XIMERA

Interactive
Mathematics
Education
Resources for
All

User Manual

Fowler • Obbels • Nowell • Snapp

This document was typeset on 2024/12/18.
GIT COMMIT: 97ca259eb9f56cb6de2218559b36c055dca3fc36
Source: https://github.com/ximeraProject/ximeraManuals

To lovers of mathematics everywhere.

Contents

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

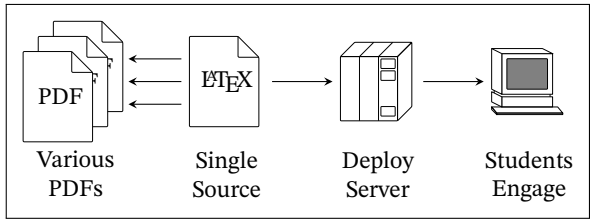
1 Introduction and setup

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, Marcus Bishop, Oscar Levin, Matt Thomas, and Hans Parshall authored parts of either the document class or original documentation.

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

1.2 First steps in Ximera

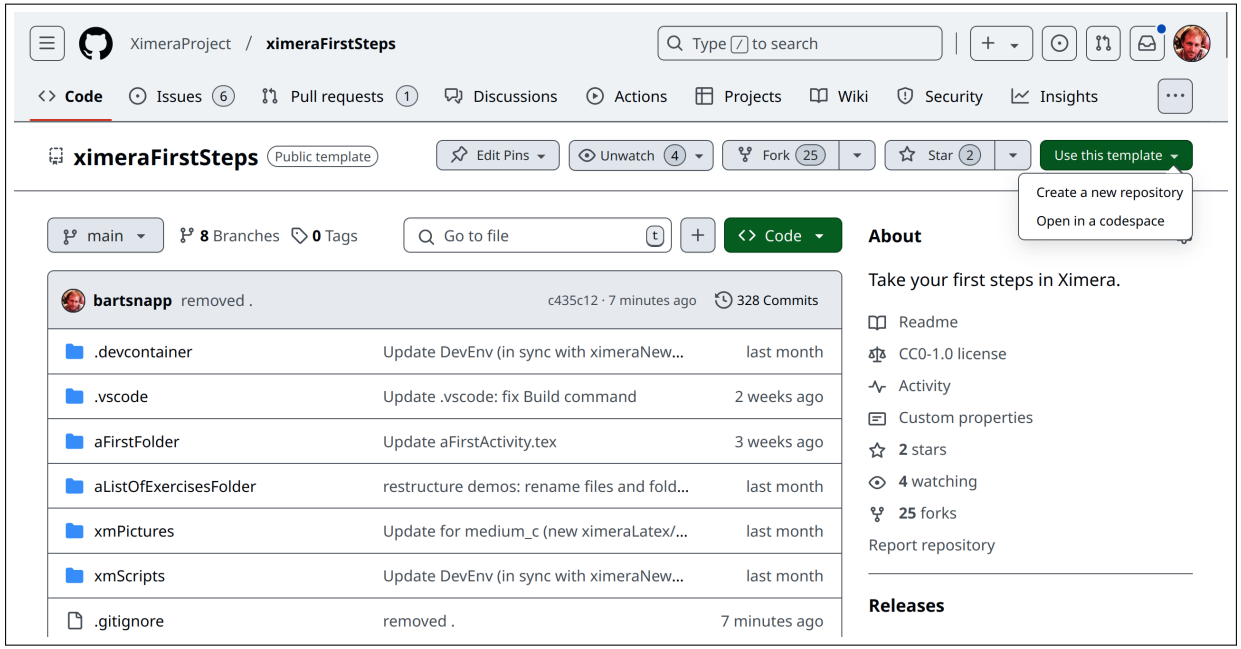
Try out Ximera!

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:

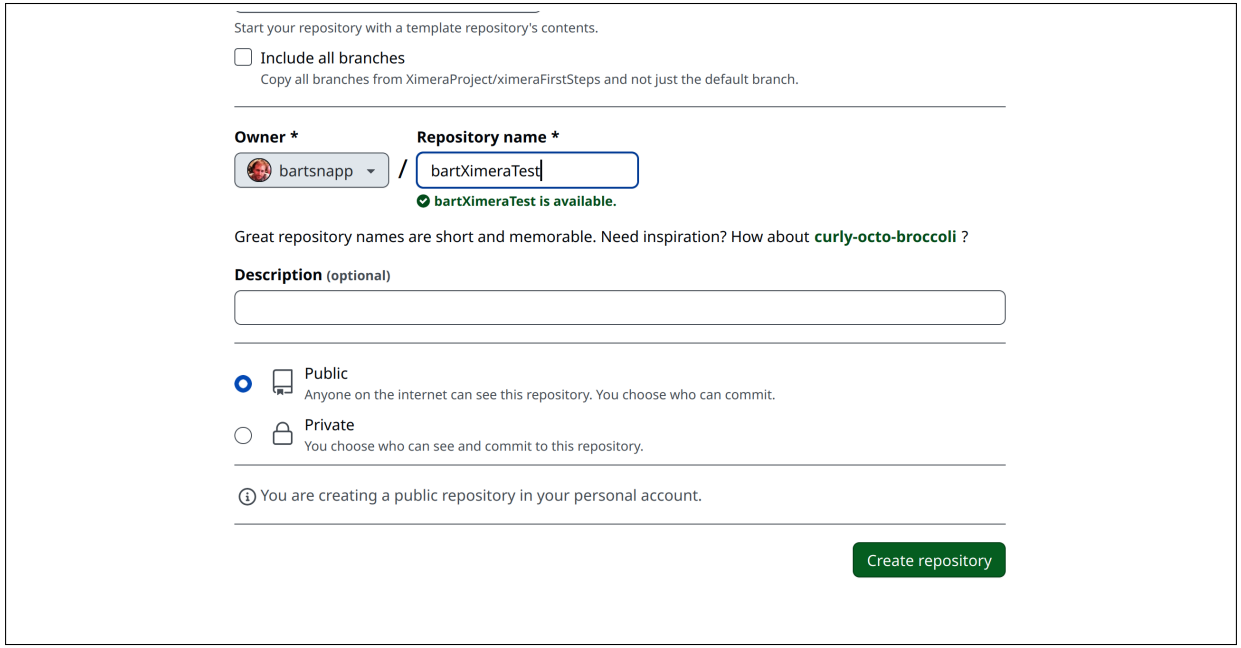
`https://github.com/ximeraProject/ximeraFirstSteps`

You will see something like this:



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

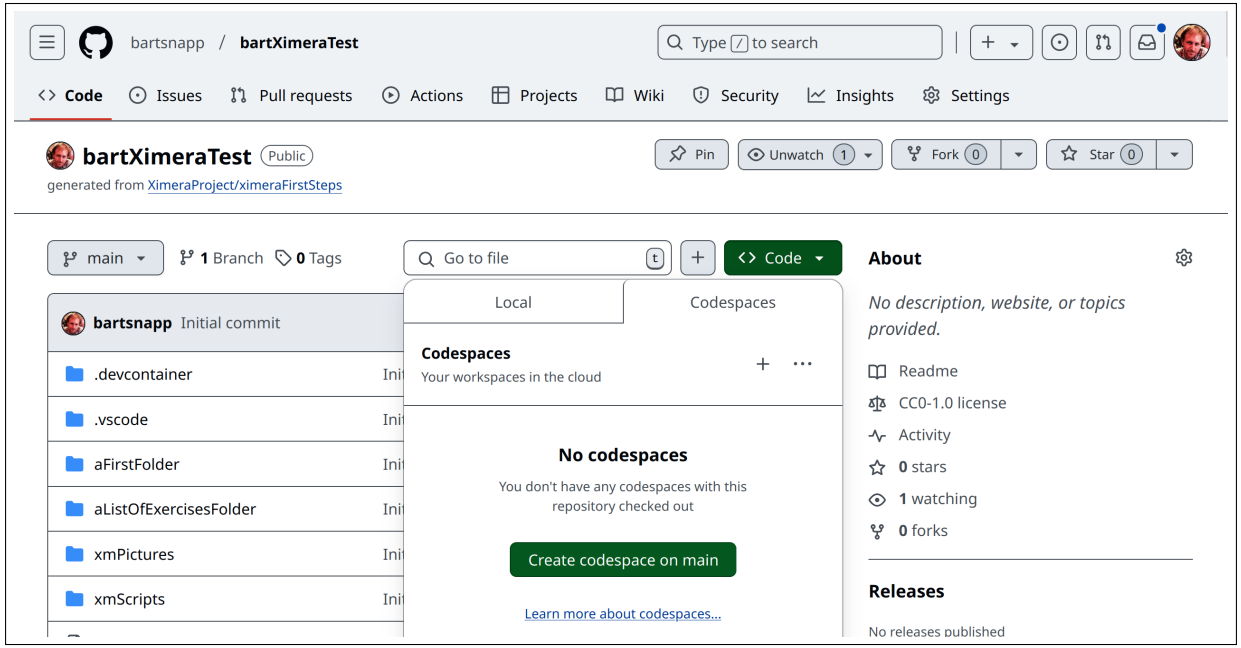
Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/



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

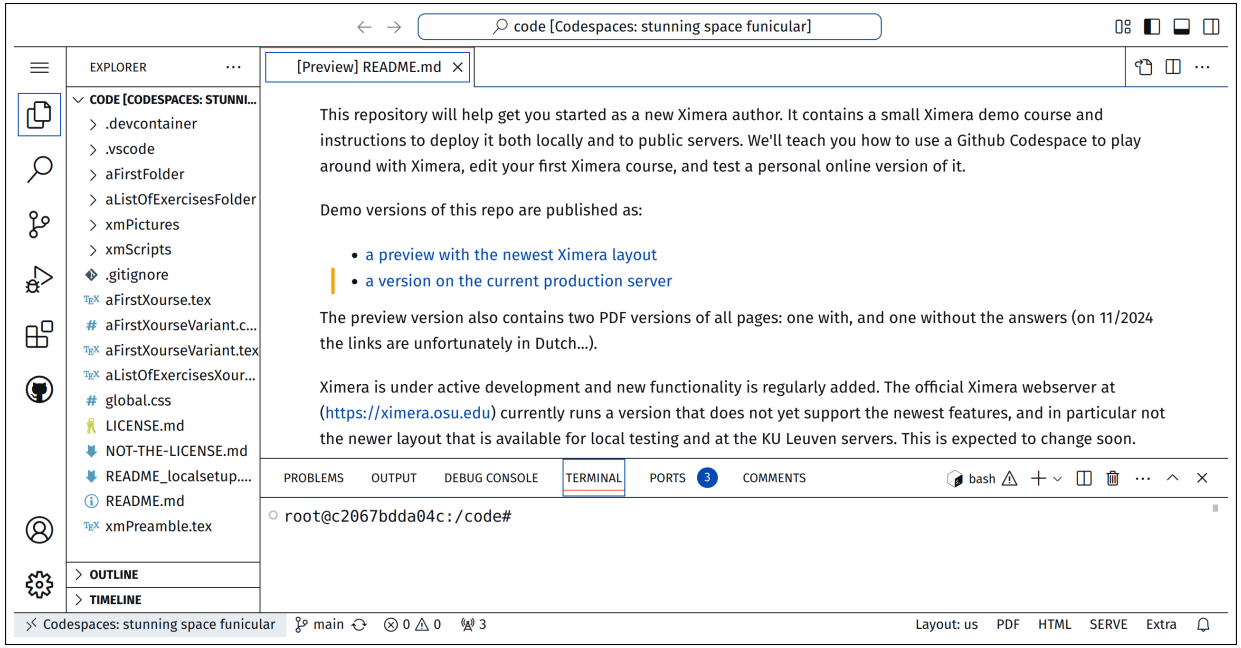
```
https://github.com/YOUR-GIT-USER-NAME/YOUR-REPO-NAME
```

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.**

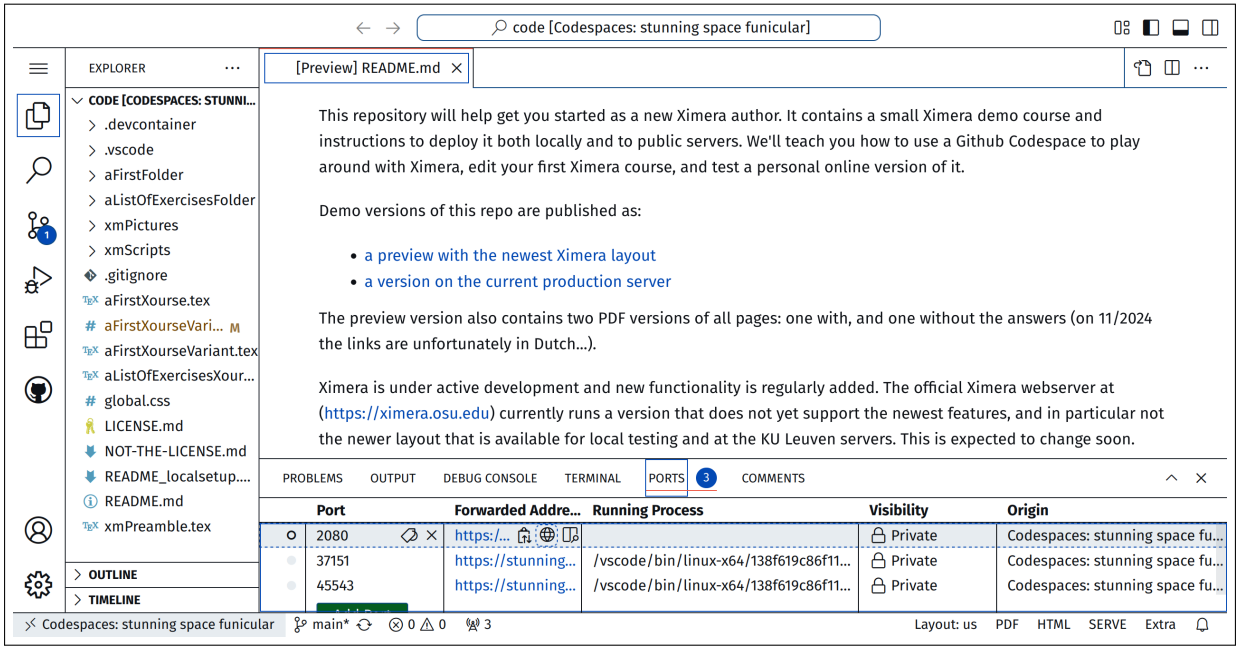
Email: ximera@math.osu.edu Website: https://github.com/ximeraProject/



