

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}{\xaketrue}
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   \xaketrue%
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" type="text/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{Ximera}
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 \ConfigureQuestionEnv{hint}{hint}
388 %%%\ConfigureQuestionEnv{shuffle}{shuffle}
389 \end{Ximera}

```

2.4.6 Hints

hint Hint environments can be embedded inside problems.

```

390 \begin{classXimera}

```

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

```

391 \newcounter{hintLevel}
392 \setcounter{hintLevel}{0}

```

Create an empty shell to renew

```

393 \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.

```

394 \renewenvironment{hint}
395 {
396 \ifhandout
397 \setbox0\vbox\bgroup
398 \else
399 \begin{trivlist}\item[\hspace{1em}\labelsep\small\slshape\bfseries Hint:\hspace{2ex}]
400 \small\slshape
401 \fi

```

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

```

402 \stepcounter{hintLevel}
403 }
404 {
405 \ifhandout
406 \egroup\ignorespacesafterend
407 \else
408 \end{trivlist}
409 \fi

```

Detract from hint level counter to track hint nested level

```

410 \addtocounter{hintLevel}{-1}
411 }
412
413 \ifhints
414 \renewenvironment{hint}{

```

```

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

```

2.4.7 Solution

solution The solution to a problem.

```

421 \begin{classXimera}
422 %% solution environment
423 \ifhandout % what follows is handout behavior
424 \newenvironment{solution}%
425     {%
426     \setbox0\vbox\bgroup
427     }
428     {%
429     \egroup
430     }
431 \else
432 \newenvironment{solution}%
433     {%
434     \begin{trivlist}
435     \item[\hskip \labelsep\bfseries Solution:\hspace{2ex}]
436     }
437     % %% line at the bottom}
438     {
439     \end{trivlist}
440     \par\addvspace{.5ex}\nobreak\noindent\hung
441     }
442 \fi
443
444
445
446 \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.

```

447 \begin{classXimera}
448 \DefineVerbatimEnvironment{code}{Verbatim}{numbers=left,frame=lines,label=Code,labelposition=left}
449 \end{classXimera}

```

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

```

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

```

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

```

453 \begin{classXimera}
454 \DefineVerbatimEnvironment{javascriptCode}{Verbatim}{numbers=left,frame=lines,label=JavaScript,labelposition=left}
455 \end{classXimera}
456 \begin{cfgXimera}
457 \renewenvironment{javascriptCode}{\NoFonts}{\EndNoFonts}
458 \ScriptEnv{javascriptCode}{\stepcounter{identification}\ifmode \IgnorePar\fi \EndP\HCode{<div>
459 \end{cfgXimera}

```

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

```

460 \begin{cfgXimera}
461 \ConfigureEnv{verbatim}{\ifmode\IgnorePar\fi\EndP\HCode{<pre>}}{\ifmode\IgnorePar\fi\EndP\HCode{</pre>}}

```

```

462 \ConfigureEnv{lstlisting}{\ifvmode\IgnorePar\fi\EndP\HCode{<pre>}}{\ifvmode\IgnorePar\fi\EndP\HCode{</pre>}}
463 \</cfgXimera>

```

2.4.9 Dialogues

dialogue A dialogue between people.

```

464 \*classXimera>
465 \newenvironment{dialogue}{%
466   \renewcommand\descriptionlabel[1]{\hspace{\labelsep}\textbf{##1:}}
467   \begin{description}%
468 }{%
469   \end{description}%
470 }
471 \</classXimera>

```

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

```

472 \*htXimera>
473 \renewenvironment{dialogue}{\begin{description}}{\end{description}}
474
475 \ConfigureList{dialogue}%
476   {\EndP\HCode{<dl \a:LRdir class="dialogue">}}%
477   \PushMacro\end:itm
478 \global\let\end:itm=\empty
479   {\PopMacro\end:itm \global\let\end:itm \end:itm \end:itm
480 \EndP\HCode{</dd></dl>}}\ShowPar}
481   {\end:itm \global\def\end:itm{\EndP\Tg</dd>}\HCode{<dt
482     class="actor">}\bgroup \bf}
483   {\egroup\EndP\HCode{</dt><dd\Hnewline class="speech">}}
484 \</htXimera>

```

2.4.10 Instructor notes

```

485 \*classXimera>
486
487 %% instructor intro/instructor notes
488 %%
489 \ifhandout % what follows is handout behavior
490 \ifinstructornotes
491 \newenvironment{instructorIntro}%
492   {%
493   \begin{trivlist}
494   \item[\hspace{\labelsep}\bfseries Instructor Introduction:\hspace{2ex}]
495   }
496   % %% line at the bottom}
497   {
498   \end{trivlist}
499   \par\addvspace{.5ex}\nobreak\noindent\hung
500   }
501 \else
502 \newenvironment{instructorIntro}%
503   {%
504   \setbox0\vbox\bgroup
505   }
506   {%If this mysteriously starts breaking
507   % remove \ignorespacesafterend
508   \egroup\ignorespacesafterend
509   }
510   \fi
511 \else% for handout, so what follows is default
512 \ifinstructornotes
513 \newenvironment{instructorIntro}%
514   {%
515   \setbox0\vbox\bgroup
516   }

```

```

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

```

```

580                                     \fi
581
582 \endclassXimera

```

2.4.11 Only

prompt The prompt part for mathmode

```

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

```

onlineOnly Only display it online

```

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

```

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.

```

612 %\iftikzexport\else\RequirePackage[framemethod=TikZ]{mdframed}\fi

```

foldable Does it fold?

```

613 \beginclassXimera
614
615 \colorlet{textColor}{black} % since textColor is referenced below
616 \colorlet{background}{white} % since background is referenced below
617
618 % The core environments. Find results in 4ht file.
619 %% pretty-foldable
620 %\iftikzexport
621 \newenvironment{foldable}{%
622 }{%
623 }
624 %\else
625 %\renewmdenv[
626 % font=\upshape,
627 % outerlinewidth=3,
628 % topline=false,
629 % bottomline=false,

```

```

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

```

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

```

666 \begin{htXimera}
667 \renewenvironment{foldable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<
668
669 \renewenvironment{expandable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{
670
671 }}{\HCode{</div>}\IgnoreIndent}
672
673 \renewcommand{\unfoldable}[1]{\HCode{<span class="unfoldable">#1\HCode{</span>}}}
674 \end{htXimera}

```

2.4.13 Leashes

leash Put content inside a scrollable box.

