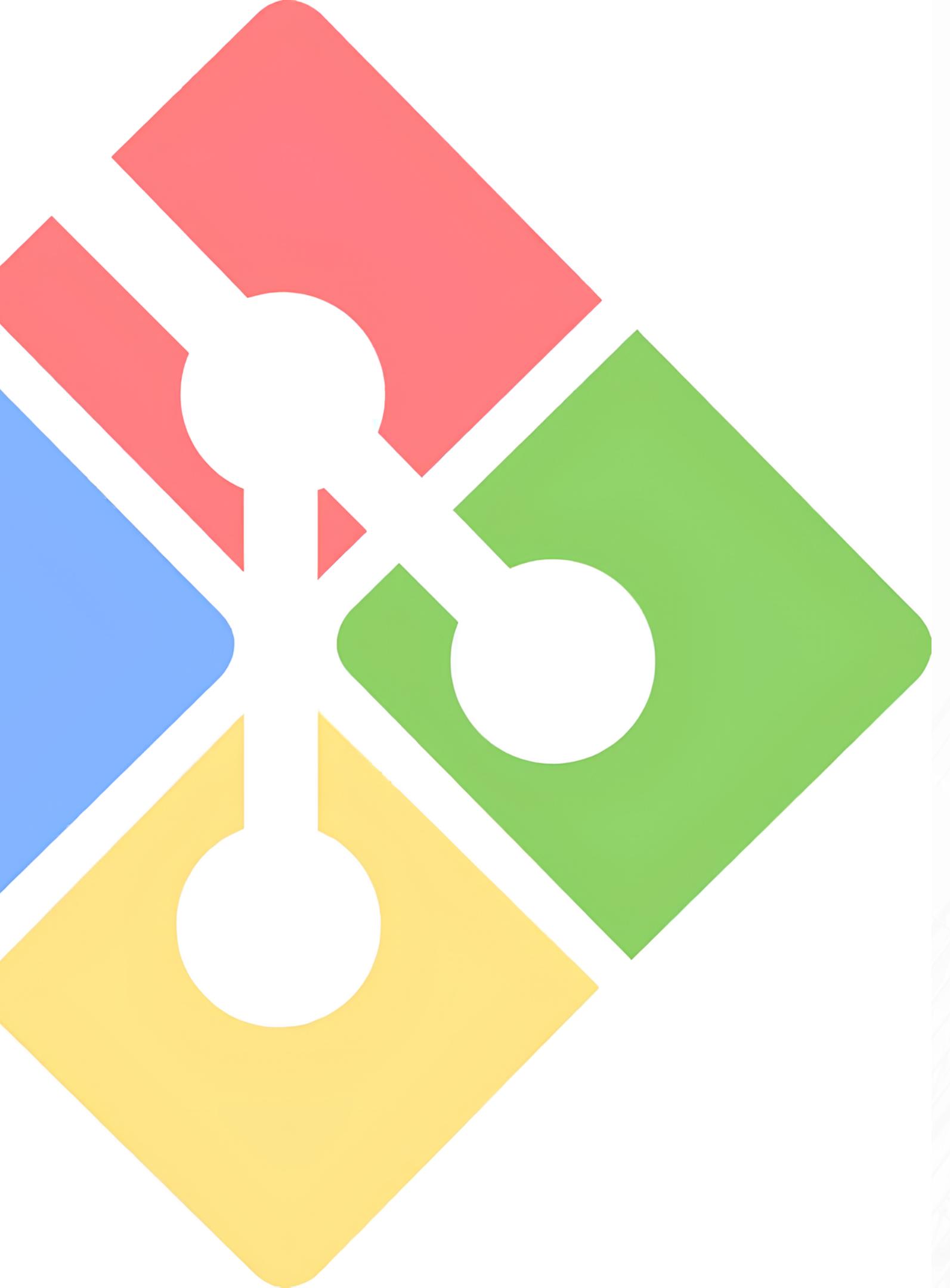


# WORK-CASE №1

---

Git is a powerful tool for software version control. Git has become not only an integral part of software development, but also a significant element of the modern IT industry. Get ready to learn more about its core concepts, benefits, and secrets to successfully using Git in your work!

