

ximera — Simultaneously write print and online interactive materials.*

Jim Fowler Jeramiah Hocutt Oscar Levin Jason Nowell
Wim Obbels Hans Parshall Bart Snapp

Released 2024/05/12

Abstract

“Ximera begins where \TeX ends.” The `ximera` class aids in the creation of hand-outs, worksheets, exercises, and sections of textbooks to be used either individually or “glued” together via a `xourse` file. All `ximera` documents can be deployed in an online interactive form via `xake`. See: [Ximera Project](#) and the source code on [GitHub](#).

1 Introduction

Ximera, pronounced “chimera,” (**X**imera: **I**nteractive, **M**athematics, **E**Resources, 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. The Ximera document class provides the following features:

Formatting for different domains The Ximera document class provides built-in support for formatting documents in both PDF and online formats, which can be a big time-saver for authors. Additionally, it allows for the simultaneous creation of solution manuals and teaching editions, which can be especially useful for educators.

Compiling individually or as a whole With the Ximera document class, authors can easily compile individual documents or an entire collection of documents. This flexibility can be helpful when making changes to specific documents without having to re-compile the entire collection. Moreover, this allows an author to share large portions of a text with another, with minimal changes.

Interactive content The Ximera document class allows for the inclusion of interactive content, such as answer boxes that are validated by a client-side computer algebra system. Additionally, it allows for the embedding of YouTube videos, Desmos graphs, and GeoGebra interactives.

All content displayed By default, the Ximera document class displays all content to the author. This means the author sees what the students see, along with answers and solutions, and links (that can be checked) to various interactive elements (when deployed, the interactive elements are truly embedded). This can be especially helpful for catching errors or inconsistencies in the content.

Online examples can be found at

<https://go.osu.edu/ximera-examples>

*This file describes version v1.5.1, last revised 2024/05/12.

2 ximera.cls

2.1 Options for the class

We start by listing the options for the ximera document class. Note, since the xourse class is based on the ximera class, all listed options are available there too.

```
1 \*classXimera\
```

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.”

```
2 \newif\ifhandout
3 \handoutfalse
4 \DeclareOption{handout}{\handouttrue}
```

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.

```
5 \newif\ifnoauthor
6 \noauthorfalse
7 \DeclareOption{noauthor}{\noauthortrue}
```

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.

```
8 \newif\ifnooutcomes
9 \nooutcomesfalse
10 \DeclareOption{nooutcomes}{\nooutcometrue}
```

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

```
11 \newif\ifinstructornotes
12 \instructornotesfalse
13 \DeclareOption{instructornotes}{\instructornotetrue}
```

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

```
14 \DeclareOption{noinstructornotes}{\instructornotetrue}
```

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

```
15 \newif\ifhints
16 \hintsfalse
17 \DeclareOption{hints}{\hintstrue}
```

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

```
18 \newif\ifnewpage
19 \newpagefalse
20 \DeclareOption{newpage}{\newpagetrue}
```

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

```
21 \newif\ifnumbers
22 \numbersfalse
23 \DeclareOption{numbers}{\numberstrue}
```

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

```
24 \newif\ifwordchoicegiven
25 \wordchoicegivenfalse
26 \DeclareOption{wordchoicegiven}{\wordchoicegiventrue}
27 \newif\iffirstinlinechoice% Support for other wordchoice command contents.
28 \firstinlinechoicetrue
```

```

29
30 \newif\ifxake
31 \xakefalse
32 \DeclareOption{xake}{\xakettrue}
33
34 \newif\iftikzexport
35 \tikzexportfalse
36 \DeclareOption{tikzexport}{%
37   \tikzexporttrue%
38   \handoutfalse%
39   \numbersfalse%
40   \newpagefalse%
41   \hintsfalse%
42   \nooutcomesfalse%
43 }
44
45 \DeclareOption*{%
46   \PassOptionsToClass{\CurrentOption}{article}%
47 }
48 \ProcessOptions\relax
49 \LoadClass{article}
50
51 \ifdefined\HCode
52   \xakettrue%
53   \tikzexporttrue%
54   \handoutfalse%
55   \numbersfalse%
56   \newpagefalse%
57   \hintsfalse%
58   \nooutcomesfalse%
59 \fi
60
61 </classXimera>
62 <*classXimera>

```

2.2 Loading packages

Since we want `\cancel` to work, we load it here to avoid polluting the `.jax` output.

```
63 \RequirePackage[makeroom]{cancel}
```

Quite a few packages are required by the document class. This is a list of required packages. As packages are added to this list, we should include a comment as to where they are being utilized. This will help keep this list from being redundant and/or outdated.

```

64 \RequirePackage[inline]{enumitem}
65 \RequirePackage[pagestyles]{titlesec}
66 \RequirePackage{titletoc}
67 \RequirePackage{titling}
68 \RequirePackage{url}
69 \RequirePackage[table]{xcolor}
70 \RequirePackage{tikz}
71 \RequirePackage{pgfplots}
72 \usepgfplotslibrary{groupplots}
73 \usetikzlibrary{calc}
74 \RequirePackage{fancyvrb}

```

Load `forloop` for the problem environment dynamic naming and building.

```
75 \RequirePackage{forloop}
```

Now we load even more packages.

```

76 \RequirePackage{environ}% Included to allow saving of environment contents. This does *not* p
77 \RequirePackage{amssymb}% Included to have access to math typeset.
78 \RequirePackage{amsmath}% Included to have access to math typeset.
79 \RequirePackage{amsthm}% Included to have access to math typeset.
80 \RequirePackage{xifthen}% http://ctan.org/pkg/xifthen

```

```

81 \RequirePackage{multido}% http://ctan.org/pkg/multido
82 \RequirePackage{listings} %% is this required???
83
84 \RequirePackage{xkeyval}
85
86 \RequirePackage{comment}
87 \end{classXimera}

```

Various packages must be loaded early to avoid polluting the .jax file.

```

88 \begin{classXimera}
89 \RequirePackage{getttitlestring}
90 \RequirePackage{nameref}
91 \RequirePackage{epstopdf}
92 \end{classXimera}

```

2.3 Page setup

We want non-indented spaced-out paragraphs.

```

93 \begin{classXimera}
94 \setlength{\parindent}{0pt}
95 \setlength{\parskip}{5pt}
96 \end{classXimera}

```

To avoid weird margins in 2-sided mode, change the margins.

```

97 \begin{classXimera}
98 \oddsidemargin 62pt
99 \evensidemargin 62pt
100 \textwidth 345pt
101 \headheight 14pt
102 \end{classXimera}

```

On the HTML side, there is more complicated page setup to perform.

```

103 \begin{cfgXimera}
104 \Preamble{xhtml,mathjax}
105
106 % We don't want to translate font suggestions with ugly wrappers like
107 % <span class="cmti-10"> for italic text
108 \NoFonts
109
110 % Don't output xml version tag
111 % \Configure{VERSION}{}
112
113 % Output HTML5 doctype instead of the default for HTML4
114 % \Configure{DOCTYPE}{\HCode{<!doctype html>\Hnewline}}
115
116 % Custom page opening
117 % \Configure{HTML}{\HCode{<html lang="en">\Hnewline}}{\HCode{\Hnewline</html>}}
118
119 % Reset <head>, aka delete all default boilerplate; alternatively set up new content
120 % \Configure{@HEAD}{\HCode{<meta name="generator" content="TeX4ht (http://www.cse.ohio-state.edu/~jim/TeX4ht/)>\Hnewline}}
121 \Configure{@HEAD}{\HCode{<meta name="ximera" content="version 2.0.1" />\Hnewline}}
122 \Configure{@HEAD}{\HCode{<link href="https://ximera.osu.edu/public/stylesheets/standalone.css" />\Hnewline}}
123 \Configure{@HEAD}{\HCode{<script type="text/javascript" async src="https://ximera.osu.edu/public/js/ximera.js">\Hnewline}}
124 \end{cfgXimera}

```

Disable certain ligatures in HTML.

```

125 \begin{htXimera}
126 \usepackage{microtype}
127 \DisableLigatures[f]{encoding=*}
128 \end{htXimera}

```

I am not sure what this does.

```

129 \begin{htXimera}
130 \NewEnviron{html}{\HCode{\BODY}}
131 \end{htXimera}

```

2.4 Structure

2.4.1 Macros

Makes everymath display style even when inline, could be optional.

```
132 \*classXimera>
133 \everymath{\displaystyle}
134 \*classXimera>
```

Ok not everything, we also need to configure “display style” limits.

```
135 \*classXimera>
136 \let\prelim\lim
137 \renewcommand{\lim}{\displaystyle\prelim}
138 \*classXimera>
```

2.4.2 Theorem and theorem-like environments

On the web, a theorem is emitted as a special <div>.

```
139 \*htXimera>
140 \newcommand{\ConfigureTheoremEnv}[1]{%
141 \renewenvironment{#1}[1][\refstepcounter{problem}%
142 \ifthenelse{\equal{##1}{}}{}{}%
143 \HCode{<span class="theorem-like-title">}##1\HCode{</span>}}%
144 }{}
145 \ConfigureEnv{#1}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div class=
146 }
147 \*htXimera>
148 \*classXimera>\theoremstyle{definition} % No italic (because this makes also text in TikZ italic)
```

The key is to make sure that the theorem environments are defined in a corresponding fashion on the web and on paper.

