XIMERA

Interactive
Mathematics
Education
Resources for
All

User Manual

Fowler • Obbels • Nowell • Snapp

This document was typeset on 2024/12/16.
GIT COMMIT: d0d11e67a0d49a79f113507bdcc1514ea778bb70
Source: https://github.com/ximeraProject/ximeraManuals

To lovers of mathematics everywhere.

Contents

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

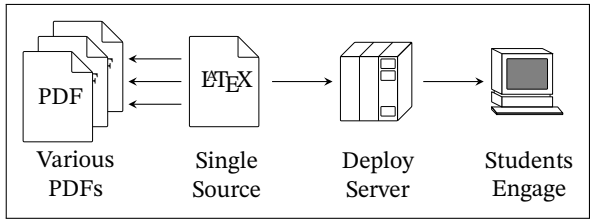
1 Introduction and setup

1.1 About Ximera

What is Ximera? What is it supposed to do? Who is it for?

Ximera, pronounced "chimera," (**X**imera: **I**nteractive, **M**athematics, **E**ducation, **R**esources, for **A**ll) is an open-source platform that provides tools for authoring and publishing (PDF and Online), open-source, interactive educational content, such as textbooks, assessments, and online courses.

Authors write and store their content on their own machines and GitHub repositories. Authors own their content and decide how to license their content. From a single source written in LaTeX, Ximera generates various output: PDF worksheets, PDF textbooks, and PDF solution manuals, and so on. Of most interest, Ximera can also create online interactive activities:



The source code used to produce PDFs can also create interactive online activities when deployed to a Ximera server. Students access this content via a URL or an assignment in their LMS.

Students interact with the *content* produced within Ximera, hence their experience is highly dependent on the *quality* of this content. Research shows that students find Ximera materials to be more readable than traditional course materials and perform equivalently to those using proprietary textbooks and online homework systems. While students typically encounter Ximera through their courses, many discover it via web-search and use the platform as independent learners. In 2023, Ximera has over one million unique visitors. Since Ximera materials are free, they are accessible to anyone, regardless of enrollment in official courses.

Get involved by contributing as an instructor, author, or developer. To get started with Ximera, visit our *First Steps in Ximera* GitHub repository:

https://go.osu.edu/xfs

This document assumes you have completed the instructions there, and have successively deployed Ximera content online.

Funding for the Ximera Project is provided by a U.S. Department of Education Open Textbooks Pilot Program grant in the amount of $2,125,000, from 2024–2026, with no external funding. In the past, the Ximera Project has also received support from NSF Grant DUE-1245433, the Shuttleworth Foundation, the Ohio State University Department of Mathematics, and the Affordable Learning Exchange at Ohio State.

As a token of our appreciation, **consider applying for a Ximera Flash-Grant Stipend:**

https://go.osu.edu/ximera-flash-grant

Thank you for your interest in Ximera. We encourage you to contact the team with any questions you may have.

The authors listed on the cover are the current Ximera lead developers. In reality, this document has many authors as it is part of an evolution of Ximera documentation. Rodney Austin, Oscar Levin, Matt Thomas, and Hans Parshall authored parts of the either the document class or original documentation.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