```

675 \begin{classXimera}
676
677 \newenvironment{leash}[1]{%
678 }{%
679 }
680
681
682 \end{classXimera}
683 \begin{htXimera}
684 \renewenvironment{leash}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div style="overflow: auto; h
685 \end{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.

```
686 \*classXimera>
687 \newcommand{\license}{\excludecomment}
688 \*classXimera<
```

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

```
689 \*classXimera>
690 \newcommand{\acknowledgement}{\excludecomment}
691 \*classXimera<
```

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

```
692 \*classXimera>
693 \renewcommand{\tag}{\excludecomment}
694 \*classXimera<
```

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

```
695 \*htXourse>
696 % Mark this as a xourse file
697 \Configure{@HEAD}{\HCode{<meta name="description" content="xourse" />\Hnewline}}
698 \*htXourse<
```

2.5.2 Abstract

`abstract` Every activity should include a short abstract.

```
699 \*classXimera>
700 \let\abstract\relax
701 \let\endabstract\relax
702 % Use of environ package, may want to find a better way.
703 \NewEnviron{abstract}{\protected@xdef\theabstract{\BODY}}
704 \*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.

```
705 \*cfgXimera>
706 \let\abstract\relax
707 \let\endabstract\relax
708 \*cfgXimera<
```

2.5.3 Titles and authors

2.5.4 Authors

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

```
709 \*classXimera>
710 \let\@emptyauthor\@author
711 \def\author#1{\gdef\@author{#1}}
712 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}
713 \*classXimera<
```

Include author name in meta tags

```
714 \*htXimera>
715 \Configure{@HEAD}{\HCode{<meta name="author" content="}\@author\HCode{" />\Hnewline}}
716 \*htXimera<
```

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

```
717 \*htXimera | classXimera>\def\and{and }
```


2.5.5 Title

```

\title Activities have titles.
718 \classXimera
719 \let\title\relax
720 \newcommand{\title}[1] []{{\protected@xdef\@prettitle{#1}}\protected@xdef\@title{
721
722 \title{
723
724 \newcounter{titlenumber}
725 \renewcommand{\thetitlenumber}{\arabic{titlenumber}}
726 %\renewcommand{\thesection}{\arabic{titlenumber}} %% Makes section numbers work
727 \setcounter{titlenumber}{0}
728
729 \newpagestyle{main}{
730 \sethead[\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}] [] [] % even
731 {}{}{\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}} % odd
732 \setfoot[\thepage] [] [] % even
733 {}{}{\thepage} % odd
734 }
735 \pagestyle{main}

\maketitle In a ximera document, redefine \maketitle and put them in a table of contents. The
\phantomsection is to fix the hrefs.
736 \renewcommand\maketitle{%
737 \addtocounter{titlenumber}{1}%
738 {\flushleft\large\bfseries \@prettitle\par\vspace{-1em}}
739 {\flushleft\LARGE\bfseries {\ifnumbers\thetitlenumber\fi}{\ifnumbers\hspace{1em}\else\hspa
740 \phantomsection%
741 \ifnumbers\addcontentsline{toc}{section}{\thetitlenumber~\@title}\else\addcontentsline{toc}
742 \vskip .6em\noindent\textit{theabstract}\setcounter{problem}{0}\setcounter{section}{0}\setco
743 \ifnooutcomes\else\let\thefootnote\relax\footnote{Learning outcomes: \theoutcomes}\fi
744 \ifnoauthor\else\let\thefootnote\relax\footnote{Author(s):~\@author}\fi
745 \aftergroup\@afterindentfalse
746 \aftergroup\@afterheading}
747
748 \ifnumbers
749 \setcounter{secnumdepth}{2}
750 \renewcommand{\thesection}{\arabic{titlenumber}.\arabic{section}}
751 \renewcommand{\thesubsection}{\arabic{titlenumber}.\arabic{section}.\arabic{subsection}}
752 \else
753 \setcounter{secnumdepth}{-2}
754 \fi
755
756 \def\activitystyle{}
757 \newcounter{sectiontitlenumber}
758 \setcounter{secnumdepth}{2}
759 \setcounter{tocdepth}{2}
760 \newcommand\chapterstyle{%
761 \def\activitystyle{activity-chapter}
762 \def\maketitle{%
763 \addtocounter{titlenumber}{1}%
764 {\flushleft\small\sffamily\bfseries\@prettitle\par\vspace{-1.5em}}%
765 {\flushleft\LARGE\sffamily\bfseries\thetitlenumber\hspace{1em}\@title \pa
766 {\vskip .6em\noindent\textit{theabstract}\setcounter{problem}{0}\setcounte
767 \par\vspace{2em}
768 \phantomsection\addcontentsline{toc}{section}{\textbf{\thetitlenumber\hsp
769 }}
770
771
772 \newcommand\sectionstyle{%
773 \def\activitystyle{activity-section}
774 \def\maketitle{%
775 \addtocounter{section}{1}

```


2.5.7 Labels and references

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

```

829 \*htXimera>
830 \let\oldlabel\label
831 \renewcommand{\label}[1]{\oldlabel{#1}\HCode{<a class="ximera-label" id="#1"></a>}}
832 \*htXimera>

\ref A \ref can connect one TEX file to another if they are in the same xourse.
833 \*htXimera>
834 \renewcommand{\ref}[1]{\HCode{<a class="reference" href="\##1">#1</a>}}
835 \*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.

```

836 \*classXimera>
837 %\newenvironment{image}[1][\begin{center}]{\end{center}}
838 \NewEnviron{image}[1][3in]{%
839   \begin{center}\resizebox{#1}{!}{\BODY}\end{center}% resize and center
840 }
841 \*classXimera>

```

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

```

842 \*classXimera>
843 \newcommand{\alt}[1]{%
844 }
\*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.

