ECAST
Thapathali Campus

SUPPORTED BY: Tensor

WELCOME TO

"my-first-commit"

We'll Be Starting Soon

ECAST
Thapathali Campus

SUPPORTED BY: Tensor

“my-first-commit”

DAY 1
Git and Open Source

Instructor:
Sushovan Shakya (BEI075)

Objectives - Why Are We Here?

- **My First Commit** is two week-long workshop conducted by ECAST, to provide a preliminary concept on several technologies in industry not taught in the college curriculum
- Something of a **Missing Semester** - an extra semester (within a semester) - to provide you with an introduction to some critical skills needed in your field

So What Skills Do We Cover?

- The Command Line
- Git and Version Control
- Linux
- Computer Networks
- Cloud Computing
- Web Development

Prerequisite For Today

- VS Code, Sublime Text, or any equivalent code editor installed
- Github Desktop installed

Prerequisite For Today

- Git installed
  - For Windows and Mac users:
    https://git-scm.com
  - For Linux users:
    - Ubuntu, Debian and derivatives:

      ```
      sudo apt-get update -y && sudo apt-get install git-all
      ```

    - Fedora:

      ```
      sudo dnf install git-all
      ```

    - Arch:

      ```
      sudo pacman -S git
      ```

Part 1:
The Open Source

What is Open Source?

- Open Source is code made available to the public for use, modification and redistribution
- Developed and maintained by a community of developers who work together to improve the software
- Secure and reliable than proprietary software because it is subject to more scrutiny from a larger group of developers

What is Open Source?

- Often free or low-cost, and it can be used for a variety of purposes, including personal, commercial, and educational use.
- The opposite of Open Source is Closed Source or Proprietary - the code is kept away from the public by developers or corporation

“Free as in Free Speech, not as in Free Beer”

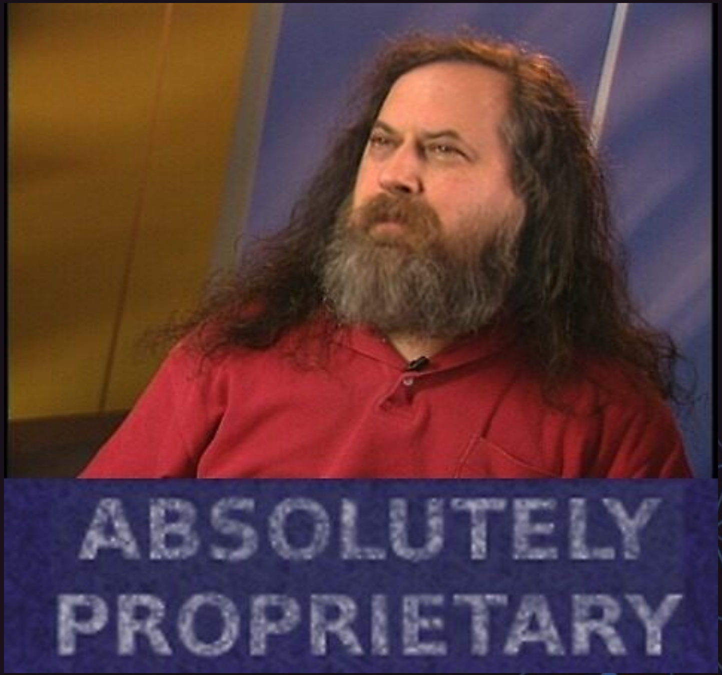
- Open Source is actually about **freedom of code** - hence “Free Speech”
- The developers and contributors are free to view, contribute and modify the code - thus making the project transparent
- Not always that open-source projects are free of cost - some cost money

“Free as in Free Speech, not as in Free Beer”

- In contrast to **proprietary code** - since a corporation or an organization only has access, there’s lack of transparency
- Proprietary software has a huge drawback in that corporations has the right full right to include pieces of code compromising the end user’s integrity and privacy
  - Most common example is Microsoft, Apple and Google taking user data without the user’s consent

Disadvantages of Proprietary Software

- Cost
- Lack of flexibility
- Vendor lock-in
- Security risks
- Less innovation
- Less control



Richard Stallman and the Free Software Movement

- Richard Stallman is a computer programmer, software freedom activist, and the founder of the Free Software Foundation.
- He is best known for his work on the GNU Project, which is an effort to create a free operating system.
- Stallman is a vocal critic of proprietary software, which he believes is a form of digital serfdom.