Once the codespace is created, you will see something like what we have above. 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/

When 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]

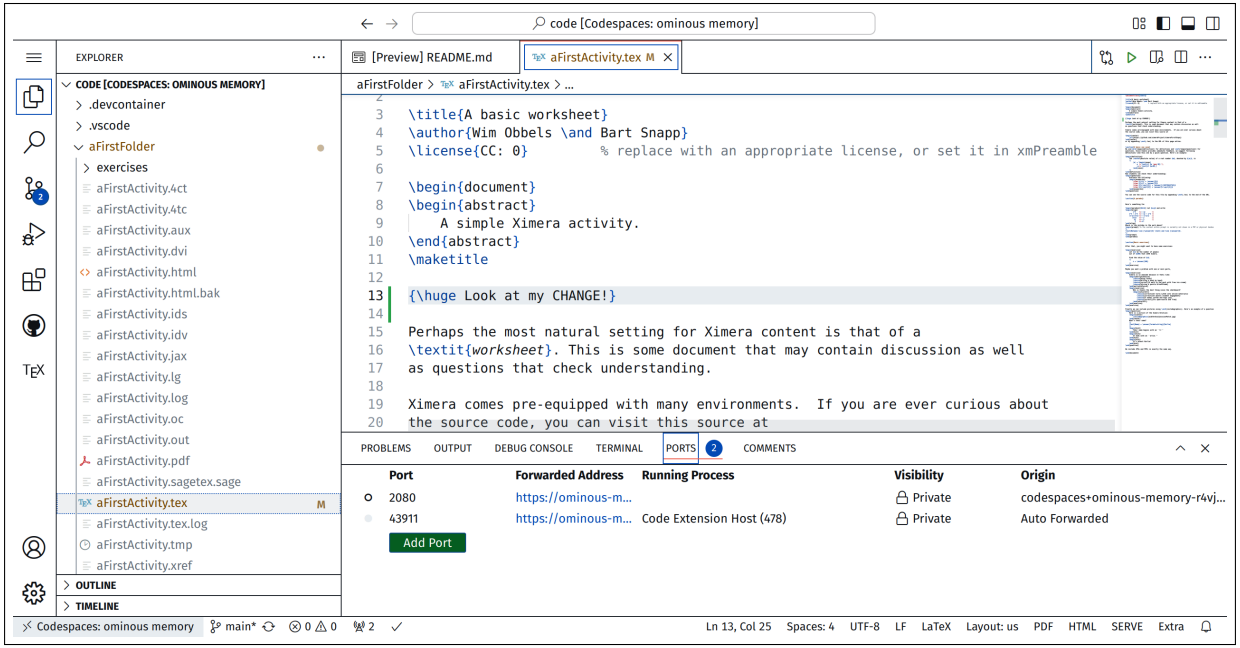
[2]See a preview with the newest Ximera layout at https://set.kuleuven.be/voorkennis/firststeps24/aFirstXourseVariant/aFirstFolder/aFirstActivityVariant
[3]See a version on the current production server at https://ximera.osu.edu/firststeps24/aFirstXourse/aFirstFolder/aFirstActivity

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.

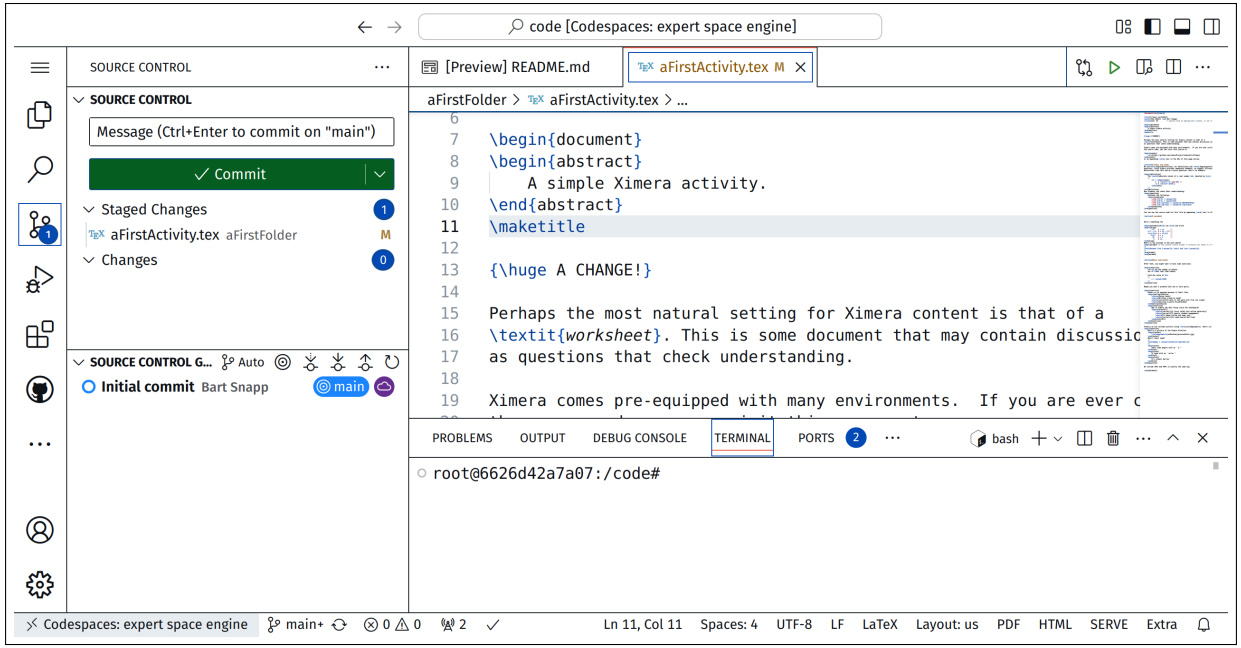
Once you've deployed our content in a GitHub codespace, you'll surely want to change it and make it your own.



