

ORGANIZED BY
IEEE CS CUI

GIT & GITHUB

conducted by
Haris Ayyaz Khan

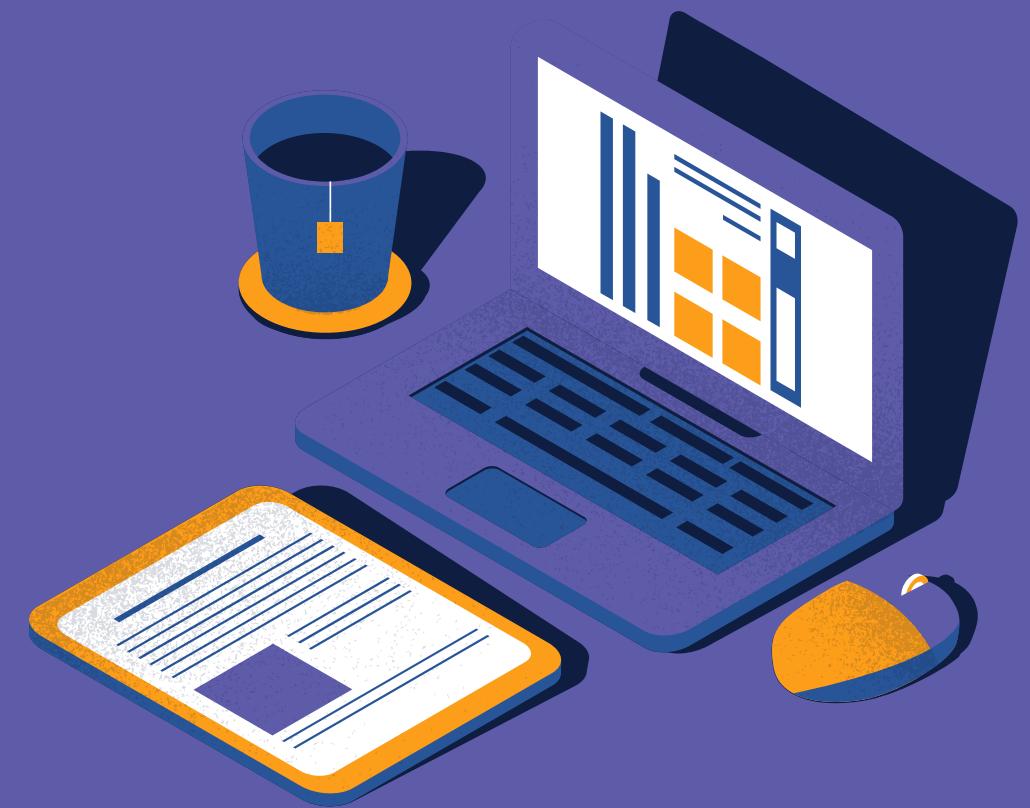


**OPEN
SOURCE**

HARIS AYYAZ

Former Chairperson IEEE CS CUI
Winner at FYP Expo (Web Dev Category)
Front end Developer Hayaland Inc.

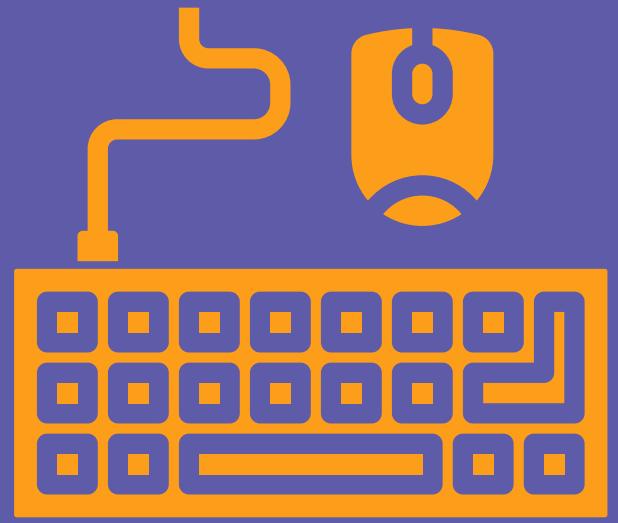




WORKSHOP OVERVIEW

- Understanding Version Control
- Introduction to Git & GitHub
- Hands-on Practice with Git Commands
- Q&A Session





WHAT IS VERSION CONTROL

- Like a “save game” feature for your code or documents.
- You can go back to any previous version anytime
- You can see who changed what and when.