theorem	Theorem	
	149 *classXimera>	\newtheorem{theorem}{Theorem}
	150 *htXimera>	\ConfigureTheoremEnv{theorem}
algorithm	Algorithm	
	151 *classXimera>	\newtheorem{algorithm}{Algorithm}
	152 *htXimera>	\ConfigureTheoremEnv{algorithm}
axiom	Axiom	
	153 *classXimera>	\newtheorem{axiom}{Axiom}
	154 *htXimera>	\ConfigureTheoremEnv{axiom}
claim	Claim	
	155 *classXimera>	\newtheorem{claim}{Claim}
	156 *htXimera>	\ConfigureTheoremEnv{claim}
conclusion	Conclusion	
	157 *classXimera>	\newtheorem{conclusion}{Conclusion}
	158 *htXimera>	\ConfigureTheoremEnv{conclusion}
condition	Condition	
	159 *classXimera>	\newtheorem{condition}{Condition}
	160 *htXimera>	\ConfigureTheoremEnv{condition}
conjecture	Conjecture	
	161 *classXimera>	\newtheorem{conjecture}{Conjecture}
	162 *htXimera>	\ConfigureTheoremEnv{conjecture}
corollary	Corollary	
	163 *classXimera>	\newtheorem{corollary}{Corollary}
	164 *htXimera>	\ConfigureTheoremEnv{corollary}
criterion	Criterion	
	165 *classXimera>	\newtheorem{criterion}{Criterion}
	166 *htXimera>	\ConfigureTheoremEnv{criterion}

definition	Definition	
	167 <code>\classXimera</code>	<code>\newtheorem{definition}{Definition}</code>
	168 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{definition}</code>
example	Example	
	169 <code>\classXimera</code>	<code>\newtheorem{example}{Example}</code>
	170 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{example}</code>
explanation	Explanation	
	171 <code>\classXimera</code>	<code>\newtheorem*{explanation}{Explanation}</code>
	172 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{explanation}</code>
fact	Fact	
	173 <code>\classXimera</code>	<code>\newtheorem{fact}{Fact}</code>
	174 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{fact}</code>
lemma	Lemma	
	175 <code>\classXimera</code>	<code>\newtheorem{lemma}{Lemma}</code>
	176 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{lemma}</code>
formula	Formula	
	177 <code>\classXimera</code>	<code>\newtheorem{formula}{Formula}</code>
	178 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{formula}</code>
idea	Idea	
	179 <code>\classXimera</code>	<code>\newtheorem{idea}{Idea}</code>
	180 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{idea}</code>
notation	Notation	
	181 <code>\classXimera</code>	<code>\newtheorem{notation}{Notation}</code>
	182 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{notation}</code>
model	Model	
	183 <code>\classXimera</code>	<code>\newtheorem{model}{Model}</code>
	184 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{model}</code>
observation	Observation	
	185 <code>\classXimera</code>	<code>\newtheorem{observation}{Observation}</code>
	186 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{observation}</code>
proposition	Proposition	
	187 <code>\classXimera</code>	<code>\newtheorem{proposition}{Proposition}</code>
	188 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{proposition}</code>
paradox	Paradox	
	189 <code>\classXimera</code>	<code>\newtheorem{paradox}{Paradox}</code>
	190 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{paradox}</code>
procedure	Procedure	
	191 <code>\classXimera</code>	<code>\newtheorem{procedure}{Procedure}</code>
	192 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{procedure}</code>
remark	Remark	
	193 <code>\classXimera</code>	<code>\newtheorem{remark}{Remark}</code>
	194 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{remark}</code>
summary	Summary	
	195 <code>\classXimera</code>	<code>\newtheorem{summary}{Summary}</code>
	196 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{summary}</code>
template	Template	
	197 <code>\classXimera</code>	<code>\newtheorem{template}{Template}</code>
	198 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{template}</code>
warning	Warning	
	199 <code>\classXimera</code>	<code>\newtheorem{warning}{Warning}</code>
	200 <code>\htXimera</code>	<code>\ConfigureTheoremEnv{warning}</code>

2.4.3 Enumerate fixes

Make enumerate use a letter

```
201 \*classXimera>
202 \renewcommand{\theenumi}{\textup{(\alph{enumi})}}
203 \renewcommand{\labelenumi}{\theenumi}
204 \renewcommand{\theenumii}{\textup{(\roman{enumii})}}
205 \renewcommand{\labelenumii}{\theenumii}
206 \*classXimera>
```

2.4.4 Proofs

proof A mathematical proof environment.

```
207 \*classXimera>
208 \renewcommand{\qedsymbol}{\blacksquare}
209 \renewenvironment{proof}[1][\proofname]
210 {\begin{trivlist}\item[\hskip \labelsep \itshape \bfseries #1]{\hspace{2ex}}}
211 {\qed\end{trivlist}}
212 \*classXimera>
213 \*htXimera>
214 % Mmm, (why) do we want/need this ...?
215 \ConfigureTheoremEnv{proof}
216 \ConfigureEnv{proof}{\ifvmode\IgnorePar\fi\EndP\HCode{<div class="proof">}}
217 \ConfigureList{trivlist}{\ifvmode\IgnorePar\fi\EndP}{\}{\}{}
218 {\ifvmode\IgnorePar\fi\EndP\HCode{</div>}}{\}{\}{}
219 \*htXimera>
```

2.4.5 Problem environments

These are problem environment decorations (these should be user invoked, not default). The decoration for these environments were inspired by <http://tex.stackexchange.com/questions/11098/nice-formatting-for-theorems>

```
220 \*classXimera>
```

latexProblemContent Added for those that want to use UF problems without using the problem filter code. This command is renewed into something meaningful in the 'ProblemSelector.sty'.

```
221 \providecommand{\latexProblemContent}[1]{#1}
222 % Iterate count for problem counts.
223 \Make@Counter{Iteration@probCnt}

224 \newcommand{\hang}{% top theorem decoration
225   \begingroup%
226   \setlength{\unitlength}{.005\linewidth}% \linewidth/200
227   \begin{picture}(0,0)(1.5,0)%
228     \linethickness{1pt} \color{black!50}%
229     \put(-3,2){\line(1,0){206}}% Top line
230     \multido{\iA=2+-1,\iB=50+-10}{5}{% Top hangs
231       \color{black!\iB}%
232       \put(-3,\iA){\line(0,-1){1}}% Top left hang
233       \put(203,\iA){\line(0,-1){1}}% Top right hang
234     }%
235   \end{picture}%
236   \endgroup%
237 }%

238 \newcommand{\hung}{% bottom theorem decoration
239   \nobreak
240   \begingroup%
241   \setlength{\unitlength}{.005\linewidth}% \linewidth/200
242   \begin{picture}(0,0)(1.5,0)%
243     \linethickness{1pt} \color{black!50}%
244     \put(60,0){\line(1,0){143}}% Bottom line
245     \multido{\iA=0+1,\iB=50+-10}{5}{% Bottom hangs
246       \color{black!\iB}%
```

```

247      %\put(-3,\iA){\line(0,1){1}}% Bottom left hang
248      \put(203,\iA){\line(0,1){1}}% Bottom right hang
249      \put(\iB,0){\line(60,0){10}}% Left fade out
250    }%
251    \end{picture}%
252  \endgroup%
253}%

Configure environment configuration commands

The command \problemNumber contains all the format code to determine the number
(and the format of the number) for any of the problem environments.

254 \MakeCounter{problem}
255 \newcommand{\problemNumber}{%
256 % First we determine if we have a counter for this question depth level.
257 \ifcsname c@depth\Roman{problem@Depth}Count\endcsname% Check to see if counter exists
258 %If so, do nothing.
259 \else
260 %If not, create it.
261 \expandafter\newcounter{depth\Roman{problem@Depth}Count}
262 \expandafter\setcounter{depth\Roman{problem@Depth}Count}{0}
263 \fi
264
265 \expandafter\stepcounter{depth\Roman{problem@Depth}Count}
266 \arabic{depthICount}% The first problem depth, what use to be |\theproblem|.
267
268 \forloop{Iteration@probCnt}{2}{\arabic{Iteration@probCnt} < \numexpr \value{problem@Depth} +
269   \expandafter\arabic{depth\Roman{Iteration@probCnt}Count}}% Get the problem number of the
270 }
271 %\@ifpackageloaded{shuffle}{<true>}{<false>}% Check if Shuffle has been added. If so, add sp
272 %\ifhandout % Currently handout mode doesn't allow hints. Putting this code in place in case
273 % \theproblem
274 %\else
275 % \theproblem
276 %\fi
277 }
278
279
280 %%%% Configure various problem environment commands
281 \Make@Counter{problem@Depth}
282
283
284
285 %%%% Configure environments start content
286
287 \newcommand{\problemEnvironmentStart}[2]{%
288 % This takes in 2 arguments.
289 % The first is optional and is the old optional argument from existing environments.
290 % This is passed down to the associated problem environment name in case you want a global v
291 % The second argument is mandatory and is the name of the 'problem' environment,
292 % such as problem, question, exercise, etc.
293 % It then configures everything needed at the start of that environment.
294
295 \stepcounter{problem@Depth}% Started a problem, so we've sunk another problem layer.
296 \def\spaceatend{#1}%
297 \begin{trivlist}%
298 \item%
299   [%
300     \hskip\labelsep\sffamily\bfseries
301     #2 \problemNumber% Determine the correct number of the problem, and the format of that n
302   ]%
303 \slshape
304 }
305
306

```



```

307
308 %%%% Configure environments end content
309
310 \newcommand{\problemEnvironmentEnd}{%This configures all the end content for a problem.
311 %
312 % First we need to see if we've dropped fully out of a depth level,
313 % so we can reset that counter back to zero for the next time we enter that depth level.
314 \stepcounter{problem@Depth}
315 \ifcsname c@depth\Roman{problem@Depth}Count\endcsname
316 \expandafter\ifnum\expandafter\value{depth\Roman{problem@Depth}Count}>0
317 \expandafter\setcounter{depth\Roman{problem@Depth}Count}{0}
318 \fi
319 \fi
320
321 \addtocounter{problem@Depth}{-2}% Exited a problem so we've exited a problem layer. Need -2
322
323 \par\addvspace{.5ex}\nobreak\noindent\hung %% line at the bottom
324
325 \ifhandout
326 \ifnewpage
327 \newpage
328 \fi
329 \fi
330 \end{trivlist}
331 }
332
333
334
335 %%%% Now populate the old environment names
336 %
337 % Old environments were "problem", "exercise", "exploration", and "question".
338 % Note that you can add content to the start/end code on top of these base code pieces if you
339
340
341 \newenvironment{problem}[1][2in]%
342 {%Env start code
343 \problemEnvironmentStart{#1}{Problem}
344 }
345 {%Env end code
346 \problemEnvironmentEnd
347 }
348
349 \newenvironment{exercise}[1][2in]%
350 {%Env start code
351 \problemEnvironmentStart{#1}{Exercise}
352 }
353 {%Env end code
354 \problemEnvironmentEnd
355 }
356
357 \newenvironment{exploration}[1][2in]%
358 {%Env start code
359 \problemEnvironmentStart{#1}{Exploration}
360 }
361 {%Env end code
362 \problemEnvironmentEnd
363 }
364
365 \newenvironment{question}[1][2in]%
366 {%Env start code
367 \problemEnvironmentStart{#1}{Question}
368 }
369 {%Env end code

```

```

370 \problemEnvironmentEnd
371 }
372 \end{classXimera}

```

Use an “identification” counter to assign IDs to the various problem-related DOM elements

```

373 \begin{htXimera}
374 \newcounter{identification}
375 \setcounter{identification}{0}
376
377 \newcommand{\ConfigureQuestionEnv}[2]{%
378 % refstepcounter ensures that labels get updated within these environments
379 \renewenvironment{#1}{\refstepcounter{problem}}{}}%
380 \ConfigureEnv{#1}{\stepcounter{identification}\ifmode \IgnorePar\fi \EndP\HCode{<div role="
381 }
382
383 \ConfigureQuestionEnv{problem}{problem}
384 \ConfigureQuestionEnv{exercise}{exercise}
385 \ConfigureQuestionEnv{question}{question}
386 \ConfigureQuestionEnv{exploration}{exploration}
387
388 \ifhintAsExpandable\else
389 \ConfigureQuestionEnv{hint}{hint} % 2024: hint is no longer a 'question-environment'.
390 \fi
391 %%%\ConfigureQuestionEnv{shuffle}{shuffle}
392 \end{htXimera}