```

845 \*htXimera>
846 \newcounter{imagealt}
847 \setcounter{imagealt}{0}
848 \renewenvironment{image}[1][\stepcounter{imagealt}%
849   \ifvmode \IgnorePar\fi \EndP%
850   \HCode{<div class="image-environment" role="img" aria-labelledby="image-alt-\arabic{imagealt}>}
851 }{\HCode{</div>}}
852 \renewcommand{\alt}[1]{\HCode{<div style="display: none;" id="image-alt-\arabic{imagealt}">}}
853 \*htXimera>
854 \*cfgXimera>
855 %% Although we accept many formats, SVG is preferred on the web.
856 %% Since we have a different mechanism for producing |alt| text, we
857 %% want to ignore tex4ht's own method fo producing alt text.
858 %% 2024: is now in TeX4ht ...
859 \DeclareGraphicsExtensions{.jpg,.png,.gif,.svg}
860 % \Configure{graphics*}
861 % {svg}{
862 %   {\Configure{Needs}{File: \Gin@base.svg}\Needs{}}
863 %   \Picture[]{\csname Gin@base\endcsname.svg \csname a:Gin-dim\endcsname}%
864 % }
865 \*cfgXimera>

```

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

```

866 \*cfgXimera>
867 \ifcsname ifstandalone\endcsname
868   \ifstandalone
869     \renewcommand\includegraphics[2][{}]{%
870   \fi

```

```
871 </cfgXimera>
```

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

```
872 <*htXimera>
873 \providecommand{\pgfsyspdfmark}[3]{-}
874 </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.

```
875 <*classXimera>
876 % everything skipped, assume TeX4ht does the jbb now
877 \ifdefined\reallyneverever
878
879 \ifdefined\HCode
880 \tikzexporttrue
881 \fi
882
883 \iftikzexport
884 \usetikzlibrary{external}
885
886 \ifdefined\HCode
887 % in htlatex, just include the svg files
888 \def\pgfsys@imagesuffixlist{.svg}
889
890 \tikzexternalize[prefix=./,mode=graphics if exists]
891 \else
892 % in pdflatex, actually generate the svg files
893 \tikzset{
894 /tikz/external/system call={
895 pdflatex \tikzexternalcheckshellescape
896 -halt-on-error -interaction=batchmode
897 -jobname "\image" "\PassOptionsToClass{tikzexport}{ximera}\texsource";
898 mutool draw -F svg \image.pdf > \image.svg ; % mutool adds "1" to filename ???
899 mutool draw -o \image.svg \image.pdf ;
900 mutool draw -r 150 -c rgbalpha -o \image.png \image.pdf ;
901 ebb -x \image.png
902 }
903 }
904 \tikzexternalize[optimize=false,prefix=./]
905 \fi
906
907 \fi
908 \fi
909 </classXimera>
```

2.6.3 XKCD

\xkcd Reference an XKCD cartoon.

```
910 <*classXimera>
911 \newcommand{\xkcd}[1]{#1}
912 </classXimera>
```

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

```
913 <*htXimera>
914 \renewcommand{\xkcd}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{
```

2.8.3 Geogebra

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

```
953 \*classXimera>
954 %Geogebra link
955 \newcommand{\geogebra}[3]{Geogebra link: \url{https://www.geogebra.org/m/#1}}
956 \*classXimera>
```

Define keys for answer geogebra key=value pairs.

```

957 \*htXimera>
958 \define@key{geogebra}{rc}[true]{\def\geo@rc{#1}}
959 \define@key{geogebra}{sdz}[true]{\def\geo@sdz{#1}}
960 \define@key{geogebra}{smb}[true]{\def\geo@smb{#1}}
961 \define@key{geogebra}{stb}[true]{\def\geo@stb{#1}}
962 \define@key{geogebra}{stbh}[true]{\def\geo@stbh{#1}}
963 \define@key{geogebra}{ld}[true]{\def\geo@ld{#1}}
964 \define@key{geogebra}{sri}[true]{\def\geo@sri{#1}}
965 %set default key values
966 \setkeys{geogebra}{rc=false,sdz=false,smb=false,stb=false,stbh=false,ld=false,sri=false}
967 %command definition
968 \renewcommand{\geogebra}[4][]{%
969   \setkeys{geogebra}{#1}% Set new keys
970   \HCode{<iframe scrolling="no" src="https://www.geogebra.org/material/iframe/id/#2/width/#3
971 \*htXimera>

```

2.8.4 Desmos

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

```

972 \*classXimera>
973 \newcommand{\desmos}[3]{Desmos link: \url{https://www.desmos.com/calculator/#1}}
974 \newcommand{\desmosThreeD}[3]{Desmos3D link: \url{https://www.desmos.com/3d/#1}}
975 \*classXimera>
976 \*htXimera>
977 \renewcommand{\desmos}[3]{\HCode{<iframe src="https://www.desmos.com/calculator/#1" width="1
978 \renewcommand{\desmosThreeD}[3]{\HCode{<iframe src="https://www.desmos.com/3d/#1" width="#2p
979 \*htXimera>

```

2.8.5 Graphs

`\graph` An embedded graph (in math mode).

```

980 \*classXimera>
981 \newcommand{\graph}[2][]{\text{Graph of $#2$}}
982 \*classXimera>
983 \*htXimera>
984 \renewcommand{\graph}[2][]{\HCode{<div class="graph" data-options="#1">#2\HCode{</div>}}
985 \*htXimera>

```

2.8.6 Video

`\youtube` Youtube command. Requires id.

```

986 \*classXimera>
987 \newcommand{\youtube}[1]{YouTube link: \url{https://www.youtube.com/watch?v=#1}}
988 \*classXimera>
989 \*htXimera>
990 \renewcommand{\youtube}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="video youtube-play
991 \*htXimera>

```

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

```

992 \*htXourse>
993 \renewcommand\youtube[1]{%
994 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="youtube" href="https://www.youtube.com/watch?v=
995 }
996 \*htXourse>

