

---

# Getting started with Ximera

---

January 23, 2018

## Contents

What is Ximera? . . . . .	3
Introduction . . . . .	3
Basic workflow . . . . .	3
Trying it out . . . . .	4
Examples of Ximera courses . . . . .	4
CoCalc . . . . .	5
Using your CoCalc account . . . . .	5
Editing existing activity files . . . . .	6
Stuck? . . . . .	7
Installing Locally . . . . .	8
Install ximeraLaTeX . . . . .	8
Install xake . . . . .	8
Installing xake on Arch . . . . .	8
Installing xake on Ubuntu . . . . .	8
Installing xake on Mac . . . . .	9
Setting up a course . . . . .	11
Setting up a Ximera course . . . . .	11
Creating further activities . . . . .	14
Other ways to set up a Ximera repository . . . . .	15
Question and answer types . . . . .	16
Basic Function Answer Type . . . . .	16
Choice Answer Type . . . . .	17
Multiple Choice . . . . .	17
Select All . . . . .	17
free-response . . . . .	18
Options for the documentclass . . . . .	19

# What is Ximera?

*An introduction to the Ximera system.*

## Introduction

Ximera (pronounced [kɪmɪrə]) is an open-source software project that seeks to help course instructors create learning materials. The materials take the form of interactive web pages and high quality PDF documents. While these formats are very different, with Ximera an author is able to separate content from deployment, and write the source for both types of materials simultaneously.

## Basic workflow

An author writes content as a  $\text{\LaTeX}$  document. This produces a PDF handout that can be distributed to students. Next the author uploads the same  $\text{\LaTeX}$  file to github.com, a free web-based service providing a number of features to developers and authors. In turn github.com delivers the file to the Ximera interpreter, which posts the file on the web. The web page has essentially the same content as the handout. However, the web page typically has interactive features not possible in the handout due to the physical limitations of paper. For example, the web page might pose a question that if answered incorrectly, would offer hints or further questions to the student. This process is illustrated in the figure below.

---

Learning outcomes: Understand what Ximera is and how it can be used

Author(s):

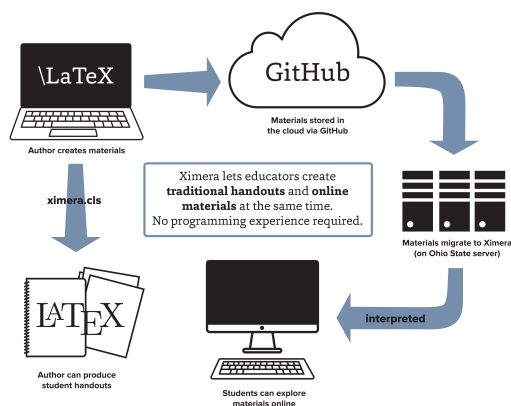
See Ximera at <http://ximera.osu.edu>

See github.com at <http://github.com>

See github.com at <http://github.com>

See Ximera at <http://ximera.osu.edu>

## What is Ximera?



## Trying it out

At this point you need to make a decision, do you want to try out Ximera on CoCalc (a cloud-based solution) or do you want to install on your own machine? If you just want to try this out with as little investment as possible, CoCalc may be the way to go. The downside is that if you become serious about developing in Ximera, you will probably want to use your own machine.

## Examples of Ximera courses

Perhaps the best way to learn is by looking at examples. We recommend looking at mooculus and the associated GitHub repository mooculus/calculus.

---

See mooculus at <https://ximera.osu.edu/mooculus>

See mooculus/calculus at <https://github.com/mooculus/calculus>

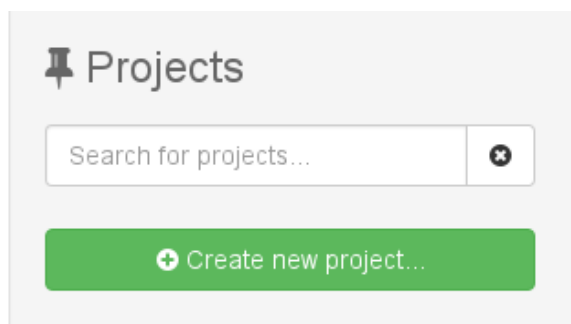
# CoCalc

*CoCalc is a way for a new author to publish.*

To publish your work to the internet, you will need a computer that can run our tools. The developers of Ximera use Arch Linux, but we believe that to be too difficult for the new user. Instead, we suggest you use CoCalc.

