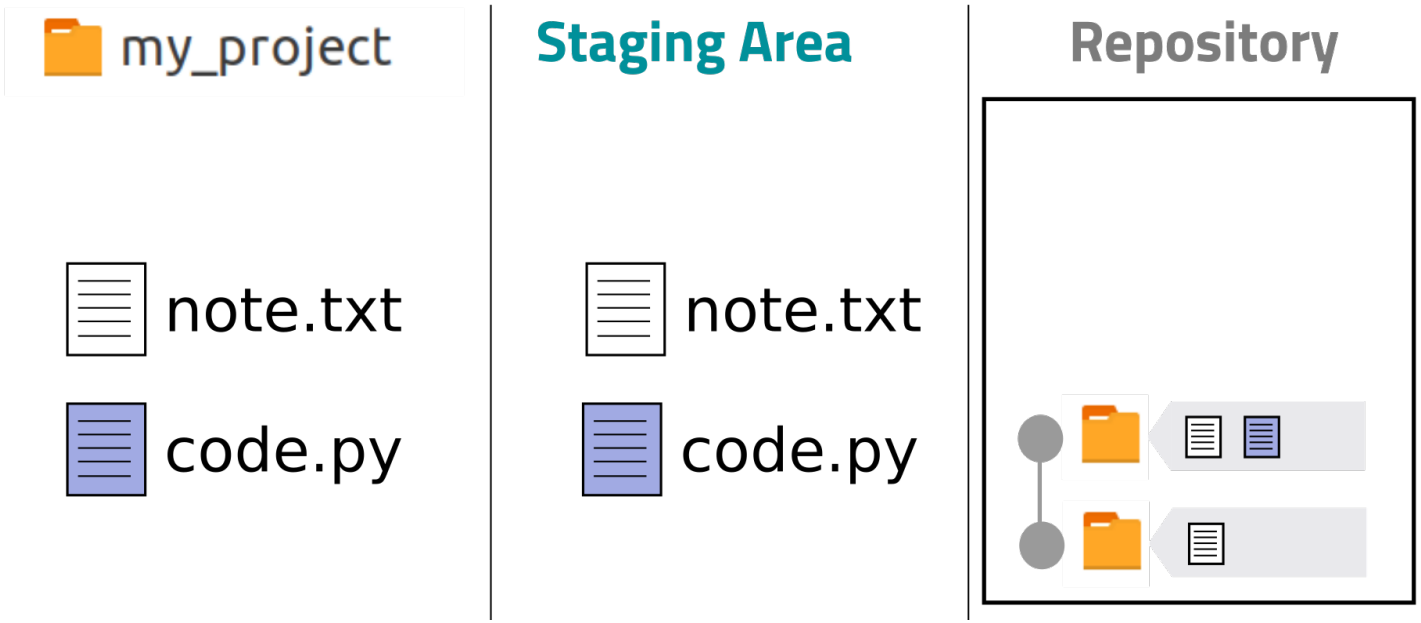


How to create a snapshot (commit)?

