



Get to know your neighbors!

Meet the people
around you!

- Name
- School
- One Course you're teaching
- Field of Study
- Familiarity with using GitHub.

Agenda

Morning

	What?	How?	How Long?
	Intro/Agenda/Who's who	Lecture	15 minutes
	What is Version Control/Git/GitHub? GitHub Tour!	Lecture	40 minutes
	Profile Readme	Activity(solo)	45 minutes
	Break		15 minutes
	Collaboration in GitHub	Activity(team)	45 minutes

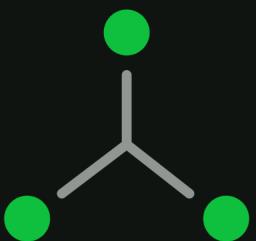
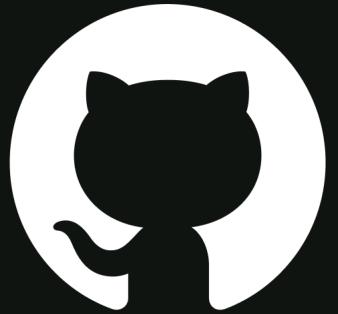


Agenda

Afternoon

What?	How?	How Long?
GitHub in your Curriculum	Lecture	30 minutes
Copilot: A Primer	Lecture	45 minutes
Parking Lot Questions	Q&A	30 minutes
Break		15 minutes
Time to play/experiment/work/ ask questions of our experts!	Activity(solo)	60 minutes



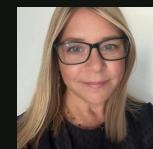


SIGCSE 2026

GitHub for Teachers



PJ METZ
Developer Advocate
He/him



PHOEBE QUINCY
Senior Program Manager
She/her



AVNI KHATRI
Senior Director Education



JACQUELINE RUSSELL
Product Manager
She/her



CHRISTOPHER HARRISON
Senior Developer Advocate
He/him

Attendees will be able to...

CREATE

- Issues
- Pull Requests
- Repositories
- Commits
- A Profile Readme
- A New Branch
- A Discussion

USE

- Web Editor
- GitHub Web
- Basic Copilot

DESCRIBE

- Version Control
- Git
- GitHub
- Git Workflow



"We are currently preparing students for jobs that don't yet exist, using technologies that haven't been invented, in order to solve problems we don't even know are problems yet,"

Richard Riley (former Sec. of Education) said this in:

2004

We've known that we can't rely on technology to sit still in education since 2004 and well before 2004. We knew this time would come. So how do we adapt?

We learn about the newest tech and help students learn to use them correctly.

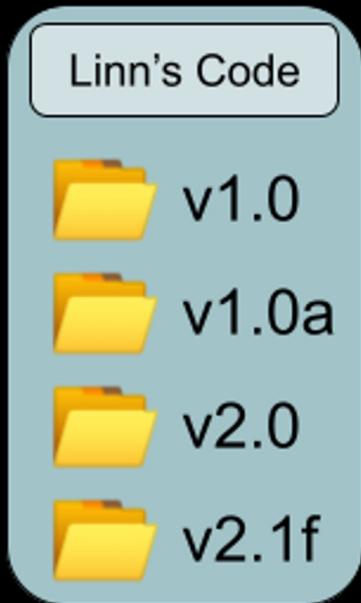
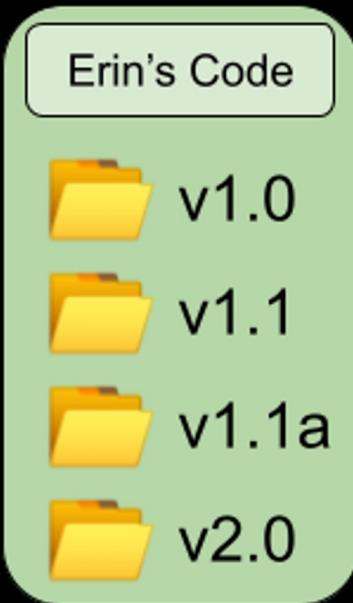
What is Version control?

Version control keeps track of changes made to code, or the different versions of code.



FinalEdit_v2_USETHISONE_Repaired_FixedV4_WITH-GFX_v6
Document - 17.6 MB

What is Git?



Git is version control software created by Linus Torvalds, the inventor of Linux. Made in 2005.

Git is software that you download and control through the command line. It tracks changes to files in specified locations and allows you to keep track of who made the change and when.

Requires setting up a remote server and a knowledge of git commands for the terminal.

Git keeps track of changes using “commits,” user controlled “snapshots” of the current state of the code.

A user makes changes to code that is being watched by Git, creates a commit that adds those changes to the repository, and now their code is ready to share with others on the project.

Quick Vocab



Repository (local or remote)

Stores files for a project as well as all the necessary tracking of changes made to that project.
“Repo” for short.

Branch

A special copy of the codebase that is separate from the version of the code deployed to users.

Commit

Adds the changes a developer made to a specific branch so everyone can see the changes.