```

2.8.7 JavaScript

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

```

997 \*classXimera>
998 \DefineVerbatimEnvironment{javascript}{Verbatim}{numbers=left,frame=lines,label=JavaScript,1
999 \*classXimera>

1000 \*htXimera>
1001 % for programming javascript
1002 \renewenvironment{javascript}{\NoFonts}{\EndNoFonts}
1003 \ScriptEnv{javascript}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div c
1004 \*htXimera>

```

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

```

1005 \*classXimera>
1006 \def\js#1{\mbox{\texttt{\detokenize{#1}}}}
1007 \*classXimera>

1008 \*htXimera>
1009 \def\js#1{\stepcounter{identification}\HCode{<span class="inline-javascript" id="javascript\
1010 \*htXimera>

```

2.9 SageMath support

Load SageTeX if it exists.

```

1011 \*classXimera>
1012 \IfFileExists{sagetex.sty}{\RequirePackage{sagetex}}{}
1013 \*classXimera>

```

sageCell Create an interactive SageMath widget.

```

1014 \*classXimera>
1015 \DefineVerbatimEnvironment{sageCell}{Verbatim}{numbers=left,frame=lines,label=SAGE,labelpositi
1016 \*classXimera>

1017 \*htXimera>
1018 \renewenvironment{sageCell}{\NoFonts}{\EndNoFonts}
1019 \ScriptEnv{sageCell}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sage"><script type="text
1020 \*htXimera>

```

sageOutput Execute SageMath code and output the result.

```

1021 \*classXimera>
1022 \DefineVerbatimEnvironment{sageOutput}{Verbatim}{numbers=left,frame=lines,label=SAGE-Output,
1023 \*classXimera>

1024 \*htXimera>
1025 \renewenvironment{sageOutput}{\NoFonts}{\EndNoFonts}
1026 \ScriptEnv{sageOutput}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sageOutput"><script ty
1027 \*htXimera>

```

sageSilent Execute SageMath code without outputting the result.

```

1028 \*htXimera>
1029 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1030 \ifdefined\sagesilent
1031 \renewenvironment{sagesilent}{\NoFonts}{\EndNoFonts}
1032 \fi
1033 \ScriptEnv{sagesilent}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="text/sagemath">}\Htm
1034 \*htXimera>

```

2.10 Answerables

2.10.1 Answers

\answer A math answer

```

1035 \*classXimera>
1036
1037 \ifdefined\HCode

```

```

1038 \newcommand{\recordvariable}[1]{}
1039 \else
1040 \newwrite\idfile
1041 \immediate\openout\idfile=\jobname.ids
1042 \newcommand{\recordvariable}[1]{\ifthenelse{\equal{#1}{}}{\immediate\write\idfile{var #1};}}
1043 \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

1044 \define@key{answer}{given}[true]{\def\ans@given{#1}}

Used for setting numeric answer tolerance for online student input.

1045 \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.

1046 \define@key{answer}{validator}{}

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

1047 \define@key{answer}{id}{\def\ans@id{#1}}

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

1048 \define@key{answer}{format}{}

Used to hide the answer input box on the web.

1049 \define@key{answer}{onlinenoinput}[false]{}

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

1050 \define@key{answer}{onlineshowanswerbutton}[false]{}

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

1051 \setkeys{answer}{id=,given=false,onlinenoinput=false,onlineshowanswerbutton=false}

Basic code for \answer.

1052
1053 % Options for handout
1054 \newcommand{\answerFormatLength}{2cm}
1055
1056 \newcommand{\answerFormatDots}[1]{\ldots\ldots}
1057 \newcommand{\answerFormatLine}[1]{\protect\rule{\answerFormatLength}{0.4pt}}
1058 \newcommand{\answerFormatFlexibleLine}[1]{\protect\rule{\widthof{#1$}*2}{0.4pt}}
1059 \newcommand{\answerFormatFlexibleBox}[1]{\fbox{\scalebox{2}{\phantom{#1$}}}}
1060
1061 % options for default (i.e with answers filled in)
1062 \newcommand{\answerFormatPlain}[1]{\ensuremath{#1}}
1063 \newcommand{\answerFormatBlue}[1]{\color{blue}\ensuremath{#1}}
1064 \newcommand{\answerFormatBoxed}[1]{\fbox{\ensuremath{#1}}}
1065 \newcommand{\answerFormatBoxedGiven}[1]{\underset{\scriptstyle\mathrm{given}}{\fbox{\ensuremath{#1}}}}
1066
1067 % defaults for handout and default mode, and for \answer[given]
1068 \let\handoutAnswerFormat\answerFormatDots
1069 \let\defaultAnswerFormat\answerFormatBlue
1070 \let\givenAnswerFormat\answerFormatBoxedGiven
1071
1072 \newcommand{\answer}[2][\%]
1073 \ifmmode%
1074 \setkeys{answer}{#1}%
1075 \recordvariable{\ans@id}
1076 \ifthenelse{\boolean{\ans@given}}
1077 {% Start then statement
1078 \ifhandout
1079 #2
1080 \else
1081 \givenAnswerFormat{#2} %% in case the argument helps formatting
1082 \fi
1083 }% End then statement

```



```

1084 {% Start else statement
1085 \ifhandout
1086 \handoutAnswerFormat{#2} %% in case the argument helps formatting
1087 \else% show answer in box outside handout mode
1088 \defaultAnswerFormat{#2} %% in case the argument helps formatting
1089 \fi
1090 }% End else statement
1091 \else%
1092 \GenericError{\space\space\space\space}% Throw an error based on... something? -- Jason
1093 {Attempt to use \@backslashchar answer outside of math mode}
1094 {See https://github.com/ximeraProject/ximeraLatex for explanation.}
1095 {Need to use either inline or display math.}%
1096 \fi
1097 }
1098 \endclassXimera

```

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

```

1099 \beginXimera
1100 \renewcommand{\answer}[2][false]{\HCode{<span class="answer_respondable">}#2\HCode{</span>}}
1101
1102 \def\validator[#1]{\stepcounter{identification}\HCode{<div class="validator" id="validator\arabic{identification}>#1</div>}}
1103 \def\endvalidator{\HCode{</div>}}
1104
1105 \endXimera

```

2.10.2 Multiple choice and the like

`multipleChoice` Multiple choice

```

1106 \beginXimera
1107 % Jim: Originally this was \renewcommand{\theenumi}{$(\mathrm{\alph{enumi}})$}
1108 % but that breaks tex4ht because mathmode can only be processed by mathjax.
1109 % so now I made this just italicized.

