# Setting Up Your Python Simulation Environment

# GitHub Desktop + PyCharm Tutorial

Everything you need to clone, run, and contribute to the simulation project

Session 1

Unnati Akhouri, QuNet Bootcamp 2025

# Agenda

- **GitHub Desktop** - Download and installation
- **PyCharm IDE** - Setup for Python development
- **Repository Cloning** - Getting the simulation code
- **Git Basics** - Push and Pull requests explained
- **Workflow Demo** - Putting it all together

**Goal:** By the end, you'll be able to download, run, and contribute to the Python simulation project!

# Why These Tools?

## GitHub Desktop

- Visual interface for Git
- No command line needed
- Easy repository management
- Seamless collaboration

## PyCharm

- Powerful Python IDE
- Built-in debugging tools
- Code completion & suggestions
- Integrated version control

**Perfect Combo:** GitHub Desktop handles version control, PyCharm handles coding - both with user-friendly interfaces!

# Github/PyCharm download

## Step 1: Download

- Go to desktop.github.com and https://www.jetbrains.com/pycharm/download/?section=mac
- Click "Download for Windows/Mac"
- File size: ~100MB

## Step 2: Install & Setup

- Run the installer (admin rights may be needed)
- Sign in with your GitHub account
- Configure your name and email
- Choose your preferred editor (we'll use PyCharm)

# PyCharm tips

## Step 1: Choose Your Version

- **Community Edition** - Free, perfect for our needs
- **Professional** - Paid, extra features for web development
- Download from jetbrains.com/pycharm

## Step 2: Installation

- Run installer (1GB+ download)
- Create desktop shortcut
- Associate .py files with PyCharm
- Add to PATH (recommended)

**Pro Tip:** Students can get PyCharm Professional free with GitHub Student Pack!

# Config

Initial Configuration

- **Theme:** Choose Darcula (dark) or Light theme
- **Keymap:** Default or match your preferred editor
- **Python Interpreter:** PyCharm will detect automatically
- **Plugins:** Enable Git integration (usually pre-enabled)

Essential Settings

- File → Settings → Version Control → Git
- Verify Git executable path is detected
- Enable "Use credential helper"

# Cloning: Downloading a complete project folder with its entire history

What you get:

- All source code files
- Complete version history
- Branch information
- README and documentation

What you can do:

- Run the code locally
- Make modifications
- Create new features
- Submit improvements

**Key Point:** Cloning creates a local copy that stays connected to the original repository!

# Cloning

Method 1: From GitHub.com

- Navigate to the simulation repository
- Click green "Code" button
- Select "Open with GitHub Desktop"
- Choose local folder location

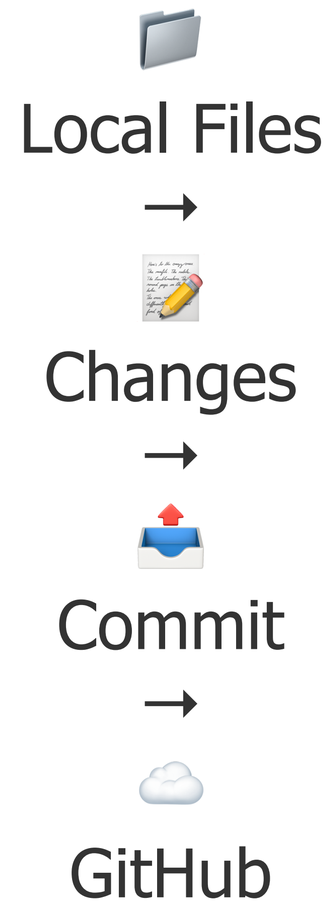
Method 2: From GitHub Desktop

- File → Clone Repository
- Enter repository URL or search by name
- Choose local path (avoid spaces in folder names)
- Click "Clone"

# Github workflow

Key Concepts

- **Commit:** Save a snapshot of your changes
- **Push:** Upload your commits to GitHub
- **Pull:** Download latest changes from GitHub
- **Branch:** Work on features separately



# Push and Pull

Push

- **Uploads** your local commits
- Shares your work with others
- Updates the online repository
- Click "Push origin" in GitHub Desktop

**When:** After you commit changes locally

Pull

- **Downloads** latest changes
- Gets others' contributions
- Updates your local repository
- Click "Pull origin" in GitHub Desktop

**When:** Before starting new work

# Pull requests

A formal way to propose changes to the main project

Why Use PRs?

- Code review process
- Discussion and feedback
- Quality control
- Documentation of changes

PR Workflow

- Create feature branch
- Make and commit changes
- Push branch to GitHub
- Open Pull Request
- Review and merge

# Common Issues & Solutions

## Python Not Found

**Solution:** Install Python from python.org, restart PyCharm

## Module Import Errors

**Solution:** Install requirements with pip install -r requirements.txt

## Merge Conflicts

**Solution:** Pull latest changes first, resolve conflicts in PyCharm

## Can't Push Changes

**Solution:** Check if you're on the right branch, ensure you're signed in to GitHub