
🕒 0:00 – 2:00 | Intro & Structure

Slide: Welcome & Agenda

- "Hi everyone! Today is about demystifying Git. We're going to spend the first 15 minutes understanding the 'Why' and 'What,' then we'll jump into a 30-minute workshop where you'll actually get your hands dirty."
- **The Schedule:**
  - 15m: Presentation
  - 5m: Quick Q&A
  - 30m: Hands-on Workshop
  - 10m: Wrap up & Feedback

🕒 2:00 – 4:00 | What is Git? (The 5-Year-Old Version)

Slide: The Magical Save Button

- "Think of Git as a **Time Machine** for your work. Normally, when you save a file, you overwrite what was there before. If you make a mistake, it's gone."
- "Git is different. It's like a 'Save' button that remembers every single version of your project since the day you started. If you break something today, you can literally travel back to yesterday and start over."
- "It's been around since 2005. It's 21 years old and is used by almost every professional developer in the world. It's tried, tested, and it works."

🕒 4:00 – 8:00 | The Problems We're Solving

Slide: Working Together Without Chaos

- "Without Git, if two people edit the same file, the person who saves last 'wins' and the other person's work is deleted. Git stops that."
- **What it gives us:**
  - **History:** We can see who changed what and why (no more 'who deleted my code?').
  - **Feature Branches:** You can create a 'copy' of the project to experiment in. If it works, we keep it. If it doesn't, we delete the branch and the main project stays perfect.
  - **Code Review:** Before your work goes 'Live,' a teammate can look at it, leave comments, and suggest improvements. It's like a second pair of eyes for quality control.
  - **Clean Production:** Our 'Live' code stays clean while we do all the messy work in the background.

🕒 8:00 – 11:00 | How it Works (The Graph)

Slide: Commits, Branches, and Merges

- "Look at this graph. Think of it like a subway map."
  - **The Main Line:** This is your 'Live' product. It should always work.
  - **The Commit (The Dots):** Each dot is a 'Snapshot.' It's a moment in time we've saved.
  - **The Branch:** This is when you 'turn off' the main track to build something new (like a 'Payment' feature).
  - **The Merge:** This is the bridge. Once the 'Payment' feature is tested, we merge it back into the Main Line.

🕒 11:00 – 13:00 | Git vs. GitLab

Slide: The Engine vs. The Garage

- "People often use these words interchangeably, but they are different."
- **Git:** This is the tool. It's the engine under the hood that tracks the changes.
- **GitLab:** This is the website (the Garage). It's where our code lives online so we can share it, talk about it in comments, and manage our projects.
- *Note: Just mention that GitLab can also do fancy things like 'GitOps'—which automates putting your code onto servers—but we'll save that for the advanced class!*

🕒 13:00 – 15:00 | Opening a PR (The Merge Request)

Slide: The Handshake

- "The most important part of our workflow is the **Merge Request (MR)** or **Pull Request (PR)**."
- "This is basically you saying: *'Hey team, I've finished my feature on my branch. Here is what I changed. Can someone check it before I merge it into the Main code?'*"
- "In the workshop, we're going to practice opening one of these together."

💡 Quick Tips for the Speaker:

- **Don't use Terminal yet:** If they are beginners, keep everything in the **GitLab Web UI** during the presentation to keep it less scary.
- **Empathize:** Remind them that *everyone* finds Git confusing for the first few days. It's normal!

Would you like me to create a "Cheat Sheet" PDF layout that you can hand out to them after the presentation?