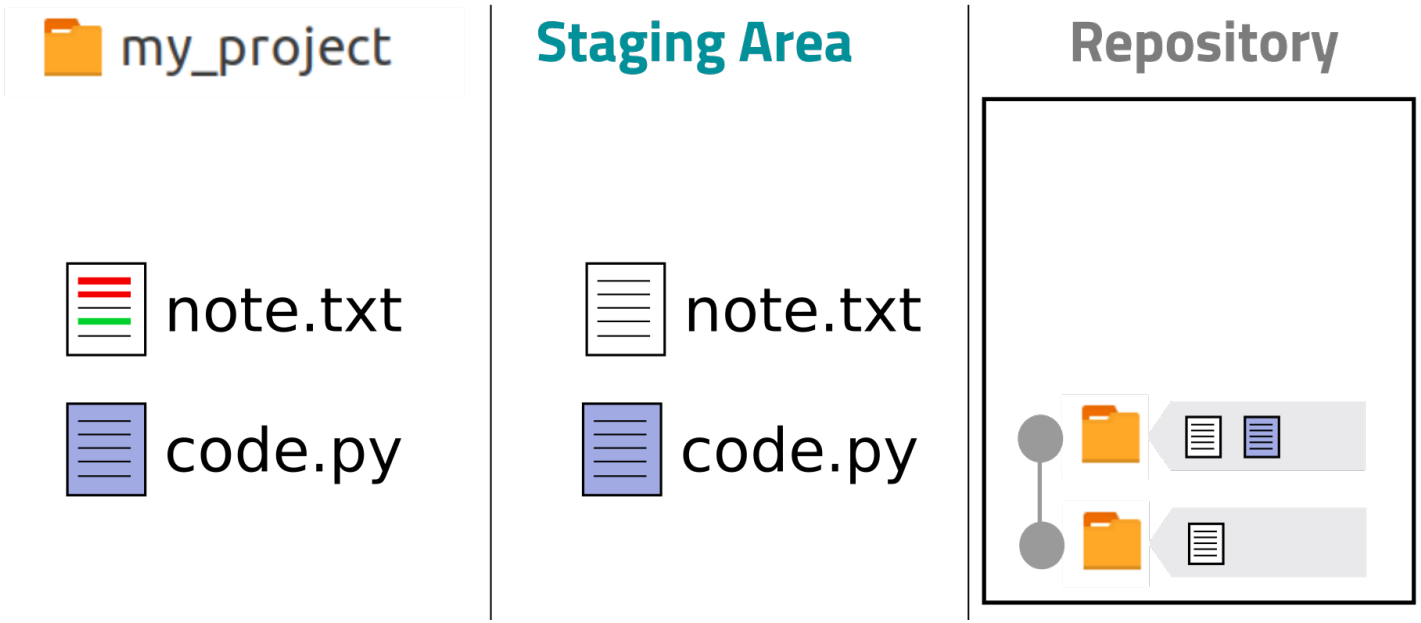



```
git commit -m "Implement some solver"
```

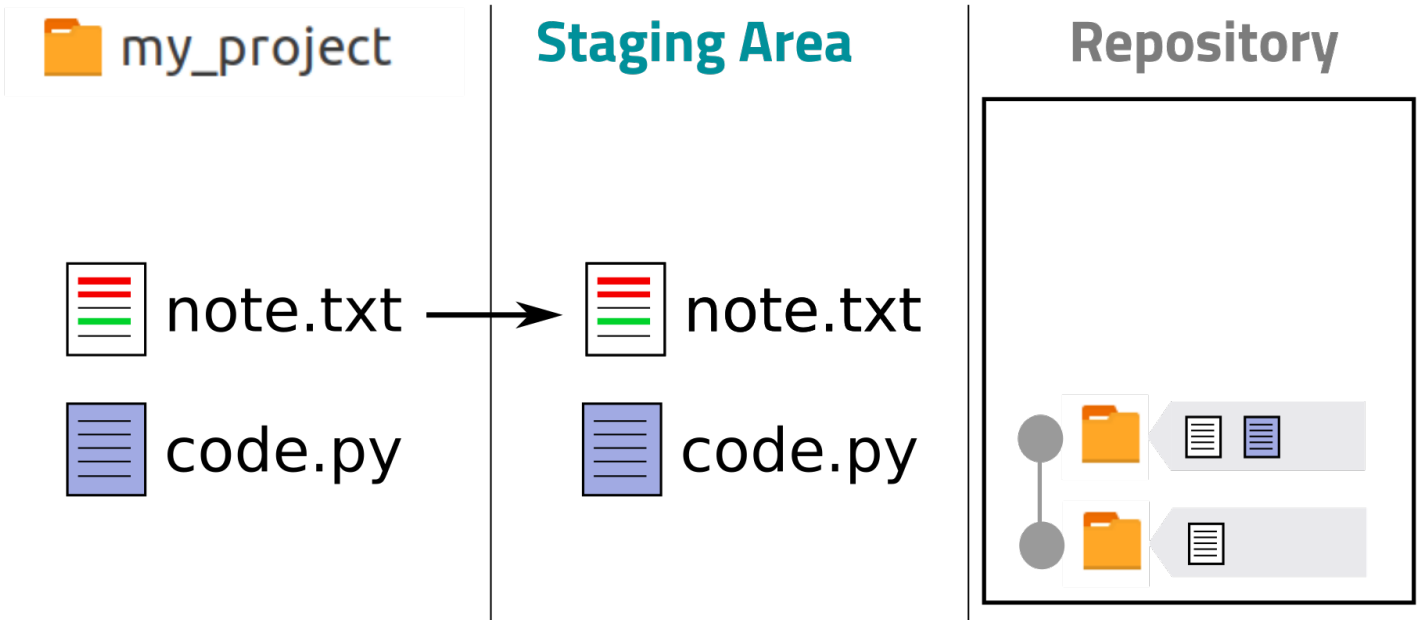
How to create a snapshot (commit)?