```

2.4.6 Hints

hint Hint environments can be embedded inside problems.

```

393 \begin{classXimera}

```

Create a counter that will track how deeply nested the current hint is

```

394 \newcounter{hintLevel}
395 \setcounter{hintLevel}{0}

```

Create an empty shell to renew

```

396 \newenvironment{hint}{}{}

```

Now we renew the environment as needed, this should allow support for any transition code that treats some parts as a “handout” and some parts as non-handout. renewing the environment on the fly is a bit hacky.

```

397 \renewenvironment{hint}
398 {
399 \ifhandout
400 \setbox0\vbox\bgroup
401 \else
402 \begin{trivlist}\item[\hskip \labelsep\small\slshape\bfseries Hint:\hspace{2ex}]
403 \small\slshape
404 \fi

```

Step up hint level to track the nested level of the hint. This will be used for problem numbering.

```

405 \stepcounter{hintLevel}
406 }
407 {
408 \ifhandout
409 \egroup\ignorespacesafterend
410 \else
411 \end{trivlist}
412 \fi

```

Detract from hint level counter to track hint nested level

```

413 \addtocounter{hintLevel}{-1}
414 }

```

```

415
416 \ifhints
417 \renewenvironment{hint}{
418 \begin{trivlist}\item[\hskip \labelsep\small\slshape\bfseries Hint:\hspace{2ex}]
419 \small\slshape}
420 {\end{trivlist}}
421 \fi
422
423 \end{classXimera}

```

2.4.7 Solution

solution The solution to a problem.

```

424 \begin{classXimera}
425 %% solution environment
426 \ifhandout % what follows is handout behavior
427 \newenvironment{solution}%
428     {%
429     \setbox0\vbox\bgroup
430     }
431     {%
432     \egroup
433     }
434 \else
435 \newenvironment{solution}%
436     {%
437     \begin{trivlist}
438     \item[\hskip \labelsep\bfseries Solution:\hspace{2ex}]
439     }
440     % %% line at the bottom}
441     {
442     \end{trivlist}
443     \par\addvspace{.5ex}\nobreak\noindent\hung
444     }
445 \fi
446
447
448
449 \end{classXimera}

```

2.4.8 Code listing environments

code A code answer environment You cannot use Environ with the fancyvrb/listings package if you want nested environments.

```

450 \begin{classXimera}
451 \DefineVerbatimEnvironment{code}{Verbatim}{numbers=left,frame=lines,label=Code,labelposition=
452 \end{classXimera}

```

python A python answer environment You cannot use Environ with the fancyvrb/listings package if you want nested environments

```

453 \begin{classXimera}
454 \DefineVerbatimEnvironment{python}{Verbatim}{numbers=left,frame=lines,label=Python,labelposition=
455 \end{classXimera}

```

javascriptCode A JavaScript answer environment Unfortunately the name javascript is already used for the actual, executed (!) JavaScript interactive. environments

```

456 \begin{classXimera}
457 \DefineVerbatimEnvironment{javascriptCode}{Verbatim}{numbers=left,frame=lines,label=JavaScript
458 \end{classXimera}
459 \begin{classXimera}
460 \renewenvironment{javascriptCode}{\NoFonts}{\EndNoFonts}
461 \ScriptEnv{javascriptCode}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<d
462 \end{classXimera}

```

On the web, translate verbatim and lstlisting blocks into <pre> elements.

```

463 \*cfgXimera>
464 \ConfigureEnv{verbatim}{\ifvmode\IgnorePar\fi\EndP\HCode{<pre>}}{\ifvmode\IgnorePar\fi\EndP\HCode{</pre>}}
465 \ConfigureEnv{lstlisting}{\ifvmode\IgnorePar\fi\EndP\HCode{<pre>}}{\ifvmode\IgnorePar\fi\EndP\HCode{</pre>}}
466 \*cfgXimera>

```

2.4.9 Dialogues

dialogue A dialogue between people.

```

467 \*classXimera>
468 \newenvironment{dialogue}{%
469   \renewcommand\descriptionlabel[1]{\hspace{\labelsep}\textbf{##1:}}
470   \begin{description}%
471 }{%
472   \end{description}%
473 }
474 \*classXimera>

```

On the web, the resulting <dl> should have an appropriate class set.

```

475 \*htXimera>
476 \renewenvironment{dialogue}{\begin{description}}{\end{description}}
477
478 \ConfigureList{dialogue}%
479   {\EndP\HCode{<dl \a:LRdir class="dialogue">}}%
480   \PushMacro\end:itm
481 \global\let\end:itm=\empty
482 {\PopMacro\end:itm \global\let\end:itm \end:itm}
483 \EndP\HCode{</dd></dl>}\ShowPar}
484 {\end:itm \global\def\end:itm{\EndP\Tg</dd>}\HCode{<dt
485   class="actor">}\bgroup \bf}
486 {\egroup\EndP\HCode{</dt><dd\Hnewline class="speech">}}
487 \*htXimera>

```

2.4.10 Instructor notes

```

488 \*classXimera>
489
490 %% instructor intro/instructor notes
491 %%
492 \ifhandout % what follows is handout behavior
493 \ifinstructornotes
494 \newenvironment{instructorIntro}%
495   {%
496   \begin{trivlist}
497   \item[\hspace{\labelsep}\bfseries Instructor Introduction:\hspace{2ex}]
498   }
499   % %% line at the bottom}
500   {
501   \end{trivlist}
502   \par\addvspace{.5ex}\nobreak\noindent\hung
503   }
504 \else
505 \newenvironment{instructorIntro}%
506   {%
507   \setbox0\vbox\bgroup
508   }
509   {%If this mysteriously starts breaking
510     % remove \ignorespacesafterend
511   \egroup\ignorespacesafterend
512   }
513   \fi
514 \else% for handout, so what follows is default
515 \ifinstructornotes
516 \newenvironment{instructorIntro}%

```

```

517         {%
518         \setbox0\vbox\bgroup
519         }
520 {%
521 \egroup
522 }
523         \else
524         \newenvironment{instructorIntro}%
525 {%
526 \begin{trivlist}
527 \item[\hskip \labelsep\bfseries Instructor Introduction:\hspace{2ex}]
528 }
529 % %% line at the bottom}
530 {
531 \end{trivlist}
532 \par\addvspace{.5ex}\nobreak\noindent\hung
533 }
534         \fi
535 \fi
536
537
538
539 %% instructorNotes environment
540 \ifhandout % what follows is handout behavior
541 \ifinstructornotes
542 \newenvironment{instructorNotes}%
543 {%
544 \begin{trivlist}
545 \item[\hskip \labelsep\bfseries Instructor Notes:\hspace{2ex}]
546 }
547 % %% line at the bottom}
548 {
549 \end{trivlist}
550 \par\addvspace{.5ex}\nobreak\noindent\hung
551 }
552 \else
553 \newenvironment{instructorNotes}%
554 {%
555 \setbox0\vbox\bgroup
556 }
557 {%
558 \egroup
559 }
560 }
561 \fi
562 \else% for handout, so what follows is default
563 \ifinstructornotes
564 \newenvironment{instructorNotes}%
565 {%
566 \setbox0\vbox\bgroup
567 }
568 {%
569 \egroup
570 }
571 \else
572 \newenvironment{instructorNotes}%
573 {%
574 \begin{trivlist}
575 \item[\hskip \labelsep\bfseries Instructor Notes:\hspace{2ex}]
576 }
577 % %% line at the bottom}
578 {
579 \end{trivlist}

```

```

580      \par\addvspace{.5ex}\nobreak\noindent\hung
581      }
582      \fi
583      \fi
584
585 \end{classXimera}

```

2.4.11 Only

prompt The prompt part for mathmode

```

586 \begin{classXimera}
587 \ifxake
588   \newenvironment{prompt}{}{}
589 \else
590 \ifhandout
591 \NewEnviron{prompt}{}
592 % Currently breaks when put in mathmode!
593 % \newenvironment{prompt}{\suppress}{\endsuppress}
594 \else
595 \newenvironment{prompt}
596   {\bgroup\color{gray!50!black}}
597   {\egroup}
598 \fi
599 \fi

```

onlineOnly Only display it online

```

600 \ifhandout
601 \NewEnviron{onlineOnly}{
602 \iftikzexport
603 \BODY
604 \else
605 \fi
606 }
607 \else
608 \newenvironment{onlineOnly}
609   {\bgroup\color{red!50!black}}
610   {\egroup}
611 \fi
612
613 \newcommand{\pdfOnly}[1]{\iftikzexport\else #1\fi}
614 \end{classXimera}

```

2.4.12 Foldable

The package `mdframed` is used to make pretty foldable, but the `amsthm/mdframed` conflict also messes up the `.jax` file so we don't load `mdframed` when performing the `xake` step. But even the below isn't enough to fix this.

```

615 \iftikzexport\else\RequirePackage[framemethod=TikZ]{mdframed}\fi

```

foldable Does it fold?

```

616 \begin{classXimera}
617
618 \colorlet{textColor}{black} % since textColor is referenced below
619 \colorlet{background}{white} % since background is referenced below
620
621 % The core environments. Find results in 4ht file.
622 %% pretty-foldable
623 \iftikzexport
624 \newenvironment{foldable}{%
625 }{%
626 }
627 \else
628 \renewmdenv[
629 % font=\upshape,

```

```

630 % outerlinewidth=3,
631 % topline=false,
632 % bottomline=false,
633 % leftline=true,
634 % rightline=false,
635 % leftmargin=0,
636 % innertopmargin=0pt,
637 % innerbottommargin=0pt,
638 % skipbelow=\baselineskip,
639 % linecolor=textColor!20!white,
640 % fontcolor=textColor,
641 % backgroundcolor=background
642 %]{foldable}%
643 %\fi
644
645 %% pretty-expandable
646 %\iftikzexport
647 %% Overwritten in .4ht, but probably also in accordion!
648 \newenvironment{expandable}[2]{%
649 }{%
650 }
651 %\else
652 %\newmdenv[
653 % font=\upshape,
654 % outerlinewidth=3,
655 % topline=false,
656 % bottomline=false,
657 % leftline=true,
658 % rightline=false,
659 % leftmargin=0,
660 % innertopmargin=0pt,
661 % innerbottommargin=0pt,
662 % skipbelow=\baselineskip,
663 % linecolor=black,
664 %]{expandable}%
665 %\fi
666
667 \newcommand{\unfoldable}[1]{#1}
668
669 \end{classXimera}

```

On the web, these foldable elements could be HTML5 details and summary.

```

670 \begin{htXimera}
671 \renewenvironment{foldable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div>}}{\HCode{</div>}}\IgnoreIndent
672
673 \renewenvironment{expandable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div>}}{\HCode{</div>}}\IgnoreIndent
674
675 \renewcommand{\unfoldable}[1]{\HCode{<span class="unfoldable">}#1\HCode{</span>}}
676
677 \end{htXimera}

```

2.4.13 Leashes

leash Put content inside a scrollable box.

```

679 \begin{classXimera}
680
681 \newenvironment{leash}[1]{%
682 }{%
683 }
684
685
686 \end{classXimera}

```

```

687 <*htXimera>
688 \renewenvironment{leash}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div style="overflow: auto; h
689 </htXimera>

```

2.5 Document metadata

2.5.1 Metadata

To encourage authors to include relevant parseable metadata in the preamble, we define some currently ignored commands.