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 document `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 temporary codespace**, a virtual 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/



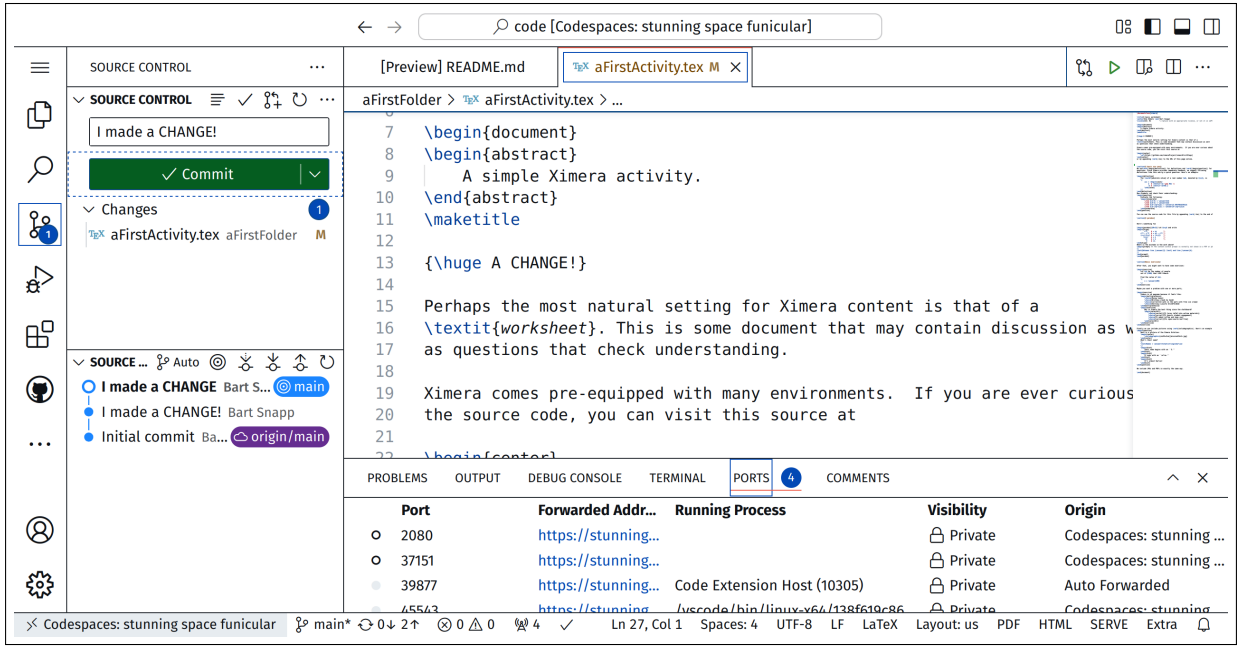
Remember Git is like a magical notebook that remembers every change you make to your project. It helps you go back in time if something breaks and lets you share your work with others. For this reason, it makes you do a little "dance" to ensure good code hygiene.

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

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

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/



Step 2: Committing Changes (Saving Your Work) After staging your files, you

(a) **type a message** directly above the green "Commit" button.

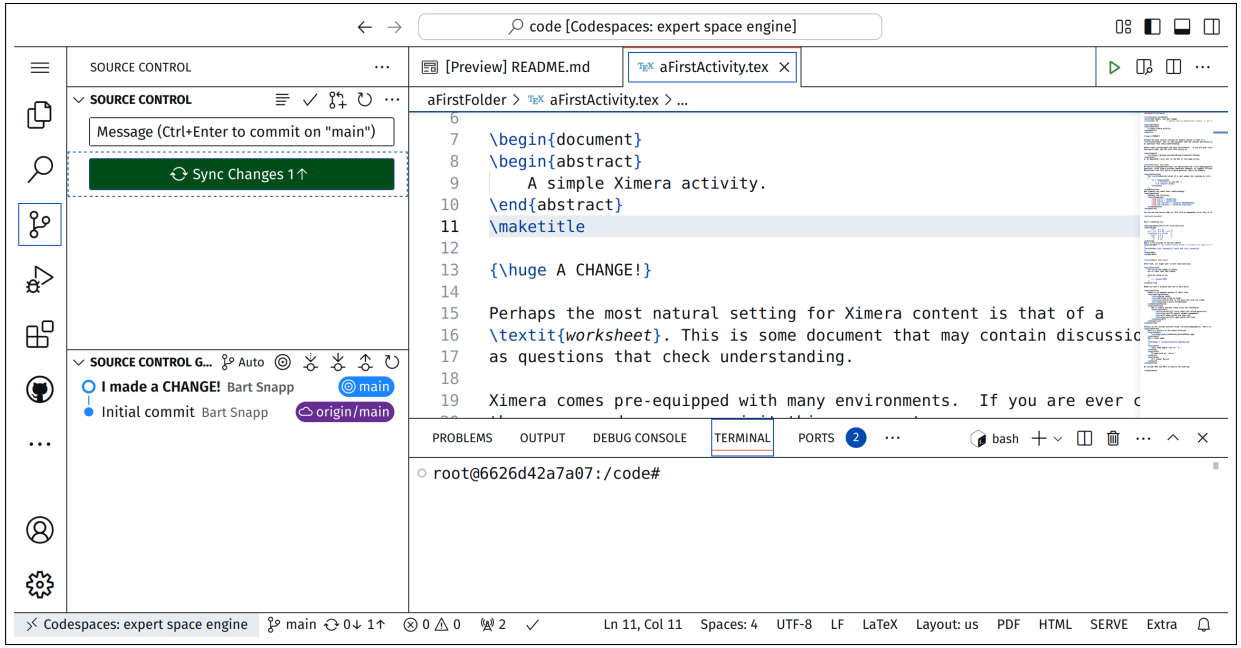
(b) Click "Commit."

This tells Git:

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

Git takes a snapshot of the staged files and saves them with your message. If you would ever need to undo your work from this point, this message will help guide your future-self.

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



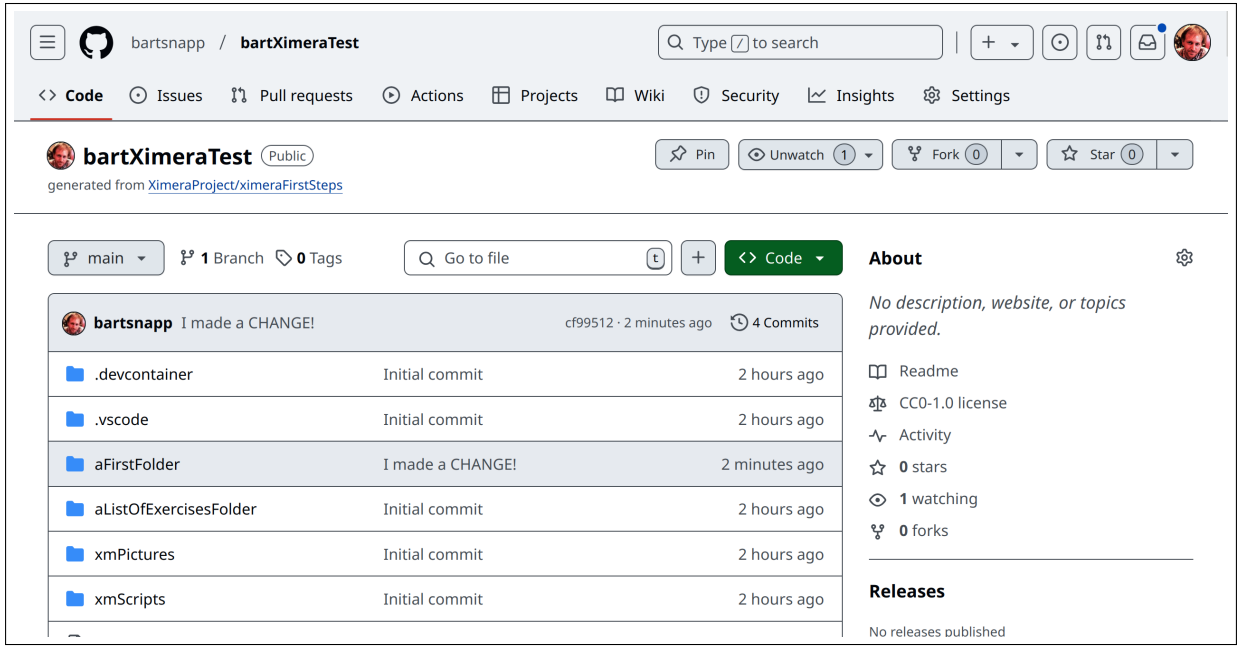
Step 3: Syncing with GitHub (Sending Your Work Online) When you click **"Commit and Sync"**, you're telling Git:

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

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. At this point, VS Code in your codespace will ask you if you periodically want to sync. You can click "Yes."