Created by Max Karpenko, Vlad Sapozhnyk and Dmytro Onufriiev

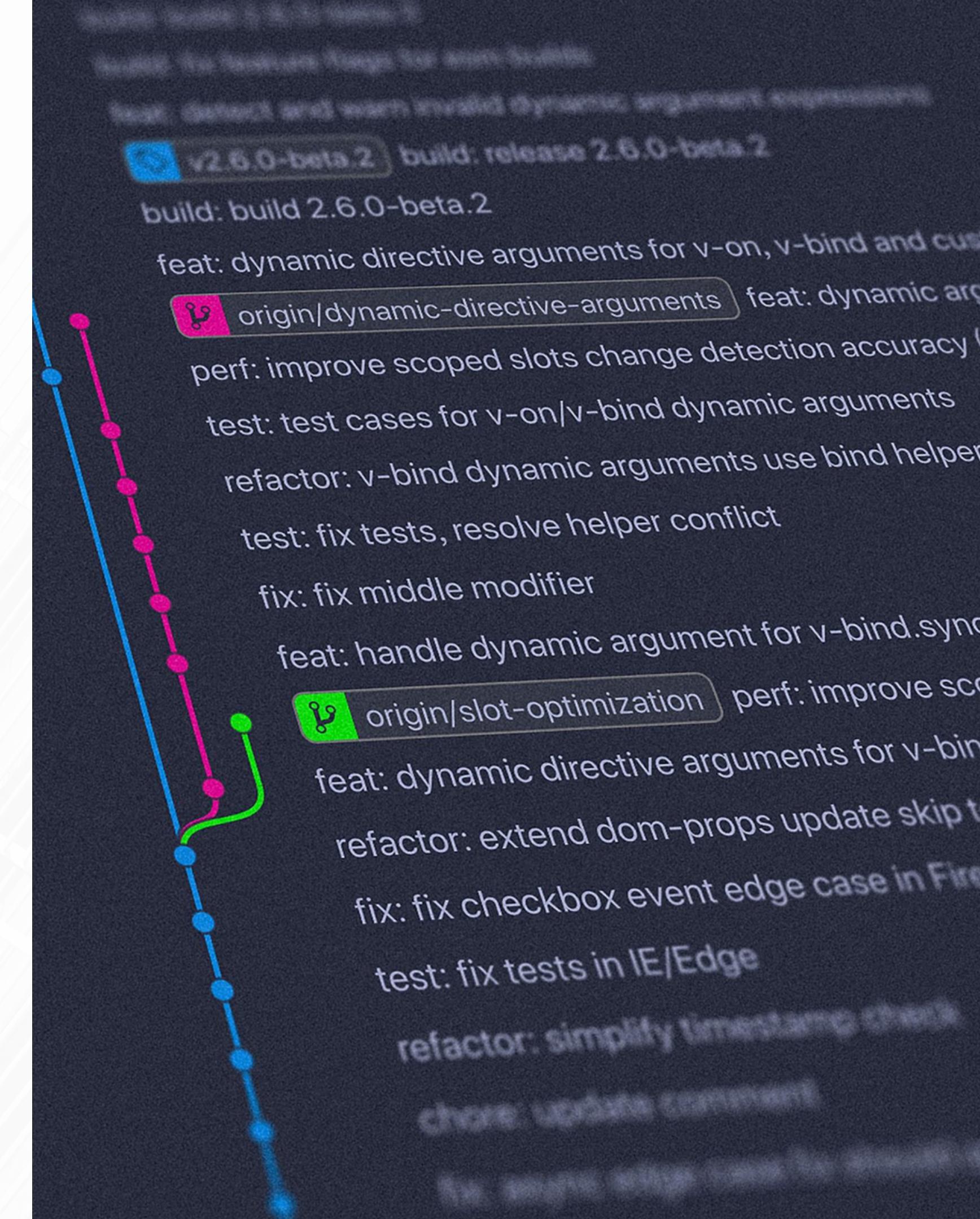


# WHAT IS GIT?

Git is a distributed version control system used primarily for tracking changes in source code during software development, but it can also be used to keep track of changes in any set of files. It facilitates collaboration among developers and allows multiple people to work on the same project simultaneously without interfering with each other's work. Here are the key purposes and basic actions/commands in Git