1.2 First steps in Ximera

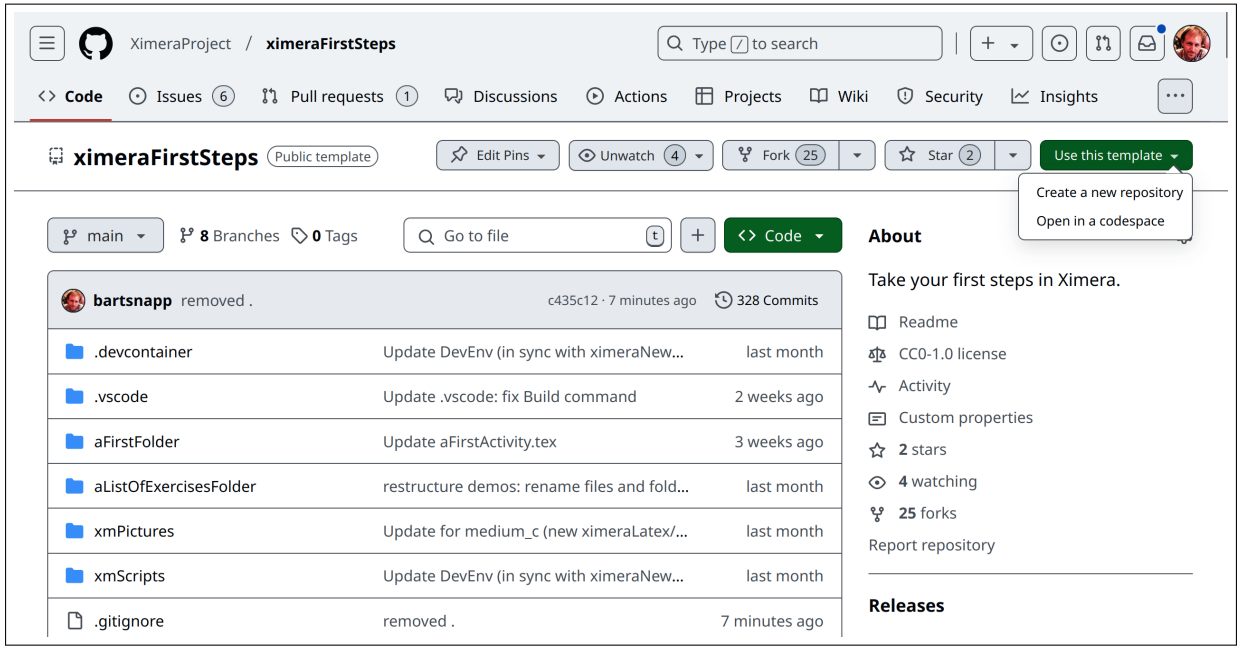
Try out Ximera!

To use Ximera, you must have a GitHub[1] account. GitHub is a web platform where developers can store, share, and manage their code. It uses `git`, popular software for version control, to help teams work together simultaneously without overwriting each other's changes. GitHub has issue tracking, pull requests for proposing changes, and other project management tools. It's like a shared folder for coding, designed to help teams work smarter and track progress. Go to `https://github.com` and either sign-up or log-in. Note, you must know your **username** and **password**, so store them in a safe place; like in a safe, or under your bed.

After you have a GitHub account, log-in and go to:

`https://github.com/ximeraProject/ximeraFirstSteps`

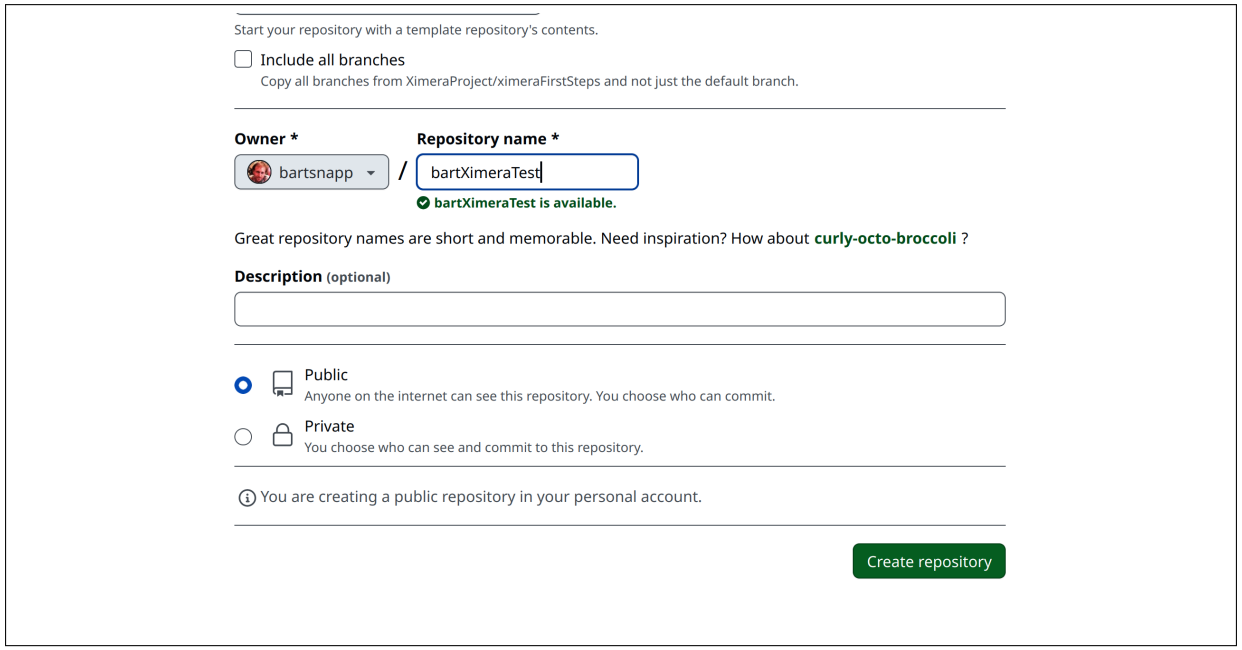
You will see something like this:



[1] See GitHub at `https://github.com`

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

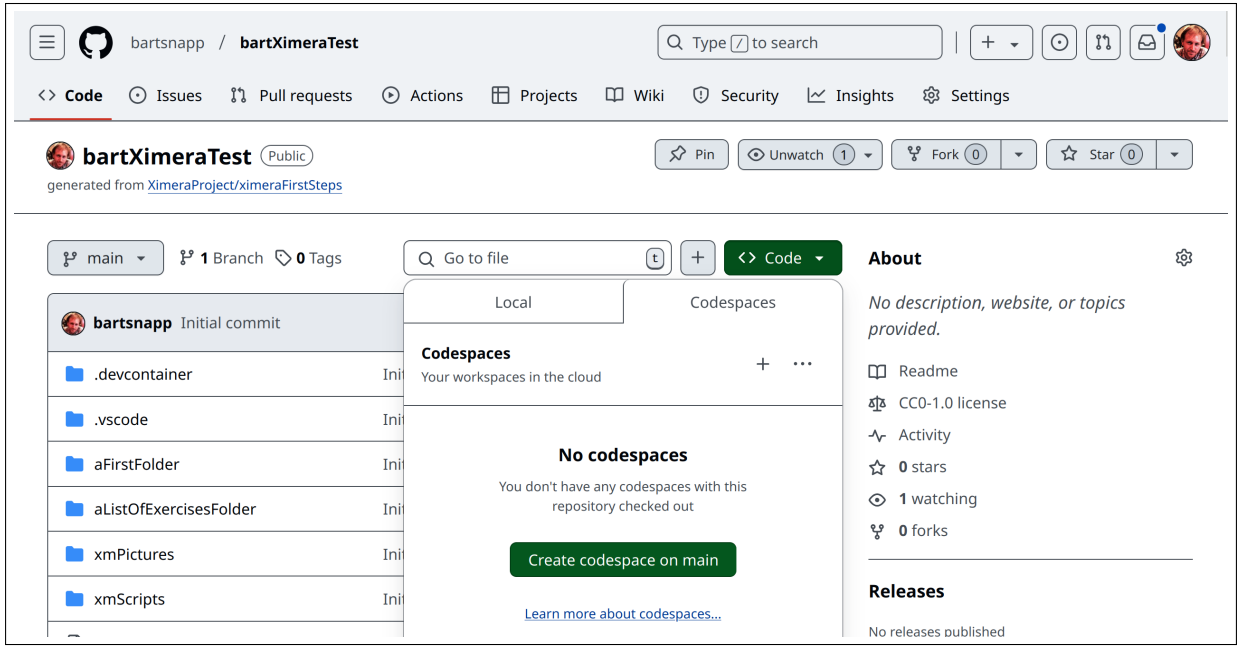
Click on the green “Use this template” button and select “Create a new repository.” Give it a fun repository name, and push the button “Create repository.”



At this point you have your own personal copy of our repository `XimeraFirstSteps`. In fact, after you create it, GitHub will take you to it. This copy can always be found at