How to create a snapshot (commit)?

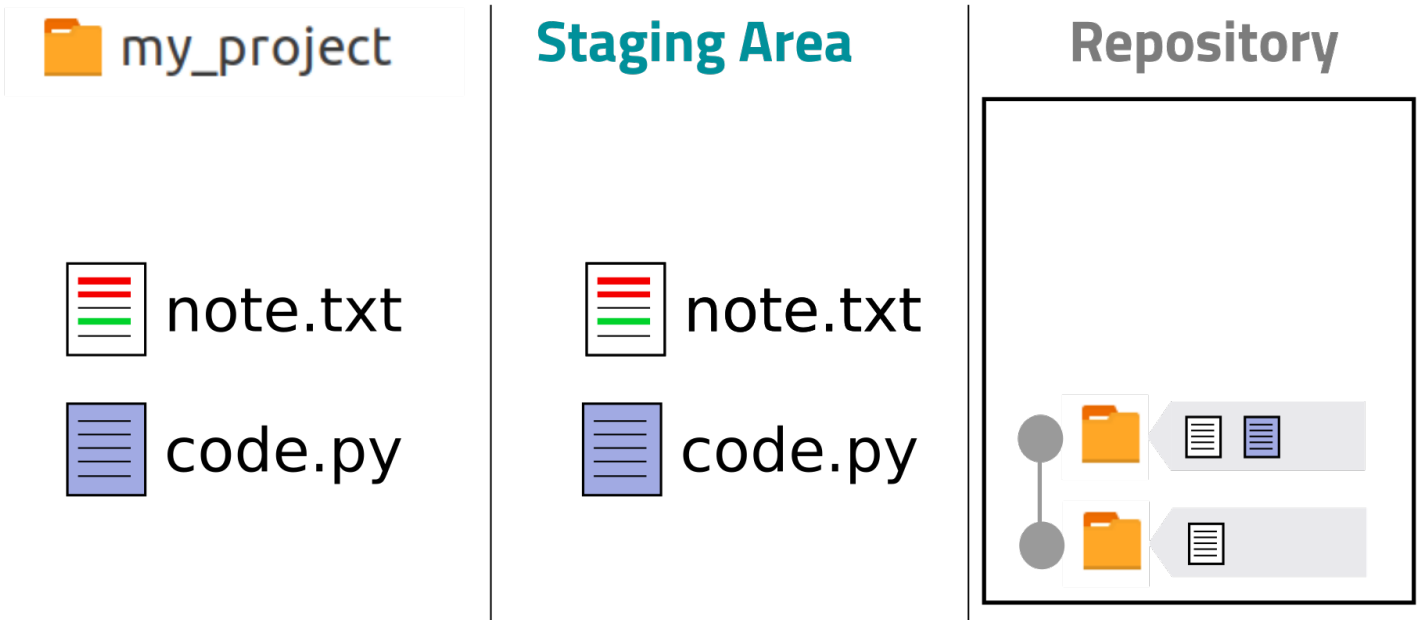


How to create a snapshot (commit)?