# BASIC COMMANDS

- **git init:** Initializes a new Git repository. This command creates a new .git directory in the project, tracking all subsequent changes.
- **git clone [url]:** Creates a local copy of a remote repository. This command is used to start working locally on an existing project that's hosted elsewhere.
- **git add [file]:** Adds a file or changes in a file to the staging area. It's the first step in the workflow to commit changes.
- **git commit -m "[commit message]":** Commits the staged changes to the repository with a descriptive message explaining what changes were made and why.
- **git status:** Shows the status of changes as untracked, modified, or staged.
- **git push [remote] [branch]:** Pushes committed changes from your local repository to a remote repository. The remote is usually called origin, and the branch could be main or any other branch.
- **git pull [remote] [branch]:** Fetches changes from the remote repository to your local repository and immediately updates your current branch to match that remote. It's a combination of git fetch and git merge.



# WHAT IS COMMIT



A "commit" in Git is a snapshot of your repository at a specific point in time. It acts as a checkpoint to which you can return later if needed. Each commit contains a set of changes to files in your repository, including new files added, existing files that were modified, or files that were deleted. Commits allow you to track the evolution of your project over time.

## HOW A COMMIT TRACKS CHANGES IN FILES:

**UNIQUE IDENTIFIER:** EACH COMMIT IS IDENTIFIED BY A UNIQUE SHA-1 HASH. THIS HASH REFLECTS THE CONTENTS OF THE COMMIT, INCLUDING CHANGES TO FILES, THE COMMIT MESSAGE, AUTHOR, AND DATE. THE HASH ENSURES THE INTEGRITY OF YOUR PROJECT'S HISTORY.

**PARENT REFERENCE:** COMMITS IN GIT MAINTAIN A REFERENCE TO THEIR PARENT COMMIT(S) (THE COMMIT(S) THAT DIRECTLY PRECEDED THEM). THIS LINKAGE CREATES A HISTORY OR A CHAIN OF COMMITS THAT CAN BE FOLLOWED BACK TO THE INITIAL COMMIT IN THE REPOSITORY. FOR MERGE COMMITS, THERE ARE MULTIPLE PARENTS.



## Snapshot of Changes

When you make a commit, Git calculates the differences (or "diffs") between the files in the commit and their counterparts in the previous commit. Rather than storing entire files for every commit, Git stores these diffs, which is space-efficient. However, for ease of access and performance, Git also maintains internal representations (trees and blobs) that allow it to reconstruct any version of the project at any commit point.

## Commit Message:

Every commit includes a commit message, which is a brief description written by the user making the commit. This message usually describes the changes made and the reason for them, providing context for the commit. This is crucial for understanding the project's history and the purpose of each change.