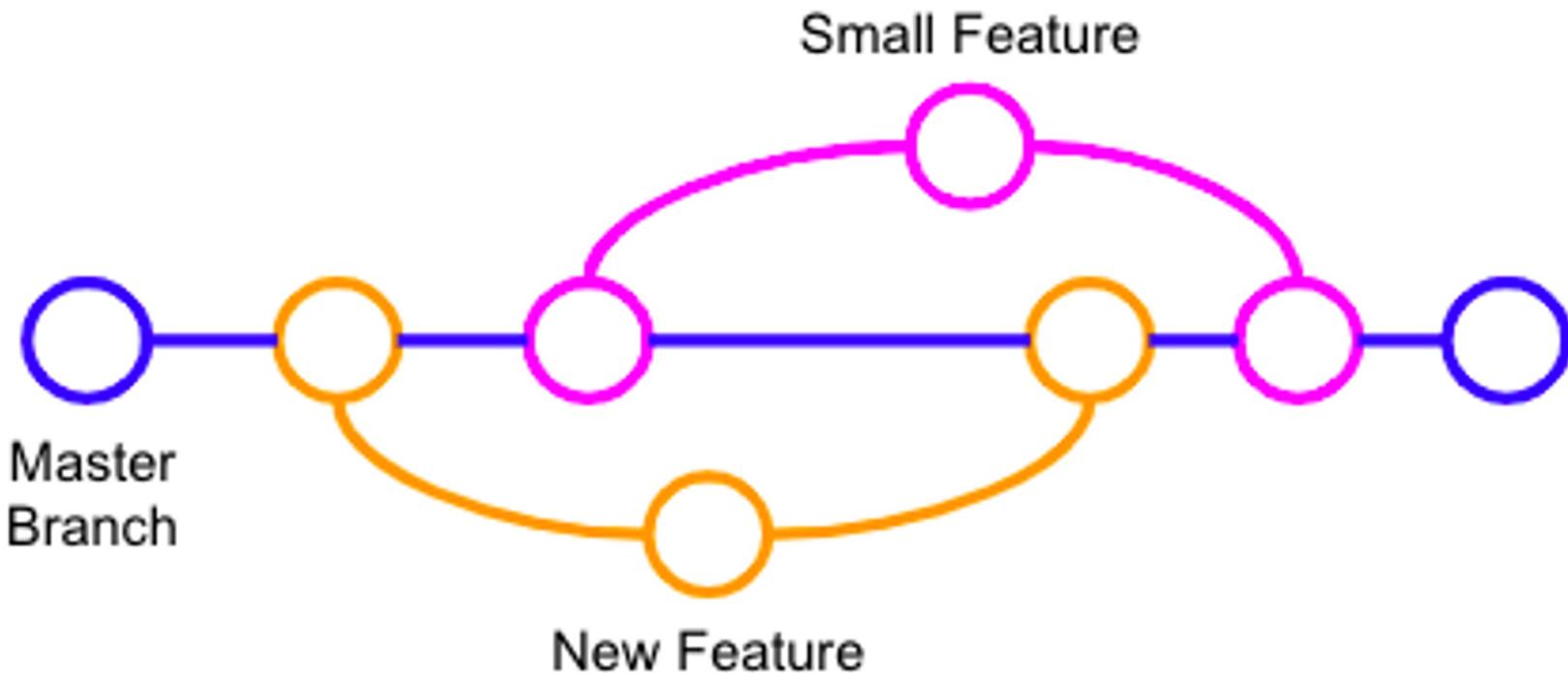


Git: Distributed Version Control

With Git, each developer has the whole codebase on their local machine (local repository) and there's a copy on a server (Remote repository).

To work on the codebase means first updating your local machine with the changes your team made without you (**PULL**), and then updating the server so other collaborators can see your changes (**PUSH**).

Branching



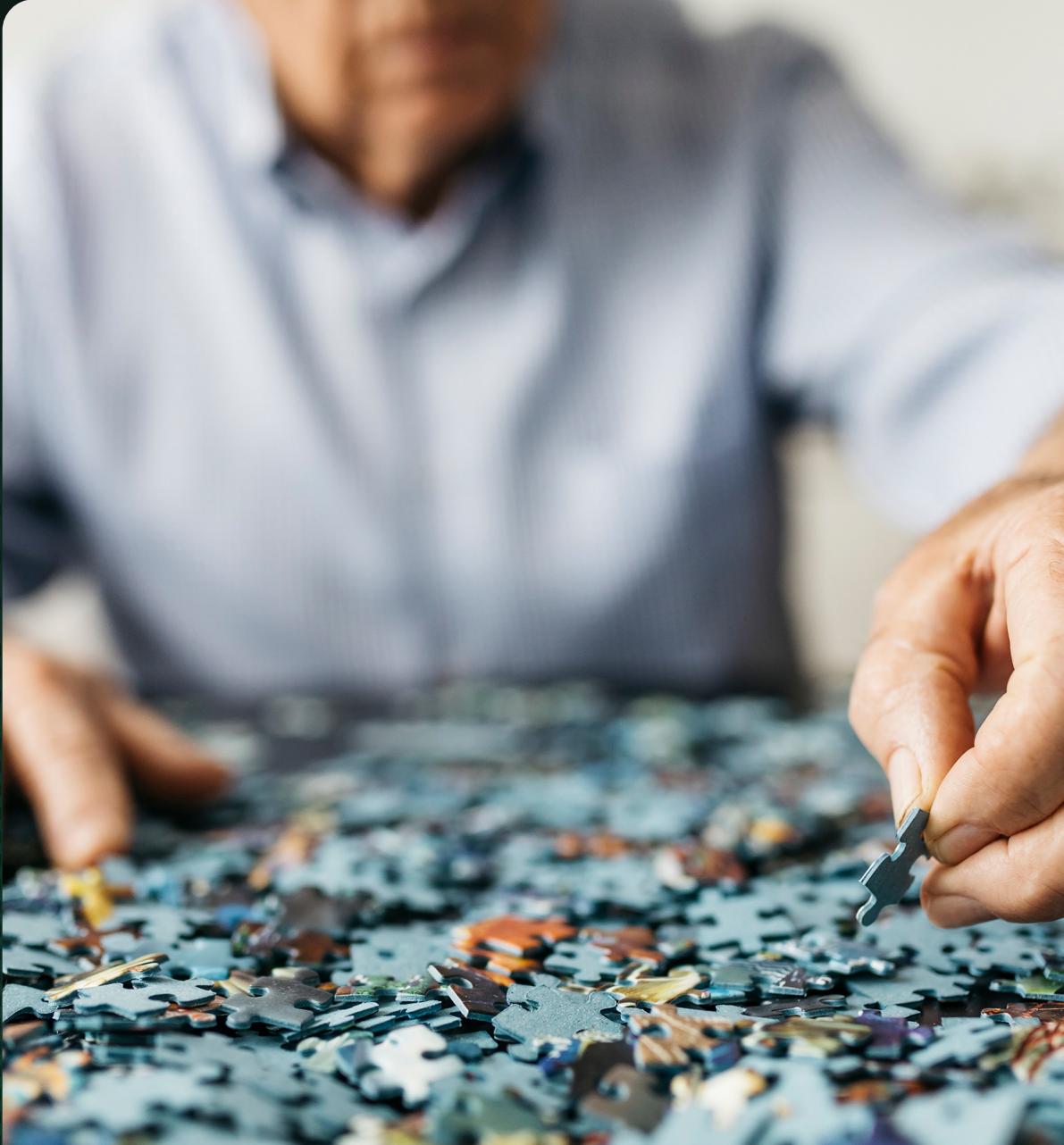
A Branch is a designated copy of the codebase where changes are made without disrupting the main/master branch, which is the official current version of the software.