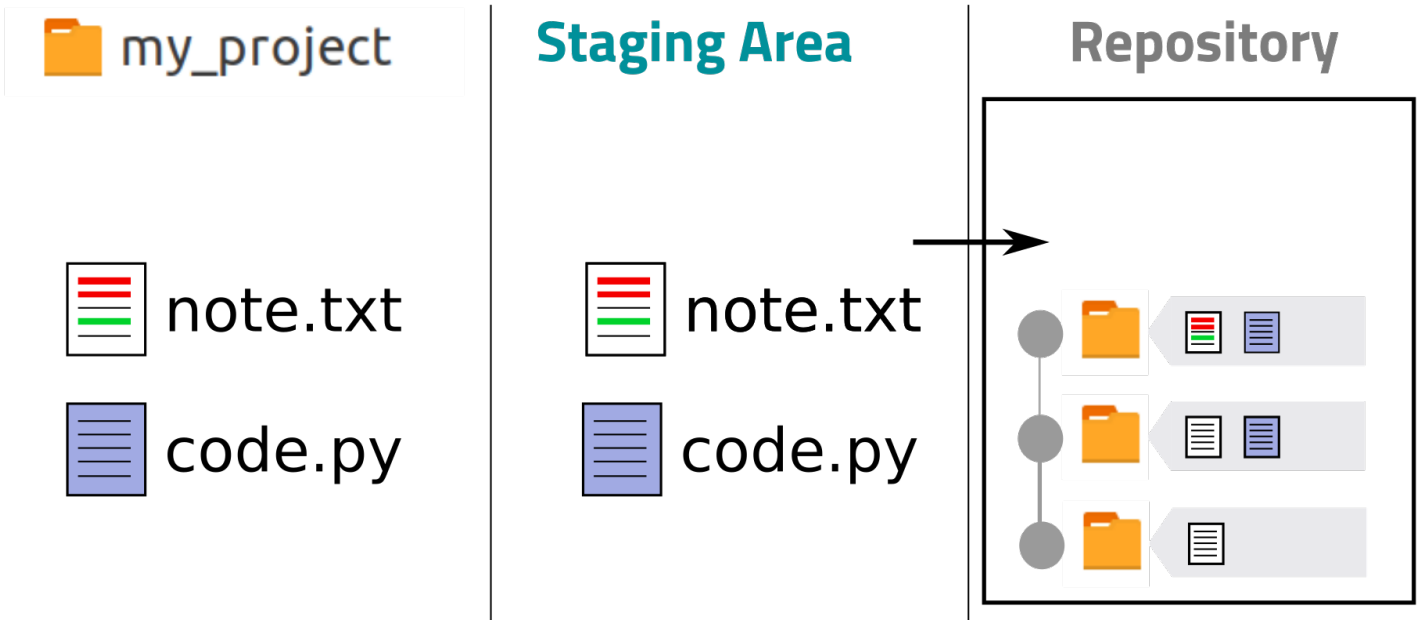



```
git add note.txt
```

How to create a snapshot (commit)?