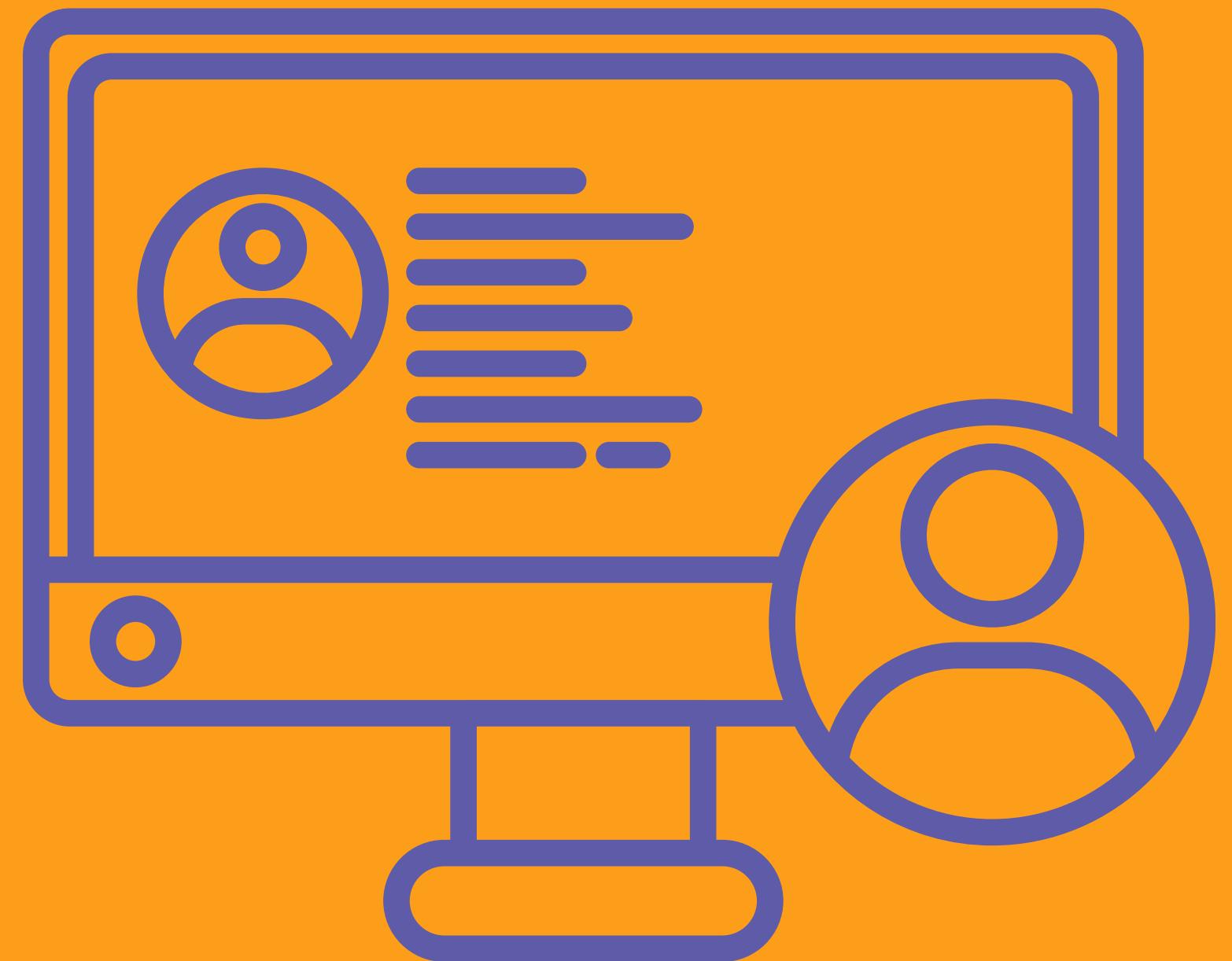
WHAT IS GIT?

- Git = A tool to save versions of your project.
- You can make “snapshots” (called commits).
- You can go back in time if something breaks.
- It works offline, on your computer.



WHY USE GIT

- Prevents losing progress during development
- Enables multiple people to work on a project simultaneously
- Keeps history of all changes
- Facilitates collaboration and teamwork





WHAT IS GITHUB?

- GitHub = a website where you can keep your Git projects online.
- You can share your projects with others.
- Work together without sending files on WhatsApp 😊.
- Employers and open-source projects live there.
- Git = saves your work | GitHub = shows it to the world.