```

2.10.3 Options

```

1110 \define@key{choice}{value}[]{\def\choice@value{#1}}

```

This flags the answer as the correct answer

```

1111 \define@boolkey{choice}{correct}[true]{\def\choice@correct{#1}}

```

Use an ID to refer to the choice.

```

1112 \define@key{multipleChoice}{id}{\def\mc@id{#1}}

```

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

```

1113 \define@key{otherchoice}{value}[]{\def\otherchoice@value{#1}}

```

```

1114 \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".

```

1115 \setkeys{choice}{correct=false,value=}

```

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

```

1116 \setkeys{multipleChoice}{id=}

```

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

```

1117 \setkeys{otherchoice}{correct=false,value=}

```

```

1118 \endclassXimera

```

2.10.4 Choices

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

```

1119 \beginXimera
1120 \newcommand{\choice}[2][]{%
1121 \setkeys{choice}{#1}%
1122 \item{#2}

```

```

1123 \ifthenelse{\boolean{\choice@correct}}
1124   {% Begin then result
1125   \ifhandout% if it's a handout do nothing.
1126   \else% otherwise place a checkmark when you select the "correct choice"... maybe? -- Jason
1127   \,\checkmark\,\setkeys{choice}{correct=false}
1128   \fi
1129   }% End then result
1130   {}% Begin/End else result.
1131 }
1132
1133 %Define an expandable version of choice Not really meant to be used outside this package (used
1134 % Is there a reason we can't just always use this as default? -- Jason
1135 \newcommand{\choiceEXP}[2][]{%
1136 \expandafter\setkeys\expandafter{choice}{#1}%
1137 \item{#2}
1138 \ifthenelse{\boolean{\choice@correct}}
1139 {% Begin then result
1140 \ifhandout
1141 \else
1142 \,\checkmark\,\setkeys{choice}{correct=false}
1143 \fi
1144 }% End then result
1145 {}% Begin/End else result.
1146 } %% note all the {} are needed in case the choice has [] in it.
1147
1148 % \otherchoice is the \choice used in wordChoice command.
1149 \newcommand{\otherchoice}[2][]{%
1150 \ignorespaces%
1151 \setkeys{otherchoice}{#1}%
1152 \ifthenelse{\boolean{\otherchoice@correct}}%
1153 {% Start then result
1154 #2\ignorespaces\setkeys{otherchoice}{correct=false}\ignorespaces%
1155 }% End then result
1156 {}% Start/End else result
1157 \ignorespaces%
1158 }%
1159 \newcommand{\inlinechoice}[2][]{%
1160 \setkeys{choice}{#1}%
1161 \iffirstinlinechoice
1162 (\hspace{-.25em}
1163 \firstinlinechoicetrue
1164 \else
1165 /
1166 \fi
1167 #2
1168 \ifthenelse{\boolean{\choice@correct}}%
1169 {% Start then result
1170 \ifhandout\else\checkmark\ignorespaces\setkeys{choice}{correct=false}\ignorespaces\fi%
1171 }% End then result
1172 {}% Start/End else result
1173 \hspace{-.25em}\ignorespaces%
1174 }
1175
1176 </classXimera>
On the HTML side, \choice emits <span>s.
1177 <*htXimera>
1178 \newcounter{choiceId}
1179 \renewcommand{\choice}[2][]{%
1180 \setkeys{choice}{correct=false}%
1181 \setkeys{choice}{#1}%
1182 \stepcounter{choiceId}\IgnorePar%
1183 \HCode{<span class="choice }%
1184 \ifthenelse{\boolean{\choice@correct}}{\HCode{correct}}{}

```

```

1185 \HCode{" }
1186 \ifthenelse{\equal{\choice@value}{}}{\HCode{data-value="\choice@value" }}
1187 \HCode{id="choice\arabic{choiceId}">}%
1188 #2\HCode{</span>}}
1189 \let\inlinechoice\choice
1190 </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.

```

1191 \*classXimera>
1192 \newenvironment{multipleChoice}[1]{}
1193 {% Environment Start Code
1194 \setkeys{multipleChoice}{#1}%
1195 \recordvariable{\mc@id}%
1196 \begin{trivlist}
1197 \item[\hspace{1cm}\labelsep\small\bfseries Multiple Choice:] \hfil
1198 \begin{enumerate}
1199 }% Note this means that \item has to be the first line after \begin{multipleChoice}.
1200 {% Environment End Code
1201 \end{enumerate}
1202 \end{trivlist}
1203 }
1204
1205 %multipleChoice@ is for internal use only! (used in wordChoice)
1206 %this is simply a wrapper for the sole showing (other)choice.
1207 \newenvironment{multipleChoice@}[1]{}{}
1208 </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.