XIMERA FORM 9745 GIT COMMIT 97ca259eb9f56cb6de2218559b36c055dca3fc36 PREVIOUS EDITION IS OBSOLETE.

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.

The Git Workflow For a new author, this can be overwhelming. Here are the steps once again.

(a) **Stage:** Use the "+" button to pick what you want to stage for commit (save).

```
git add FILE-NAME-1 FILE-NAME-2  # You can list multiple files or
git add -u                       # Add all modified files
```

(b) **Commit:** Write a message and click "commit" to save your changes.

```
git commit -m 'A GOOD MESSAGE'
```

(c) **Sync:** Send your changes to GitHub and get updates from your team.

```
git pull && git push
```

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

Checking Your Work After syncing, go to:

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

You should see your changes there!

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

2 Getting work done

2.1 GitHub setup

How to create a new Ximera project.

All Ximera files must be hosted in a git repository. As an author, you have some choices as to how you proceed.

Starting fresh is easy. We have two repositories potential authors can use as a template:

- https://github/ximeraProject/ximeraFirstSteps for a template with multiple files that you can edit.
- https://github/ximeraProject/ximeraNewProject for a more minimal template.

If you go to either of these repositories when you are logged in to GitHub, you will have the option to make a new repository using the green "Use this template" button. Once the template is made, you are ready to go!

Deploy an existing repository from a codespace requires some extra files. Copy the following files from either ximeraFirstSteps or ximeraNewProject into the repository you wish to deploy.

.devcontainer/ (a folder you need if you want to deploy from a GitHub codespace)

xmScripts/ This is a folder that contains our Ximera scripts.

.vscode/ This is a folder that contains our VS Code settings, including the buttons that allow Ximera compilation.

.xmKeyFile This is a 'dummy' gpg-key file. If you want to deploy from a GitHub codespace and preview your work, you need this file.

.gitignore If you already have a .gitignore file, we suggest you **replace yours with ours.**

Commit your changes to GitHub and view your repository online via a web browser to ensure that the files were added. You can further check your work by starting a codespace in your repository.

Deploy an existing repository from your machine is for experienced users who can install software on their own computer. In particular, you will need to install:

Docker is a development utility allow you to easily run self-contained applications on your computer. You must install and start Docker before you can deploy. You can just let it run in the background.

VS Code Desktop is a popular text-editor. It is has strong LaTeX, git, and Docker integration. It provides a UNIX-like command line interface on Windows machines via Microsoft's WSL (Windows Subsystem for Linux).

Once these applications are installed, a typical workflow will be:

(a) Open Docker and minimize the window.

(b) Open Visual Studio Code, do File → Open Folder, and select the folder of your git repository.

(c) To open files, do Ctrl-p and start typing file names. Any file committed to your git repository will be found, and files in .gitignore, will not be shown.

(d) To run a special command (like search and replace) do Ctrl-Shift-p, and search for the command.

(e) To toggle a UNIX-like terminal, use Ctrl-~.

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

Now, copy the following files from either `ximeraFirstSteps` or `ximeraNewProject` into the repository you wish to deploy.

`.devcontainer/` (only needed if you want to additionally deploy from a GitHub codespace)

`xmScripts/` This is a folder that contains our Ximera scripts.

`.vscode/` This is a folder that contains our VS Code settings, including the buttons that allow Ximera compilation.

`.gitignore` If you already have a `.gitignore` file, we suggest you **replace yours with ours.**

Commit your changes to GitHub and view your repository online via a web browser to ensure that the files were added. Now, move the `.xmKeyFile` to your repository. **If you are deploying from your machine, do not commit the `.xmKeyFile` to the repository.**

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

2.2 Create an activity

How to set up an activity.

Ximera content consists of two distinct types of LaTeX files. Those with the `xourse` document class and those with the `ximera` document class. Files with the `xourse` document class glue-together Ximera content via the commands `\activity` and `\practice`. Ximera content is written using the `ximera` document class. The basic structure of a Ximera activity is as follows:

```
\documentclass{ximera}
\begin{document}
%%
%% Content goes here
%%
\end{document}
```

A Ximera activity may be a single problem, a list of exercises, a review of a concept, a section of a book—you are only limited by your imagination. Examples of individual Ximera activities can be found here:

Single problem See:

https://ximera.osu.edu/mooculus/limitLaws/exercises/exerciseList/limitLaws/exercises/limitLaw2

List of exercises See:

https://ximera.osu.edu/mooculus/limitLaws/exercises/exerciseList/limitLaws/exercises/limitLaw7

Review of concept See:

https://xronos.clas.ufl.edu/mac1140nowell/PrecalculusXourse/explorePolynomials/Practice/factoringGeneral-Practice1

Section of book See:

https://ximera.osu.edu/mooculus/calculus1/limitLaws/digInLimitLaws

You can see the source code for any of the activities above by appending `.tex` to the URL.

2.2.1 Tips for Ximera activities

We wish to write Ximera documents as simply as possible. Here are some tips.

Use basic LaTeX environments and commands like: `enumerate`, `itemize`, `description`, `align*`, `$...$`, `\[...\]`, and so on. Macros made of basic LaTeX commands are fine as long as you store them in `xmPreamble.tex` or another file on your input path. This helps keep the headers very clean.

Avoid manual formatting. At the core of the Ximera philosophy is the principle of separating content from deployment. This means that the intellectual substance—the ideas, explanations, and activities—should remain independent of the specific presentation or layout in which they are delivered. This separation ensures adaptability, longevity, and accessibility, enabling content to evolve without being constrained by the limitations of any particular delivery method.

With that said, our strategies for creating PDFs are to load a separate style file at the level of a `xourse` file. Further formatting can be achieved via the command `\pdfOnly`.

Special Ximera documents and folders are **automatically** loaded by all files in the repository (assuming they are no more than three folders deep).

`xmPreamble.tex` This is a place to put custom commands to help your document compile. It is not intended to be used for cosmetic changes to Ximera. For making pretty documents, see the section REFERENCE SECTION

`xmPictures/` This is a folder that you can place JPGs, PNGs, and PDFs, for inclusion into Ximera documents.

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

2.3 Create a xourse

We group Ximera activities into a collection.

Ximera documents can be *glued* together using the `xourse` document class. A `xourse` file is basically a list of other `ximera` files and even other `xourse` files. The `xourse` file for *A First Step in Ximera* can be found at the top level of the GitHub repository `ximeraFirstSteps` here

https://github.com/XimeraProject/ximeraFirstSteps/blob/main/aFirstStepInXimera.tex

The `xourse` documentclass specifies information such as the name of the document, the names of the authors, a description of the content, a license and the names of all Ximera LaTeX files comprising the whole document.