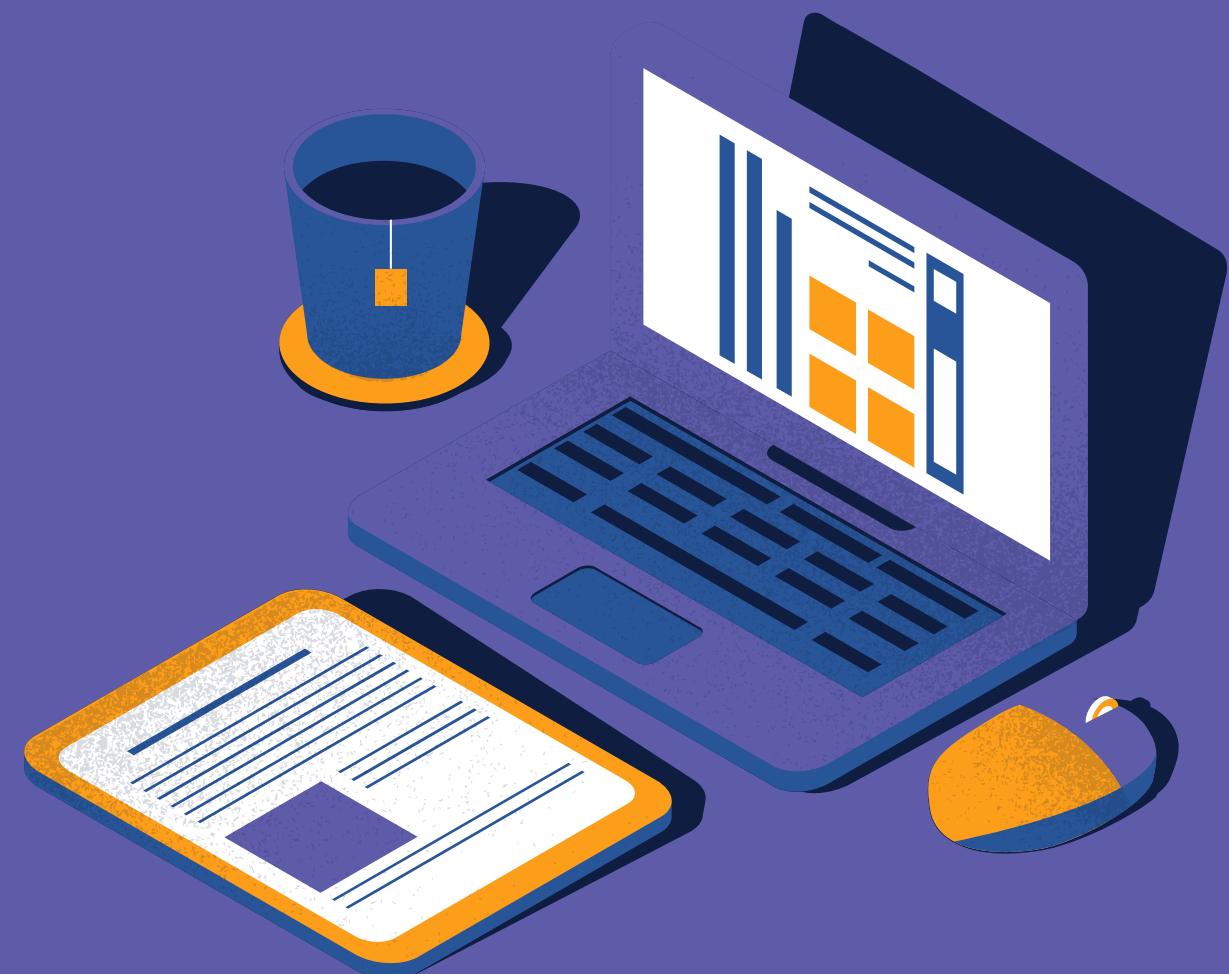
SETTING UP GIT

- Download and install Git from git-scm.com
- Set username and email using terminal commands:
 - `git config --global user.name "Your Name"`
 - `git config --global user.email "your@email.com"`
- Check configuration: `git config --list`



BASIC GIT WORKFLOW

1. Initialize a repository → git init
2. Add files to staging → git add filename
3. Commit changes → git commit -m "message"
4. Check status → git status
5. View history → git log



CONNECTING TO GITHUB

- Create a GitHub account
- Create a new repository on GitHub
- Link local repo to GitHub using:
 - `git remote add origin <repo-url>`
- Push changes online: `git push -u origin main`