\license In the preamble, use `\license` with an SPDX license expression.

```

690 <*classXimera>
691 \newcommand{\license}{\excludecomment}
692 </classXimera>

```

\acknowledgement In the preamble, use `\acknowledgement` to credit others who contributed to the intellectual content beside the author.

```

693 <*classXimera>
694 \newcommand{\acknowledgement}{\excludecomment}
695 </classXimera>

```

\tag In the preamble, a `\tag` provides a free-form taxonomy.

```

696 <*classXimera>
697 \renewcommand{\tag}{\excludecomment}
698 </classXimera>

```

On the HTML side, we mark the file as the appropriate kind of object—either activity or xourse.

```

699 <*htXourse>
700 % Mark this as a xourse file
701 \Configure{@HEAD}{\HCode{<meta name="description" content="xourse" />\Hnewline}}
702 </htXourse>

```

2.5.2 Abstract

abstract Every activity should include a short abstract.

```

703 <*classXimera>
704 \let\abstract\relax
705 \let\endabstract\relax
706 % Use of environ package, may want to find a better way.
707 \NewEnviron{abstract}{\protected@xdef\theabstract{\BODY}}
708 </classXimera>

```

The abstract has been stored in `\theabstract` and should be emitted as a div. The code below is required for the abstract to show online.

```

709 <*cfgXimera>
710 % contents of abstract is saved to a macro, but there are still tags for abstract,
711 % so we need to remove it
712 \ConfigureEnv{abstract}{\}{\}{\}{}
713 </cfgXimera>

```

2.5.3 Titles and authors

2.5.4 Authors

\author Activities have authors. Warn the user if no author is provided.

```

714 <*classXimera>
715 \let\@emptyauthor\@author
716 \def\author#1{\gdef\@author{#1}}
717 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}
718 </classXimera>

```


Include author name in meta tags

```
719 \*htXimera>
720 \Configure{@HEAD}{\HCode{<meta name="author" content="}\@author\HCode{" />\Hnewline}}
721 \</htXimera>
```

The `\and` command would emit tabular environments which really should not appear in a meta tag.

```
722 \htXimera | classXimera)\def\and{and }
```

2.5.5 Title

`\title` Activities have titles.

```
723 \*classXimera>
724 \let\title\relax
725 \newcommand{\title}[1] [] {{\protected@xdef\@pretitle{#1}}\protected@xdef\@title{
726
727 \title{
728
729 \newcounter{titlenumber}
730 \renewcommand{\thetitlenumber}{\arabic{titlenumber}}
731 %\renewcommand{\thesection}{\arabic{titlenumber}} %% Makes section numbers work
732 \setcounter{titlenumber}{0}
733
734 \newpagestyle{main}{
735 \sethead[\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}] [] [] % even
736 {}{}{\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}} % odd
737 \setfoot[\thepage] [] [] % even
738 {}{}{\thepage} % odd
739 }
740 \pagestyle{main}
```

`\maketitle` In a ximera document, redefine `\maketitle` and put them in a table of contents. The `\phantomsection` is to fix the hrefs.

```
741 \renewcommand\maketitle{%
742   \addtocounter{titlenumber}{1}%
743   {\flushleft\large\bfseries \@pretitle\par\vspace{-1em}}
744   {\flushleft\LARGE\bfseries {\ifnumbers\thetitlenumber\fi}{\ifnumbers\hspace{1em}\else\hspa
745   \phantomsection%
746   \ifnumbers\addcontentsline{toc}{section}{\thetitlenumber~\@title}\else\addcontentsline{toc}
747   \vskip .6em\noindent\textit{theabstract}\setcounter{problem}{0}\setcounter{section}{0}\setco
748   \ifnooutcomes\else\let\thefootnote\relax\footnote{Learning outcomes: \theoutcomes}\fi
749   \ifnoauthor\else\let\thefootnote\relax\footnote{Author(s):~\@author}\fi
750   \aftergroup\@afterindentfalse
751   \aftergroup\@afterheading}
752
753 \ifnumbers
754 \setcounter{secnumdepth}{2}
755 \renewcommand{\thesection}{\arabic{titlenumber}.\arabic{section}}
756 \renewcommand{\thesubsection}{\arabic{titlenumber}.\arabic{section}.\arabic{subsection}}
757 \else
758 \setcounter{secnumdepth}{-2}
759 \fi
760
761 \def\activitystyle{}
762 \newcounter{sectiontitlenumber}
763 \setcounter{secnumdepth}{2}
764 \setcounter{tocdepth}{2}
765 \newcommand\chapterstyle{%
766   \def\activitystyle{activity-chapter}
767   \def\maketitle{%
768     \addtocounter{titlenumber}{1}%
769     {\flushleft\small\sffamily\bfseries\@pretitle\par\vspace{-1.5em}}%
770     {\flushleft\LARGE\sffamily\bfseries\thetitlenumber\hspace{1em}\@title \par
771     {\vskip .6em\noindent\textit{theabstract}\setcounter{problem}{0}\setcounte
```

```

772             \par\vspace{2em}
773             \phantomsection\addcontentsline{toc}{section}{\textbf{\thetitlenumber\hspace{1em}\thechapter}}
774 }}
775
776
777 \newcommand\sectionstyle{%
778   \def\activitystyle{activity-section}
779   \def\maketitle{%
780     \addtocounter{section}{1}
781     \setcounter{sectiontitlenumber}{\value{section}}
782     {\flushleft\small\sffamily\bfseries\@pretitle\par\vspace{-1.5em}}%
783     {\flushleft\Large\sffamily\bfseries\thetitlenumber.\thesectiontitlenumber\hspace{1em}\thechapter}%
784     {\vskip .6em\noindent\textit{\theabstract}\setcounter{subsection}{0}}%
785     \par\vspace{2em}
786     \phantomsection\addcontentsline{toc}{section}{\thetitlenumber.\thesectiontitlenumber\hspace{1em}\thechapter}
787   \renewcommand\section{\@startsection{section}{2}{\z@}%
788                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
789                                     {1.5ex \@plus .2ex}%
790                                     {\normalfont\large\bfseries}}
791
792   \renewcommand\subsection{\@startsection{subsubsection}{3}{\z@}%
793                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
794                                     {1.5ex \@plus .2ex}%
795                                     {\normalfont\normalsize\bfseries}}
796
797 }}
798
799
800 \iftikzexport%% allows xake to handle \chapterstyle and \sectionstyle
801 \renewcommand\chapterstyle{\def\activitystyle{chapter}}
802 \renewcommand\sectionstyle{\def\activitystyle{section}}
803 \else
804 \fi
805
806 \end{classXimera}

```

Eliminate some formatting that we'll handle later with CSS

```

807 \begin{htXimera}
808 \renewcommand\maketitle{}
809 \end{htXimera}

```

2.5.6 Learning Outcomes

`\outcome` Specify a learning outcome, either at the level of a problem or an entire document in the preamble.

```

810 \begin{classXimera}
811 \def\theoutcomes{}
812
813 \ifdefined\HCode%
814   \newcommand{\outcome}[1]{}
815 \else%
816   \newwrite\outcomefile
817   \immediate\openout\outcomefile=\jobname.oc
818
819   \newcommand{\outcome}[1]{\edef\theoutcomes{\theoutcomes #1~}%
820   \immediate\write\outcomefile{\unexpanded{\outcome}{#1}}}
821 \fi%
822 \end{classXimera}

```

These can appear in either the preamble or in problem environments. with pdf_latex, we produce the .oc file which includes ALL the outcomes; in the tex4ht world, we just produce spans for the specific outcomes.

```

823 \begin{cfgXimera}
824 \renewcommand{\outcome}[1]{

```

```

825 \Configure{@HEAD}{\HCode{<meta name="learning-outcome" content="#1"/>\Hnewline}}
826 }
827 % Sometimes there are no outcomes at all
828 \IfFileExists{\jobname.oc}{\input{\jobname.oc}}{}
829
830 \renewcommand{\outcome}[1]{%
831 \HCode{<span class="learning-outcome">#1</span>}
832 }
833 </cfgXimera>

```

2.5.7 Labels and references

\label Labels and refs both generate anchors. A **\label** can be referenced from any file in the xourse.

```

834 <*htXimera>
835 \let\oldlabel\label
836 \renewcommand{\label}[1]{\oldlabel{#1}\HCode{<a class="ximera-label" id="#1"></a>}}
837 </htXimera>

```

\ref A **\ref** can connect one T_EX file to another if they are in the same xourse.

```

838 <*htXimera>
839 \renewcommand{\ref}[1]{\HCode{<a class="reference" href="#1">#1</a>}}
840 </htXimera>

```

2.6 Images

2.6.1 Images

image Place images inside an **image** environment. On paper, this centers the image. On the web, this provides additional benefits.

```

841 <*classXimera>
842 % Provide a default graphicspath
843 % (somewhat tricky: an activity can be included in a xourse in a wildly different path !)
844 % Suggested convention: put all images in i /pictures folder in the root of your project
845 \graphicspath{ %% When looking for images,
846 {./} %% look here first,
847 {./pictures/} %% then look for a pictures folder,
848 {../pictures/} %% which may be a directory up.
849 {../../pictures/} %% which may be a directory up.
850 {../../../pictures/} %% which may be a directory up.
851 }
852 %\newenvironment{image}[1][\begin{center}]{\end{center}}
853 \NewEnviron{image}[1][3in]{%
854 \begin{center}\resizebox{#1}{!}{\BODY}\end{center}% resize and center
855 }
856 </classXimera>

```

\alt Inside an **image** environment, **\alt** provides alt-text for assistive technology like screen-readers.

```

857 <*classXimera>
858 \newcommand{\alt}[1]{}
859 </classXimera>

```

The **image** environment doesn't actually work in tex4ht as defined with **NewEnviron**; so this **renewenvironment** is needed. **image-environment** also gets formatted in a well, and when the user clicks on the image, it zooms in.

```

860 <*htXimera>
861 \newcounter{imagealt}
862 \setcounter{imagealt}{0}
863 \renewenvironment{image}[1][\stepcounter{imagealt}%
864 \ifvmode \IgnorePar\fi \EndP%
865 \HCode{<div class="image-environment" role="img" aria-labelledby="image-alt-\arabic{imagealt}>}
866 }{\HCode{</div>}}
867 \renewcommand{\alt}[1]{\HCode{<div style="display: none;" id="image-alt-\arabic{imagealt}>#1}}

```

```

868 </htXimera>
869 <*cfgXimera>
870 %% Although we accept many formats, SVG is preferred on the web.
871 %% Since we have a different mechanism for producing |alt| text, we
872 %% want to ignore tex4ht's own method fo producing alt text.
873 %% 2024: is now in TeX4ht ...
874 % \DeclareGraphicsExtensions{.jpg,.png,.gif,.svg}
875 % \Configure{graphics*}
876 % {svg}{
877 %   {\Configure{Needs}{File: \Gin@base.svg}\Needs{}}
878 %   \Picture[]{\csname Gin@base\endcsname.svg \csname a:Gin-dim\endcsname}%
879 % }
880 </cfgXimera>

```

This is a hack to kill includegraphics commands in \documentclass{standalone} files

```

881 <*cfgXimera>
882 \ifcsname ifstandalone\endcsname
883   \ifstandalone
884     \renewcommand\includegraphics[2][]{ }
885   \fi
886 </cfgXimera>

```

PGF sometimes causes trouble, but we simply don't care in tex4ht mode.

```

887 <*htXimera>
888 \providecommand{\pgfsyspdfmark}[3]{}
889 </htXimera>

```

2.6.2 TikZ export

2024: We DON NOT ANYMORE generate SVGs and PNGs for any TikZ images, via the “externalize” feature of TikZ.