## Using your CoCalc account

Once you have a CoCalc account, you will need to start a new project



And when you create this new project, give it a Title, and a Description. This project **must** have internet access.

---

Learning outcomes:

Author(s):

See Arch Linux at <http://www.archlinux.org/>

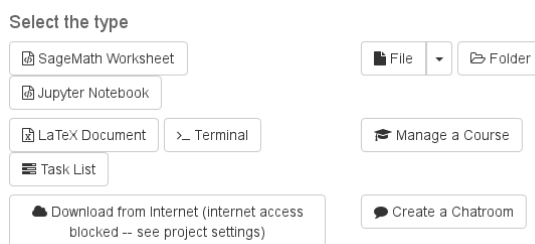
See CoCalc at <http://cocalc.com/>

Now you will need to add a terminal to your CoCalc account, click on

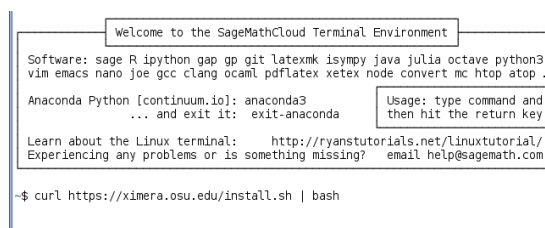
No Files Found

**+ Create or upload files...**

and select “Terminal.”



After a few seconds, a window will appear



Then run the two commands

```
curl -OL http://xandbox.github.io/cocalc/install.sh
source install.sh
```

This will install all the required tools (like the Ximera class file, the ‘xake’ build tool to compile and publish to the server, and mutool for rasterizing graphics).

## Editing existing activity files

- Click on “Files” to view the files that have been downloaded into your project.
- Choose the directory “xandbox” from the list of files. If you are working on your own computer, “xandbox” could be any directory.

- (c) Choose “first.tex”
- (d) Make some edits and click “Save.”
- (e) Click on the terminal in your project.
- (f) At this point you need to set-up a git repository. This is not difficult, but it is dependent on what you name your git repository. See this.
- (g) Now you will want to clone your repo into your xandbox directory.
- (h) You’ll need to pull your new repo. Move “first.tex” into your repo.
- (i) Type `git add first.tex` to stage the changes you’ve made to the first.tex file.
- (j) Type `git commit -m "My first edit"` to commit the staged changes.
- (k) Type `xake name` followed by a space and then a short lowercase name for your xandbox. This could be your first name, for instance. The name you choose must be globally unique, so be creative! For instance, `xake name turnloon` is what I will choose.
- (l) Type `xake bake` to compile your first.tex into an html file. If you run into errors, you can go back to your first.tex file and make additional edits.
- (m) Type `xake frost` to create a publication commit on top of your source commit.
- (n) Type `xake serve` to share your content with the world. For instance, my content will appear at <https://ximera.osu.edu/turnloon/first>

## Stuck?

If you are stuck, please contact us at [ximera@math.osu.edu](mailto:ximera@math.osu.edu) to get help.

---

See this at <https://help.github.com/articles/creating-a-new-repository/>

## Installing Locally

*Instructions for installing Ximera locally*

Ximera can also be run on your own machine rather than through CoCalc. Note that you will need to initialize your own git repos if you are not running through the CoCalc xandbox using `git init`.

### Install ximeraLaTeX

This can be done by going to XimeraProject/ximeraLatex GitHub page, and scrolling down to the directions for your particular platform.

### Install xake

Xake is Ximera’s version of “make” and it converts the L<sup>A</sup>T<sub>E</sub>X source into HTML.

#### Installing xake on Arch

`yaourt -S xake-git` should work assuming you use `yaourt`.

#### Installing xake on Ubuntu

```
sudo apt-get install golang-go
```

Corresponding `libgit2` and `git2go` installations are needed:

```
sudo apt-get install libgit2-dev
sudo apt-get install golang-git2go-dev
mkdir -p ~/go/src/github.com/ximeraproject
export GOPATH=$HOME/go
cd ~/go/src/github.com/ximeraproject
git clone https://github.com/XimeraProject/xake.git
cd ~/go/src/github.com/libgit2/git2go
```

---

Learning outcomes:

Author(s):

See Ximera at <http://ximera.osu.edu>

See XimeraProject/ximeraLatex GitHub page at <https://github.com/XimeraProject/ximeraLatex>



```
git checkout v24
cd ~/go/src/github.com/ximeraproject/xake
go get .
go build .
Use command ./xake or better yet, add xake to PATH by...
Also set your PATH so that ~/go/bin/ is in it: (edit .profile)
PATH=$PATH:~/go/bin
```