CLONING & PULLING

- Clone existing repo: `git clone <repo-url>`
- Fetch updates: `git pull origin main`
- Keep your project synced with collaborators





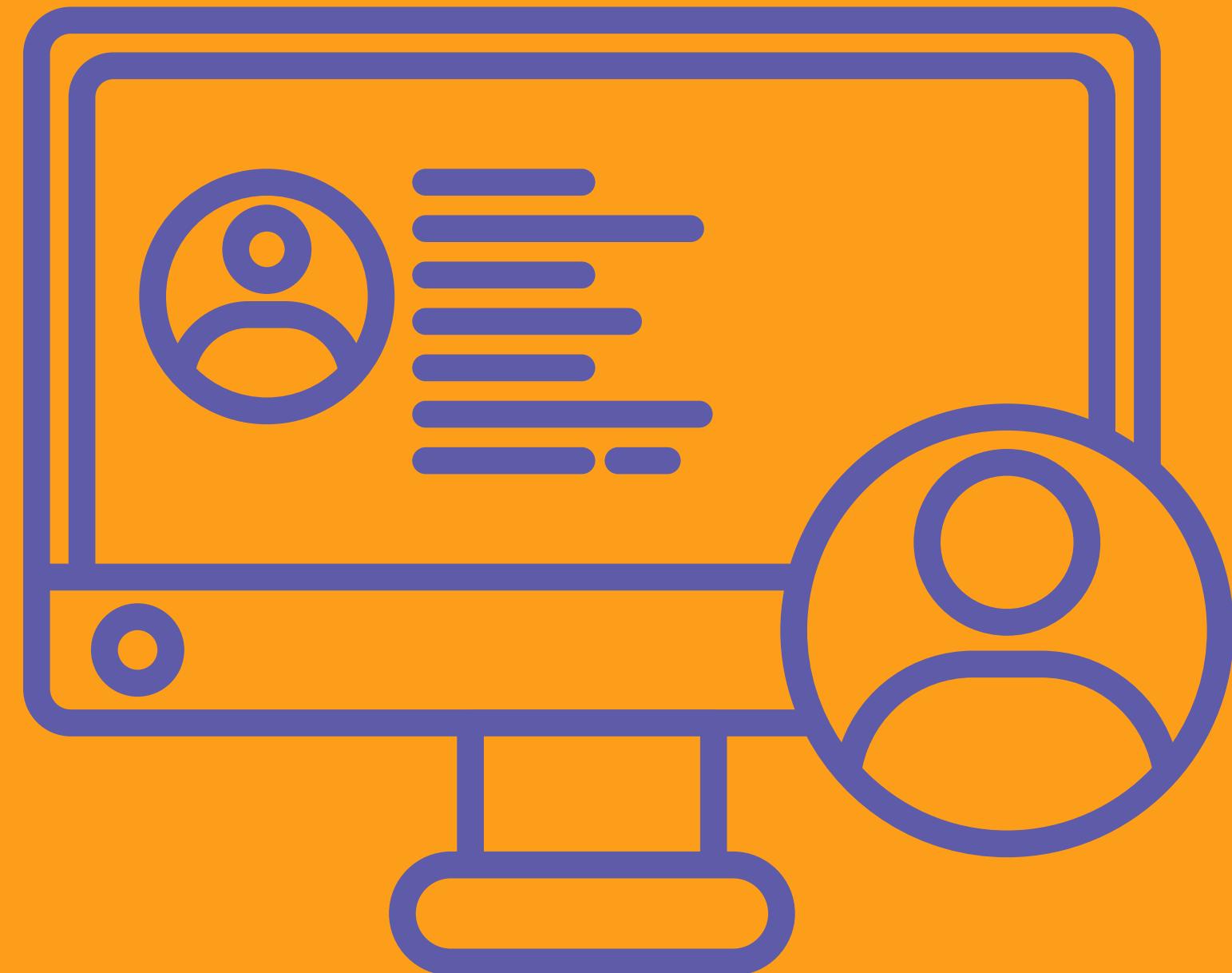
WORKING WITH BRANCHES

- Create a branch: `git branch feature-1`
- Switch branches: `git checkout feature-1`
- Merge branch to main: `git merge feature-1`
- Delete a branch: `git branch -d feature-1`



COMMON GIT COMMANDS

- `git status` — Check file status
- `git diff` — Show changes made
- `git reset` — Unstage or undo changes
- `git rm` — Remove a file from repo
- `git log` — View commit history