Previously TikZ didn't compile natively into the website because of how the xake bake compilation works. In order to make Tikz work, you need to get the tool mutool on the machine that is performing xake bake.

```

890 <*classXimera>
891 % everything skipped, assume TeX4ht does the job now
892 \ifdefined\reallyneverever
893
894 \ifdefined\HCode
895   \tikzexporttrue
896 \fi
897
898 \iftikzexport
899   \usetikzlibrary{external}
900
901   \ifdefined\HCode
902     % in htlatex, just include the svg files
903     \def\pgfsys@imagesuffixlist{.svg}
904
905     \tikzexternalize[prefix=./,mode=graphics if exists]
906   \else
907     % in pdflatex, actually generate the svg files
908     \tikzset{
909       /tikz/external/system call={
910         pdflatex \tikzexternalcheckshellescape
911         -halt-on-error -interaction=batchmode
912         -jobname "\image" "\PassOptionsToClass{tikzexport}{ximera}\texsource";
913         mutool draw -F svg \image.pdf > \image.svg ;      % mutool adds "1" to filename ???
914         mutool draw -o \image.svg \image.pdf ;
915         mutool draw -r 150 -c rgbalpha -o \image.png \image.pdf ;
916         ebb -x \image.png
917       }
918     }

```

```

919 \tikzexternalize[optimize=false,prefix=./]
920 \fi
921
922 \fi
923 \fi
924 \endclassXimera

```

2.6.3 XKCD

`\xkcd` Reference an XKCD cartoon.

```

925 \classXimera
926 \newcommand{\xkcd}[1]{#1}
927 \endclassXimera

```

On the web, this should be an image linked to the actual XKCD website.

```

928 \htXimera
929 \renewcommand{\xkcd}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{

```

2.8.3 Geogebra

\geogebra Geogebra command. Requires id, width, and height as arguments.

```

968 <*classXimera>
969 %Geogebra link
970 \newcommand{\geogebra}[3]{Geogebra link: \url{https://www.geogebra.org/m/#1}}
971 </classXimera>

Define keys for answer geogebra key=value pairs.

972 <*htXimera>
973 \define@key{geogebra}{rc}[true]{\def\geo@rc{#1}}
974 \define@key{geogebra}{sdz}[true]{\def\geo@sdz{#1}}
975 \define@key{geogebra}{smb}[true]{\def\geo@smb{#1}}
976 \define@key{geogebra}{stb}[true]{\def\geo@stb{#1}}
977 \define@key{geogebra}{stbh}[true]{\def\geo@stbh{#1}}
978 \define@key{geogebra}{ld}[true]{\def\geo@ld{#1}}
979 \define@key{geogebra}{sri}[true]{\def\geo@sri{#1}}
980 %set default key values
981 \setkeys{geogebra}{rc=false,sdz=false,smb=false,stb=false,stbh=false,ld=false,sri=false}
982 %command definition
983 \renewcommand{\geogebra}[4][ ]{%
984   \setkeys{geogebra}{#1}% Set new keys
985   \HCode{<iframe scrolling="no" src="https://www.geogebra.org/material/iframe/id/#2/width/#3
986 </htXimera>

```

2.8.4 Desmos

\desmos Desmos command. Requires id, width, and height as arguments.

```

987 <*classXimera>
988 \newcommand{\desmos}[3]{Desmos link: \url{https://www.desmos.com/calculator/#1}}
989 \newcommand{\desmosThreeD}[3]{Desmos3D link: \url{https://www.desmos.com/3d/#1}}
990 </classXimera>

991 <*htXimera>
992 \renewcommand{\desmos}[3]{\HCode{<iframe src="https://www.desmos.com/calculator/#1" width="1
993 \renewcommand{\desmosThreeD}[3]{\HCode{<iframe src="https://www.desmos.com/3d/#1" width="#2p
994 </htXimera>

```

2.8.5 Graphs

\graph An embedded graph (in math mode).

```

995 <*classXimera>
996 \newcommand{\graph}[2][ ]{\text{Graph of $#2$}}
997 </classXimera>

998 <*htXimera>
999 \renewcommand{\graph}[2][ ]{\HCode{<div class="graph" data-options="#1">#2\HCode{</div>}}
1000 </htXimera>

```

2.8.6 Video

\youtube Youtube command. Requires id.

```

1001 <*classXimera>

```

```

1002 \newcommand{\youtube}[1]{YouTube link: \url{https://www.youtube.com/watch?v=#1}}
1003 \end{classXimera}

1004 \begin{htXimera}
1005 %% \renewcommand{\youtube}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="video youtube-p
1006 % Fixes no-youtube-when-no-cookies-accepted. Class xmyoutube allows for css customization.
1007 \renewcommand{\youtube}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<iframe class="xmyoutube" src=
1008
1009 \end{htXimera}

```

Video commands are also emitted, slightly differently, when placed at top-level in a xourse file.

```

1010 \begin{htXourse}
1011 \renewcommand{\youtube}[1]{%
1012 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="youtube" href="https://www.youtube.com/watch?v=
1013 }
1014 \end{htXourse}

```

2.8.7 JavaScript

javascript Code inside a javascript environment is printed on paper, but executed on the web.

```

1015 \begin{classXimera}
1016 \DefineVerbatimEnvironment{javascript}{Verbatim}{numbers=left,frame=lines,label=JavaScript,la
1017 \end{classXimera}

1018 \begin{htXimera}
1019 % for programming javascript
1020 \renewenvironment{javascript}{\NoFonts}{\EndNoFonts}
1021 \ScriptEnv{javascript}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div c
1022 \end{htXimera}

```

\js Code inside a \js macro is evaluated and replaced with its value.

```

1023 \begin{classXimera}
1024 \def\js#1{\mbox{\texttt{\detokenize{#1}}}}
1025 \end{classXimera}

1026 \begin{htXimera}
1027 \def\js#1{\stepcounter{identification}\HCode{<span class="inline-javascript" id="javascript\
1028 \end{htXimera}

```

2.9 SageMath support

Load SageTeX if it exists.

```

1029 \begin{classXimera}
1030 \IfFileExists{sagetex.sty}{\RequirePackage{sagetex}}{}
1031 \end{classXimera}

```

sageCell Create an interactive SageMath widget.

```

1032 \begin{classXimera}
1033 \DefineVerbatimEnvironment{sageCell}{Verbatim}{numbers=left,frame=lines,label=SAGE,labelpositi
1034 \end{classXimera}

1035 \begin{htXimera}
1036 \renewenvironment{sageCell}{\NoFonts}{\EndNoFonts}
1037 \ScriptEnv{sageCell}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sage"><script type="text
1038 \end{htXimera}

```

sageOutput Execute SageMath code and output the result.

```

1039 \begin{classXimera}
1040 \DefineVerbatimEnvironment{sageOutput}{Verbatim}{numbers=left,frame=lines,label=SAGE-Output,
1041 \end{classXimera}

1042 \begin{htXimera}
1043 \renewenvironment{sageOutput}{\NoFonts}{\EndNoFonts}
1044 \ScriptEnv{sageOutput}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sageOutput"><script typ
1045 \end{htXimera}

```

sageSilent Execute SageMath code without outputting the result.

```

1046 \*htXimera>
1047 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1048 \ifdefined\sagesilent
1049 \renewenvironment{sagesilent}{\NoFonts}{\EndNoFonts}
1050 \fi
1051 \ScriptEnv{sagesilent}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="text/sagemath">}\Html
1052 </htXimera>

```

2.10 Answerables

2.10.1 Answers

\answer A math answer

```

1053 \*classXimera>
1054
1055 \ifdefined\HCode
1056 \newcommand{\recordvariable}[1]{}
1057 \else
1058 \newwrite\idfile
1059 \immediate\openout\idfile=\jobname.ids
1060 \newcommand{\recordvariable}[1]{\ifthenelse{\equal{#1}{}}{\}\{\immediate\write\idfile{var #1};}
1061 \fi

```

Determines if answer is shown in handout mode. when `given=true`, show answer in handout mode, show answer in “given box” outside handout mode. When `given=false`, do not show answer in handout mode, show answer outside handout mode

```
1062 \define@key{answer}{given}[true]{\def\ans@given{#1}}
```

Used for setting numeric answer tolerance for online student input.

```
1063 \define@key{answer}{tolerance}{\def\ans@tol{#1}}
```

Used to run dynamic js code on student provided answers. Note: currently pdf outputs the validator code itself.

```
1064 \define@key{answer}{validator}{}

```

Used for assigning a js ID to answer for dynamic code (eg validators).

```
1065 \define@key{answer}{id}{\def\ans@id{#1}}
```

Used to set anticipated input format; eg “string”.

```
1066 \define@key{answer}{format}{}

```

Used to hide the answer input box on the web.

```
1067 \define@key{answer}{onlinenoinput}[false]{}

```

Used to add a ‘show answer’ button to the answer blank.

```
1068 \define@key{answer}{onlineshowanswerbutton}[false]{}

```

Set default values for `\answer` command key=value pairs. Default values are `given = false`.

```
1069 \setkeys{answer}{id=,given=false,onlinenoinput=false,onlineshowanswerbutton=false}
```

