

# MASTERING GIT

**Soumaya Erradi**

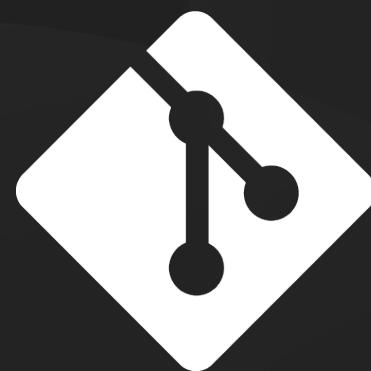
*Senior Software Developer & IT and Electronics Instructor*

# Soumaya Erradi

- Senior software developer @ Atlantis
- Frontend and Web3 specialist
- International speaker
- IT and electronics divulger



# Software versioning



git

# Version Control System

Git is a distributed version control system (DVCS) that allows multiple developers to work on a project simultaneously.



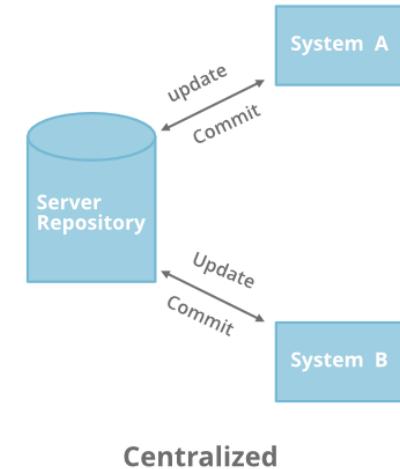
# **Created by Linus Torvalds**

Originally developed in 2005 to manage the Linux kernel's development.

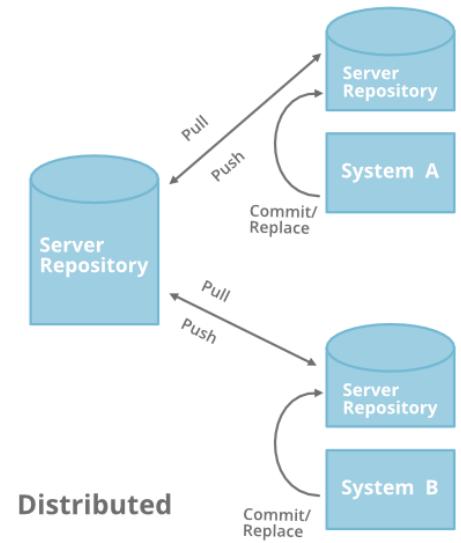


# Distributed Architecture

Unlike centralized version control systems, every developer has a full copy of the repository, including the entire history.