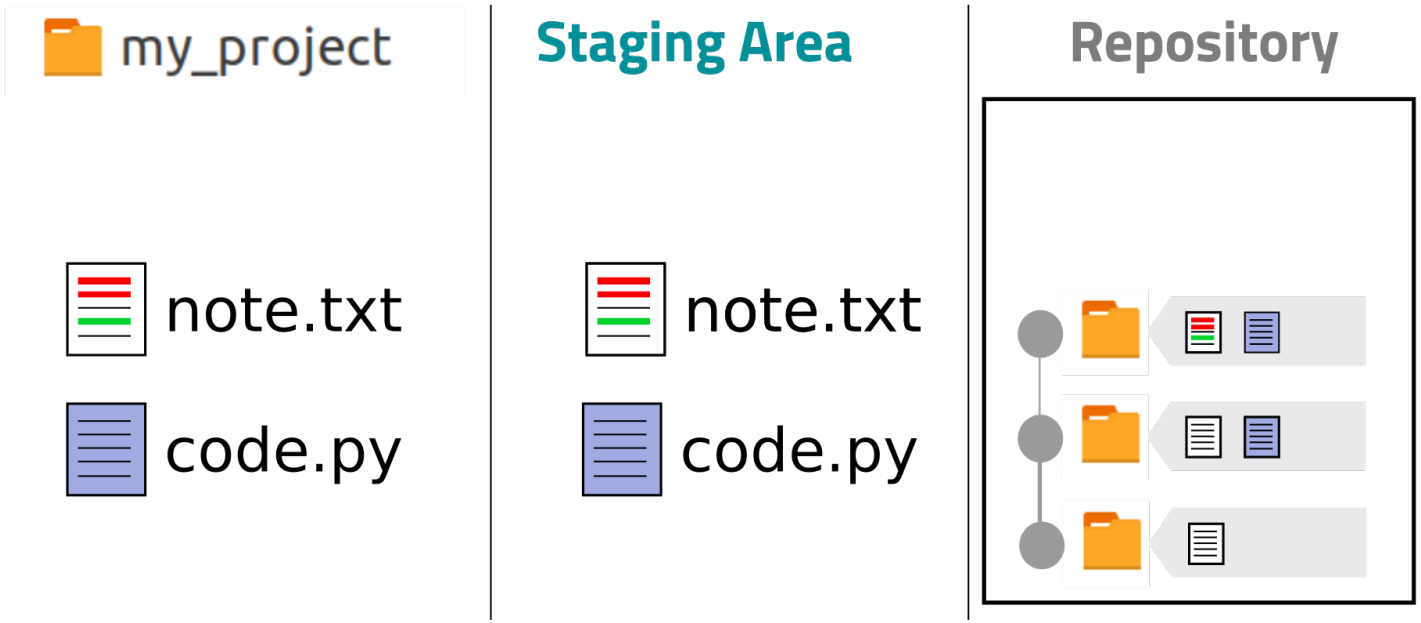


How to create a snapshot (commit)?




```
git commit -m "Fix typo in note"
```

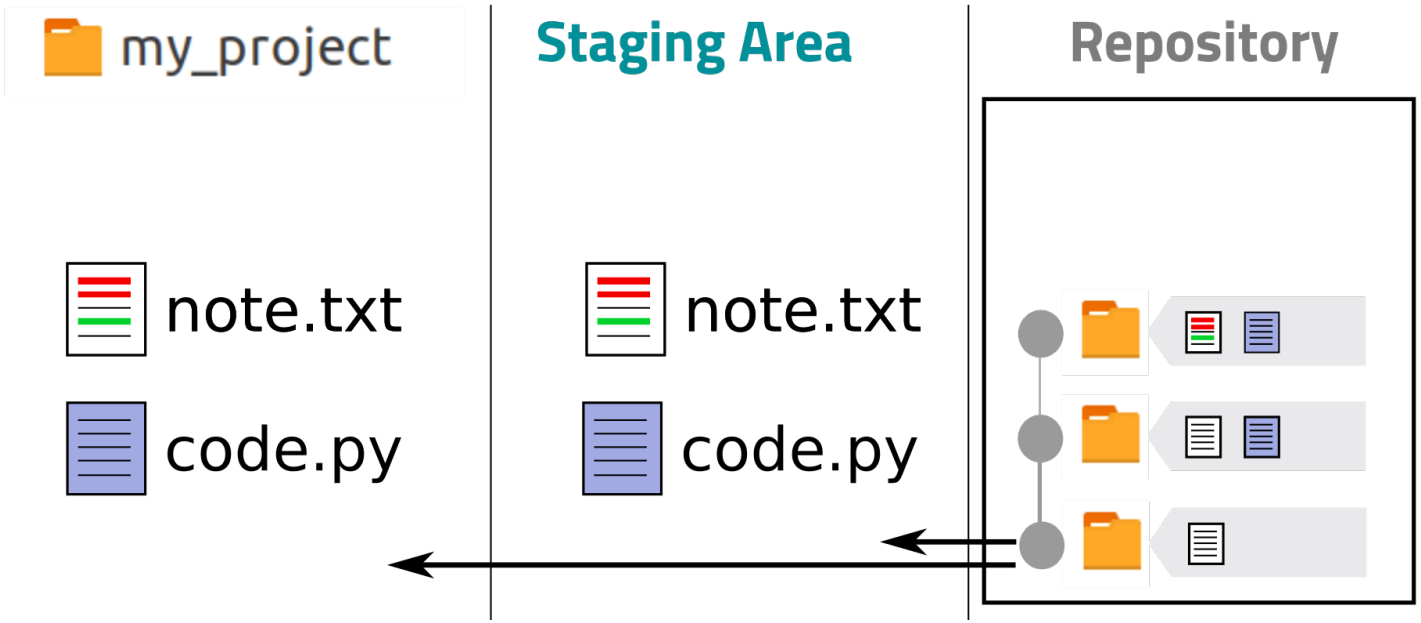
How to create a snapshot (commit)?