Reboot! Now you should be able to use the command xake

## Installing xake on Mac

- (a) First you need HomeBrew. The following command, typed in a terminal window

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master"
```

Should be all you need.

- (b) Then you need to install Go. The following command, typed in a new terminal window

```
brew install go --cross-compile-common
```

again should be all you need.

- (c) Install "pkg-config and "libgit2. This can be done via HomeBrew:

```
brew install pkg-config
brew install libgit2
brew install gpg
brew install mupdf
```

- (d) Now perform the following commands in terminal:

```
mkdir -p ~/go/src/github.com/ximeraproject
export GOPATH=$HOME/go
cd ~/go/src/github.com/ximeraproject
git clone https://github.com/XimeraProject/xake.git
brew install xake/xake
```

- (e) If you dont already have a GPG key, then do

```
gpg --gen-key
```

---

See HomeBrew at <https://brew.sh>

See Go at [http://www.golangbootcamp.com/book/get\\_setup](http://www.golangbootcamp.com/book/get_setup)

### *Installing Locally*

answer the questions, but **leave the passphrase blank** and copy the long hex string as YOUR-GPG-KEY-ID ( ABCD3562DBF9929292 or whatever)

```
gpg --keyserver hkps://ximera.osu.edu/ --send-key YOUR-GPG-KEY-ID  
PATH=$PATH:~/go/bin  
xake -k YOUR-GPG-KEY-ID name yourxakename
```

You will need to enter your passkey; prompt may be broken but try typing it and hitting enter anyway

- (f) You may get vague error messages here if you have not successfully set up your git name and email; you can do this with:

```
git config --global user.name "YOUR-NAME"  
git config --global user.email "YOUR-EMAIL"
```

## Setting up a course

*Instructions for setting up a course.*

### Setting up a Ximera course

Now that you can create a course. A Ximera course consists of a directory (`anExampleCourse`) containing a set of directories (`theFirstActivity`, `theSecondActivity`) that all contain  $\text{\LaTeX}$  files using the `ximera` document class. This directory also contains one or more  $\text{\LaTeX}$  files using the `xourse` document class, each of which is a launch page for the activities in the course(s). This section describes how to create a course.

- (a) In this example, we will create a directory called `anExampleCourse` that contains a directory called `theFirstActivity`. We will also create the `anExampleCourse.tex` file. Open a terminal session and type the following commands to do that.

```
mkdir anExampleCourse
cd anExampleCourse
mkdir theFirstActivity
touch anExampleCourse.tex
```

- (b) Next we will modify the `anExampleCourse.tex` file in the (`anExampleCourse`) directory. Click “Files,” click on the (`xandbox`) directory and click on the (`anExampleCourse`) directory, then click the file `anExampleCourse.tex`. Copy and paste the following in the left-hand side of the window and click “Save.” It may complain that you haven’t created the `theFirstActivity.tex` file yet, but that’s fine.

```
\documentclass{xourse}

\title{An example course}
%% This is the Name of your course
%% personalize it

\begin{document}
\begin{abstract} %% This describes your course
This is a Ximera activity explaining how to get started with Ximera for course instruct
```

---

Learning outcomes:  
 Author(s):  
 See Ximera at <http://ximera.osu.edu>

```

\end{abstract}
\maketitle

%% Here we have a listing of the activities.
\activity{theFirstActivity/theFirstActivity}

\end{document}

```

Now return to the terminal window, which should still be in the directory `theExampleCourse`. You need to add the file (and the directory containing it) to git and commit your changes, and also tell it who you are.

```

git add anExampleCourse.tex
git commit -m "this is my first course"

```

Every time you add or change files, you will need to run `git add` and `git commit -m "short description of changes"` to commit the changes to the server. The description of changes is necessary to commit, as is the `-m`. If you are changing multiple files, you can use `git add *`.

**Remark 1.** *In general the file with the `xourse` document class specifies course information such as the name of the course, a description of the course, and the names of all  $\text{\LaTeX}$  activity files comprising the course, in the order they should be presented to students. In addition to a name and a description, `anExampleCourse.tex` above specifies that there is one activity file `theFirstActivity.tex`, written with or without the extension `.tex`, and located in a directory called `theFirstActivity`. We will create this file and directory in the following step.*

*Generally courses should contain more than one activity. We recommend placing each activity in a directory of the same name. This facilitates sharing activities among collaborators and makes reusing existing activities easier. We also recommend that the directory and the  $\text{\LaTeX}$  file have exactly the same name as the title of the activity, with all spaces removed and all words other than the first word capitalized. So for example, if the title of the activity were `Plants native to Ohio` the  $\text{\LaTeX}$  file `plantsNativeToOhio.tex` would be located in a directory called `plantsNativeToOhio`.*

- (c) Now create your first activity. In your terminal, move to the `theFirstActivity` directory and create a file called `theFirstActivity.tex`. This can be accomplished by executing the commands below.