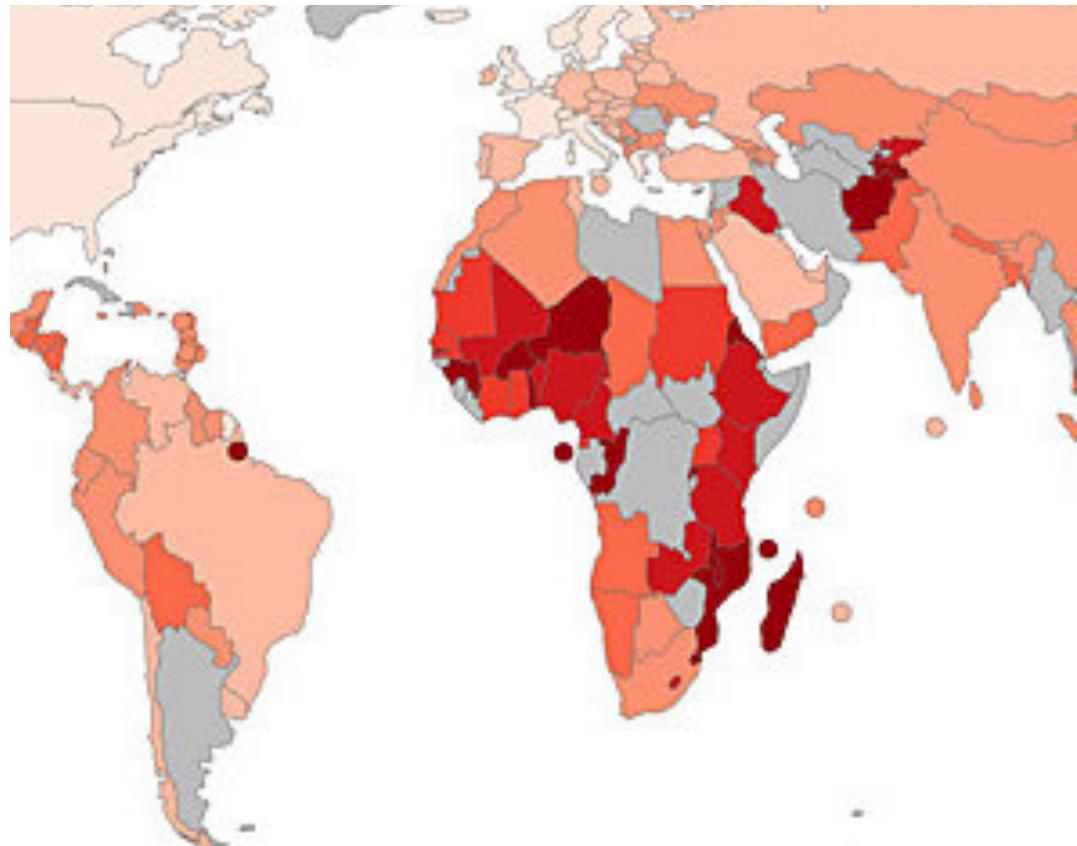
Centralized



Distributed

# Widely Used

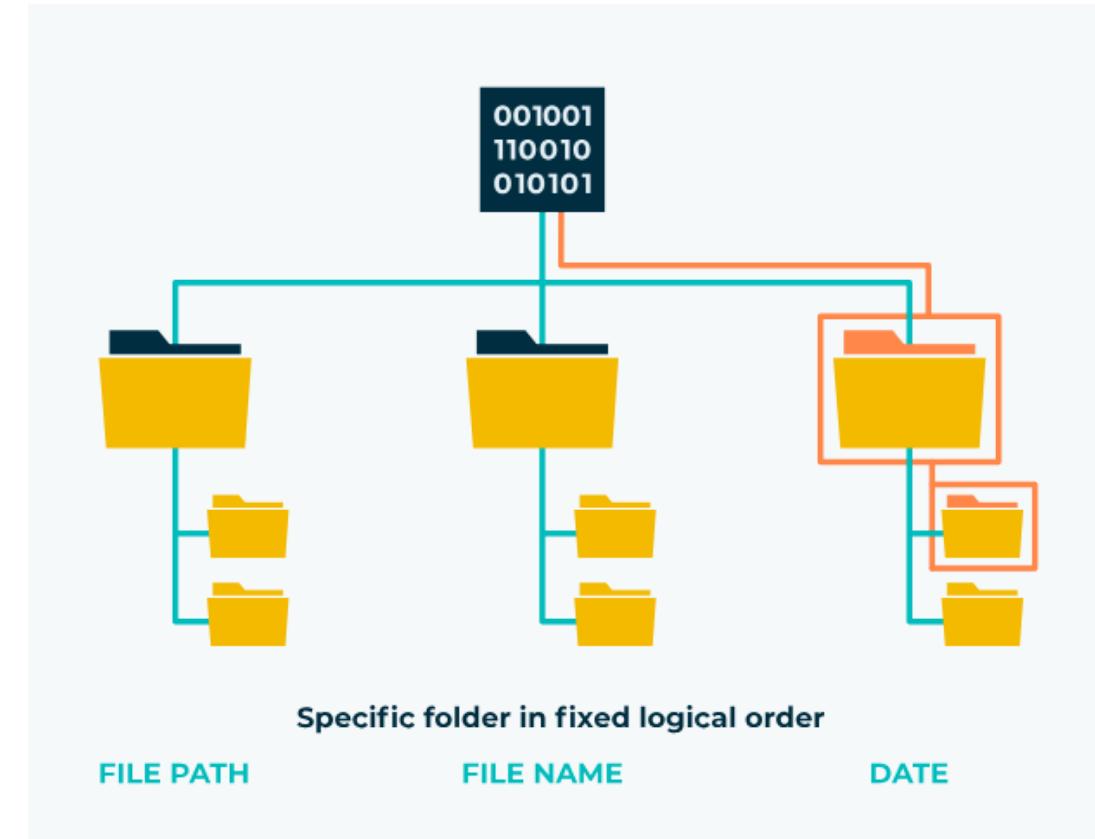
Git is the most popular version control system today, used by millions of developers worldwide.



# Understanding Git Objects

# Git's Content-Addressable Filesystem

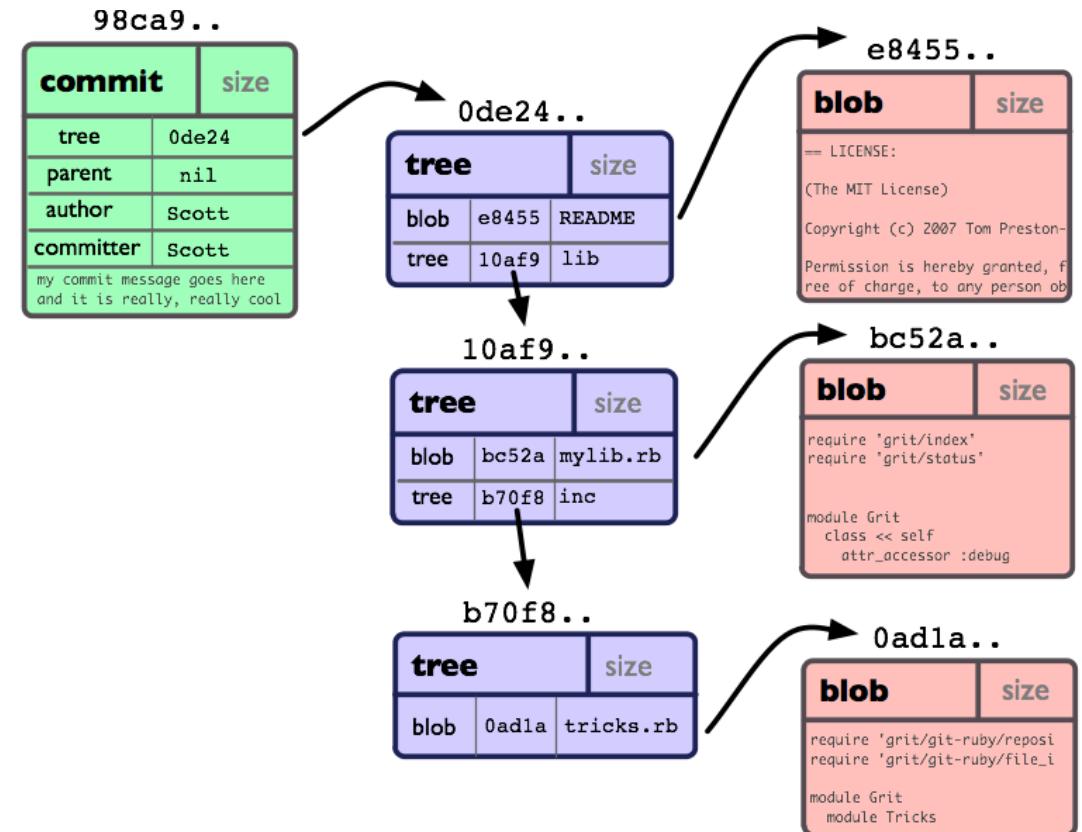
Git is fundamentally a content-addressable filesystem where all the data is stored as objects.