Basic code for `\answer`.

```

1070
1071 % Options for handout
1072 \newcommand{\answerFormatLength}{2cm}
1073
1074 \newcommand{\answerFormatDots}[1]{\ldots\ldots}
1075 \newcommand{\answerFormatLine}[1]{\protect\rule{\answerFormatLength}{0.4pt}}
1076 \newcommand{\answerFormatFlexibleLine}[1]{\protect\rule{\widthof{#1$}*2}{0.4pt}}
1077 \newcommand{\answerFormatFlexibleBox}[1]{\fbox{\scalebox{2}{\phantom{#1$}}}}
1078
1079 % options for default (i.e with answers filled in)
1080 \newcommand{\answerFormatPlain}[1]{\ensuremath{#1}}
1081 \newcommand{\answerFormatBlue}[1]{\color{blue}\ensuremath{#1}}
1082 \newcommand{\answerFormatBoxed}[1]{\fbox{\ensuremath{#1}}}
1083 \newcommand{\answerFormatBoxedGiven}[1]{\underset{\scriptstyle\mathrm{given}}{\fbox{\ensuremath{#1}}}}

```



```

1084
1085 % defaults for handout and default mode, and for \answer[given]
1086 \let\handoutAnswerFormat\answerFormatDots
1087 \let\defaultAnswerFormat\answerFormatBlue
1088 \let\givenAnswerFormat\answerFormatBoxedGiven
1089
1090 \newcommand{\answer}[2] [] {%
1091 \ifmode%
1092 \setkeys{answer}{#1}%
1093 \recordvariable{\ans@id}
1094 \ifthenelse{\boolean{\ans@given}}
1095 {% Start then statement
1096 \ifhandout
1097 #2
1098 \else
1099 \givenAnswerFormat{#2} %% in case the argument helps formatting
1100 \fi
1101 }% End then statement
1102 {% Start else statement
1103 \ifhandout
1104 \handoutAnswerFormat{#2} %% in case the argument helps formatting
1105 \else% show answer in box outside handout mode
1106 \defaultAnswerFormat{#2} %% in case the argument helps formatting
1107 \fi
1108 }% End else statement
1109 \else%
1110 \GenericError{\space\space\space\space}% Throw an error based on... something? -- Jason
1111 {Attempt to use \@backslashchar answer outside of math mode}
1112 {See https://github.com/ximeraProject/ximeraLatex for explanation.}
1113 {Need to use either inline or display math.}%
1114 \fi
1115 }
1116 \end{classXimera}

```

On the HTML side, `\answer` emits spans—but it is usually just handled directly by MathJax.

```

1117 \begin{classXimera}
1118 \renewcommand{\answer}[2] [false]{\HCode{<span class="answer_respondable">}#2\HCode{</span>}}
1119
1120 \def\validator[#1]{\stepcounter{identification}\HCode{<div class="validator" id="validator\arabic{identification}">}#1\HCode{</div>}}
1121 \def\endvalidator{\HCode{</div>}}
1122
1123 \end{classXimera}

```

2.10.2 Multiple choice and the like

`multipleChoice` Multiple choice

```

1124 \begin{classXimera}
1125 % Jim: Originally this was \renewcommand{\theenumi}{\mathrm{\alph{enumi}}}$}
1126 % but that breaks tex4ht because mathmode can only be processed by mathjax.
1127 % so now I made this just italicized.

```

2.10.3 Options

```

1128 \define@key{choice}{value}[]{\def\choice@value{#1}}

```

This flags the answer as the correct answer

```

1129 \define@boolkey{choice}{correct}[true]{\def\choice@correct{#1}}

```

Use an ID to refer to the choice.

```

1130 \define@key{multipleChoice}{id}{\def\mc@id{#1}}

```

`\otherchoice` outputs the item if correct and nothing if incorrect.

```

1131 \define@key{otherchoice}{value}[]{\def\otherchoice@value{#1}}

```

```

1132 \define@boolkey{otherchoice}{correct}[true]{\def\otherchoice@correct{#1}}

```

Default key choices for multiple choice options. Default for choice pairs. Default: answers without the option "correct=true" is "incorrect".

```
1133 \setkeys{choice}{correct=false,value=}
```

Defaults for multipleChoice pairs. Default to no id? – Jason

```
1134 \setkeys{multipleChoice}{id=}
```

Defaults for otherchoice pairs. Default "otherchoice" to behave like "choice" for error checking.

```
1135 \setkeys{otherchoice}{correct=false,value=}
```

```
1136 \endclassXimera
```

2.10.4 Choices

\choice Like **\item** but for choice environments. **choice** command denotes a possible answer choice for the multiple choice question.

```
1137 \beginclassXimera
```

```
1138 \newcommand{\choice}[2] [] {%
```

```
1139 \setkeys{choice}{#1}%
```

```
1140 \item{#2}
```

```
1141 \ifthenelse{\boolean{\choice@correct}}{
```

```
1142   {% Begin then result
```

```
1143   \ifhandout% if it's a handout do nothing.
```

```
1144   \else% otherwise place a checkmark when you select the "correct choice"... maybe? -- Jason
```

```
1145     \,\checkmark\,\setkeys{choice}{correct=false}
```

```
1146   \fi
```

```
1147   }% End then result
```

```
1148   }% Begin/End else result.
```

```
1149 }
```

```
1150
```

```
1151 %Define an expandable version of choice Not really meant to be used outside this package (use
```

```
1152 % Is there a reason we can't just always use this as default? -- Jason
```

```
1153 \newcommand{\choiceEXP}[2] [] {%
```

```
1154 \expandafter\setkeys\expandafter{choice}{#1}%
```

```
1155 \item{#2}
```

```
1156 \ifthenelse{\boolean{\choice@correct}}{
```

```
1157   {% Begin then result
```

```
1158   \ifhandout
```

```
1159   \else
```

```
1160     \,\checkmark\,\setkeys{choice}{correct=false}
```

```
1161   \fi
```

```
1162 }% End then result
```

```
1163 }% Begin/End else result.
```

```
1164 } %% note all the {} are needed in case the choice has [] in it.
```

```
1165
```

```
1166 % \otherchoice is the \choice used in wordChoice command.
```

```
1167 \newcommand{\otherchoice}[2] [] {%
```

```
1168 \ignorespaces%
```

```
1169 \setkeys{otherchoice}{#1}%
```

```
1170 \ifthenelse{\boolean{\otherchoice@correct}}{%
```

```
1171   {% Start then result
```

```
1172 #2\ignorespaces\setkeys{otherchoice}{correct=false}\ignorespaces%
```

```
1173 }% End then result
```

```
1174 }% Start/End else result
```

```
1175 \ignorespaces%
```

```
1176 }%
```

```
1177 \newcommand{\inlinechoice}[2] [] {%
```

```
1178 \setkeys{choice}{#1}%
```

```
1179 \iffirstinlinechoice
```

```
1180 (\hspace{-.25em}
```

```
1181 \firstinlinechoicetrue
```

```
1182 \else
```

```
1183 /
```

```
1184 \fi
```

```
1185 #2
```

```

1186 \ifthenelse{\boolean{\choice@correct}}{%
1187 {% Start then result
1188 \ifhandout\else\checkmark\ignorespaces\setkeys{choice}{correct=false}\ignorespaces\fi%
1189 }% End then result
1190 }{% Start/End else result
1191 \hspace{-.25em}\ignorespaces%
1192 }
1193
1194 \end{classXimera}

```

On the HTML side, `\choice` emits `s`.

```

1195 \begin{htXimera}
1196 \newcounter{choiceId}
1197 \renewcommand{\choice}[2][ ]{%
1198 \setkeys{choice}{correct=false}%
1199 \setkeys{choice}{#1}%
1200 \stepcounter{choiceId}\IgnorePar%
1201 \HCode{<span class="choice }%
1202 \ifthenelse{\boolean{\choice@correct}}{\HCode{correct}}{}
1203 \HCode{" }
1204 \ifthenelse{\equal{\choice@value}{} }{\HCode{data-value="\choice@value" }}
1205 \HCode{id="choice\arabic{choiceId}">}%
1206 #2\HCode{</span>}}
1207 \let\inlinechoice\choice
1208 \end{htXimera}

```

2.10.5 Environment(s)

multipleChoice The environment `multipleChoice@` is for internal use only. Wrap `\choices` in a `multipleChoice` environment to make a multiple choice question.

```

1209 \begin{classXimera}
1210 \newenvironment{multipleChoice}[1][ ]
1211 {% Environment Start Code
1212 \setkeys{multipleChoice}{#1}%
1213 \recordvariable{\mc@id}%
1214 \begin{trivlist}
1215 \item[\hskip \labelsep\small\bfseries Multiple Choice:] \hfil
1216 \begin{enumerate}
1217 }% Note this means that \item has to be the first line after \begin{multipleChoice}.
1218 {% Environment End Code
1219 \end{enumerate}
1220 \end{trivlist}
1221 }
1222
1223 %multipleChoice@ is for internal use only! (used in wordChoice)
1224 %this is simply a wrapper for the sole showing (other)choice.
1225 \newenvironment{multipleChoice@}[1][ ]{}{}
1226 \end{classXimera}

```

On the web, you might also expect these to be "problem environments" but they aren't – they're *responsibles*. You might expect a `\setcounter{choiceId}{0}` here — that would be wrong, because then the generated IDs would no longer be unique.

```

1227 \begin{htXimera}
1228 \renewenvironment{multipleChoice}[1][ ]
1229 {\setkeys{multipleChoice}{#1}%
1230 \stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div class="multiple-choice" }
1231 \ifthenelse{\equal{\mc@id}{} }{\HCode{data-id="\mc@id" }}%
1232 \HCode{id="problem\arabic{identification}">}%
1233 }{\HCode{</div>}\IgnoreIndent}
1234 \ConfigureEnv{multipleChoice}{}{}{}
1235 \end{htXimera}

```

2.11 Word choice

`\wordChoice` An in-line version of `multipleChoice`: uses `enumitem` package note, it is coded as a single line to avoid unwanted spaces in “given” mode.

```

1236 \*classXimera>
1237 \newcommand{\wordChoice}[1]{%
1238 \let\choicetemp\choice% Assign a "choicetemp" command to duplicate choice.
1239 \ifwordchoicegiven% If wordchoice option is on, we need to juggle around some definitions.
1240 \let\choice\otherchoice%
1241 %\begin{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1242 #1
1243 %\end{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1244 \else% If it isn't the regular "choice" command should work.
1245 \let\choice\inlinechoice%
1246 \begin{multipleChoice@}%
1247 #1%
1248 \end{multipleChoice@}%
1249 \fi%
1250 \let\choice\choicetemp% Now that choicetmp has been manipulated to what we want, replace choi
1251 }%
1252
1253
1254 \*classXimera>

```

This is actually just word choice

```

1255 \*htXimera>
1256 \renewenvironment{multipleChoice@}{\refstepcounter{problem}}{}%
1257 \ConfigureEnv{multipleChoice@}{\stepcounter{identification}\IgnorePar\HCode{<span class="word
1258 \*htXimera>

```

2.12 Select all

`selectAll` A multiple-multiple choice question

```

1259 \*classXimera>
1260 \newenvironment{selectAll}[1]{}
1261 {\begin{trivlist}\item[\hskip \labelsep\small\bfseries Select All Correct Answers:]\hfil\begin{
1262 \end{enumerate}\end{trivlist}}
1263 \*classXimera>

```

In the future we need this to (optionally) be displayed in the problem, while the actual code lives in the solution. Here is how this could be implemented: Like the `title/maketitle` commands, the multiple-choice could be stored in `\themultiplechoice`, flip a boolean, and execute `\makemultiplechoice` at the `\end` of the problem. We should also make a command called `\showchoices` that will show choices in the handout.