```

cd theFirstActivity
touch theFirstActivity.tex

```

- (d) Click “Files” again and navigate to the (`theFirstActivity`) folder, open `theFirstActivity.tex`, and paste in the following text. Then save the file.

```
\documentclass{ximera}
\title{The First Activity}
\begin{document}
\begin{abstract}
This activity deals with Ximera activities.
\end{abstract}
\maketitle
\end{document}
```

**Remark 2.** *An activity should be composed as a regular  $\text{\LaTeX}$  file in the document class `ximera`. It should contain the title of the activity and an abstract. These will both appear on the course website in the navigation area, so the abstract should be short. At this stage your activity contains a title and an abstract, but is otherwise blank.*

- (e) Now add this file to git and commit your changes. Back in the terminal, change to the directory `anExampleCourse` and execute the following commands

```
git add theFirstActivity.tex
git commit -m "Added first activity file"
```

- (f) Type `xake bake` to compile the tex documents, then `xake frost` to create a publication commit on top of your source commit. Finally type `xake serve` to share your content with the world. For instance, my content will appear at <https://ximera.osu.edu/turnloon/anExampleCourse>
- (g) This is a good point to add some further content in the form of a simple exercise. Update the file `theFirstActivity.tex` you created above so that it looks like the following.

```
\documentclass{ximera}
\title{The First Activity}
\begin{document}
\begin{abstract}
This activity deals with Ximera activities.
\end{abstract}
\maketitle
This activity is about creative work.
\begin{exercise}
Choose the best place to work on mathematics.
\begin{multipleChoice}
\choice{At the library}
```

```

\choice[correct]{At the caf\'e}
\choice{In your office}
\end{multipleChoice}
\end{exercise}
\end{document}

```

**Remark 3.** *The edits above insert a multiple-choice question into the `theFirstActivity` activity. See the `Question and answer types` activity later in this tutorial for more information on creating exercises.*

- (h) Change to the directory `anExampleCourse` and execute the following commands.

```

git add theFirstActivity.tex
git commit -m "Added an exercise"
git push
xake bake
xake frost
xake serve

```

If everything went well, you should see a URL printed on the terminal. If not, see the `Troubleshooting` activity in this tutorial or send your questions to `ximera@math.osu.edu`.

**Remark 4.** *The commands above inform `git` that changes have been made to your repository and communicates them to `github.com` which in turn communicates them to `ximera.osu.edu`. You should execute similar commands whenever you change files in your repository.*

## Creating further activities

From here you can create further Ximera activities as in step item (d). You should issue a `git add` command after creating a new file or directory and a `git commit` command followed by a `git push` command periodically to transmit your most recent changes to `github.com`. You should also add the name of your activity file to the `xourse` file in the position relative to other activities where you want the activity to appear. Observe however that once the filename appears in the `xourse` file the corresponding activity will appear to students. It might therefore be preferable to create a separate branch on GitHub until the activity is ready for students. During the editing phase you still view the activity by processing it with  $\text{\LaTeX}$  and inspecting the resulting PDF file, which might be helpful in any case for finding and correcting mistakes.

---

See `ximera@math.osu.edu` at `mailto:ximera@math.osu.edu`  
 See `git` at `http://git-scm.com`  
 See `github.com` at `http://github.com`  
 See `ximera.osu.edu` at `http://ximera.osu.edu`  
 See Ximera at `http://ximera.osu.edu`  
 See `github.com` at `http://github.com`

## Other ways to set up a Ximera repository

There are other ways to create a Ximera course. One possibility is to begin by creating the repository on github.com. Then instead of executing the commands to initialize the local copy of the repository, you could *clone* the copy on github.com using a `git clone` command. Alternately you could *fork* an existing repository, either your own or someone else's. See the git manual for more information about the `clone` and `fork` commands. Both possibilities above obviate step [item \(a\)](#) since cloning or forking a git repository creates a local directory and initializes it as a git repository.

---

See Ximera at <http://ximera.osu.edu>  
See github.com at <http://github.com>  
See github.com at <http://github.com>  
See git manual at <http://git-scm.com>  
See git at <http://git-scm.com>  
See git at <http://git-scm.com>

## Question and answer types

*This section explores the different types of exercises Ximera supports.*

Ximera supports many exercise types.

### Basic Function Answer Type

The basic way of including a answerable item in Ximera is to use the `\answer` command. The `\answer` **must** be inside of an environment.