# Three Main Types of Git Objects

There are three main types of objects:

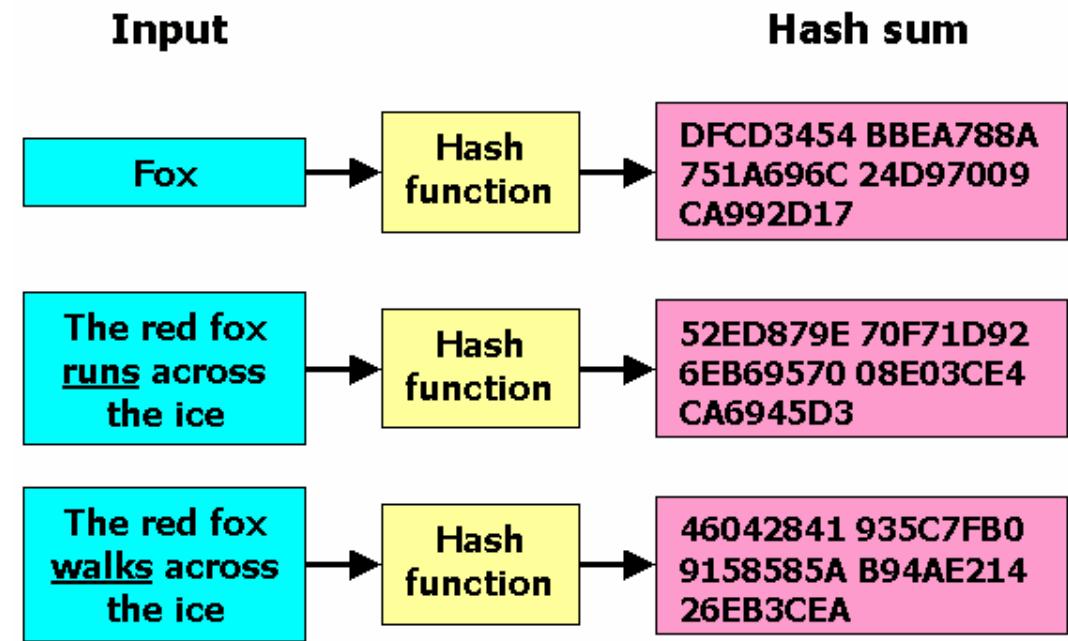
- commit
- tree
- blob



# SHA-1 Hashes in Git

Each of these objects is identified by a unique SHA-1 hash, which is a 40-character string.

This hash ensures that even the smallest change in content results in a completely different hash.





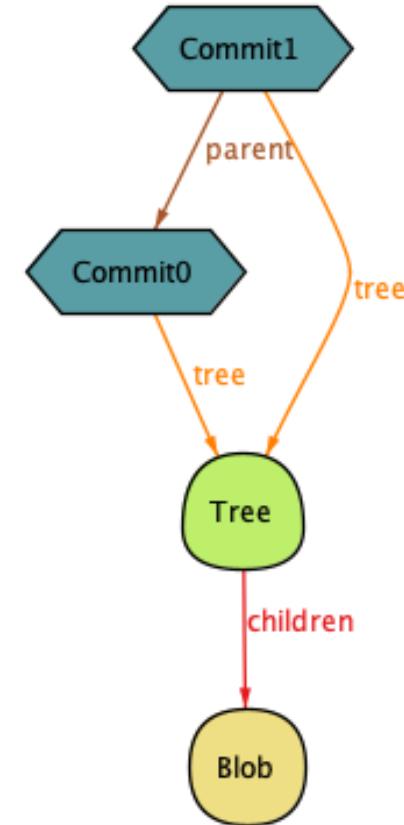
# Demo

# The Commit Object

# What is a Commit object?

A commit object in Git is one of the most important objects. It represents a snapshot of the project at a specific point in time.