To check commit history

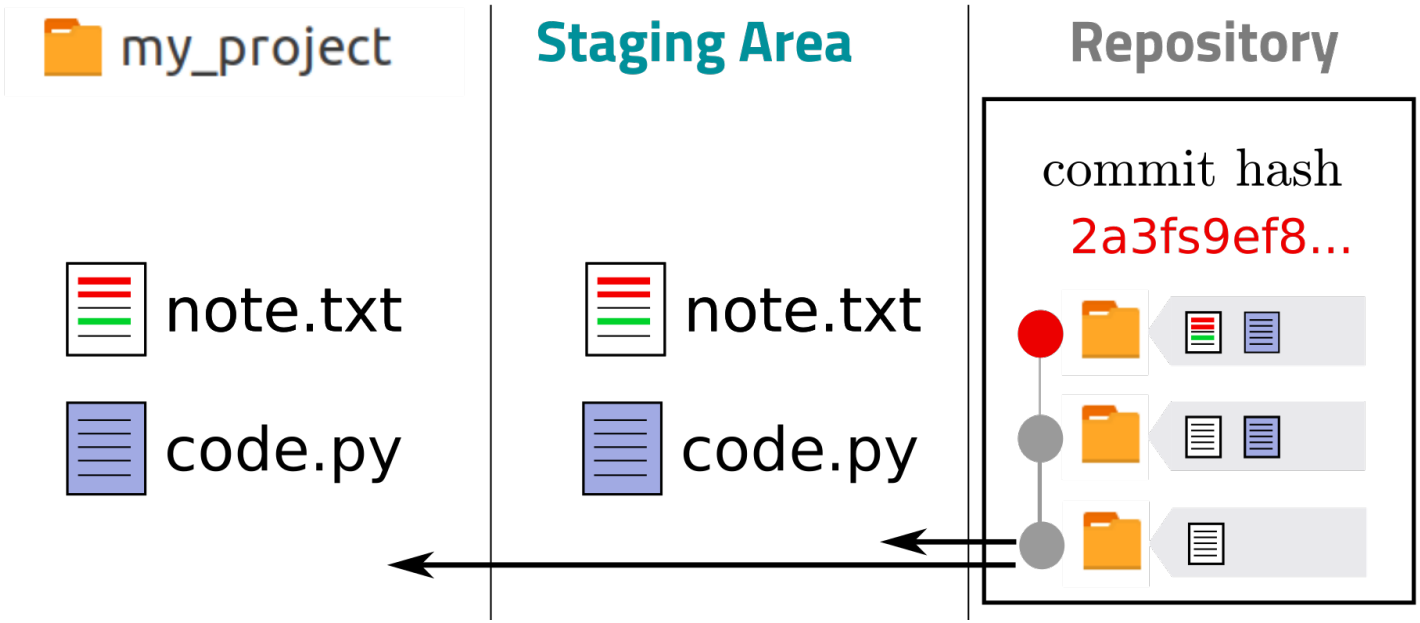
```
git log --all
```

Change between versions



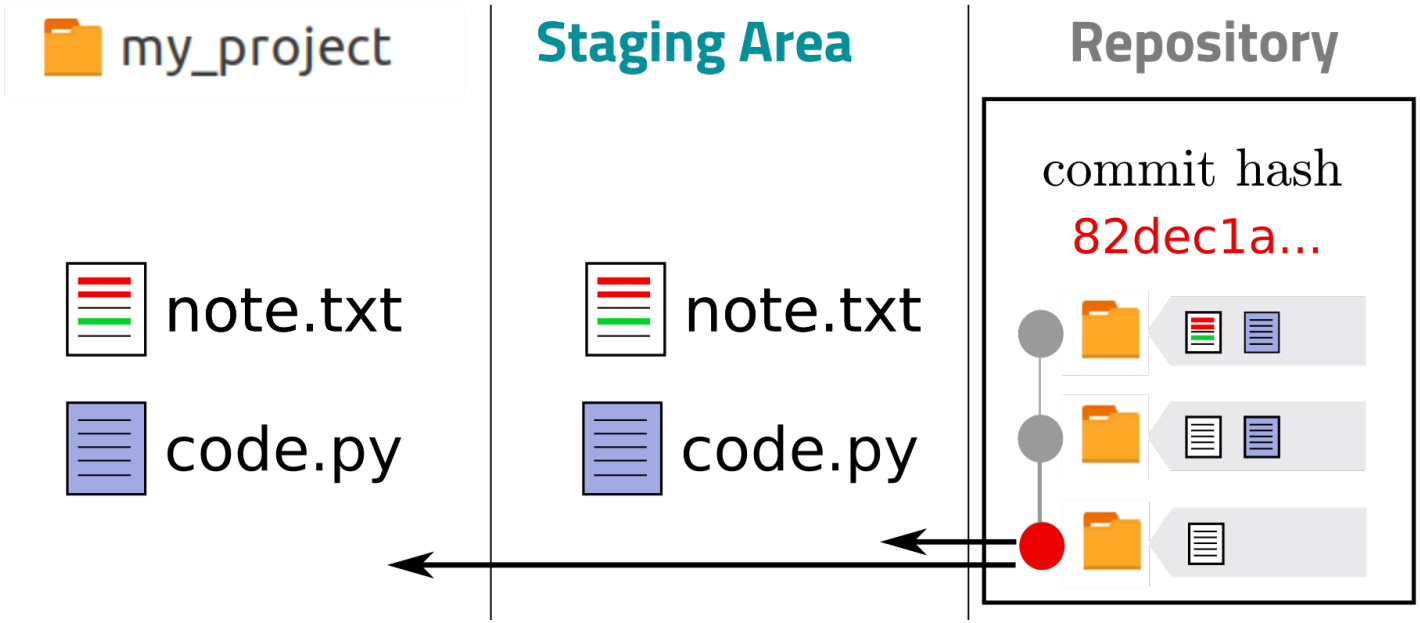
How to obtain a specific version of the project from history?