Richard Stallman and the Free Software Movement

- He argues that free software is essential for freedom and democracy.
- The free software movement has had a significant impact on the software industry, and it has helped to make free software more popular and accessible.

Free Software Movement

- In 1983, Stallman founded the Free Software Foundation (FSF).
- The FSF's mission is to promote the freedom to use, study, copy, modify, and distribute software.
- Stallman is the author of the GNU General Public License (GPL), which is a free software license that protects users' freedom to modify and redistribute software.

FSF
FREE SOFTWARE
FOUNDATION

Free Software Movement

- The GPL is one of the most widely used free software licenses in the world.
- The free software movement has had a significant impact on the software industry.
- It has helped to make free software more popular and accessible.
- It has also helped to raise awareness of the importance of software freedom.

Open Source Licenses

- Type of software license that gives users the freedom to run, study, share, and modify the software.
- Open source licenses are designed to promote collaboration and innovation by allowing anyone to use, modify, and redistribute the software.
- Popular open source licenses:
    - GNU General Public License (GPL)
    - Apache License
    - MIT License

Part 2:
Git and Version Control

What is Version Control?

- Think of a project you’re doing
  - You made a huge change in your code, and the program doesn’t run anymore
  - You know where you made the changes but you can’t figure out what changes you made
  - You need to move back to the previous code, but you can’t, because you don’t have a backup
  - What will you do?

What is Version Control?

- This is where Version Control comes into play
- Think of Version Control as a time machine for your project
- In simple terms: version control is a system that enables you to track changes in your code, documents, or any project, over time

So, why do we need Version Control?

- **Reasons to use Version Control:**
  - Track changes over time
  - Work on multiple projects at the same time
  - Collaborate with other developers
- **Benefits of using Version Control**
  - Improved Code Quality
  - Increased Productivity
  - Reduced Risk of Data Loss

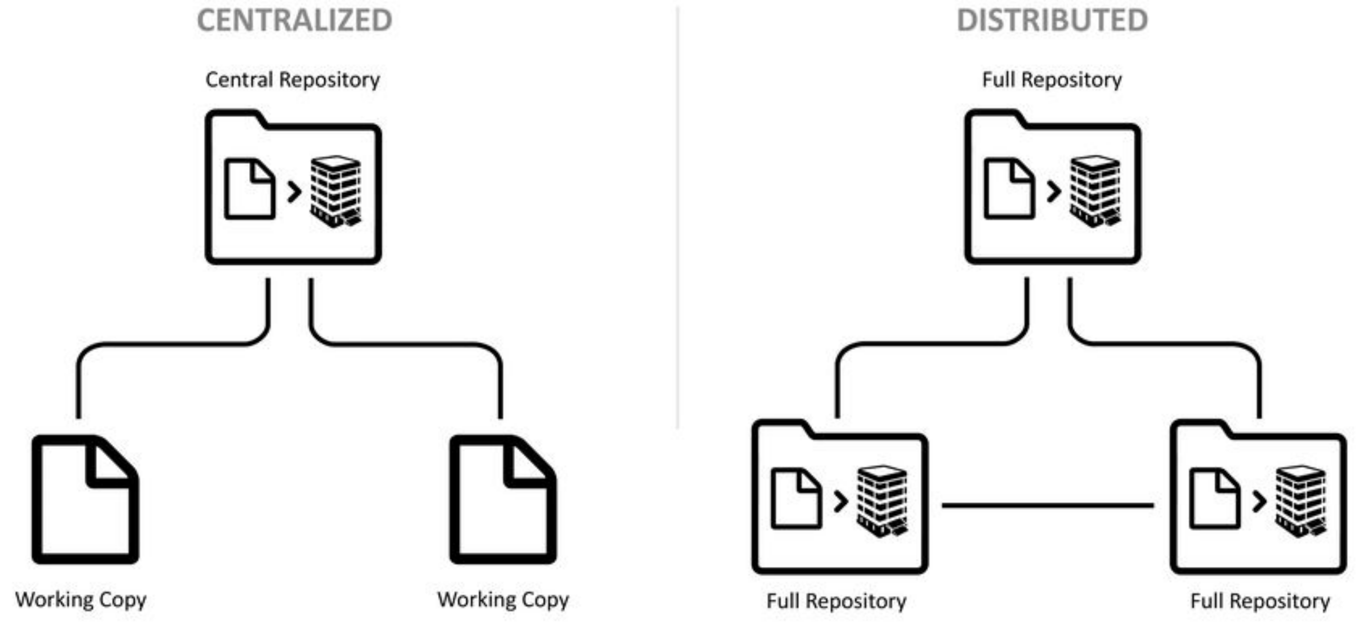
Centralized and Decentralized Version Control