```

1209 \*htXimera>
1210 \renewenvironment{multipleChoice}[1]{}
1211 {\setkeys{multipleChoice}{#1}%
1212 \stepcounter{identification}\ifmode \IgnorePar\fi \EndP\HCode{<div class="multiple-choice" }
1213 \ifthenelse{\equal{\mc@id}{}}{\HCode{data-id="\mc@id" }}%
1214 \HCode{id="problem\arabic{identification}">}%
1215 }{\HCode{</div>}\IgnoreIndent}
1216 \ConfigureEnv{multipleChoice}{}{}{}
1217 </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.

```

1218 \*classXimera>
1219 \newcommand{\wordChoice}[1]{%
1220 \let\choicetemp\choice% Assign a "choicetemp" command to duplicate choice.
1221 \ifwordchoicegiven% If wordchoice option is on, we need to juggle around some definitions.
1222 \let\choice\otherchoice%
1223 %\begin{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1224 #1
1225 %\end{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1226 \else% If it isn't the regular "choice" command should work.
1227 \let\choice\inlinechoice%
1228 \begin{multipleChoice@}%
1229 #1%
1230 \end{multipleChoice@}%
1231 \fi%
1232 \let\choice\choicetemp% Now that choicetmp has been manipulated to what we want, replace choi
1233 }%

```

```

1234
1235
1236 </classXimera>
This is actually just word choice
1237 <*/htXimera>
1238 \renewenvironment{multipleChoice@}{\refstepcounter{problem}}{}%
1239 \ConfigureEnv{multipleChoice@}{\stepcounter{identification}\IgnorePar\HCode{<span class="word
1240 </htXimera>

```

2.12 Select all

```

selectAll A multiple-multiple choice question
1241 <*/classXimera>
1242 \newenvironment{selectAll}[1]{}
1243 {\begin{trivlist}\item[\hskip \labelsep\small\bfseries Select All Correct Answers:]\hfil\begin{
1244     {\end{enumerate}\end{trivlist}}
1245 </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`.

```

1246 <*/htXimera>
1247 \renewenvironment{selectAll}{\refstepcounter{problem}}{}%
1248 \ConfigureEnv{selectAll}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div
1249 </htXimera>

```

2.12.1 Free response

```

freeResponse A freeform input box.
1250 <*/classXimera>
1251 \newboolean{given} %% required for freeResponse
1252 \setboolean{given}{true} %% could be replaced by a key=value pair later if needed
1253
1254 \ifhandout
1255 \newenvironment{freeResponse}[1][false]%
1256 {%
1257 \def\givenatend{\boolean{#1}}
1258 \ifthenelse{\boolean{#1}}
1259 {% Begin then result
1260 \begin{trivlist}
1261 \item
1262 }% End then result
1263 {% Begin else result
1264 \setbox0\vbox\bgroup
1265 }% End else result
1266 % {}% Don't think this is doing anything? -- Jason
1267 }
1268 {%
1269 \ifthenelse{\givenatend}
1270 {% Begin then result
1271 \end{trivlist}
1272 }% End then result
1273 {% Begin else result
1274 \egroup
1275 }% End else result
1276 % {}% Don't think this is doing anything? -- Jason
1277 }
1278 \else
1279 \newenvironment{freeResponse}[1][false]%

```

```

1280 {% Environment Beginning Code
1281 \ifthenelse{\boolean{#1}}{% Could probably change this with just putting the (given) in the
1282   {% Begin then result
1283   \begin{trivlist}
1284   \item[\hspace{1cm} \labelsep\bfseries Free Response (Given):\hspace{2ex}]
1285   }% End then result
1286 {% Begin else result
1287 \begin{trivlist}
1288 \item[\hspace{1cm} \labelsep\bfseries Free Response:\hspace{2ex}]
1289 }% End else result
1290 }
1291 {% Environment Ending Code
1292 \end{trivlist}
1293 }
1294 \fi
1295
1296 \end{classXimera}
1297 \begin{htXimera}
1298
1299 \renewenvironment{freeResponse}{\refstepcounter{problem}}{%
1300 \ConfigureEnv{freeResponse}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<
1301
1302 \end{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.

```

1303 \begin{classXimera}
1304 \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.

```

1305 \newenvironment{validator}[1][]{
1306 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter" to
1307 \mbox{\texttt{\detokenize\expandafter{\PH@Command}}}% Now expand PH@Command once and then det
1308 }{}

```

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

```

1309 \ifhandout%
1310 \newenvironment{feedback}
1311   {%
1312   \setbox0\vbox\bgroup
1313   }
1314   {%
1315   \egroup
1316   }

```

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.

```

1317 \else
1318 \newenvironment{feedback}[1][attempt]{
1319
1320 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter" to
1321

```

```

1322 \begin{trivlist}% Begin the trivlist to use formatting of the "Feedback" label.
1323 \item[\hspace{1cm}\labelsep\small\slshape\bfseries Feedback% Format the "Feedback" label. Don't fo
1324 (\texttt{\detokenize\expandafter{\PH@Command}}):% Format (and detokenize) the condition for
1325 \hspace{2ex}]\small\slshape% Insert some space before the actual feedback given.
1326 }{
1327 \end{trivlist}
1328 }
1329
1330 \fi
1331 \end{classXimera}

```

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

```

1332 \begin{classXimera}
1333 \def\feedback{\@ifnextchar[{\@feedbackcode}{\@feedbackattempt}}
1334 \def\@feedbackattempt{\@feedbackcode[attempt]}
1335 \def\@feedbackcode[#1]{\stepcounter{identification}%
1336 \ifvmode \IgnorePar\fi \EndP%
1337 \ifthenelse{\equal{#1}{attempt}}{\HCode{<div class="feedback" data-feedback="attempt" id="fe
1338 {\ifthenelse{\equal{#1}{correct}}{\HCode{<div class="feedback" data-feedback="correct" id="f
1339 {\HCode{<div class="feedback" data-feedback="script" id="feedback\arabic{identification}"><s
1340 \def\endfeedback{\HCode{</div>}}\IgnoreIndent}
1341 \end{classXimera}

```

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.

```

1342 \begin{classXimera}
1343 \newenvironment{ungraded}{}{}
1344 \end{classXimera}

```

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

```

1345 \begin{classXimera}
1346 \renewenvironment{ungraded}{%
1347 \ifvmode \IgnorePar\fi \EndP\HCode{<div class="ungraded">}}\IgnoreIndent%
1348 }{
1349 \ifvmode \IgnorePar\fi \EndP\HCode{</div>}}\IgnoreIndent%
1350 }
1351 \end{classXimera}

```

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.