In this way, changes are made to the software by implementing features in copies of the code, and then replacing the main code with the copy once it's clear that it won't break the software.

Commit: Snapshots along the way

Imagine working on a puzzle.

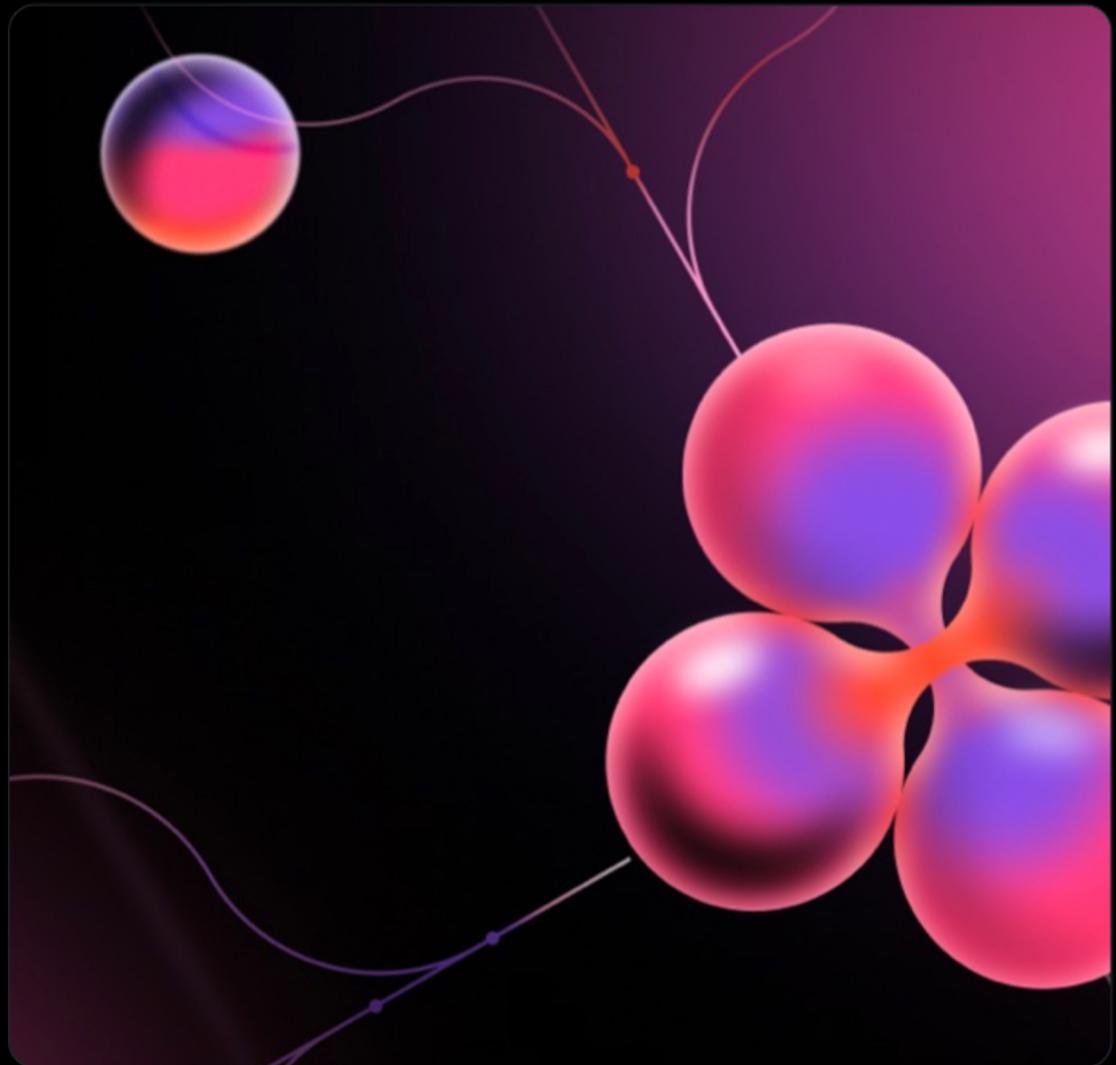
- Let's do the edges first.
- Then section by section
- Each time, you get the puzzle to a state that you're happy with and can put it away, that's a commit