## Author and Committer Information:

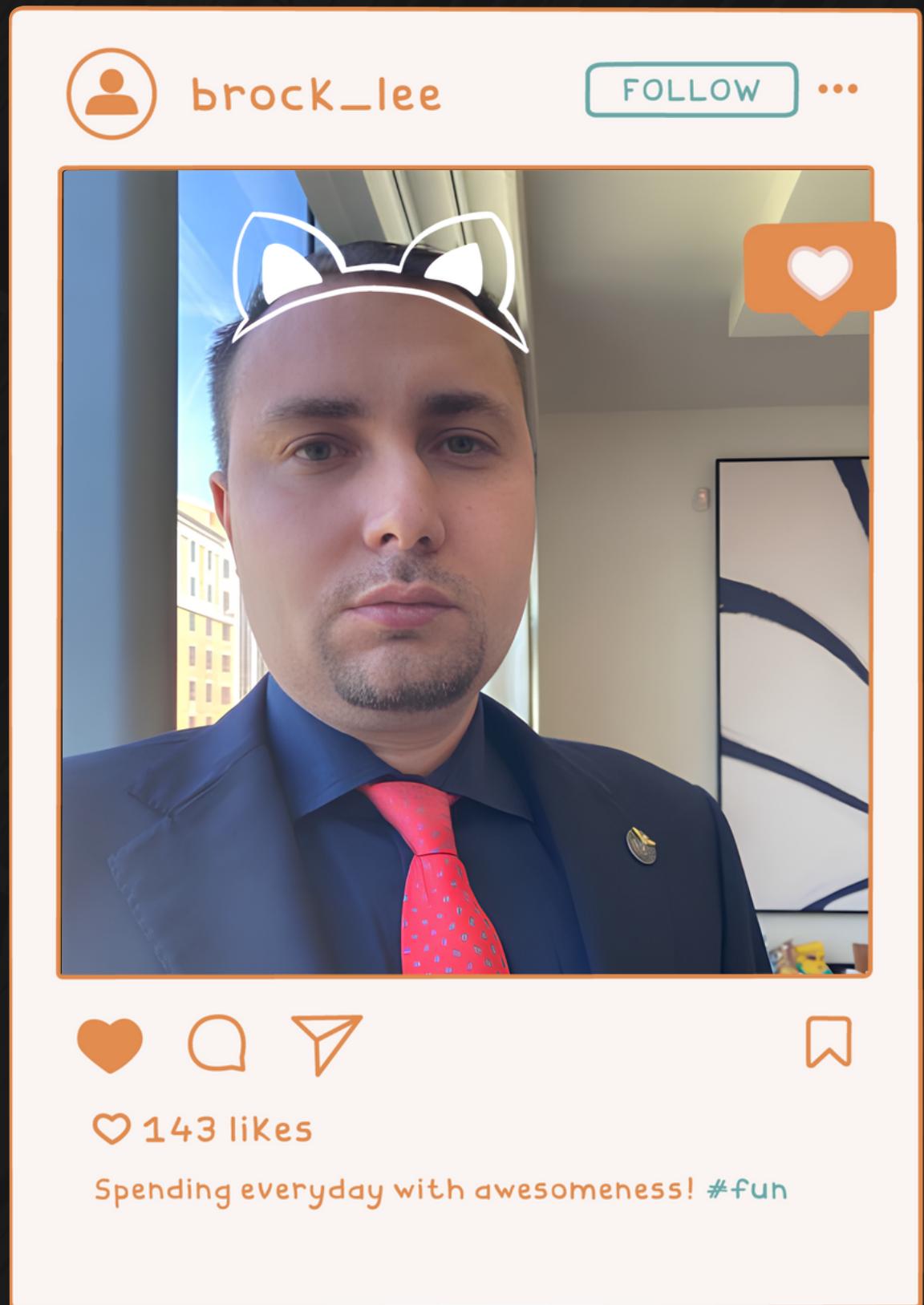
Git commits also store the name and email address of the person who made the changes (author) and the person who committed those changes to the repository (committer), along with timestamps for both actions. In most cases, the author and committer are the same person, but they can differ in collaborative workflows.