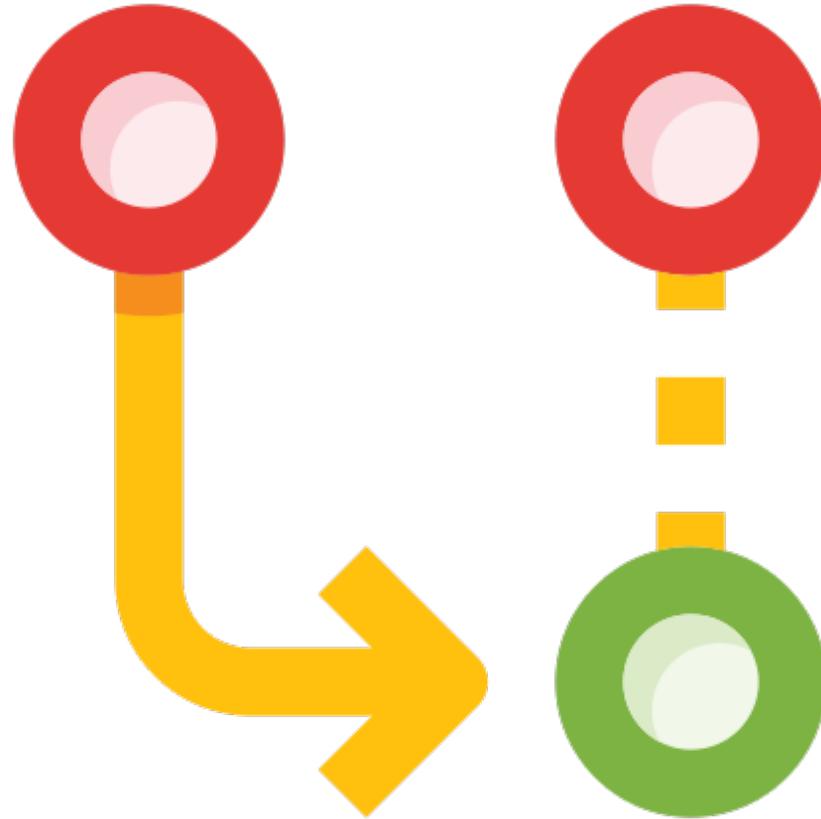
Each commit contains the commit message, a pointer to a tree object, and references to parent commits.



# Commit as a Snapshot

Unlike some version control systems, a commit in Git captures the entire working directory at a specific point in time, not just the changes.

This snapshot includes the state of every file and directory in the project.



# Anatomy of a Commit object

A commit object contains:

- **Commit message**: Describes what this commit does.
- **Tree pointer**: Points to a tree object representing the state of the project.
- **Parent commits**: References to previous commits, forming the commit history.

ae668..	
commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



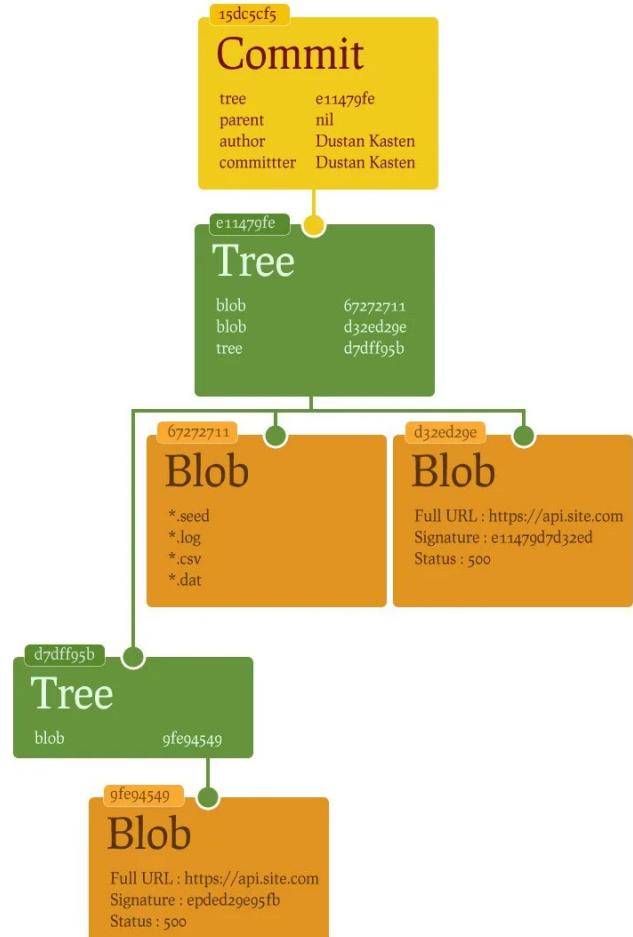
# Demo

# The Tree Object

# What is a Tree object?

A tree object in Git represents a directory. It holds references to blobs (files) and possibly other tree objects (subdirectories).