`https://github.com/YOUR-GIT-USER-NAME/YOUR-REPO-NAME`

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

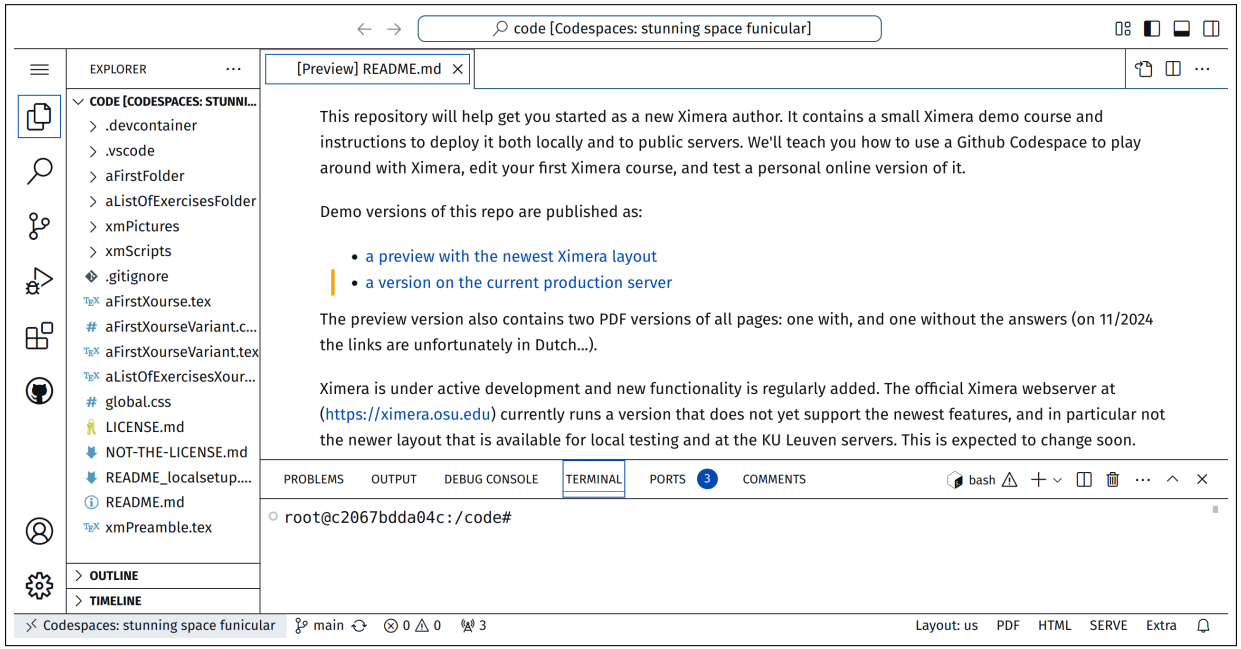


Once there, click the green “code” button, select the “Codespaces” tab, and click “Create codespace on main.” A GitHub codespace is like a remote computer set up specifically for coding. It’s a cloud-based environment where you can write, test, and run your code, just like on your own computer, but everything happens on remote servers. It comes preconfigured with all the tools, libraries, and settings you need for your project. You connect to it through your browser or favorite editor, and because it’s tied to your GitHub projects, you can instantly start working without worrying about setting up software on your local machine. It’s like having a ready-to-use, fully equipped coding computer that you can access from anywhere. **It will take around 5 minutes for your codespace to be created.**

XIMERA FORM 3193 GIT COMMIT d0d11e67a0d49a79f113507bdcc1514ea778bb70 PREVIOUS EDITION IS OBSOLETE.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