- **Centralized**
  - All data is stored on a central server.
  - All users must connect to the central server to access data.
  - A single point of failure.
  - Can be more difficult to scale.

Centralized and Decentralized Version Control

- **Decentralized**
  - Each user has a local copy of the entire codebase.
  - Users can work offline and sync changes when they connect to the network.
  - No single point of failure.
  - Easier to scale.

Centralized and Decentralized Version Control



What is Git?

- Git is a distributed version control system that allows you to track changes in your code over time.
- It is a free and open-source software tool.
- Git was developed by Linus Torvalds, the creator of the Linux operating system.

What is Git?

- Torvalds started developing Git in 2005 after the company that developed BitKeeper, a proprietary version control system that was being used for Linux development, revoked its free license.
- Torvalds wanted a distributed version control system that would allow developers to work on the same project without having to all be connected to the same server.

What is Github?

- GitHub is a web-based hosting service for version control using Git.
- Github hosts your Git repository, which means that your project are backed up remotely in Github
- Github enables multiple developers to contribute to an open source project and collaborate

What is Github?

- Developers can create Wikis, which are websites used to document projects
- Github Marketplace - you can find third-party tools and services
- github.io domain - provides users with domain to host static sites
- Github Student Pack - additional benefits given by Github for students

Github Student Developer Pack

- A service given by Github to students
- To apply, one must have enrolled in a college and university, and must have a verifiable email address issued by the institution
- You can apply for Github Student Developer Pack through the Github Website
- Great platform for students to use tools and learn latest technologies without spending much of their own money

Benefits of Github Student Developer Pack

- **Unlimited private repositories:** You can create as many private repositories as you need to store your code and projects.
- **$100 in credit for Digital Ocean:** This credit can be used to purchase hosting, storage, and other services from Digital Ocean.
- **6 months of free access to Coursera:** This subscription gives you access to a wide range of online courses, including courses on software development, data science, and machine learning.

Benefits of Github Student Developer Pack

- **75% off a Pluralsight subscription:** This subscription gives you access to a library of video courses on software development, IT, and business.
- **Free access to GitHub Education:** This program provides resources and support for students who are learning to code.

Some Github Alternatives

GitLab

ATLASSIAN
Bitbucket

launchpad

Now, let's get into the fun stuff….