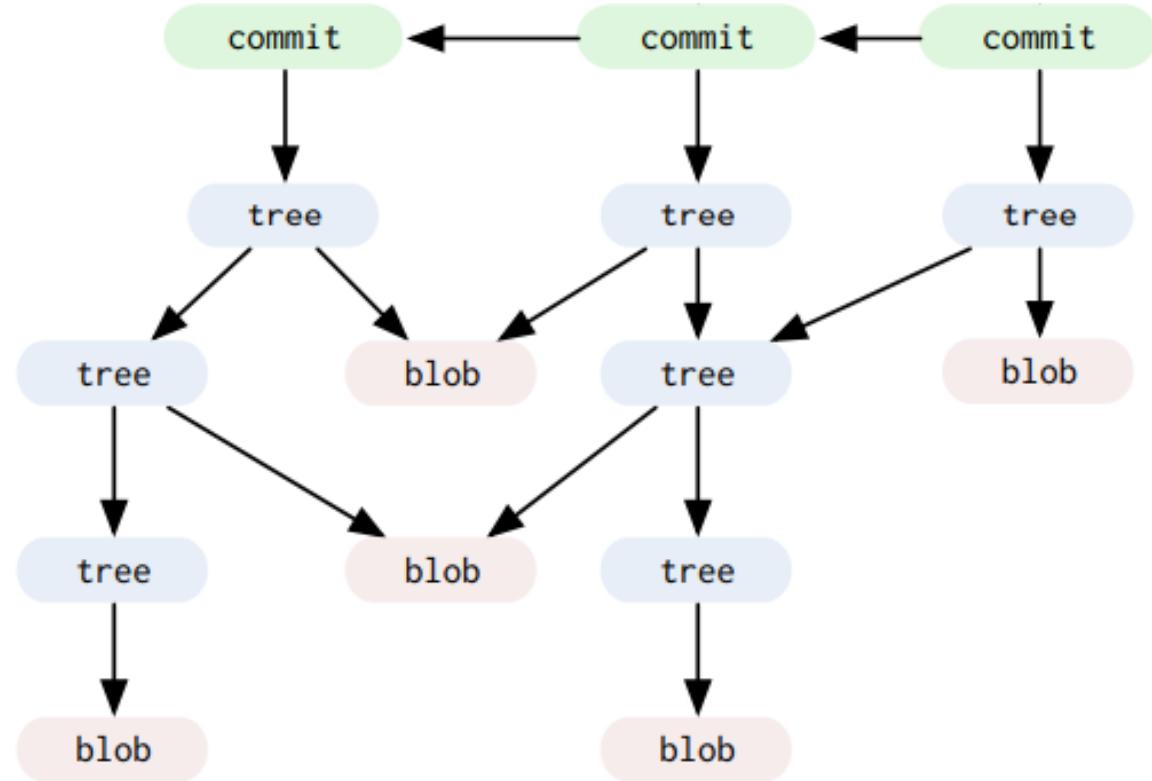
Tree objects are crucial for capturing the hierarchical structure of your project.



# How Tree objects work

When you create a commit, Git uses tree objects to store the state of the files and directories.

Each tree object points to the blobs and other trees that represent the contents of that directory at the time of the commit.



# Anatomy of a Tree object

A tree object contains:

- References to blobs (files) and their corresponding filenames.
- References to other tree objects (subdirectories) and their corresponding directory names.

The tree object provides the structure that organizes the files and directories in your project.

c36d4..

tree	size
blob	5b1d3 README
tree	03e78 lib
tree	cdc8b test
blob	cba0a test.rb
blob	911e7 xdiff



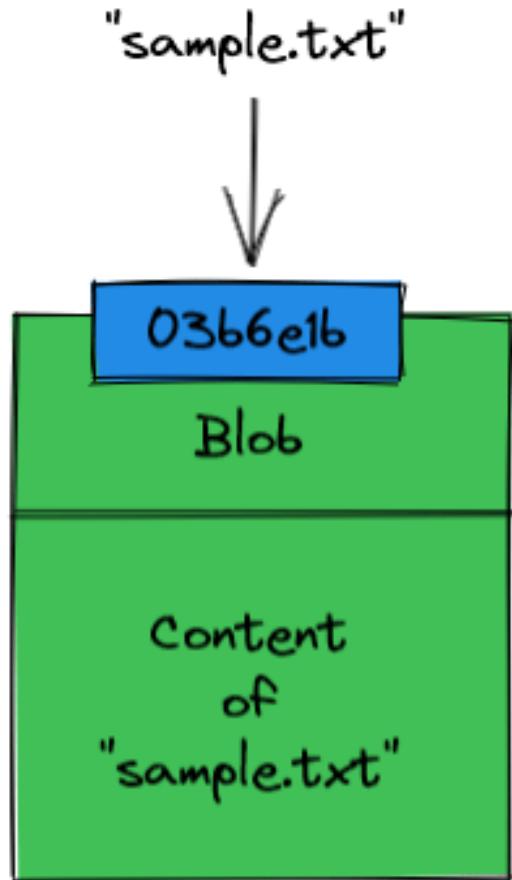
# Demo

# The Blob Object

# What is a Blob object?

A blob (Binary Large Object) in Git is used to store the contents of a file.

Blobs are the simplest type of object in Git, containing nothing but the raw data of a file.