On the web, `selectAll` is handled just like `multipleChoice`.

```

1264 \*htXimera>
1265 \renewenvironment{selectAll}{\refstepcounter{problem}}{}%
1266 \ConfigureEnv{selectAll}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div
1267 \*htXimera>

```

2.12.1 Free response

`freeResponse` A freeform input box.

```

1268 \*classXimera>
1269 \newboolean{given} %% required for freeResponse
1270 \setboolean{given}{true} %% could be replaced by a key=value pair later if needed
1271
1272 \ifhandout
1273 \newenvironment{freeResponse}[1][false]%
1274 {%
1275 \def\givenatend{\boolean{#1}}
1276 \ifthenelse{\boolean{#1}}
1277 {% Begin then result

```

```

1278 \begin{trivlist}
1279 \item
1280 }% End then result
1281 {% Begin else result
1282 \setbox0\vbox\bgroup
1283 }% End else result
1284 % {}% Don't think this is doing anything? -- Jason
1285 }
1286 {%
1287 \ifthenelse{\givenatend}
1288 {% Begin then result
1289 \end{trivlist}
1290 }% End then result
1291 {% Begin else result
1292 \egroup
1293 }% End else result
1294 % {}% Don't think this is doing anything? -- Jason
1295 }
1296 \else
1297 \newenvironment{freeResponse}[1][false]%
1298 {% Environment Beginning Code
1299   \ifthenelse{\boolean{#1}}% Could probably change this with just putting the (given) in the
1300   {% Begin then result
1301     \begin{trivlist}
1302     \item[\hskip \labelsep\bfseries Free Response (Given):\hspace{2ex}]
1303     }% End then result
1304   {% Begin else result
1305     \begin{trivlist}
1306     \item[\hskip \labelsep\bfseries Free Response:\hspace{2ex}]
1307     }% End else result
1308   }
1309   {% Environment Ending Code
1310   \end{trivlist}
1311 }
1312 \fi
1313
1314 </classXimera>
1315 < *htXimera>
1316
1317 \renewenvironment{freeResponse}{\refstepcounter{problem}}{}%
1318 \ConfigureEnv{freeResponse}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<
1319
1320 </htXimera>

```

2.12.2 Feedback

feedback An initially hidden environment that uncovers itself at an appropriate time. New Validator rewrite code added by Jason Nowell. Original code provided by Jim Fowler. Validator is an environment designed to run a custom check on answers (usually) using javascript code.

Define a placeholder command for validator and feedback.

```

1321 < *classXimera>
1322 \newcommand{\PH@Command}{}

Validator should take an argument and detokenize it and display it at the start of the
environment. The original Validator environment had everything framed in an mbox;
presumably to make the text look a bit nicer, although this seems redundant with texttt.
It shouldn't cause any harm so I have left it in for now.

1323 \newenvironment{validator}[1][]{
1324 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter" to
1325 \mbox{\texttt{\detokenize\expandafter{\PH@Command}}}% Now expand PH@Command once and then det
1326 }{}

```

First, if it's a handout, we want feedback to eat everything and then disappear entirely. So we do this:

```

1327 \ifhandout%
1328 \newenvironment{feedback}
1329     {%
1330 \setbox0\vbox\bgroup
1331     }
1332     {%
1333 \egroup
1334     }

```

If this isn't a handout, then we want to display the Feedback by using a label, positioned and formatted as a `\item` in a `trivlist`. It is important that we also detokenize the content of the optional argument, as it is likely to contain javascript or other code that latex won't be able to make sense of.

```

1335 \else

1336 \newenvironment{feedback}[1][attempt]{
1337
1338 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter" to
1339
1340 \begin{trivlist}% Begin the trivlist to use formatting of the "Feedback" label.
1341 \item[\hspace{\labelsep}\small\slshape\bfseries Feedback% Format the "Feedback" label. Don't f
1342 (\texttt{\detokenize\expandafter{\PH@Command}}):% Format (and detokenize) the condition for f
1343 \hspace{2ex}]\small\slshape% Insert some space before the actual feedback given.
1344 }{
1345 \end{trivlist}
1346 }
1347
1348 \fi
1349 \end{classXimera}

```

Feedback environments take an optional parameter (which describes when the feedback is to be provided)

```

1350 \begin{htXimera}
1351 \def\feedback{\ifnextchar[{\@feedbackcode}{\@feedbackattempt}}
1352 \def\@feedbackattempt{\@feedbackcode[attempt]}
1353 \def\@feedbackcode[#1]{\stepcounter{identification}%
1354 \ifvmode \IgnorePar\fi \EndP%
1355 \ifthenelse{\equal{#1}{attempt}}{\HCode{<div class="feedback" data-feedback="attempt" id="fe
1356 {\ifthenelse{\equal{#1}{correct}}{\HCode{<div class="feedback" data-feedback="correct" id="f
1357 {\HCode{<div class="feedback" data-feedback="script" id="feedback\arabic{identification}><s
1358 \def\endfeedback{\HCode{</div>}\IgnoreIndent}
1359 \end{htXimera}

```

2.12.3 Ungraded activities

ungraded The `ungraded` environment is used to record that certain parts of activities should not be worth points. For example, if you want to use a `multipleChoice` as a survey question, you can place it inside an `ungraded` environment. On the \LaTeX side, the `ungraded` environment does nothing.

```

1360 \begin{classXimera}
1361 \newenvironment{ungraded}{}{}
1362 \end{classXimera}

```

But on the html side, `ungraded` wraps the activities in a `div` in order to assign some weight to them for grading.

```

1363 \begin{htXimera}
1364 \renewenvironment{ungraded}{%
1365 \ifvmode \IgnorePar\fi \EndP\HCode{<div class="ungraded">}\IgnoreIndent%
1366 }{
1367 \ifvmode \IgnorePar\fi \EndP\HCode{</div>}\IgnoreIndent%
1368 }
1369 \end{htXimera}

```

2.13 Support for the web

2.13.1 MathJax support

When using mathjax, dump all the `\newcommands` to a `.jax` file.

First, create the `.jax` file.

```
1370 \*classXimera>
1371 \ifdefined\HCode
1372   \else
1373     \newwrite\myfile
1374     \immediate\openout\myfile=\jobname.jax
1375 \fi
1376 \*classXimera>
```

From `only.dtx` we must also create `prompt` on the MathJax side.

```
1377 \*classXimera>
1378 \ifdefined\HCode
1379   \else
1380     \immediate\write\myfile{\unexpanded{\newenvironment}{prompt}{}}{}
1381 \fi
1382 \*classXimera>
```

Redefine newcommand appropriately.

```
1383 \*classXimera>
1384 \ifdefined\HCode
1385   \else
1386     \let\@oldargdef\@argdef
1387     \long\def\@argdef#1[#2]#3{%
1388       \immediate\write\myfile{\unexpanded{\newcommand}{\unexpanded{#1}}[\unexpanded{#2}]{\unexpanded{#3}}}%
1389       \@oldargdef#1[#2]{#3}%
1390     }
1391
1392     \let\@oldDeclareMathOperator\DeclareMathOperator
1393     \renewcommand{\DeclareMathOperator}[2]{\@oldDeclareMathOperator{#1}{#2}\immediate\write\myfile{\unexpanded{\newcommand}{\unexpanded{#1}}[\unexpanded{#2}]{\unexpanded{#3}}}%
1394
1395   \fi
1396 \*classXimera>
```

Include the `jax`'ed newcommands

```
1397 \*cfgXimera>
1398 % Remove commands that use @
1399 \immediate\write18{sed -i "/[:*@]/d" \jobname.jax}
1400 % Replace ##1 with #1 and so forth
1401 \immediate\write18{sed -i "s/\string#\string#\string\\([0-9]\string\\)/\string#\string\\1/g"}
1402
1403 \Configure{BVerbatimInput}{}{}{}
1404
1405 \Configure{verbatiminput}{}{}{}
1406
1407 % Instead of a nonbreaking space, use a standard space
1408 \makeatletter
1409 \def\FV@Space{\space}
1410 \makeatother
1411
1412 % Include the mathjax newcommands in a math/tex script right at the beginning of the body
1413 \Configure{BODY}{%
1414   \HCode{<body>\Hnewline}%
1415   \Tg<div class="preamble">%
1416   \IfFileExists{\jobname.jax}{
1417     \Tg<script type="math/tex">%
1418     \BVerbatimInput{\jobname.jax}%
1419     \Tg</script>%
1420   }
1421   {\Hnewline\HCode{<!-- mm, no \newcommands provided -->}\Hnewline}
1422 }
```

```

1423 \IfFileExists{\jobname.ids}{\HCode{<script type="text/javascript">\Hnewline}%
1424 \BVerbatimInput{\jobname.ids}%
1425 \HCode{</script>\Hnewline}%
1426 }{}
1427 \Tg</div>%
1428 }{%
1429 \ifvmode\IgnorePar\fi\EndP\HCode{</body>\Hnewline}%
1430 }
1431
1432 % prevent spaces as in "\begin {align}" (it confuses Mathax2)
1433 \renewcommand\VerbMathToks[2]{%
1434   \HCode{\string\begin{#2}}%
1435   \alreqtoks{#1}%
1436   \HCode{\string\end{#2}}%
1437 }
1438
1439 % This is a fix for the LAODE book, which uses matlabEquation as if it were an equation
1440 \ScriptEnv{matlabEquation}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="math/tex; mode=d
1441
1442 \</cfgXimera>

```

2.13.2 Semantic HTML

\textbf Using **\textbf** emits a **** tag.

```

1443 \<*/cfgXimera>
1444 \Configure{textbf}{\ifvmode\ShowPar\fi\HCode{<strong>}}{\HCode{</strong>}}
1445 \</cfgXimera>

```

\textit Using **\textit** or similar emits an **** tag.

```

1446 \<*/cfgXimera>
1447 \Configure{textit}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1448 \Configure{emph}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1449 \</cfgXimera>