Basic Git Commands

- Setting up a Git Repository
  - git init
- Basic Git operations
  - git add
  - git rm
  - git mv
  - git commit
  - git push
  - git remote add origin

“my-first-commit”

Identifying a Git Repository

- Every git repository must have a .git folder
- .git folder is automatically created when **git init** command is run
- .git contains all information required for version control
- 4 sub-directories:
    - hooks/
    - info/
    - objects/
    - refs/

Identifying a Git Repository

- 4 files
  - HEAD
  - config
  - description
  - index

.gitignore file

- If you want to exclude certain files from your git repository, include the files in the .gitignore file
- .gitignore is a simple text file with no extensions, where you simply put the filename or the directory
- The files or directory mentioned in .gitignore will not be tracked by git

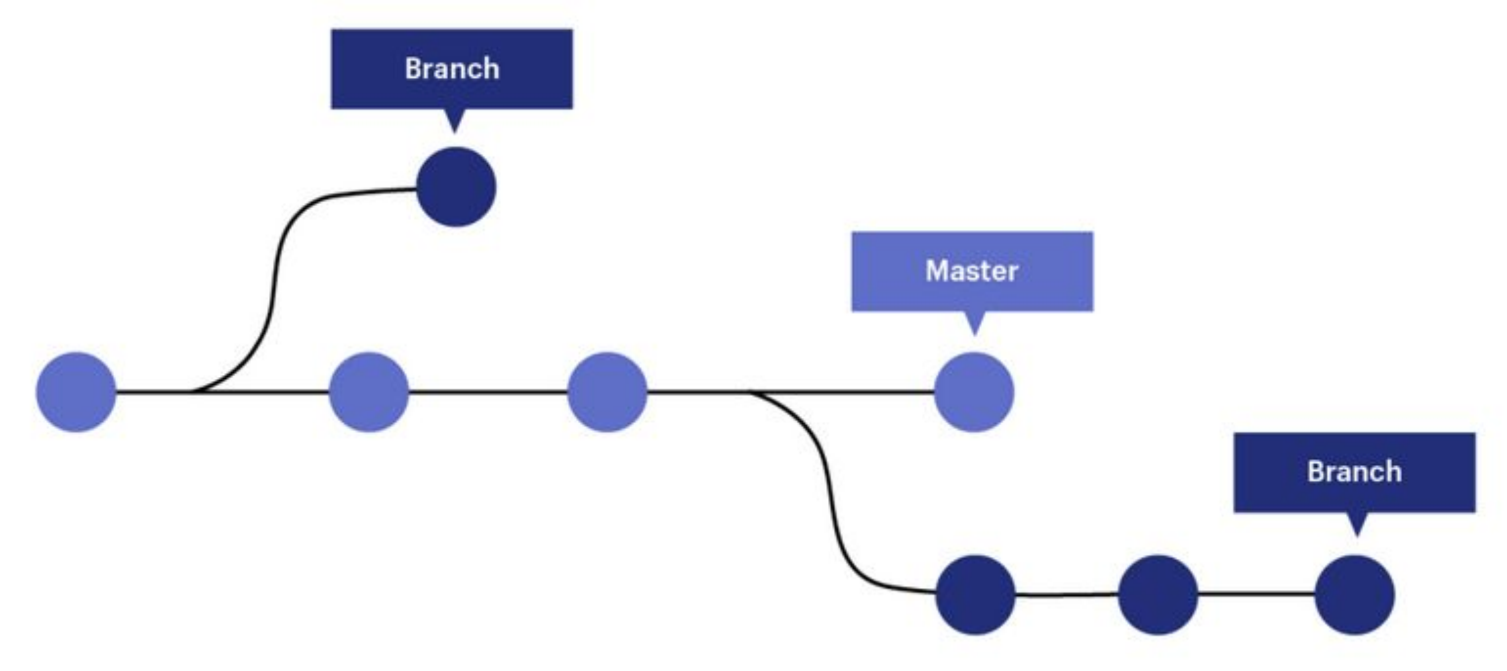
Git Branching

- Git consists of different branches, where different timelines of the project exist
- By default, git uses the **main** branch
- Branching has many different uses
- You can branch the code to fix bugs in your project
- Branching can be used to test changes
- Best for working on few features without affecting the main production line of the code