# WE USE GITHUB

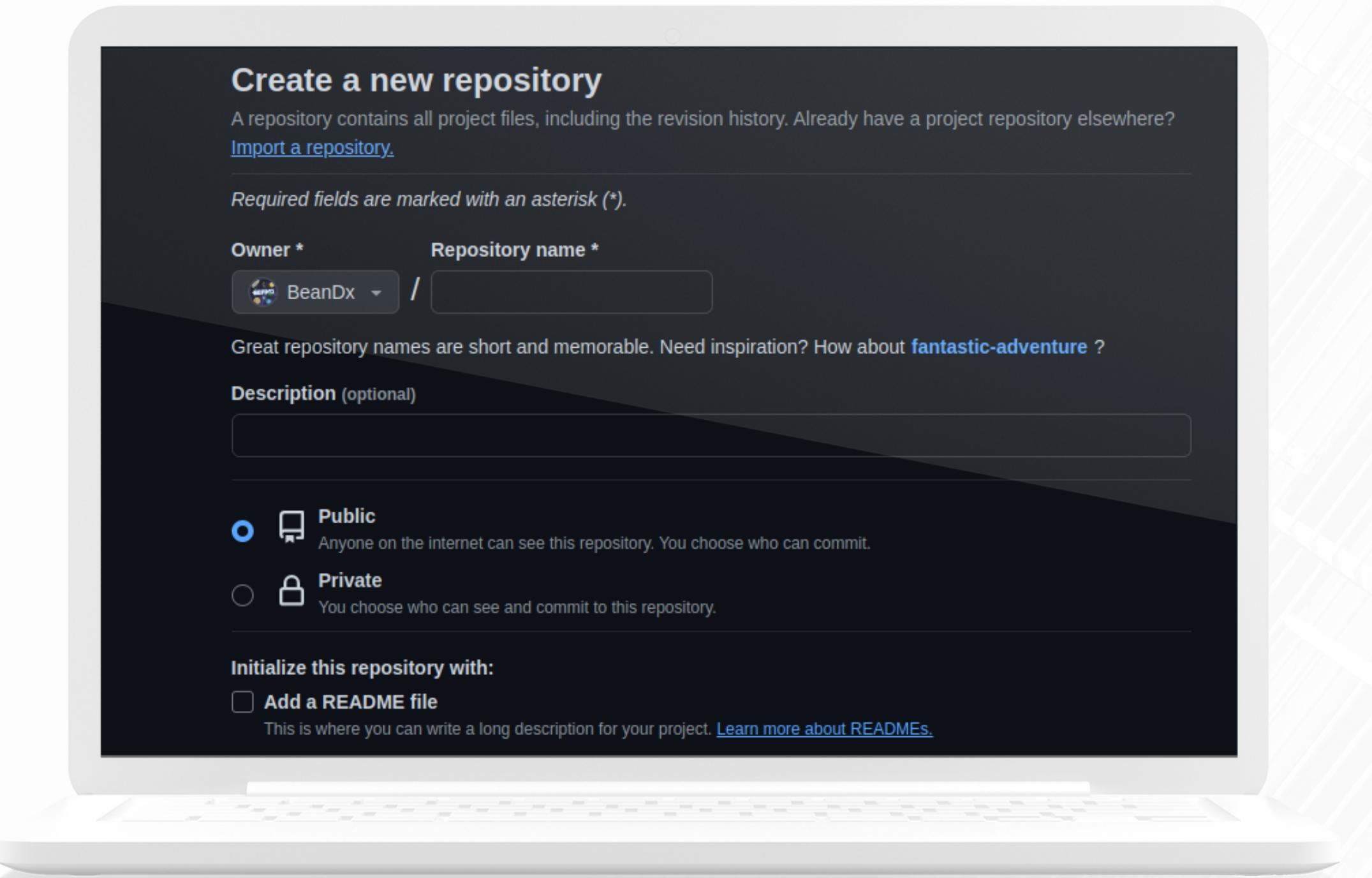
## BTW

GitHub is a web-based platform and hosting service for version control using Git. It offers a wide range of features for software development teams and individual developers, including:

- Version Control: GitHub allows developers to collaborate on projects by managing and tracking changes to their codebase using Git.
- Repository Hosting: It provides a place to store Git repositories, making it easy to share code with others and collaborate on projects.
- Collaboration Tools: GitHub offers features such as pull requests, code reviews, and issue tracking to facilitate collaboration among team members.
- Project Management: Developers can organize their work using project boards, milestones, and labels, helping to keep track of tasks and prioritize work.
- Community and Social Features: GitHub is also a social platform where developers can follow each other, star repositories, and contribute to open-source projects.
- Overall, GitHub has become a central hub for software development, fostering collaboration, innovation, and open-source contributions within the developer community.



# HOW TO CREATE REPOSITORY ON THE GITHUB



# OUR REPOSITORY



**Vlad Sapozhnyk**

vlad0syk · he/him

Keep moving forward!

[Edit profile](#)

0 followers · 5 following

[Update README.md](#)

 00f8t committed 2 minutes ago

[Update README.md](#)

 vlad0syk committed 44 minutes ago

[Update README.md](#)

 vlad0syk committed 1 hour ago

[create new dir work-case-1](#)

A circular profile picture of a cat wearing a blue baseball cap with the text "PLEASE BE PATIENT I HAVE AUTISM" printed on it. A small smiley face icon is in the bottom right corner of the image.

00f8t

[Edit profile](#)



**BeanD**

BeanDx

Pseudo programmer in html, css, cpp, python ;)

[Edit profile](#)

0 followers · 4 following

# THANK YOU