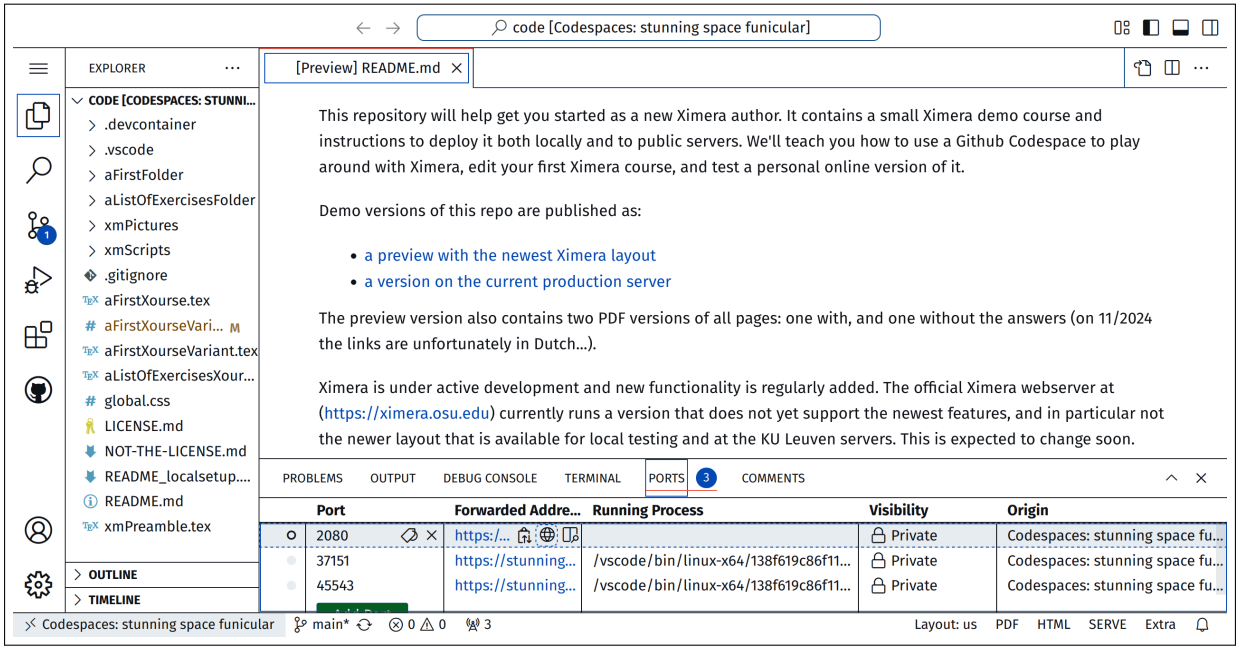
Once the codespace is created, you will see something like this:



At the bottom right-hand corner of the screen you will see a button that says “SERVE.” Press the “SERVE” button to compile Ximera content to HTML and JavaScript. This will take a few minutes.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

Once the compilation is finished, note the line that says: “PROBLEMS,” “OUTPUT,” “DEBUG CONSOLE,” “TERMINAL,” “PORTS.” You want to click on “PORTS.” The “PORTS” tab may be hidden within · · ·.



After you click on “PORTS,” click on the globe, and a webpage will open. Your content will be under the link “Content.” You should be able to see the content in your browser. Demo versions of this repo are published as:

- a preview with the newest Ximera layout[2]
- a version on the current production server[3]

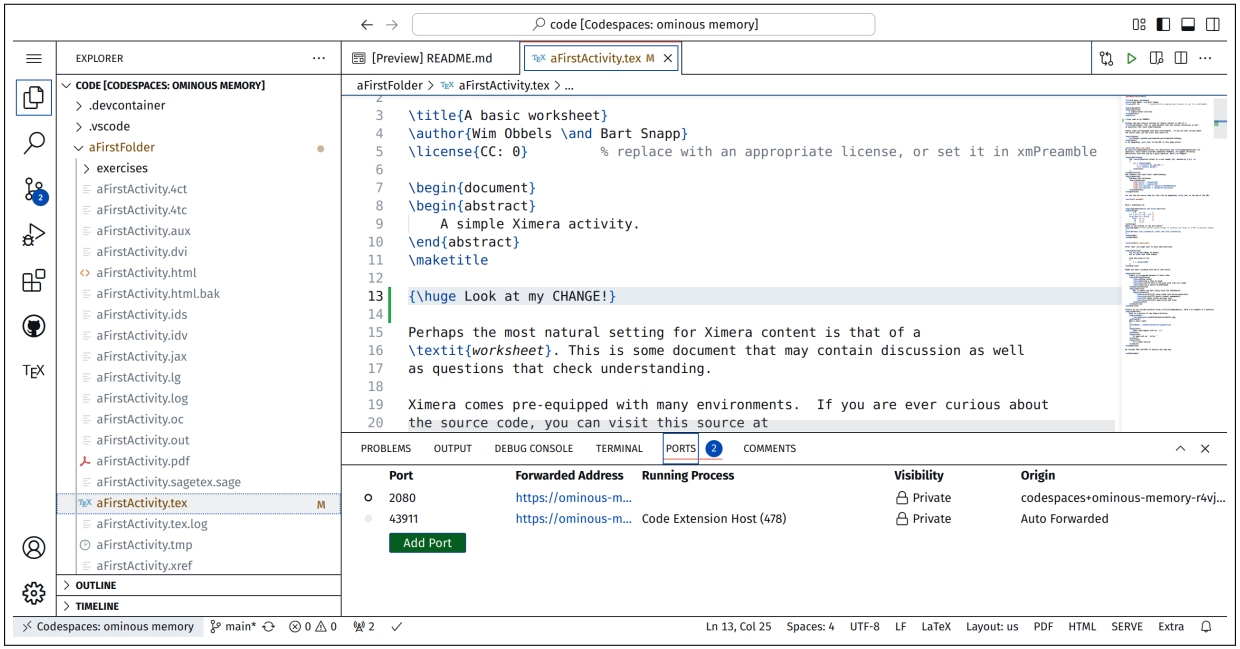
[2]See a preview with the newest Ximera layout at https://set.kuleuven.be/voorkennis/firststeps24/aFirstXourseVariant/aFirstFolder/aFirstActivityVariant
[3]See a version on the current production server at https://ximera.osu.edu/firststeps24/aFirstXourse/aFirstFolder/aFirstActivity