WARNING: All document and folder names used for Ximera must be web-safe! This means all document and folder names:

Must only use alphanumeric English characters Meaning: a,b, …, z, A,B, …, Z, 0,1, …, 9, and hyphen '-' and underscore '_' though the last two are discouraged.

Cannot use use any other characters, including spaces This means all Ximera documents and folders file names must be a single word.

This is not a limitation of Ximera, rather it is a rule that nearly all web-accessible documents must follow.

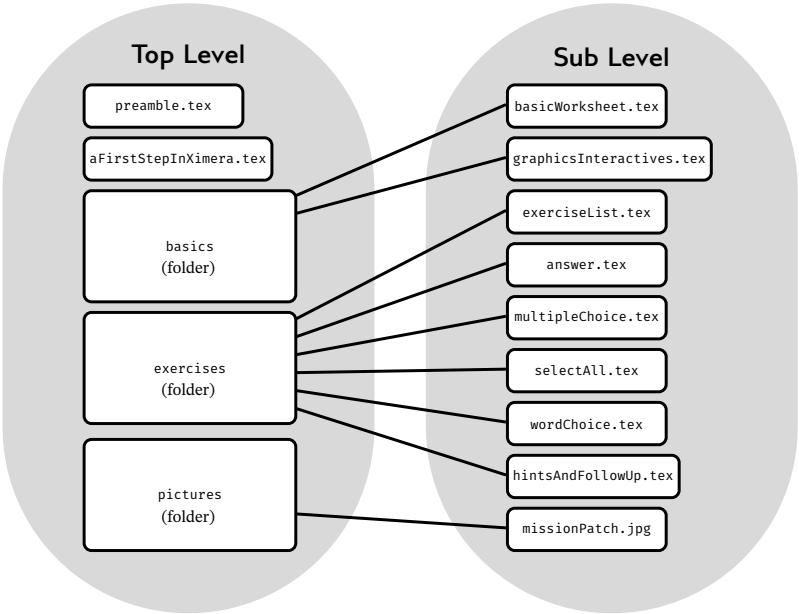
2.3.1 File structure

While one can write a Ximera document without the use of folders, this quickly turns into a mess that is difficult to understand and should only be used for the most basic Ximera content. To help others (including your future-self) work with larger projects, you should have a file structure that helps developers understand where content is stored. We recommend the following:

Group by concept by having all documents that are closely related in idea or scope in the same folder. If someone else wants to use your content, this will be a one-stop destination for them.

Descriptive file names will help you and others understand the structure of your repository. Give you documents descriptive file names like: `completeTheSquare.tex` or `derivativeRules.tex` rather than generic names like: ~~`chapter1.tex`~~, or ~~`math872ch2sec3.tex`~~. Authors find themselves reordering their content and generic names are useless for other users.

For the repository `ximeraFirstSteps` we have the following structure:



We've left some files out of this diagram; regardless, you should be able to goto

https://github.com/ximeraProject/ximeraFirstSteps

and witness this file structure. At the top level of the repository, we have the documents `preamble.tex` and `aFirstStepInXimera.tex` along with the folders `basics` and `exercises`. Inside `basics`, we have two activities and a JPG that is required by one of them. Inside `exercises` we have all the practice exercises.

A consistent and well thought out set up will allow you and others to easily understand and modify your content for years to come. Ideally a document's parent folder would contain everything that document needs to compile, except for perhaps the preamble, and we address this below.

WARNING: Every `*.tex` file in the repository with a `\documentclass` **must compile** for online deployment.

2.3.2 Including with `\activity` and `\practice`

Once you have some files and a basic directory structure, you can add them to the `xourse` document.

WARNING: The `xourse` document will only be easily accessible online if it contains a title, abstract, and `\maketitle` command.

If a `xourse` does not contain a title, abstract, and `\maketitle` it will still be deployed but you will need to know the path to the file. It will be something like:

```
https://ximera.osu.edu/YOUR-COURSE-NAME/PATH-TO-YOUR-FILE
```

It is important that there is no trailing '/' as

- https://ximera.osu.edu/moooculus is correct but
- ~~https://ximera.osu.edu/moooculus/~~ is not.

There are two different commands we use to add `ximera` documents to a `xourse` file:

`\activity` is for including Ximera documents that **include a title, abstract, and `\maketitle`**. They will be represented by their title online. This command is typically used for worksheets and sections of textbooks.

`\practice` is for including Ximera documents that **do not include** a title, abstract, and `\maketitle`. They will be represented by a number based on their order in `xourse` file. This command is used for lists of exercises and problem banks.

In `aFirstStepInXimera.tex` we write

```
\documentclass{xourse}
\input{preamble} %% Loads the graphics path
\author{Wim Obbels \and Bart Snapp}
\title{A First Step in Ximera}
\begin{document}
\begin{abstract}
    A simple collection of Ximera activities,
    to be deployed online.
\end{abstract}
\maketitle
\activity{basics/basicWorksheet}
\activity{basics/graphicsInteractives}
\practice{exercises/answer}
\practice{exercises/multipleChoice}
\practice{exercises/selectAll}
\practice{exercises/wordChoice}
\practice{exercises/hintsAndFollowUp}
\end{document}
```

Note how we give the paths to the exercises, it's the folder name, followed by the document name. The `.tex` can be left on if you like.

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

2.4 Preambles, input paths, and graphics paths

How to add extra packages and commands

The Ximera documentclass comes with many packages preloaded: `enumitem`, `titlesec`, `titletoc`, `titling`, `url`, `xcolor`, `tikz`, `pgfplots`, `fancyvrb`, `forloop`, `environ`, `amssymb`, `amsmath`, `amsthm`, `xifthen`, `multido`, `listings`, `xkeyval`, `comment`, `gettitlestring`, `nameref`, `epstopdf`, `hyperref`.

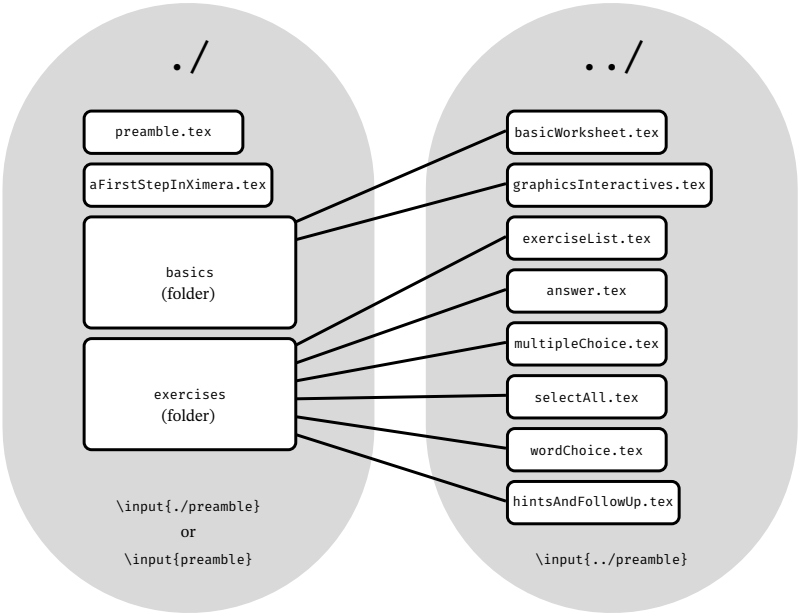
The documentclass also provides support for the following theorem-like environments: `algorithm`, `axiom`, `claim`, `conclusion`, `condition`, `conjecture`, `corollary`, `criterion`, `definition`, `example`, `explanation`, `exercise`, `exploration`, `fact`, `formula`, `hypothesis`, `idea`, `lemma`, `model`, `notation`, `observation`, `paradox`, `proof`, `problem`, `procedure`, `proposition`, `question`, `remark`, `solution`, `summary`, `template`, `theorem`, `warning`.