ECAST
Thapathali Campus

SUPPORTED BY: Tensor

Git Branching



“my-first-commit”

Git Branching (continued)

- Commands involved
  - git branch
  - git checkout

Git Merge

- Merging is the opposite operation of branching
- Merge is used when any two or more branches in the project development needs to be merged
- Commands involved:
  - git merge

Git Log

- Basic but quintessential command
- Basically, it displays an entire log of your commits - you’re literally looking at the entire history of the project
- Extremely useful in tracking changes and finding major changes
- Useful in git restore - moving behind the branch

Git Restore

- Restore is simply Git’s version of past time travel
- If anything went wrong in your project - don’t worry, git restore got your back
- Commits are simply the checkpoints in your project’s timeline - and restore enables you to travel back in time to those checkpoints

Where do we go next?

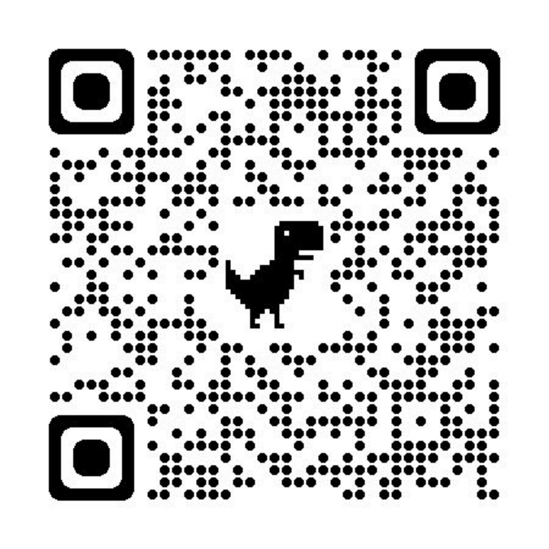
To Learn More About Git and Version Control

- Atlassian has a great Git documentation

  https://www.atlassian.com/git/tutorials

- You learn by doing - best way to learn Git is by using it in your projects

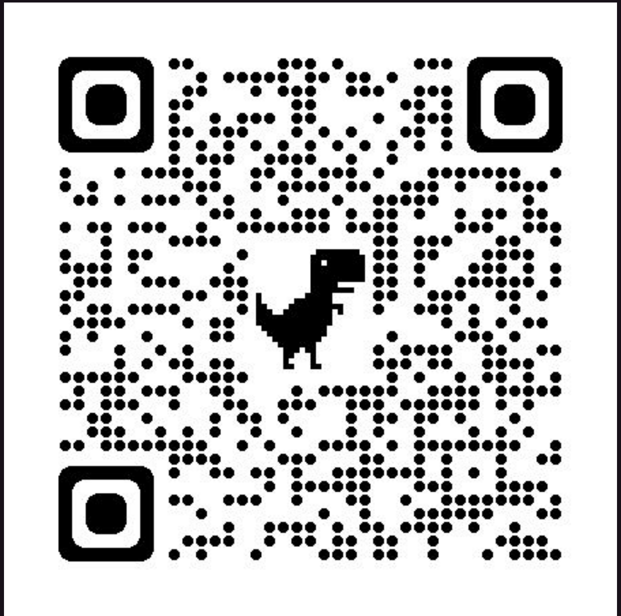
Join our Discord Server to ask your queries