```

\texttt Using **\texttt** emits a **<code>** tag.

```

1450 \<*/cfgXimera>
1451 \Configure{texttt}{\ifvmode\ShowPar\fi\HCode{<code>}}{\HCode{</code>}}
1452 \</cfgXimera>

```

2.14 Tools

2.14.1 Suppress

suppress The suppress environment is a good way to suppress output without commenting it. This way we can avoid many of the places we use environ package and this should also avoid most of the verbatim conflicts. This is code adapted from **syntonly.sty**.

```

1453 \<*/classXimera>
1454 \font\dummyft@=dummy \relax
1455 \def\suppress{%
1456   \begingroup\par
1457   \parskip\z@
1458   \offinterlineskip
1459   \baselineskip=\z@skip
1460   \lineskip=\z@skip
1461   \lineskiplimit=\maxdimen
1462   \dummyft@
1463   \count@\sixt@@n
1464   \loop\ifnum\count@ >\z@
1465     \advance\count@\m@ne
1466     \textfont\count@\dummyft@
1467     \scriptfont\count@\dummyft@
1468     \scriptscriptfont\count@\dummyft@
1469   \repeat
1470   \let\selectfont\relax

```



```

1471 \let\mathversion\@gobble
1472 \let\getanddefine@fonts\@gobbletwo
1473 \tracinglostchars\z@
1474 \frenchspacing
1475 \hbadness\@M}
1476 \def\endsuppress{\par\endgroup}
1477 \end{classXimera}

```

2.14.2 The End

It seems that some of the files need to conclude with something or another.

```

1478 \end{htXimera}
1479 \Hinput{ximera}
1480 \end{htXimera}

1481 \end{htXourse}
1482 \Hinput{xourse}
1483 \end{htXourse}

1484 \end{cfgXimera}
1485 \begin{document}
1486 \EndPreamble
1487 \end{cfgXimera}

```

3 xourse.cls

```

1488 \end{classXourse}

```

notoc The default behavior of the class is to provide a table of contents listing all activities in the course. This option will suppress this table of contents.

```

1489 \newif\ifnotoc
1490 \notocfalse
1491 \DeclareOption{notoc}{\notoctrue}

```

nonewpage The default behavior of the class is to start each activity on a new page. This option will start activities without making a new page.

```

1492 \newif\ifnonewpage
1493 \nonewpagefalse
1494 \DeclareOption{nonewpage}{\nonewpagetrue}

1495 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{ximera}}
1496 \ProcessOptions\relax
1497 \LoadClass{ximera}
1498 % \begin{macrocode}
1499 \end{classXourse}

```

3.1 Activities

The core of the **xourse** system. It works by redefining the **document** environment, thus making the **\begin** and **\end{document}** of the subfile ‘transparent’ to the inclusion. The redefinition of **\documentclass** is analogous, just having a required and an optional arguments which mean nothing to **\subfile**.

```

1500 \end{classXourse}
1501 \newcommand{\skip@preamble}{%
1502   \let\document\relax\let\enddocument\relax%
1503   \newenvironment{document}{\let\input\otherinput}{}%
1504   \renewcommand{\documentclass}[2][subfiles]{}}

```

Note that the new command **\subfile** calls for **\skip@preamble** *within a group*. The changes to **document** and **\documentclass** are undone after the inclusion of the subfile.

Numbering starts a page too soon without this:

```

1505 \let\otherinput\input

```

```

Store usual \maketitle as \othermaketitle
1506 \let\othermaketitle\maketitle
\maketitle In a xourse file, \maketitle is redefined to give course packet title page and toc.
1507 \renewcommand{\maketitle}{ %
1508 \pagestyle{empty}
1509 \begin{center}
1510 ~\ \ %puts space at top of page to move title down.
1511 \vskip .25\textheight
1512 \hrulefill\ \
1513 \vskip 1em
1514 \bfseries{\Huge \@title} \ \
1515 \hrulefill\ \
1516 \vskip 3em
1517 {\Large \@author}
1518 \vskip 2em
1519 {\large \@date}
1520 \end{center}
1521 \clearpage

When notoc option is used, we do not include a table of contents. Otherwise we include
a table of contents in every course packet.
1522 \ifnotoc
1523 \else
1524 \tableofcontents\clearpage
1525 \clearpage
1526 \fi

Switch to main pagestyle, just like a document with documentclass ximera.
1527 \pagestyle{main}

Renew maketitle to usual definition.
1528 \let\maketitle\othermaketitle

And we finish with our redefinition of \maketitle.
1529 }
1530 \relax
1531 \end{classXourse}

```

3.1.1 Regular activities

\activity Documents included with `\activity` will be included in the body of the xourse document. Any `\input` commands within included ximera documents will be ignored. Any `\usepackage` commands within included ximera documents will cause an error. Overlapping `\newcommand` definitions within multiple ximera documents included simultaneously will cause an error. The `\activity` command inputs the file name provided without `\documentclass`, without `\begin{document}`/`\end{document}` and without any inputs in the preamble of the included file.

```

1532 \begin{classXourse}
1533 \ifnonepage
1534 \newcommand{\activity}[2][]{\%
1535 \setkeys{activity}{#1}
1536 \renewcommand{\input}[1]{
1537 \begin{group}\skip@preamble\otherinput{#2}\end{group}\par\vspace{\topsep}
1538 \let\input\otherinput}
1539 \else
1540 \newcommand{\activity}[2][]{\%
1541 \setkeys{activity}{#1}
1542 \renewcommand{\input}[1]{
1543 \begin{group}\skip@preamble\otherinput{#2}\end{group}\clearpage
1544 \let\input\otherinput}
1545 \fi
1546 \relax
1547 \end{classXourse}

```

```

1548 <*htXourse>
1549 \renewcommand\activity[2] [] {%
1550 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="activity card \activitystyle" href="#2" data-opt
1551 }
1552 </htXourse>

```

When running xake, we can just ignore activities

```

1553 <*classXourse>
1554 \ifxake
1555 \renewcommand\activity[2] [] {}
1556 \fi
1557 </classXourse>

```

3.1.2 Practice activities

`\practice` Like `\activity` but not expecting a title.

```

1558 <*classXourse>
1559 \ifhandout
1560 \newcommand{\practice}[2] [] {
1561 \setkeys{practice}{#1}%!!!!
1562 \renewcommand{\input}[1] {}
1563 \begingroup\skip@preamble\otherinput{#2}\endgroup
1564 \let\input\otherinput}
1565 \else
1566 \newcommand{\practice}[2] [] {\texttt{\detokenize{#2}}}% gives file name for practice
1567 \setkeys{practice}{#1}%!!!!
1568 \renewcommand{\input}[1] {}
1569 \begingroup\skip@preamble\otherinput{#2}\endgroup
1570 \let\input\otherinput}
1571 \fi
1572 \relax
1573 </classXourse>

```

The practice environment does nothing, but will eventually produce exercises at the end of an activity

```

1574 <*classXourse>
1575 \ifxake
1576 \renewcommand\practice[2] [] {}
1577 \fi
1578 </classXourse>

```

I suppose it is reasonable for practice cards to NOT have an `activitystyle`, since the `activitystyle` is basically PRACTICE.

```

1579 <*htXourse>
1580 \renewcommand\practice[2] [] {%
1581 \ifvmode\IgnorePar\fi\EndP%
1582 \HCode{<a class="activity card practice" href="#2" data-options="#1">#2</a>}%
1583 \IgnoreIndent%
1584 }
1585 </htXourse>

```

3.2 Sectioning

`\section` Makes the table of contents look a bit better. This can be redefined in the preamble if you do not like the appearance. The name of a section inside an activity.

```

1586 <*classXourse>
1587 \renewcommand*\l@section{\@dottedtocline{1}{1.5em}{4.2em}}
1588 </classXourse>

```

`\subsection` The name of a subsection inside an activity.

```

1589 <*classXourse>
1590 \renewcommand*\l@subsection{\@dottedtocline{2}{3.8em}{4.2em}}
1591 </classXourse>

```

`\part` Xourse files can have parts. The name of a large part of a xourse.

```

1592 \*htXourse>
1593 \newcounter{ximera@part}
1594 \setcounter{ximera@part}{0}
1595 \renewcommand\part[1]{%
1596 \stepcounter{ximera@part}%
1597 \ifvmode \IgnorePar\fi \EndP%
1598 %\HCode{<h1 id="part\arabic{ximera@part}" class="card part">}#1\HCode{</h1>}}% makes cards di
1599 \HCode{<h1 id="part\arabic{ximera@part}" class="card part">}#1</h1>}}%
1600 \IgnoreIndent%
1601 }
1602 </htXourse>

```

`\paragraph` Paragraph commands emit spans. A small heading.

```

1603 \*cfgXimera>
1604 \renewcommand{\paragraph}[1]{%
1605   \HCode{<span class="paragraphHead">}}%
1606   #1%
1607   \HCode{</span>}\par\IgnorePar}
1608 </cfgXimera>

```

`\subparagraph` An even smaller heading.

```

1609 \*cfgXimera>
1610 \renewcommand{\subparagraph}[1]{%
1611   \HCode{<span class="subparagraphHead">}}%
1612   #1%
1613   \HCode{</span>}\par\IgnorePar}
1614 </cfgXimera>

```

3.3 Grading by points

`graded` The `graded` environment does nothing in latex, but in html, it wraps the activities in a div in order to assign some weight to them for grading.

```

1615 \*classXourse>
1616 \newenvironment{graded}[1]{}{}
1617 </classXourse>

```

So indeed this environment in html wraps the activities in a div in order to assign some number of points to them.

```

1618 \*htXourse>
1619 \renewenvironment{graded}[1]{%
1620 \ifvmode \IgnorePar\fi \EndP\HCode{<div class="graded" data-weight="#1">}\IgnoreIndent%
1621 }{
1622 \ifvmode \IgnorePar\fi \EndP\HCode{</div>}\IgnoreIndent%
1623 }
1624 </htXourse>

```

3.4 Logos

`\logo` A logo for the xourse.

```

1625 \*classXourse>
1626 \newcommand*\logo[1]{%
1627   \ifx\@onlypreamble\@notprerr
1628     \ClassError{xourse}{logo can only be used in the preamble}
1629     {Move your logo command to the preamble}
1630   \else %
1631     \IfFileExists{#1}%
1632     {\gdef\xourse@logo{#1}}%
1633     {\ClassError{xourse}{logo file does not exist}
1634      {To use logo, make sure that the referenced image file exists}}%
1635   \fi%
1636 }
1637
1638 </classXourse>

```

The xourse logo is an `og:image` in the opengraph taxonomy.

```
1639 <*htXourse>
1640 \Configure{@HEAD}{%
1641   \HCode{<meta name="og:image" content="}%
1642   \ifdefined\xourse@logo%
1643     \xourse@logo%
1644   \fi%
1645   \HCode{" />\Hnewline}}%
1646 </htXourse>
```