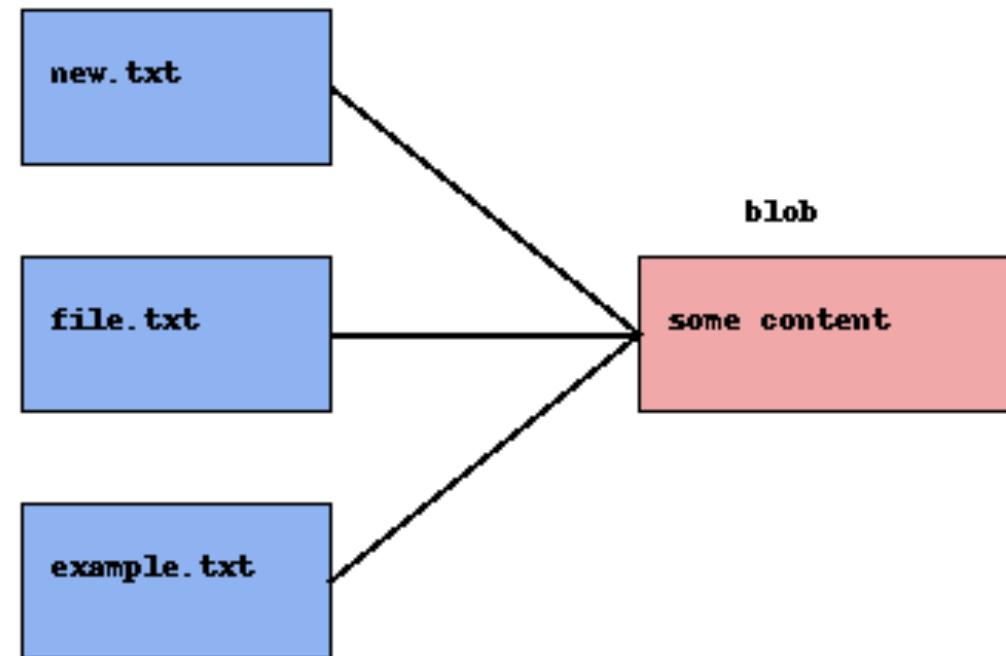


# How Blob objects work

Blob objects do not store any metadata, such as file names or directory paths.

They are identified by their SHA-1 hash, which ensures that each unique file content is stored only once.

If two files have the same content, they will share the same blob object, even if their filenames or locations differ.



# Anatomy of a Blob object

A blob object contains only the raw data of the file, with no metadata.

The size of the file and its contents are stored directly in the blob.

5b1d3..

**blob**

size

```
#ifndef REVISION_H
#define REVISION_H

#include "parse-options.h"

#define SEEN          (1u<<0)
#define UNINTERESTING (1u
#define TREESAME    (1u<<2)
```



# Demo

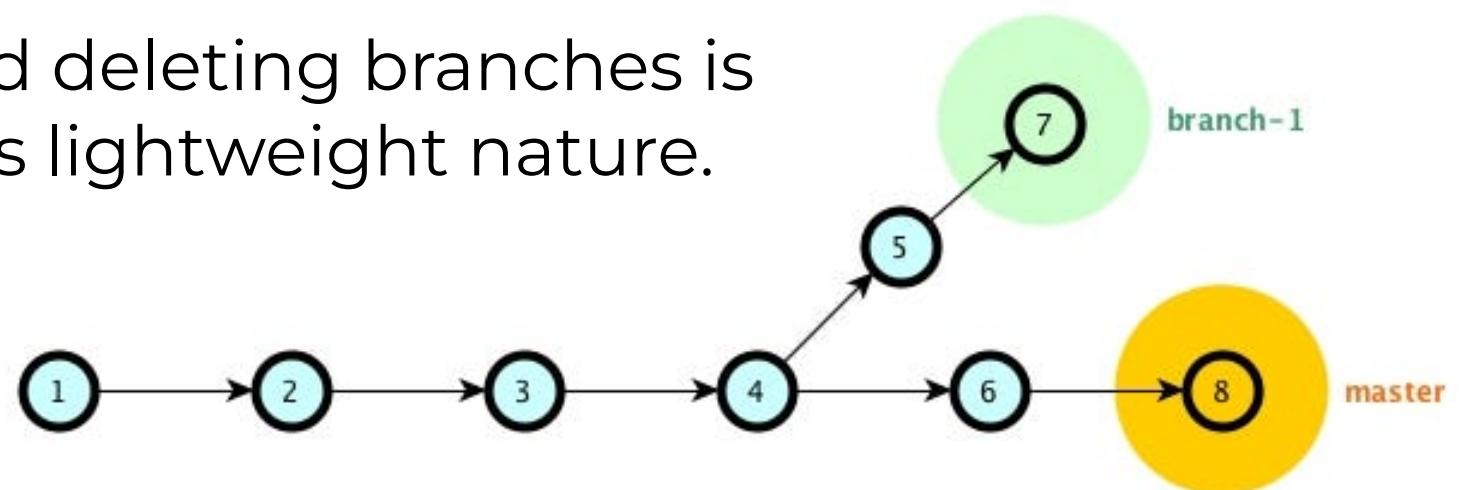
# Branches, Tags, and HEAD

# Understanding Branches

A branch in Git is simply a lightweight pointer to a commit.