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

1.3 Working with git

We introduce you to git, and help you make changes on your own.

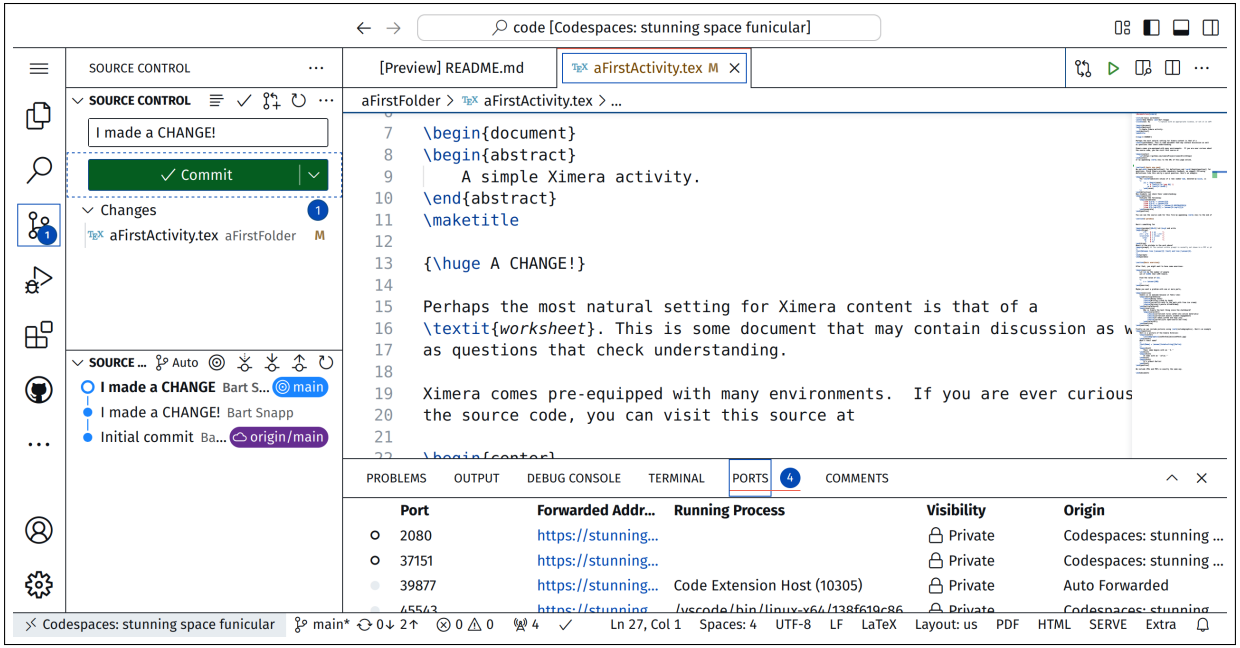
Once you've deployed our content in a GitHub codespace, you'll surely want to change it, make it your own. Within codespace, you are running VS Code, a full-fledged text editor. You can make changes directly there. Our files are on the left, and are revealed by the "pages" icon. Here, I've opened the folder aFirstFolder, and then the file aFirstActivity.tex. I made a change in the middle of the screen.



