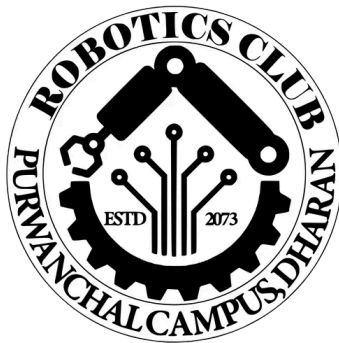


Get this  
slide and  
its source  
code ➡



# Importance of Version Control and Documentation

## TECHNOMORPH: INTRODUCTION TO ROBOTS AND ROBOTICS

Spandan Guragain (scientiac)

 [scientiac.space](http://scientiac.space)

# Why Version Control?

## ▸ The Problem

- `project_final.zip`
- `project_final_final.zip`
- `project_final_v2.zip`
- `project_final_ACTUALLY_FINAL.zip`

**Sound familiar?** Version control solves this chaos!

# What is Version Control?

## ▸ Key Benefits

- **Track changes** over time
- **Collaborate** with others seamlessly
- **Revert** to previous versions
- **Branch** for experiments
- **Merge** contributions safely

Think of it as a **time machine** for your code!

# Enter Git

## ▸ Why Git?

- **Distributed** - Everyone has full history
- **Fast** - Optimized for performance
- **Flexible** - Supports any workflow
- **Industry Standard** - Used everywhere

```
git init
```

```
git add .
```

```
git commit -m "First commit"
```

# Git Hosting Platforms

## ▸ GitHub

- Most popular for individuals
- Great for open source
- Excellent CI/CD
- Largest git hosting platform

## ▸ GitLab

- If you are feeling extra adventurous
- Most FOSS communities use this
- Free private repos
- Enterprise features

**Others:** Codeberg, SourceForge

# Learning Git

## ▸ Resources to Get Started

- **Interactive:** [learngitbranching.js.org](https://learngitbranching.js.org)
- **Official:** [git-scm.com/doc](https://git-scm.com/doc)
- **GitHub:** GitHub Skills courses
- **Practice:** Create your first repository!

**Pro tip:** Start with basic add, commit, push, pull

# Documentation: Your Future Self Will Thank You

## ▸ Why Document?

- **Remember** what you did and why
- **Share** knowledge with others
- **Professional** portfolio building
- **Learning** reinforcement
- **Debugging** becomes easier

Documentation is **not** just for others—it's for **you**!

# What to Document?

## ▸ Essential Documentation

- **README.md** - Project overview
- **Setup instructions** - How to run
- **API documentation** - How to use
- **Learning notes** - What you discovered
- **Decision logs** - Why you chose X over Y

**Remember:** Future you is a different person!



# Documentation Tools

## ▸ Typst

- Modern markup language
- Beautiful PDFs
- Fast compilation
- Great for reports

## ▸ LaTeX

- Academic standard
- Precise formatting
- Rich ecosystem
- Publication ready

**Others:** Markdown, Obsidian, Notion, Sphinx

# Typst vs LaTeX: Quick Comparison

Feature	Typst	LaTeX
Learning curve	Gentle	Steep
Compilation	Fast	Slow
Syntax	Modern	Verbose
Community	Growing	Massive
Use case	Modern docs	Academic papers

**Recommendation:** Try Typst first, learn LaTeX for academic work

# Best Practices

## ▸ Version Control

- Commit **early** and **often**
- Write **meaningful** commit messages
- Use **branches** for features
- **Pull** before you **push**

# Best Practices

## ▸ Documentation

- Write as you **code**
- Keep it **simple** and **clear**
- Update when you **change** things
- Include **examples**

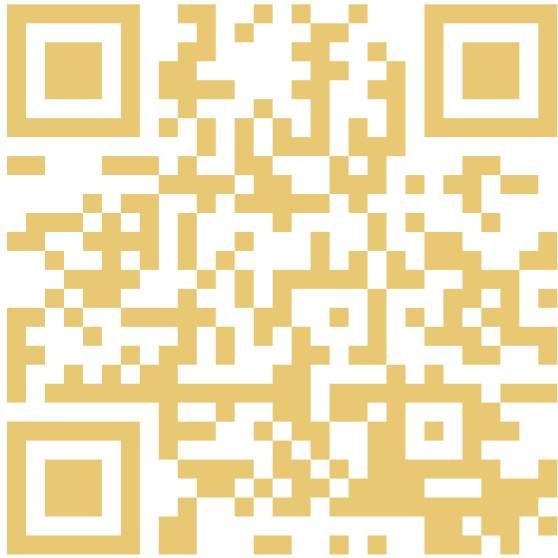
# Getting Started

## ▸ Action Items

1. Create a **GitHub** account
  2. Install **Git** on your machine
  3. Make your **first repository**
  4. Write a **README.md**
- Try **Typst** or **LaTeX** for notes

**Start small, be consistent!**

# See you soon!



↑ Find the templates  
for documenting your  
bootcamp journey here

This presentation is made with typst, you can find the source for it in the github repo linked from this QR.

✉ [spandan@scientiac.space](mailto:spandan@scientiac.space)

🌐 [scientiac.space](http://scientiac.space)

📧 [@iac@polymaths.social](https://polymaths.social/@iac)