Branches allow you to work on different features or bug fixes in isolation without affecting the main codebase.

Creating, switching, and deleting branches is very fast in Git due to its lightweight nature.

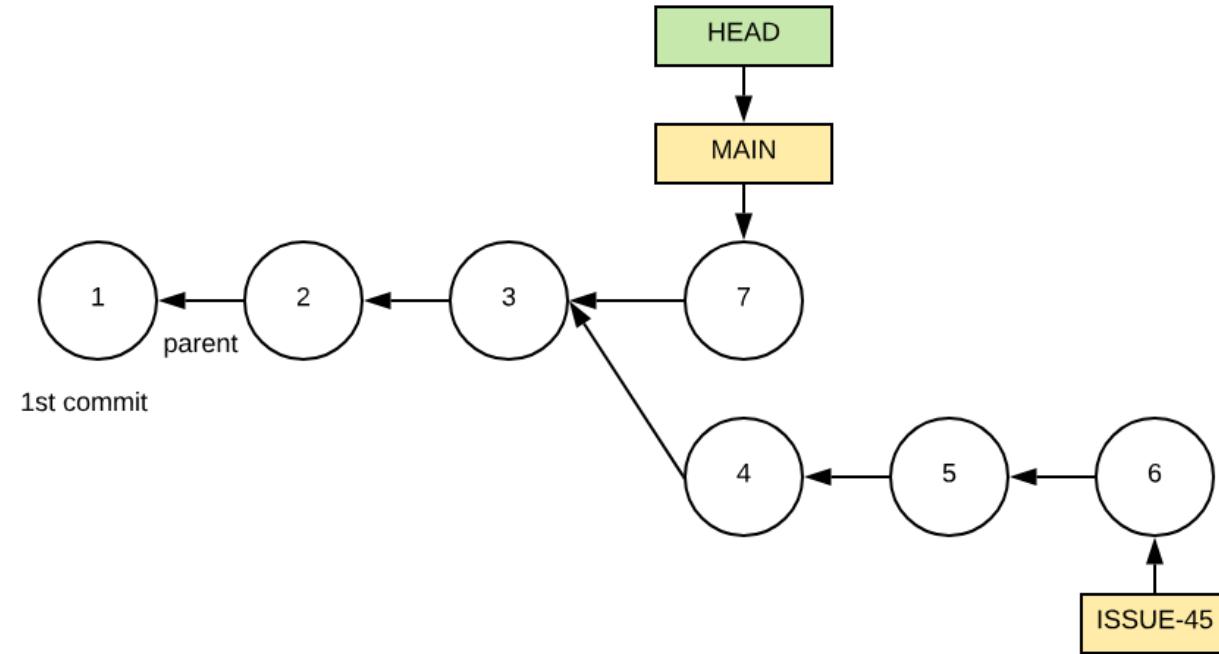


# The role of HEAD

HEAD is a special pointer that refers to the current branch or commit you are working on.

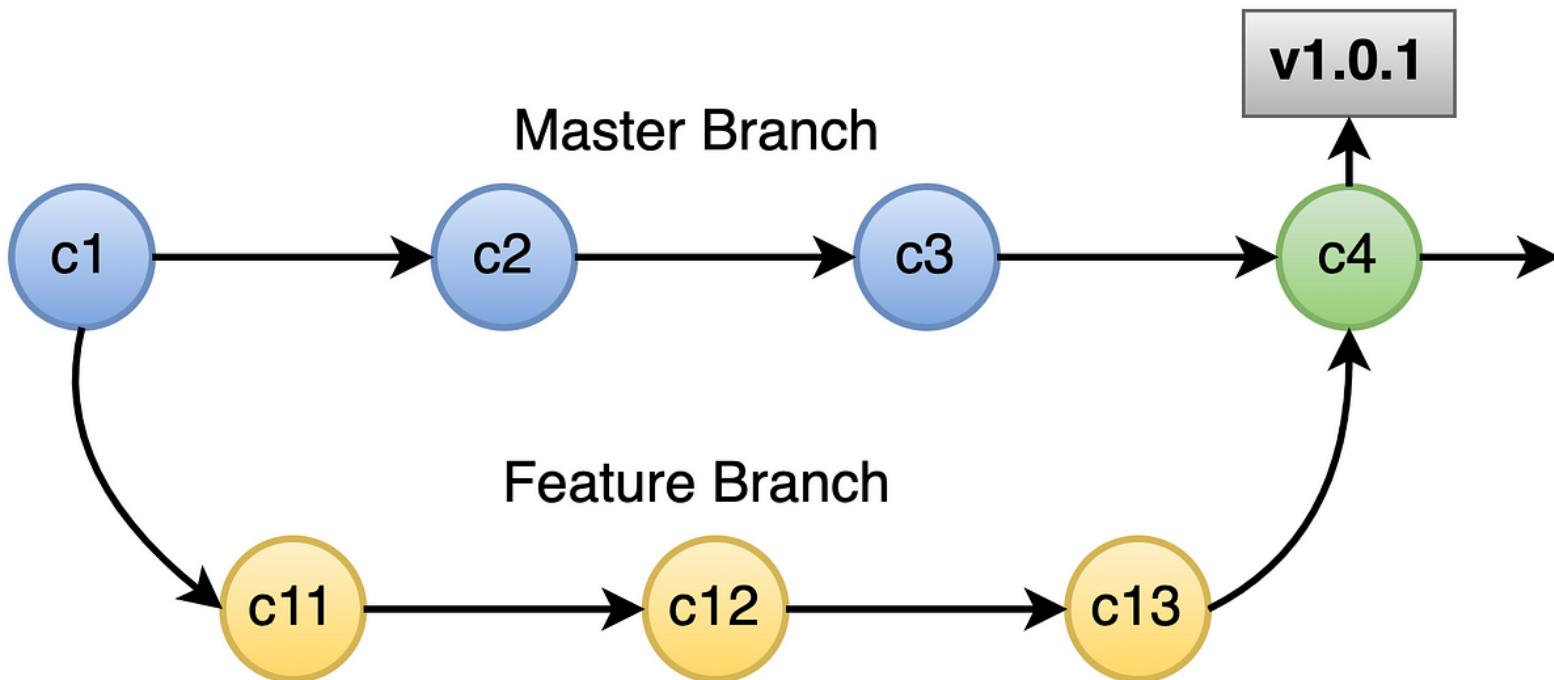
By default, HEAD points to the latest commit on the current branch.