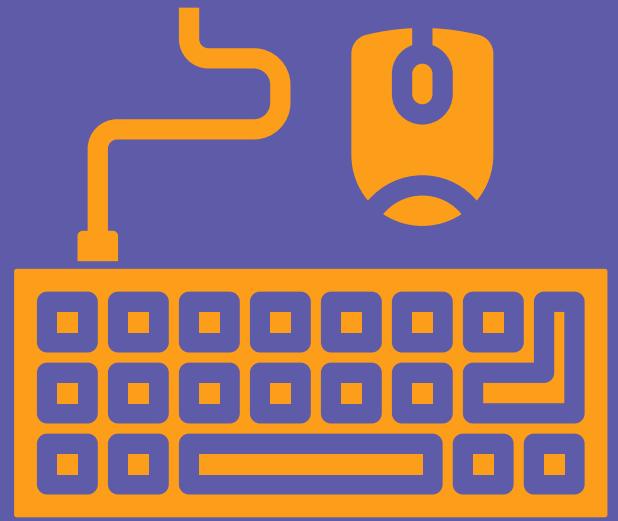


BEST PRACTICES



- Commit often with meaningful messages
- Pull before you push
- Use branches for new features
- Avoid committing unnecessary files
- Collaborate using pull requests

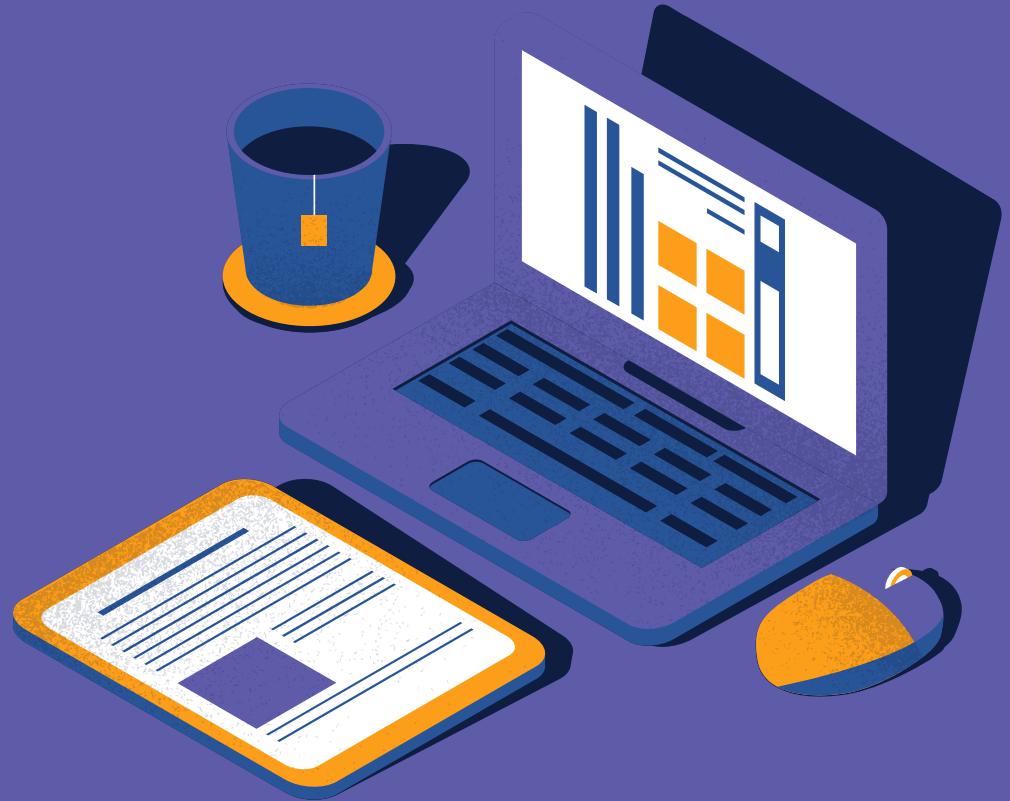




RESOURCES TO LEARN MORE

- Git Documentation: git-scm.com/docs
- GitHub Guides: docs.github.com
- Interactive Learning:
learngitbranching.js.org
- YouTube Channels: Traversy Media,
FreeCodeCamp





- Live demonstration of Git commands
- Students will create and push their first repo
- Interactive question and answer session

Q&A AND PRACTICE