Change between versions



Each commit has a unique identifier (hash code). You can check the hash code for a specific commit when you run git log command.

Change between versions



To switch to a specific commit, you only need to type just a few initial characters of its hash code.

```
git checkout 82dec1a
```

Change between versions

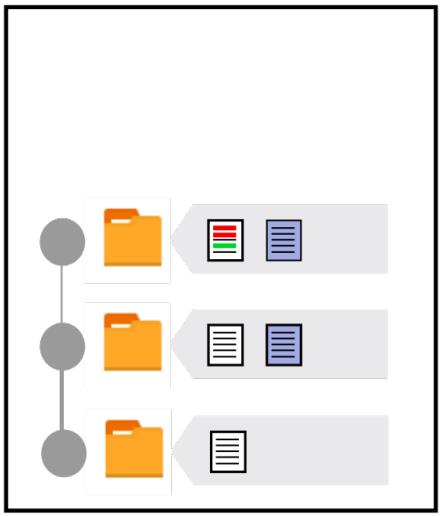
my_project

note.txt

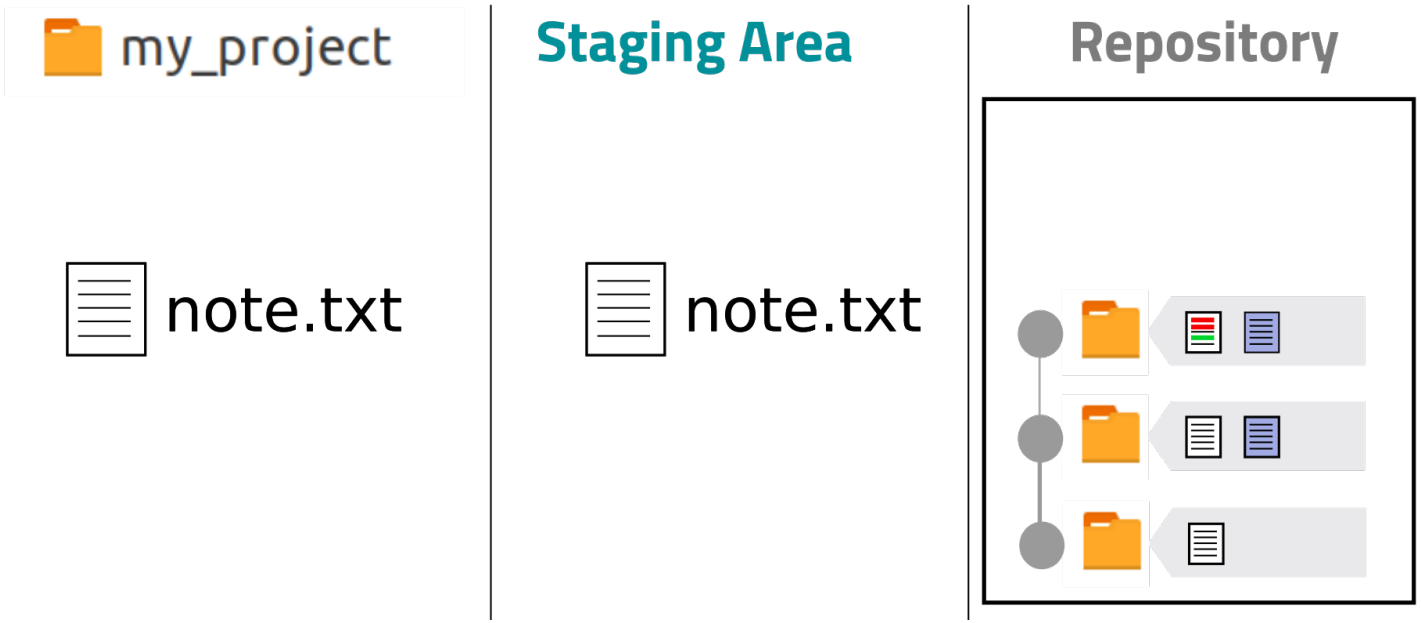
Staging Area

note.txt

Repository



Change between versions

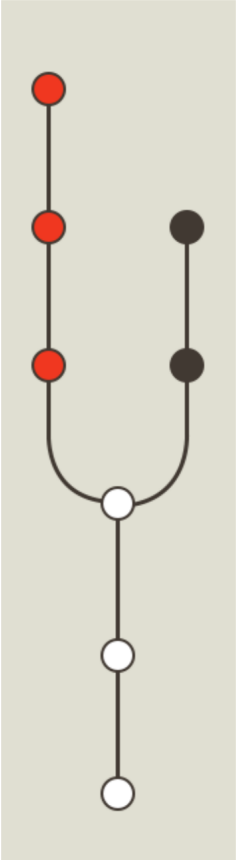


To go back to the top of the main branch

```
git checkout main
```

Branching

Branches allow you to work on a feature without affecting the rest of the project. After the feature is implemented, you can merge the changes in a branch back to the main line of development.



To create a new branch called feature_01

```
git branch feature_01
```

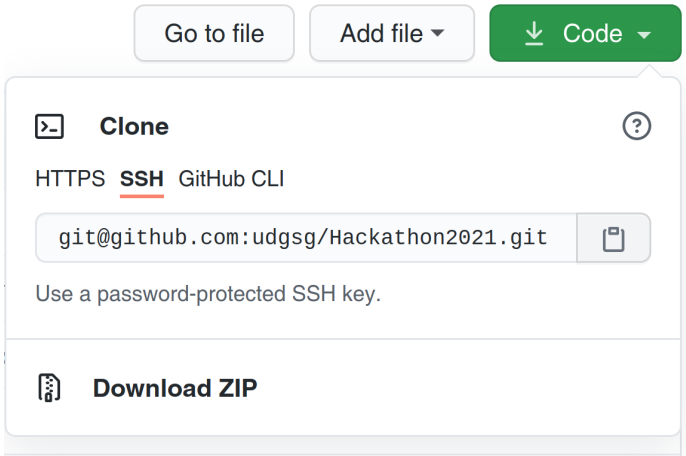
To list branches in a repository

```
git branch
```

To switch to branch

```
git checkout feature_01
```

Working with a remote repository



To create a local repository from a remote

```
git clone git@github.com:udgsg/Hackathon2021.git Hackaton
```

To push changes from your local repository to GitHub

```
git push
```

To get data from a remote repository

```
git pull
```

Summary

`git init`	Create an empty repository.
`git status`	Check for changes in the working directory or the staging area.
`git add` *somefile*	Add a file to the staging area.
`git checkout` *somefile*	Revert to a version in the staging area.
`git checkout --` *somefile*	Revert to a version in the repository (last commit).
`git commit`	Create a new commit, i.e. save the state from the staging area into the repository history.
`git log`	List the repository history.
`git push origin`	Push your current branch to a remote repository called **`origin`**.
`git pull origin`	Get the latest files from a repository called **`origin`** into your current branch.
`git help` *command*	List a **`man`** page for a specific git command (e.g. `git help log`).