ECAST
Thapathali Campus

For Attendance

SUPPORTED BY: Tensor
BEYOND DIMENSIONS

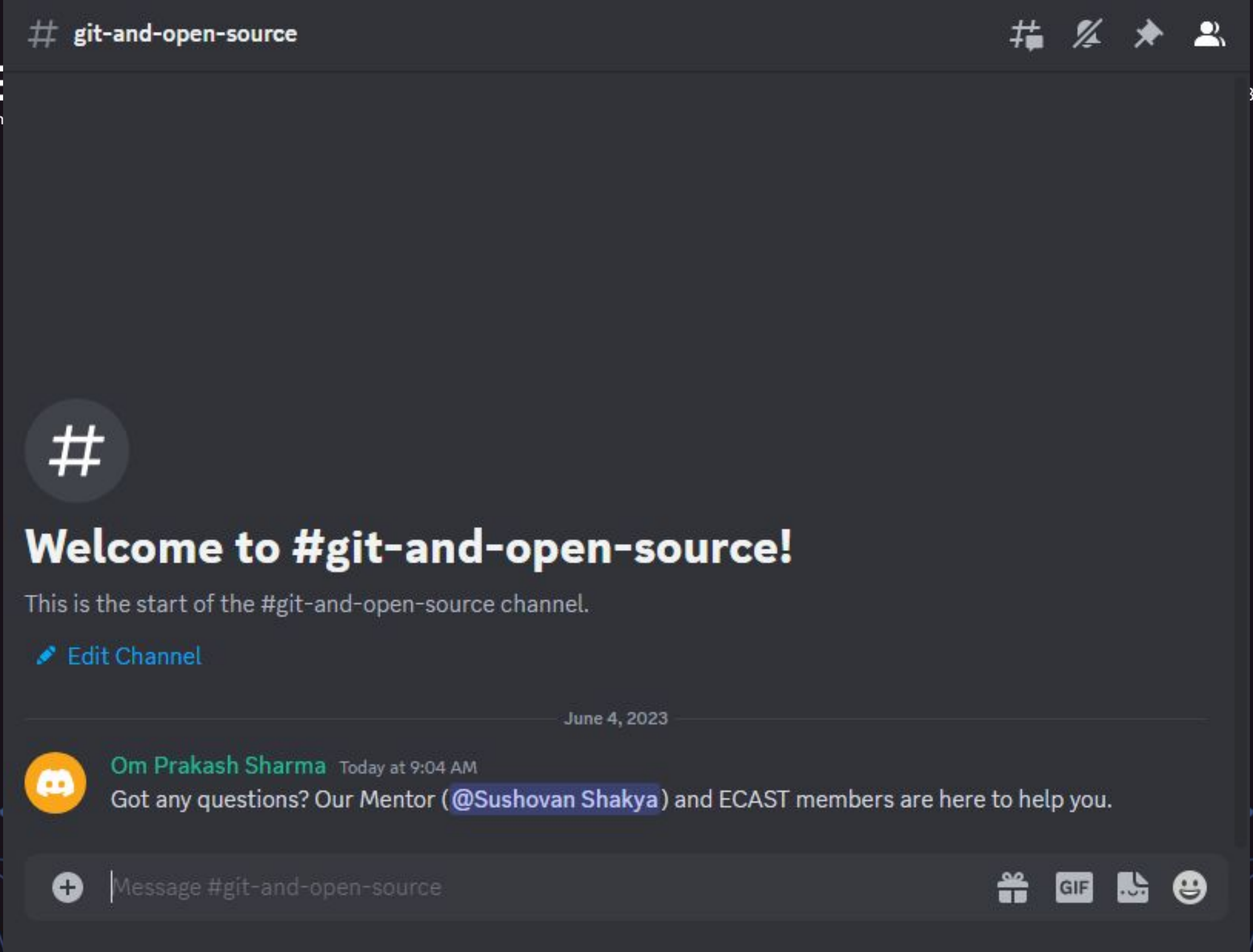


“*my-first-commit*”

E

Th

BY:



Welcome to #git-and-open-source!

This is the start of the #git-and-open-source channel.

June 4, 2023

Om Prakash Sharma Today at 9:04 AM

Got any questions? Our Mentor (@Sushovan Shakya) and ECAST members are here to help you.

ECAST
Thapathali Campus

SUPPORTED BY: Tensor

Thank You

“my-first-commit”