At this point, you may push "SERVE" and see the results of your change; however, **these changes were made only in your codespace**, a temporary computer, lost in the cloud. To make these changes to your actual GitHub repository, you need to "sync" them back.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

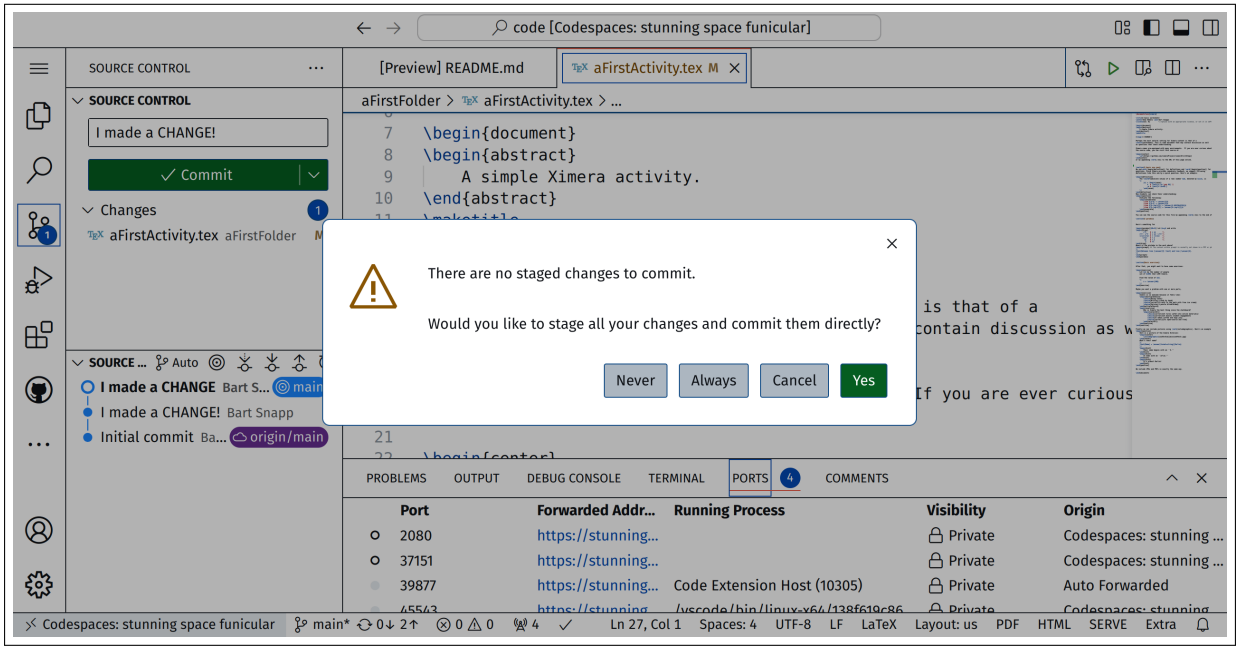
You start by clicking on the icon that looks like a poorly drawn "Y" with lines and circles on the left. Then you click on a + for every file you want to send to your repository. You must type a message. I typed the message "I made a CHANGE!"



XIMERA FORM 3193 GIT COMMIT d0d11e67a0d49a79f113507bdcc1514ea778bb70 PREVIOUS EDITION IS OBSOLETE.