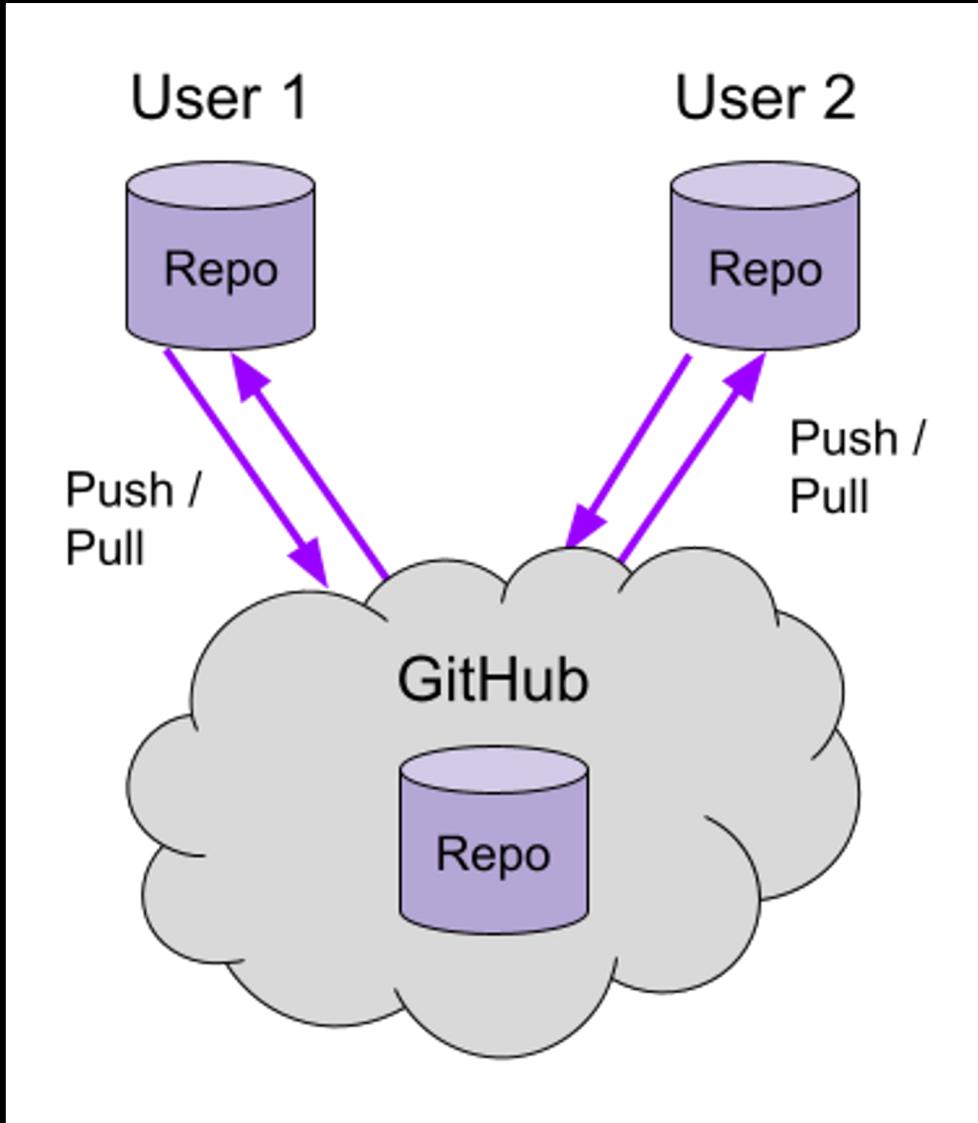
Open Source Software

Open Source Software means you can download, change, modify, or whatever you want as long as you follow the rules in the license of the software.

Git is not owned by GitHub, but because it's OSS, we have built GitHub on top of Git.



What is GitHub?



GitHub Platform

Copilot

Collaboration

- Pull requests
- Projects
- Issues
- Discussions
- Merge queue
- Search

Productivity

- CI/CD with Actions
- Automation
- Copilot
- Codespaces
- Runners
- npm
- Mobile

Security

- Advanced Security
- Secret scanning
- Code scanning
- Security overview
- Supply chain
- Dependabot

Scale

- 150M+ developers
- Source code management

Integrations & APIs

A GitHub Repository Tour!

The screenshot shows a GitHub repository page for the user 'MetzinAround'. The repository name is 'MetzinAround' and it is public. The main navigation bar includes links for Code, Issues (1), Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. A search bar at the top right allows users to search for files. Below the navigation bar, there is a profile picture of the user 'MetzinAround' and a 'Pin' button. The repository has 8 branches and 0 tags. A 'Code' dropdown menu is open, showing options to Go to file, Add file, or view the raw code. The repository's history shows the following commits:

Commit	Message	Date
dc32da5 · 2 weeks ago	MetzinAround typos and more	29 Commits
	added new pic images to update the profile readme with	2 weeks ago
	typos and more	2 weeks ago
	changing picture 🤝	7 months ago

<https://github.com/BestTeeChur/repo-tour>



Kedasha Kerr

LadyKerr

LadyKerr / README.md

Hey, I'm Kedasha! 🙌

I'm passionate about making AI accessible to developers and helping underrepresented folks thrive in tech. I've keynoted alongside tech CEOs, reached 10M+ developers globally, and created frameworks that help developers turn their expertise into career acceleration. I currently work at GitHub as a Developer Advocate.

I believe AI should amplify human potential, not replace it. Let's build the future where developers thrive alongside AI.