HEAD can also be detached, which allows you to work on a specific commit without being on any branch.



# Tags in Git

Tags are used to mark specific points in the commit history as important, such as version releases.



# Tags in Git

There are two types of tags:

- Lightweight tags are simple pointers to a commit.
- Annotated tags store additional information like a message, tagger name, and date.



# Tags in Git

```
Lenovo@LAPTOP-3ML3SK05 MINGW64 ~/desktop/tagsDemo (master)
$ git show v0.1
commit 15f7bf93b7a71ee50319d479d6cf52493d720e6e (tag: v0.1)
Author: Pankaj <pankaj@gmail.com>
Date:   Mon Jun 7 21:21:38 2021 +0530
```

Initial Commit

```
diff --git a/newFile.txt b/newFile.txt
new file mode 100644
index 000000..5b1dd02
--- /dev/null
+++ b/newFile.txt
@@ -0,0 +1 @@
+Some Text
```

```
Lenovo@LAPTOP-3ML3SK05 MINGW64 ~/desktop/tagsDemo (master)
$ git show v0.2
tag v0.2
Tagger: Pankaj <pankaj@gmail.com>
Date:   Mon Jun 7 21:23:36 2021 +0530
```

This is v0.2 and I am an annotated tag.

```
commit f95854aa621a8eadbc7de9598d9dc029d30ae648 (HEAD -> master, tag: v0.2)
Author: Pankaj <pankaj@gmail.com>
Date:   Mon Jun 7 21:23:28 2021 +0530
```

Second Commit

```
diff --git a/newFile.txt b/newFile.txt
index 5b1dd02..387682e 100644
--- a/newFile.txt
+++ b/newFile.txt
@@ -1 +1,2 @@
 Some Text
+Line Added
```

Commit to which  
the tag point

Difference in the  
files

Tag Metadata

**Light Weight Tag**

**Annotated Tag**



# Demo



**Soumaya Erradi**

soumayaerradi · she/her

Find a repository...

Type ▾

Language ▾

Sort ▾

New

## mastering-git

Public

Updated 3 minutes ago

Star ▾



@soumayaerradi



serradi92@gmail.com



@sumyerradi



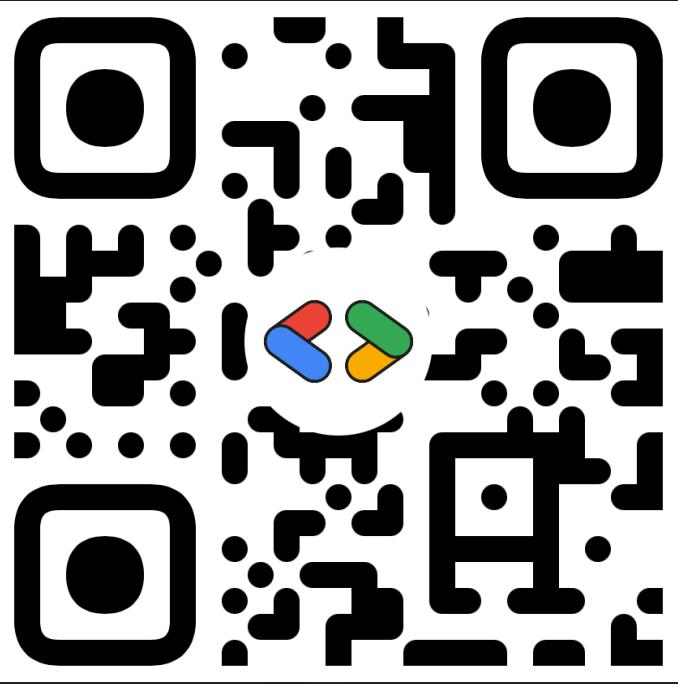
@sumy92



@soumaya-erradi



@soumayaerradi



# Thank you!

Soumaya Erradi

*[www.soumayaerradi.it](http://www.soumayaerradi.it)*