```

1352 \begin{classXimera}
1353 \ifdefined\HCode
1354 \else
1355 \newwrite\myfile
1356 \immediate\openout\myfile=\jobname.jax
1357 \fi
1358 \end{classXimera}

```

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

```

1359 \begin{classXimera}
1360 \ifdefined\HCode
1361 \else
1362 \immediate\write\myfile{\unexpanded{\newenvironment}{prompt}{}{}}
1363 \fi

```

```

1364 </classXimera>
Redefine newcommand appropriately.
1365 <*classXimera>
1366 \ifdefined\HCode
1367   \else
1368   \let\@oldargdef\@argdef
1369   \long\def\@argdef#1[#2]#3{%
1370   \immediate\write\myfile{\unexpanded{\newcommand}{\unexpanded{#1}}[\unexpanded{#2}]{\unexpanded{#3}}}%
1371   \@oldargdef#1[#2]{#3}%
1372   }
1373
1374 \let\@oldDeclareMathOperator\DeclareMathOperator
1375 \renewcommand{\DeclareMathOperator}[2]{\@oldDeclareMathOperator{#1}{#2}\immediate\write\myfile{#1 #2}%}
1376
1377 \fi
1378 </classXimera>
Include the jax'ed newcommands
1379 <*cfgXimera>
1380 % Remove commands that use @
1381 \immediate\write18{sed -i "/@/d" \jobname.jax}
1382 % Replace ##1 with #1 and so forth
1383 \immediate\write18{sed -i "s/\string#\string#\string\\([0-9]\string\\)/\string#\string\\1/g"}
1384
1385 \Configure{BVerbatimInput}{-}{-}{-}
1386
1387 \Configure{verbatiminput}{-}{-}{-}
1388
1389 % Instead of a nonbreaking space, use a standard space
1390 \makeatletter
1391 \def\FV@Space{\space}
1392 \makeatother
1393
1394 % Include the mathjax newcommands in a math/tex script right at the beginning of the body
1395 \Configure{BODY}{-%
1396 \HCode{<body>\Hnewline}%
1397 \Tg<div class="preamble">%
1398 \IfFileExists{\jobname.jax}{
1399 \Tg<script type="math/tex">%
1400 \BVerbatimInput{\jobname.jax}%
1401 \Tg</script>%
1402 }
1403 {\Hnewline\HCode{<!-- mm, no \newcommands provided -->}\Hnewline}
1404
1405 \IfFileExists{\jobname.ids}{\HCode{<script type="text/javascript">\Hnewline}%
1406 \BVerbatimInput{\jobname.ids}%
1407 \HCode{</script>\Hnewline}%
1408 }{}
1409 \Tg</div>%
1410 }{-%
1411 \ifvmode\IgnorePar\fi\EndP\HCode{</body>\Hnewline}%
1412 }
1413
1414 % This is a fix for the LAODE book, which uses matlabEquation as if it were an equation
1415 \ScriptEnv{matlabEquation}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="math/tex; mode=d
1416
1417 </cfgXimera>

```

2.13.2 Semantic HTML

`\textbf` Using `\textbf` emits a `` tag.

```

1418 <*cfgXimera>
1419 \Configure{textbf}{\ifvmode\ShowPar\fi\HCode{<strong>}}{\HCode{</strong>}}

```

```

1420 </cfgXimera>
\textit Using \textit or similar emits an <em> tag.
1421 <*cfgXimera>
1422 \Configure{textit}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1423 \Configure{emph}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1424 </cfgXimera>
\texttt Using \texttt emits a <code> tag.
1425 <*cfgXimera>
1426 \Configure{texttt}{\ifvmode\ShowPar\fi\HCode{<code>}}{\HCode{</code>}}
1427 </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`.

```

1428 <*classXimera>
1429 \font\dummyft@=dummy \relax
1430 \def\suppress{%
1431   \begingroup\par
1432   \parskip\z@
1433   \offinterlineskip
1434   \baselineskip=\z@skip
1435   \lineskip=\z@skip
1436   \lineskiplimit=\maxdimen
1437   \dummyft@
1438   \count@\sixt@@n
1439   \loop\ifnum\count@ >\z@
1440     \advance\count@\m@ne
1441     \textfont\count@\dummyft@
1442     \scriptfont\count@\dummyft@
1443     \scriptscriptfont\count@\dummyft@
1444   \repeat
1445   \let\selectfont\relax
1446   \let\mathversion\@gobble
1447   \let\getanddefine@fonts\@gobbletwo
1448   \tracinglostchars\z@
1449   \frenchspacing
1450   \hbadness\@M}
1451 \def\endsuppress{\par\endgroup}
1452 </classXimera>

```

2.14.2 The End

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

```

1453 <*htXimera>
1454 \Hinput{ximera}
1455 </htXimera>

1456 <*htXourse>
1457 \Hinput{xourse}
1458 </htXourse>

1459 <*cfgXimera>
1460 \begin{document}
1461 \EndPreamble
1462 </cfgXimera>

```


3 xourse.cls

```

1463 \*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.
1464 \newif\ifnotoc
1465 \notocfalse
1466 \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.
1467 \newif\ifnonewpage
1468 \nonewpagefalse
1469 \DeclareOption{nonewpage}{\nonewpagetrue}

1470 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{ximera}}
1471 \ProcessOptions\relax
1472 \LoadClass{ximera}
1473 % \begin{macrocode}
1474 \end{macrocode}

```

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`.

```

1475 \*classXourse\
1476 \newcommand{\skip@preamble}{%
1477   \let\document\relax\let\enddocument\relax%
1478   \newenvironment{document}{\let\input\otherinput}{}%
1479   \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:

```

1480 \let\otherinput\input
Store usual \maketitle as \othermaketitle
1481 \let\othermaketitle\maketitle

\maketitle In a xourse file, \maketitle is redefined to give course packet title page and toc.
1482 \renewcommand{\maketitle}{ %
1483 \pagestyle{empty}
1484 \begin{center}
1485 ~\ %puts space at top of page to move title down.
1486 \vskip .25\textheight
1487 \hrulefill\
1488 \vskip 1em
1489 \bfseries{\Huge \@title} \
1490 \hrulefill\
1491 \vskip 3em
1492 {\Large \@author}
1493 \vskip 2em
1494 {\large \@date}
1495 \end{center}
1496 \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.

```

1497 \ifnotoc
1498 \else
1499   \tableofcontents\clearpage

```

```

1500 \clearpage
1501 \fi

```

Switch to main pagestyle, just like a document with documentclass ximera.

```
1502 \pagestyle{main}
```

Renew maketitle to usual definition.

```
1503 \let\maketitle\othermaketitle
```

And we finish with our redefinition of \maketitle.

```

1504 }
1505 \relax
1506 \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.