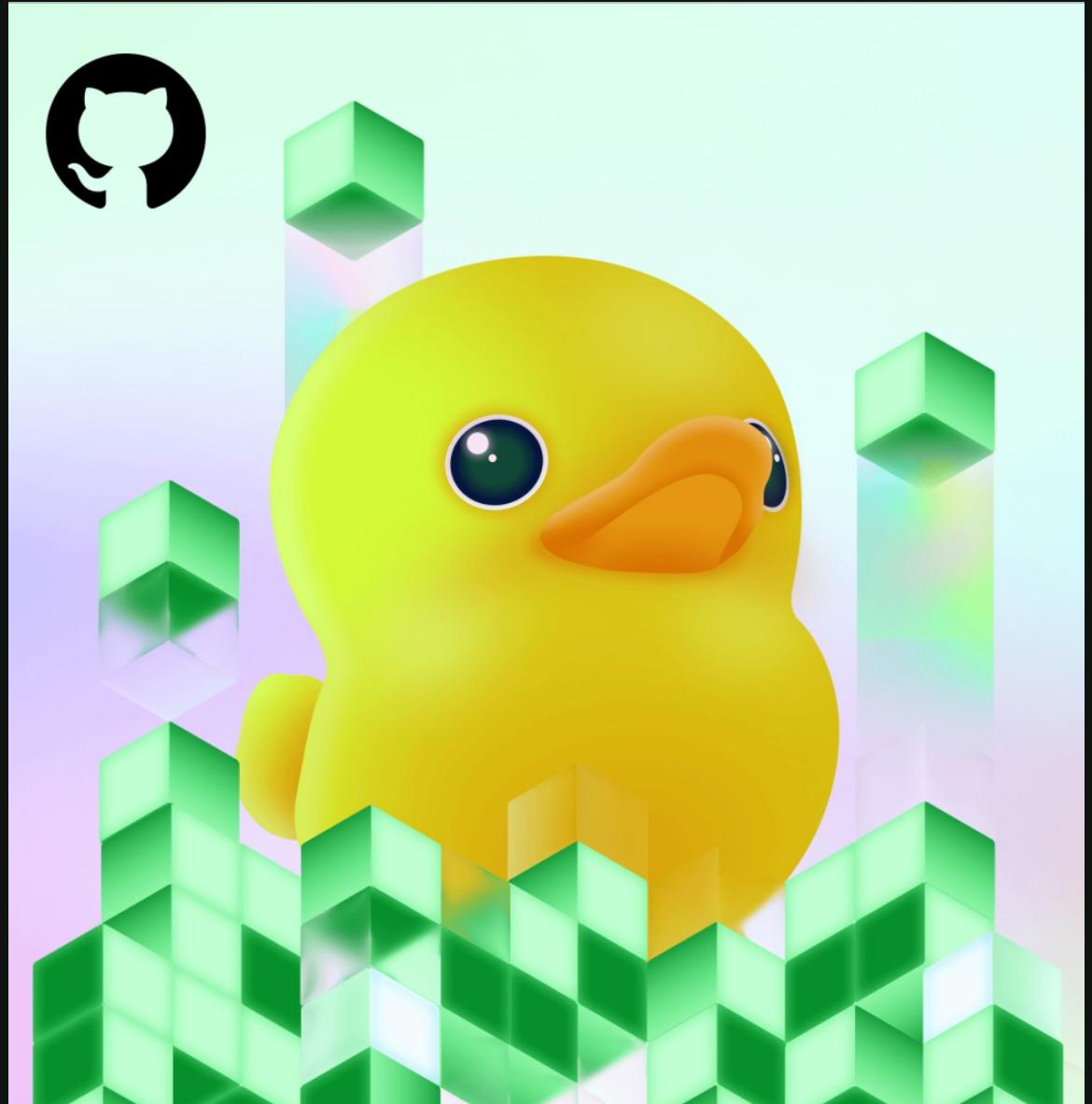
⌚ What I Do

- **AI Education:** Scaling GitHub Copilot adoption and teaching developers to build with AI
- **Developer Advocacy:** Creating content that reaches millions of developers worldwide
- **Framework Creation:** Built the C.O.D.E. Framework for strategic developer content creation
- **Community Building:** Supporting women and underrepresented developers in tech
- **Founder:** Created [VibeCodeHer Academy](#) to turn ideas into apps in 48 hours - no coding required