We can add functionality and generality, via user defined commands, to the document via a preamble, or macros, document. The purpose of a preamble is to ensure **all files compile consistently, it is not for cosmetic changes.** A typically preamble might contain things like:

```
\newcommand{\R}{\mathbb{R}}
\newcommand{\d}{\, d}
```

WARNING: Cosmetic changes to Ximera environments will result in unpredictable behavior online.

Since every `*.tex` file with a `\documentclass` must compile, the preamble file must be accessible by all files.



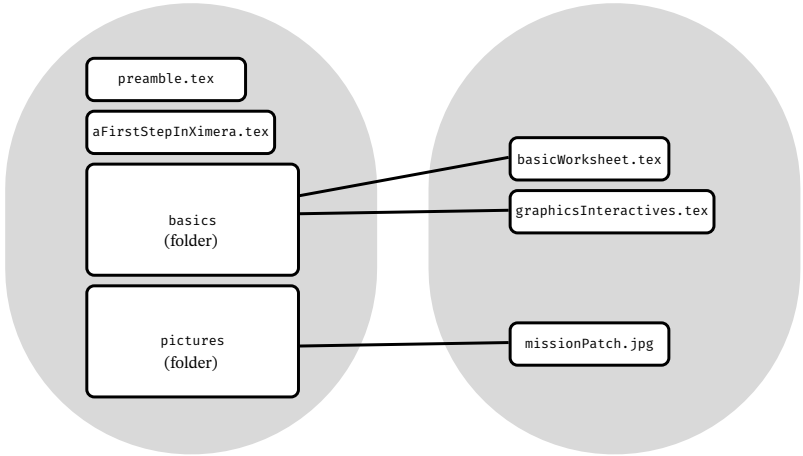
In this diagram, we attempt to show how one should modify the inputs of the preamble based on the directory structure.

Graphics paths help `\xourse` files find images. We used `\includegraphics` in `basicWorkSheet.tex` with code like:

```
\begin{center}
  \includegraphics[width=5cm]{missionPatch.jpg}
\end{center}
```

In this case, `basicWorkSheet.tex` and `missionPatch.jpg` are both in the folder `basics`. However, when this document is compiled via a `xourse` document, the LaTeX

compiler looks for the image *at the level of the* `aFirstStepInXimera.tex`.



Since there is no file `missionPatch.jpg` (which is on the right) at the level of the `xourse` document (on the left) the compilation will fail. To solve this problem we use the preamble to append the graphics path. If we add (and in reality we did!) this

```
\graphicspath{  %% When looking for images,
{./}            %% look here first,
{./pictures/}   %% then look for a "pictures" folder,
{../pictures/}  %% which may be a directory up.
}
```

to our preamble, then when compiling `aFirstStepInXimera.tex`, the compiler will know where to look.

Input paths allow you to load other documents into your files. This is a less common use case, but we'll mention it here. You can surely avoid the use of such files with an enhanced preamble. However, for this document, we needed to input various TikZ files to help us draw our diagrams. Hence we needed to add the location of these files to our input path. We did this in our preamble the following way:

```
%% Where to look for inputs
\makeatletter     %% make "@" a letter-character
\def\input@path{  %% When looking for files,
{./}              %% look first at your level
{./coverArt/}     %% then in this folder,
{./introduction/} %% then in this folder,
}
\makeatother      %% make "@" an other-character
```

With this said, modifying input paths is more of an advanced feature and one should be careful.

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

2.5 Graphics

Embed pictures in Ximera activities.

We've seen basic ways to include JPGs, PNGs, and PDFs using `\includegraphics`. However, this is not the preferred way to include graphics. Moreover, there are considerations for positioning of the graphics.

2.5.1 Positioning graphics

In LaTeX it is common to write images in the `figure` environment. We choose not to use this because `figure` 'floats' the images for 'optimal' page layout. We are not concerned with page layout. When working online, the page is essentially infinite in length. Moreover, the *consumers* of the content we create are *students*. Students are also unconcerned with awkward page layout. Students prefer to see the image exactly when it is mentioned. With this said, we suggest wrapping all images in either a `center` environment or an (Ximera-specific) `image` environment. The environment `image` centers and automatically scales the contents. If an author finds themselves printing to various page-sizes, `image` might be preferred. Moreover, `image` can be redefined globally to act identically to `center`, but `center` cannot be redefined.

If you use `center`, you would write something like:

```
\begin{center}
\includegraphics[width=5cm]{missionPatch.jpg}
\end{center}
```

If you use `image`, you would write something like:

```
\begin{image}
\includegraphics[width=5cm]{missionPatch.jpg}
\end{image}
```

The disadvantage of using `\includegraphics` is that you need to handle the paths to the images in some way. In the past we've added the files in the repository to the graphics path. However, we now suggest you use a global graphics path. The repository

https://github.com/XimeraProject/ximeraNewProject

Comes with the file `preamble.tex` with

```
\graphicspath{  %% When looking for images,
{./}            %% look here first,
{./pictures/}   %% then look for a "pictures" folder,
{../pictures/}  %% which may be a directory up.
}
```

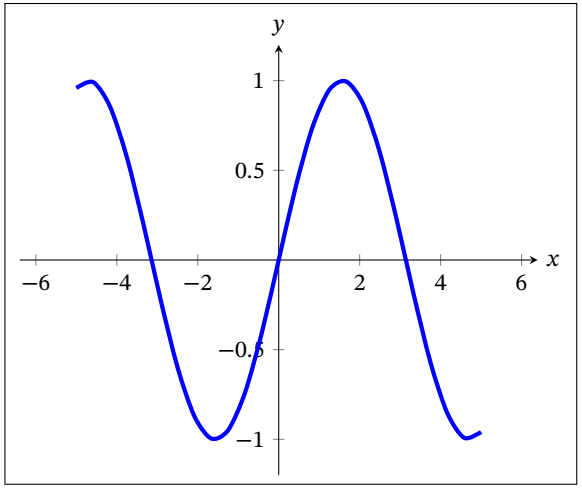
This means that any required `*.jpg`, `*.png`, and `*.pdf` can be found, assuming your file isn't nested too deep. Moreover, we've set up the `.gitignore` to not ignore `*.jpg`, `*.png`, and `*.pdf` when they are placed in the `pictures` directory. Finally, while this is currently the *easiest* method for adding graphics in Ximera, it might be better if all graphics were placed at the level of the source file, and then a soft link (using the terminal) could be made to the `pictures` directory. The advantage here being that if someone wants to borrow your material, everything they need is in the same folder.

With this said, TikZ is the preferred method for graphics because code, found in the Ximera LaTeX file, generates the image. No need to worry about graphics paths.