```

1507 \begin{classXourse}
1508 \ifnonepage
1509 \newcommand{\activity}[2][]{\%
1510 \setkeys{activity}{#1}
1511 \renewcommand{\input}[1]{
1512 \begin{group}\skip@preamble\otherinput{#2}\end{group}\par\vspace{\topsep}
1513 \let\input\otherinput}
1514 \else
1515 \newcommand{\activity}[2][]{\%
1516 \setkeys{activity}{#1}
1517 \renewcommand{\input}[1]{
1518 \begin{group}\skip@preamble\otherinput{#2}\end{group}\clearpage
1519 \let\input\otherinput}
1520 \fi
1521 \relax
1522 \end{classXourse}

1523 \begin{htXourse}
1524 \renewcommand{\activity}[2][]{\%
1525 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="activity card \activitystyle" href="#2" data-opt
1526 }
1527 \end{htXourse}

```

When running xake, we can just ignore activities

```

1528 \begin{classXourse}
1529 \ifxake
1530 \renewcommand{\activity}[2][]{\%
1531 \fi
1532 \end{classXourse}

```

3.1.2 Practice activities

\practice Like \activity but not expecting a title.

```

1533 \begin{classXourse}
1534 \ifhandout
1535 \newcommand{\practice}[2][]{\%
1536 \setkeys{practice}{#1}%!!!!
1537 \renewcommand{\input}[1]{
1538 \begin{group}\skip@preamble\otherinput{#2}\end{group}
1539 \let\input\otherinput}
1540 \else

```

```

1541 \newcommand{\practice}[2][\texttt{\detokenize{#2}}}% gives file name for practice
1542 \setkeys{practice}{#1}%!!!!
1543 \renewcommand{\input}[1]{
1544 \begingroup\skip@preamble\otherinput{#2}\endgroup
1545 \let\input\otherinput}
1546 \fi
1547 \relax
1548 \end{classXourse}

```

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

```

1549 \end{classXourse}
1550 \ifxake
1551 \renewcommand{\practice}[2][{}]{
1552 \fi
1553 \end{classXourse}

```

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

```

1554 \end{htXourse}
1555 \renewcommand{\practice}[2][{}]{
1556 \ifvmode\IgnorePar\fi\EndP%
1557 \HCode{<a class="activity card practice" href="#2" data-options="#1">#2</a>}%
1558 \IgnoreIndent%
1559 }
1560 \end{htXourse}

```

3.2 Sectioning

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.

```

\section
1561 \end{classXourse}
1562 \renewcommand*\l@section{\@dottedtocline{1}{1.5em}{4.2em}}
1563 \end{classXourse}

```

\subsection The name of a subsection inside an activity.

```

1564 \end{classXourse}
1565 \renewcommand*\l@subsection{\@dottedtocline{2}{3.8em}{4.2em}}
1566 \end{classXourse}

```

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

```

1567 \end{htXourse}
1568 \newcounter{ximera@part}
1569 \setcounter{ximera@part}{0}
1570 \renewcommand\part[1]{%
1571 \stepcounter{ximera@part}%
1572 \ifvmode \IgnorePar\fi \EndP%
1573 %\HCode{<h1 id="part\arabic{ximera@part}" class="card part">#1\HCode{</h1>}}% makes cards dis
1574 \HCode{<h1 id="part\arabic{ximera@part}" class="card part">#1</h1>}}%
1575 \IgnoreIndent%
1576 }
1577 \end{htXourse}

```

\paragraph Paragraph commands emit spans. A small heading.

```

1578 \end{cfgXimera}
1579 \renewcommand{\paragraph}[1]{%
1580 \HCode{<span class="paragraphHead">#1%
1581 #1%
1582 \HCode{</span>}}\par\IgnorePar}
1583 \end{cfgXimera}

```

\subparagraph An even smaller heading.

```

1584 \end{cfgXimera}
1585 \renewcommand{\subparagraph}[1]{%
1586 \HCode{<span class="subparagraphHead">#1%

```

```

1587 #1%
1588 \HCode{</span>}\par\IgnorePar}
1589 </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.

```

1590 <*classXourse>
1591 \newenvironment{graded}[1]{%{}
1592 </classXourse>

```

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

```

1593 <*htXourse>
1594 \renewenvironment{graded}[1]{%
1595 \ifvmode \IgnorePar\fi \EndP\HCode{<div class="graded" data-weight="#1">}\IgnoreIndent%
1596 }{
1597 \ifvmode \IgnorePar\fi \EndP\HCode{</div>}\IgnoreIndent%
1598 }
1599 </htXourse>

```

3.4 Logos

\logo A logo for the xourse.

```

1600 <*classXourse>
1601 \newcommand*{\logo}[1]{%
1602 \ifx\@onlypreamble\@notprerr
1603 \ClassError{xourse}{logo can only be used in the preamble}
1604 {Move your logo command to the preamble}
1605 \else %
1606 \IfFileExists{#1}%
1607 {\gdef\xourse@logo{#1}}%
1608 {\ClassError{xourse}{logo file does not exist}
1609 {To use logo, make sure that the referenced image file exists}}%
1610 \fi%
1611 }
1612
1613 </classXourse>

```

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

```

1614 <*htXourse>
1615 \Configure{@HEAD}{%
1616 \HCode{<meta name="og:image" content="}%
1617 \ifdefined\xourse@logo%
1618 \xourse@logo%
1619 \fi%
1620 \HCode{" />\Hnewline}}%
1621 </htXourse>

```