Profile Readme Activity

<https://learnwithpj.com/profile-readme/>





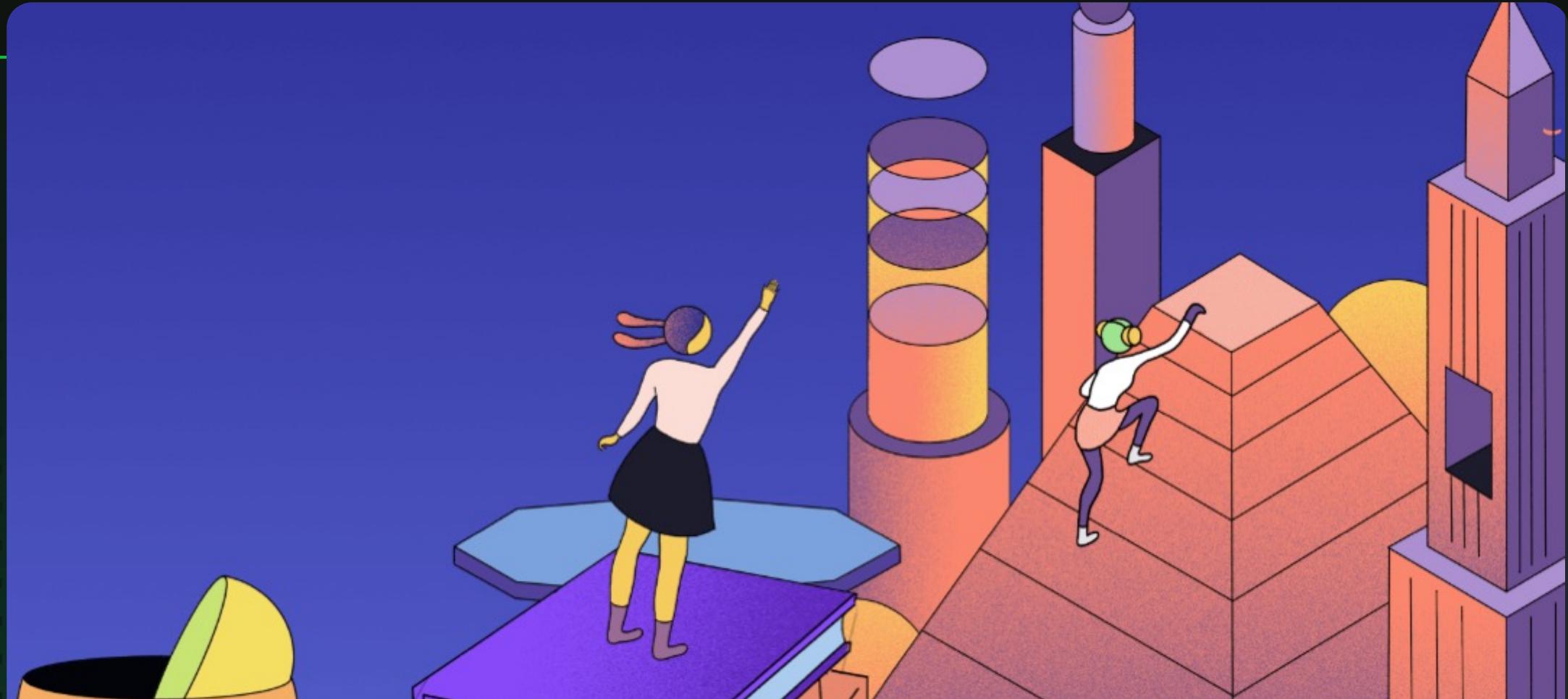
Collaborative Activity

<https://learnwithpj.com/collaborative/>



LUNCH

GitHub in your Curriculum



How can this work in your Classroom?



Groupwork in GitHub

Have them use the tool that teams already use to build software.

Bonus: Git tracks who actually contributes to the assignment.



Repos as Assignments

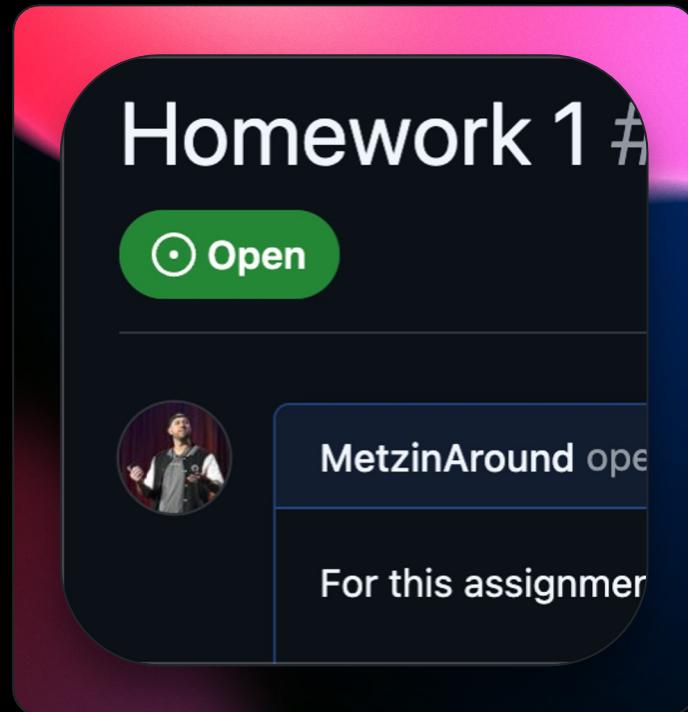
Using a Repo for distributing code to students. Add the students as collaborators or have them fork the repo to work in.



Real world experience

- Due Dates are Milestones
- Assigned issues in larger projects
- Tracking expectations while they work (Just look at their repo!)

Using Issues in class



English-III-Honors

Assignments, Information, and more.

Welcome to English III Honors! This course will cover American Lit/media from the late 1800s through the present, with a focus on close reading, analysis, and media literacy.

Inside this repository you will find the syllabus, a reading guide, and more resources available for your convenience.

Assignments will be created through Issues and subsequent repos that have files in them for you to work on. Please make sure that you always [fork](#) a repo in order to add your changes or your group's changes to the assignment.

<https://gh.io/englishclass25>

GitHub 101

Activities

[Home](#)

[Creating A Profile Readme](#)

[Collaborative Activity](#)

[Creating and Solving a Merge Conflict](#)

Use our Activities from Today!

- Have students make a Profile Readme for a quick intro to GitHub.
- Have Students learn to work collaboratively by using the Collaborate Activity
- Teach them to avoid Merge Conflicts

These are ready to go assignments for you all to use in your class OR to assign independently to students.

If you want the raw markdown to use for yourself, check out the repo for the activities:
<https://gh.io/activities-repo-26>



Fri 20 Feb 2026 13:40 - 15:00 at Meeting Room 230 - GitHub Session - 2

GitHub Education: Cultivating the Educator Community

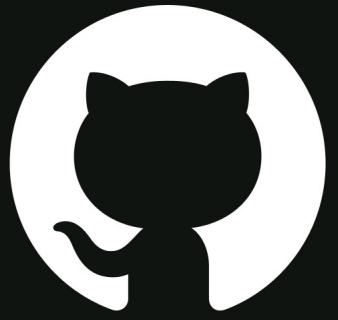
GitHub Education's mission is to bring tech and open source collaboration to students and educators across the globe. As technology, including AI, rapidly advances, computing education is racing and evolving to keep abreast of the changes. GitHub Education believes educators are key to this evolution. We will share how GitHub Education engages with and supports the educator community through products such as GitHub Classroom, programs such as the Campus Program, and our community-based initiatives.