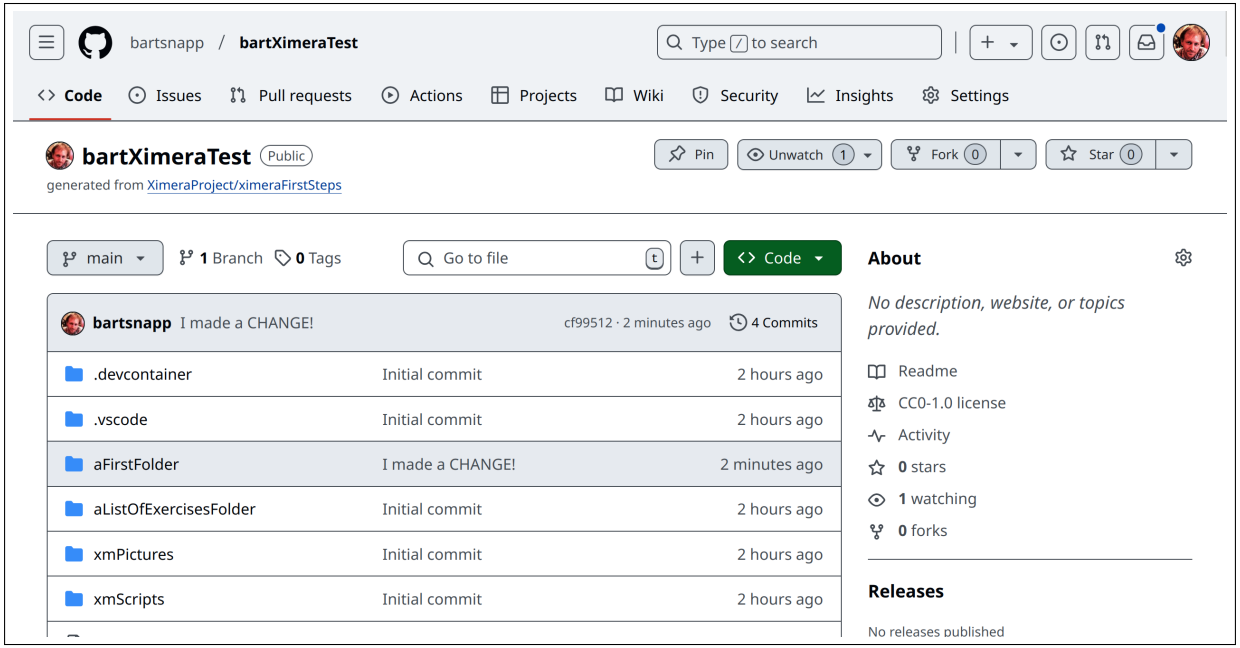
Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

Next to the “Commit” button there is a carrot down. Click it, and select “Commit and Sync.”



Click “Yes” and you should be sending your changes back to your source code.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/



To check that everything worked correctly, go to

https://github.com/YOUR-GIT-USER-NAME/YOUR-REPO-NAME

Above we see my repository, and we see that my change was indeed made.

1.3.1 Understanding Git Commands

Git is like a magical notebook that remembers every change you make to your project. It helps you go back in time if something breaks and lets you share your work with others. For this reason, it makes you do a little "dance" to ensure good code hygiene.

Step 1: Staging Files (The "+" Button) **What You're Doing:** When you click the little "+" next to a file, you're saying,

"Hey Git, this file is ready to be saved!"

What Git Does: This adds the file to a special list called the **"staging area."** Only files in this list will be saved in the next step.

Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/

Command Behind the Scenes:

```
git add <filename>
```

if you want to add all updated files, do

```
git add -u
```

Why It's Important: Imagine you're painting, but you only want to save some parts of the picture right now. Staging lets you pick exactly which files (or changes) to save.

Step 2: Committing Changes (Saving Your Work)

What You're Doing: After staging your files, you type a message and click **"Commit."** This tells Git:

> *"Save these changes forever, and here's a note about what I did."*

What Git Does: Git takes a snapshot of your files and saves them with your message. It's like taking a picture of your work and writing a caption.

Command Behind the Scenes:

```
git commit -m 'Your commit message'
```

Why It's Important: Commits are like save points in a video game. If something goes wrong later, you can go back to this point and start fresh. Your notes (commit messages) are clues as to where you should go.

Pro Tip: Always write clear commit messages, like *"Fixed the login bug"* or *"Added a new feature."* Future-you will thank you!

Step 3: Syncing with GitHub (Sending Your Work Online)

What You're Doing: When you click **"Commit and Sync"**, you're telling Git:

> *"Send my saved changes to the big Git notebook on GitHub."*

What Git Does: Git takes your saved changes and sends them to your **remote repository** (the one on GitHub). At the same time, it checks if there are any new changes from your teammates and brings them back to you.

Command Behind the Scenes:

```
git push    # Sends your changes to GitHub
git pull    # Gets changes from GitHub
```

Why It's Important: Syncing keeps everything up-to-date! You're sharing your work and making sure you have the latest updates from your team.

The Big Idea: Git Workflow

Here's the step-by-step magic:

(a) **Stage:** Use the "+" button to pick what you want to save. (`git add`)

(b) **Commit:** Click commit and write a message to save your changes. (`git commit`)

(c) **Sync:** Send your changes to GitHub and get updates from your team. (`git push`, `git pull`)

Checking Your Work

After syncing, go to:

`https://github.com/YOUR-GIT-USER-NAME/YOUR-REPO-NAME`

You should see your changes there!