2.5.2 TikZ is the preferred method for graphics

Most of TikZ is supported, and these are rendered as PNGs online. For example the image below:

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



was generated via

```
\begin{image}
\begin{tikzpicture}
  \begin{axis}[
  xmin=-6.4,
  xmax=6.4,
  ymin=-1.2,
  ymax=1.2,
  axis lines=center,
  xlabel=$x$,
  ylabel=$y$,
  every axis y label/.style=
    {at=(current axis.above origin),anchor=south},
  every axis x label/.style=
    {at=(current axis.right of origin),anchor=west},
  ]
  \addplot [ultra thick, blue, smooth] {sin(deg(x))};
  \end{axis}
  \end{tikzpicture}
  \end{image}
```

When you use TikZ for graphics, you don't need to worry about `\graphicspath`.

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

2.6 Desmos and Geogebra

Embed compelling content in Ximera activities.

2.6.1 The graph command

The easiest way to include an interactive graph is to use the `\graph` command. Unfortunately, the `\graph` command doesn't draw a graph in the PDF, rather, it states (in words) that a graph is produced. That is,

```
\[
  \graph{x^2,x^3}
\]
```

produces

Graph of x^2, x^3

There are a number of options concerning the function being graphed:

```
\graph{x^2,x^3}                    %% just x^2 and x^3
                                   %%
\graph{x^2                         %%
\left\{ 1 \leq x \leq 10 \right\}} %% restricted domain
                                   %%
\graph{\sin(x) \left\{x<0\right\},  %%
2x \left\{ x>=0 \right\} }         %% piecewise
                                   %%
\graph{r=\theta}                   %% polar
```

While the code above modifies the function being graphed, there are also several options for the display of the graph.

Optional arguments for `\graph`

`xmin, ymin, xmax, ymax` These set the size of the viewing window with `\graph[xmin=-5,xmax=5,ymin=-5,ymax=5]{y=x^2}`.

`panel` Determines if the panel is shown with `\graph[panel]{y=x^2}`.

`xAxisLabel, yAxisLabel` Gives the axes labels with `\graph[xAxisLabel="time", yAxisLabel="distance"]{y=x^2}`.

`hideXAxis, hideYAxis` Hides the axes with `\graph[hideXAxis=true, hideYAxis=true]{x^2}`.

`hideXAxisNumbers=true, hideYAxisNumbers=true` Hides the tick marks on the axes with `\graph[hideXAxisNumbers=true, hideYAxisNumbers=true]{y=x^2}`.

`polar` Shows polar grid lines with `\graph[polar]{y=x^2}`.

Graph of $x^2\{1 \leq x \leq 10\}$Graph of $\sin(x)\{x < 0\}, 2x\{x >= 0\}$

2.6.2 Desmos, Desmos 3D, and GeoGebra

If you require further features from Desmos[4] , you can sign up for an account and include your worksheets using the syntax `\desmos{ID}{width}{height}`, where ID is the widget ID and width and height are the dimensions (in pixels) you want the embedded widget to have.

```
\begin{center}
\desmos{zwywds7med}{800}{600}
\end{center}
```

which renders as:

Desmos link: https://www.desmos.com/calculator/zwywds7med

[4]See Desmos at https://www.desmos.com/

The syntax for Desmos 3D is similar. Use `\desmosThreeD{ID}{width}{height}`, where ID is the widget ID and width and height are the dimensions (in pixels) you want the embedded widget to have.

```
\begin{center}
\desmosThreeD{bb4exrhrl3}{800}{600}
\end{center}
```

Seen here:

Desmos3D link: https://www.desmos.com/3d/bb4exrhrl3

You can also use GeoGebra[5] . Embed the widget using the syntax `\geogebra{ID}{width}{height}`, where ID is the widget ID and width and height are the dimensions (in pixels) you want the embedded widget to have.

```
\begin{center}
\geogebra{XC3FXUdJ}{800}{600}
\end{center}
```

Geogebra link: https://www.geogebra.org/m/XC3FXUdJ

While we cannot get data from these sorts of interactives directly, the clever author can ask questions that **use** the interactive to find a solution.

[5] See GeoGebra at https://www.geogebra.org/

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

2.7 Videos and other interactives

Embed compelling content in Ximera activities.

2.7.1 Videos

We can embed YouTube Videos using the syntax `\youtube{ID}`, where ID is the video ID found at the end of the YouTube link after "watch?v=".

```
\begin{center}
\youtube{FvgF95i0_lw}
\end{center}
```

which would embed the video into the page, like this:

YouTube link: https://www.youtube.com/watch?v=FvgF95i0_lw

WARNING: YouTube videos count toward the completion of the activity. As such, currently, students must watch the **entire video to earn full credit**.

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

2.8 Deploying Ximera documents

Deploying Ximera documents.

Finally to deploy,

(a) Make sure all source files are committed and pushed to the repository. A quick

```
git add -u && git commit -m "this is my change" && git push
```

may help. Also, you can check your personal GitHub page to ensure files are in the repository.

(b) Ensure Docker is running.

(c) Press *Bake* (typically on the bottom taskbar in VS Code).

(d) Press *Serve* (typically on the bottom taskbar in VS Code).

We should discuss this more once the discussion in the repository `ximeraFirstSteps` is steady.

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

2.9 Common issues

Suggestions for dealing with common issues.

If the Bake fails and you are working in a Codespace, then you can use `xake` directly to debug your code. You can use `pdflatex` to compile and use

```
pdflatex FILE-NAME.tex
xake -v compile FILE-NAME.tex
more FILE-NAME.tex.log
```

If you find a file that does not compile, you can try to compile the file directly while slowly moving `\end{document}` down through the document. By starting at the top, the file should compile, and then you should be able to locate the exact location of the bug.

An svg-viewer missing error message means that there was a compilation error that confused Xake. The offending file needs to be recompiled. Make a trivial change in the file, and delete all SVG files in the directory.

Sometimes Ximera materials look good to instructors and others, but not to some individual students.

Ximera refreshes the page when submitting can be caused by several issues, each with a different solution:

Problem: Poor internet connection

Solution: Refresh the page immediately before entering each answer.

Problem: Browser settings

Solution: Try using a different browser. Different students have reported luck with different browsers, including Edge, Firefox, and Chrome.

Problem: Antivirus Software

Solution: Some antivirus software (such as McAfee) are now blocking websockets, a technology Ximera relies on. Try turning off the websockets setting on the antivirus software.

Problem: VPN

Solution: If you are using a VPN, try turning it off before using Ximera.

unknown node type: parser error message. This appears sometimes rather than the actual rendered mathematics. This is due to a corruption in the cookies/cache for the browser, and the person seeing this error need to clear their cache and then reload the page to resolve the problem.

Difficulty accessing Ximera is often caused by browser settings. The easiest solution is usually to try using a different browser. Students seem to have the best luck with Chrome, Firefox, or Safari.

Math processing error This is caused by an error in the browser setting. Two things to try:

First possible solution: Clear browser cookies. Close the browser. Open the browser and log in to the LMS (if applicable). Go to the assignment.

Second possible solution: Find one of the pages this is happening, Put your mouse over the ‘math processing error’ Now, ‘right-click’ and select ‘accessibility’ If collapsible math is checked, uncheck it. Reload the page.