We will share the broader approach guiding this work—how our programs and tools fit together, the principles shaping their evolution, and the areas where we're investing to better support students, teachers, and institutions.

Designed as a precursor to the GitHub Educator Summit in March 2026, this session gives participants an inside look at what GitHub Education offers today, what's new, and what's changing in the months ahead.

Learn more about GitHub Education here at SIGCSE

**GitHub Education: Cultivating
the Educator Community**



Copilot Primer

First Steps to new Lands

 GitHub Copilot



Where to use Copilot

- Auto Complete
- Chat
- Agent Mode
- Plan Mode
- Copilot Spaces
- In your IDE
(VSCode, Jetbrains,
& more)
- Command Line

A screenshot of a web browser window showing the GitHub Copilot interface. The browser bar at the top has a dark theme with red, yellow, and green circular icons, a back/forward button, a lock icon, the URL 'github.com', and a plus sign for new tabs. The main content area has a dark background with white text and features a large, stylized blue and white blocky font logo for 'COPILOT'. Above the logo, the text 'Welcome to GitHub' is visible. Below the logo, the text 'Command-line interface' is displayed. To the right of the logo is a pixelated illustration of a person wearing a blue and pink headband with a brain-like pattern. Below the logo and text, there is descriptive text about Copilot's capabilities, followed by a bulleted list of status items and a terminal-style command prompt.

Welcome to GitHub
COPILOT
Command-line interface

Copilot can write, test and debug code right from your terminal. Describe a task to get started or enter ? for help. Copilot uses AI, check for mistakes.

- Connected to GitHub MCP Server
- Logged in as user: MonaLisa

~/MonasRepo [./main]

> Enter @ to mention files or / for commands

Ctrl+c Exit · Ctrl+r Collapse all

But Pj, how can WE as Educators use GitHub Copilot?

Great question

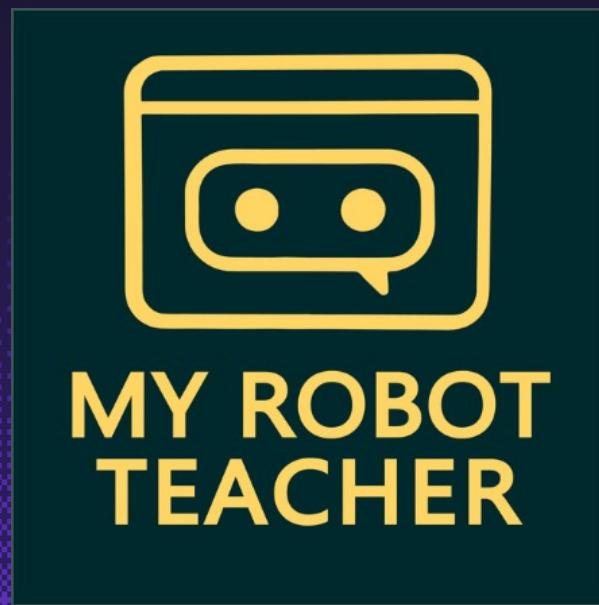
- AI Proofing your assignments
 - “Take this example project and help me brainstorm ways to keep students from using AI on it”
- Make new Lesson Plans
- Update old Lesson Plans
 - “Make this solo project into one for a group.”
 - “Update this for the latest version of Python.”
- Starter Code for assignments
- Broken Code for debugging
 - “Write me a Python terminal app that has five hard to spot errors. Include functions, global vs local variables, small formatting issues, and user input.”
- Feedback on code?
 - But why not encourage students to get feedback themselves?
- Differentiated Learning
 - “Given this test on this topic, give me 30 more test questions I could ask. Create three versions of the test with answer keys.”
- Testing the Test
 - “Are there any questions in this test that might have too hard an answer for students to get?”
- Different types of test
 - “Make this formative assessment more summative”

But Pj, how can STUDENTS use GitHub Copilot effectively?

Great question

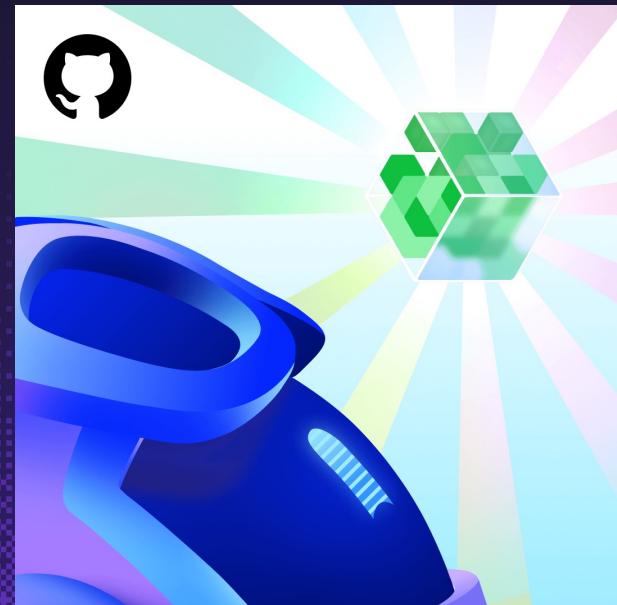
- Writing comments for code they wrote themselves.
 - “Comment this code to explain how it works”
- Guardrails (before prompt)
 - “Copilot do not give me the answer this whole session. Do not provide code or answers, only ask me questions to help me understand. Act like a tutor.”
- Error Checking
 - “What does this error mean?”
- Replace Google
 - “I forgot how to start a virtual environment in Python.”
- First Pass
 - “The code runs, but I wonder what my professor will think about it. Can you ‘grade’ this for me?”
- Additional work to check self understanding
 - “Based on this assignment, give me a project idea that will help me put this into real world practice.”
 - Or
 - “What should I go learn next now that I covered this in class?”