```
\begin{question}
$3\times 2 = \answer{6}$
\end{question}
```

Will produce the question:

**Question 1**  $3 \times 2 =$

In addition to numerical answers, we also support elementary functions:

```
\begin{question}
$ \frac{\partial}{\partial x} x^2 \sin(y) = \answer{2x \sin(y)} $
\end{question}
```

Produces:

**Question 2**  $\frac{\partial}{\partial x} x^2 \sin(y) =$

**Remark 5.** *Under the hood, Ximera is parsing the user input, producing a function, and checking the user input function against “answer” at 100 different complex numbers, and seeing if the results are “reasonably” close to each other. We compare the complex extensions of these functions to circumvent domain issues.*

---

Learning outcomes:  
Author(s):



While **any** environment can contain the command `\answer`, there are four special environments: `question`, `exercise`, `problem`, `exploration`. Each of these environments is the same, except for the name. These environments interact with the optional arguments in the documentclass in useful ways. We'll discuss this later.

## Choice Answer Type

### Multiple Choice

```
\begin{question}
Which of the following functions has a graph which is a parabola?
\begin{multipleChoice}
\choice[correct]{$y=x^2+3x-3$}
\choice{$y = \frac{1}{x+2}$}
\choice{$y=3x+1$}
\end{multipleChoice}
\end{question}
```

Produces:

**Question 3** Which of the following functions has a graph which is a parabola?

**Multiple Choice:**

(a)  $y = x^2 + 3x - 3$  ✓

(b)  $y = \frac{1}{x+2}$

(c)  $y = 3x + 1$

---

**Remark 6.** Multiple choice answers appear in the order you type them.

### Select All

```
\begin{question}
Which of the following numbers are even?
\begin{selectAll}
\choice[correct]{$2$}
\choice{$1$}
```

```
\choice[correct]{$-4$}
\choice[correct]{$0$}
\end{selectAll}
\end{question}
```

Produces:

**Question 4** Which of the following numbers are even?

**Select All Correct Answers:**

- (a) 2 ✓
- (b) 1
- (c)  $-4$  ✓
- (d) 0 ✓

---

**Remark 7.** Select All answers appear in the order you type them.

## free-response

The free response environment gives students access to a L<sup>A</sup>T<sub>E</sub>X editor.

```
\begin{question}
Question goes here!
\begin{freeResponse}
This is the model solution %You don't actually need anything in between the begin and end 1
\end{freeResponse}
\end{question}
```

**Question 5** Question goes here!

**Free Response:** This is the model solution

---

**Remark 8.** Clicking on View model solution shows the user whatever you typed in the freeResponse environment.

## Options for the documentclass

*We describe options for the Ximera documentclass.*

There are a number of options for the document class, though their effects are only seen in the PDF:

**handout** The default behavior of the class is to display **all** content. This means that if any questions are asked, all answers are shown. Moreover, some content will only have a meaningful presentation when displayed online. When compiled without any options, this content will be shown too. This option will suppress such content and generate a reasonable printable “handout.”

**noauthor** By default, authors are listed at the bottom of the first page of a document. This option will suppress the listing of the authors.

**nooutcomes** By default, learning outcomes are listed at the bottom of the first page of a document. This option will suppress the listing of the learning outcomes.

**instructornotes** This option will turn on (and off) notes written for the instructor.

**noinstructornotes** This option will turn off (and on) notes written for the instructor.

**hints** When the **handout** options is used, hints are not shown. This option will make hints visible in handout mode.

**space** This option will place space after each problem-like environment (**exercise**, **question**, **problem**, and **exploration**)

**newpage** This option will start each problem-like environment (**exercise**, **question**, **problem**, and **exploration**) start on a new page.

**numbers** This option will number the titles of the activity. By default the activities are unnumbered.

**wordchoicegiven** This option will replace the choices shown by **wordChoice** with the correct choice. No indication of the **wordChoice** environment will be shown.

---

Learning outcomes:  
Author(s):