Resources



My Robot Teacher

Excellent podcast from a Math and a Literature professor in SF. They are AI fans, but they approach it critically and with an eye for how it affects learning.



Copilot for Beginners

GitHub Developer Advocates have a complete guide on YouTube to fill in any blanks from today.
gh.io/begincopilot

A screenshot of the GitHub Copilot documentation page. It features a large title "GitHub Copilot documentation" and a subtext "You can use GitHub Copilot to enhance your productivity and assist as you work on code." Below the subtext are two buttons: a green "Overview" button and a black "Quickstart" button.

Docs A Plenty

Check out our docs to learn how to install, download, and use all versions of Copilot! Gh.io/docs

Thu 19 Feb 2026 10:40 - 12:00 at [Meeting Room 240](#) - GitHub Session - 1

★ GitHub: Everything You Wanted to Know About Agentic AI (But Were Afraid to Ask)

Feeling like AI is moving faster than you can keep up? You're not alone. With new terminology, new tools, and constant innovation, it can be hard to separate the signal from the noise. This session is designed to give educators a clear, grounded understanding of where the industry is headed.

We will break down what Agents and Model Context Protocol (MCP) servers are, why they matter, and how developers are using them in modern software development—including through new GitHub tools such as Agent HQ and Mission Control. You'll learn how GitHub Copilot has evolved beyond autocomplete-style suggestions into a broader ecosystem of agents and extensible workflows that help developers design, debug, and ship software more efficiently.

Through live demonstrations, we'll show how agents can execute multi-step tasks, troubleshoot code, interpret project context, and connect with external tools or data sources. You will walk away with a clearer understanding of the latest Copilot capabilities and how these emerging patterns can support both teaching and learning.

The session concludes with dedicated Q&A time to help translate concepts into practical strategies you can bring into your classroom.

Learn more about Copilot here at SIGCSE

[GitHub: Everything You Wanted to Know About Agentic AI \(But Were Afraid to Ask\)](#)

Q&A

Parking Lot

Questions



Free Work Time



Structured Work time

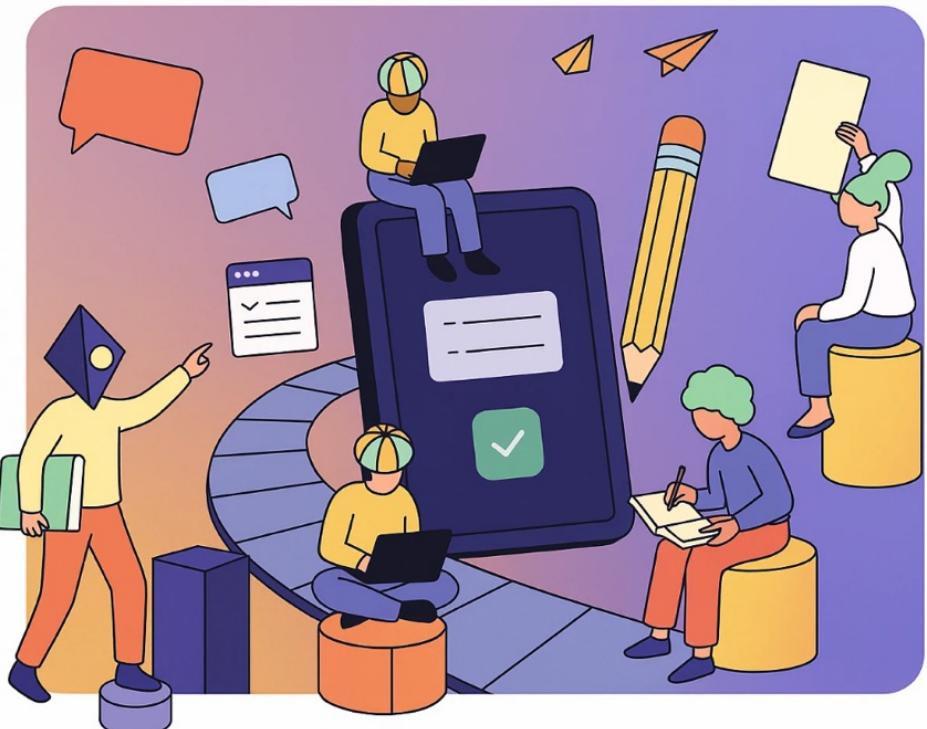
This is time for you all to strengthen the skills you just learned, try something new, or continue to ask us questions.

Your hub for teaching real-world development

The Teacher Toolbox is a resource hub for educators and administrators to get started with GitHub. It includes helpful links, guides, learning modules, sample curriculum, use cases, videos, and articles to help teachers in the classroom.

[Get verified](#)

[Browse resources](#)



Teacher Toolbox

<https://gh.io/sigcse-26-tchrs>



A chalkboard background featuring a chalk-drawn illustration of several books standing upright and an apple resting on a horizontal surface. The drawing is done in white chalk on a dark green chalkboard.

We'd love your
feedback!

Gh.io/sigcse26gh1

<https://learn.github.com/>

Start your GitHub learning journey

Explore curated learning content, interactive exercises, and certifications to enhance your GitHub skills on your preferred learning platform—and take the next step in your career.

Learn

Learn from a library of self directed learning content and hands-on exercises.

[Browse Content](#)

Practice

Practice your GitHub skills on GitHub with hands-on learning experiences.

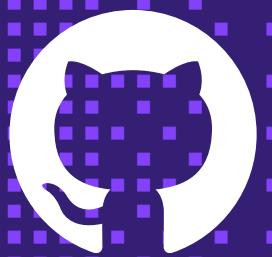
[Try GitHub Skills](#)

Credentials

Validate and showcase proficiency with GitHub products through credentials.

[Explore Credentials](#)





Thank you