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 handouts, 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, **ER**esources, for **A**ll) is an open-source platform that provides tools for authoring and publishing (PDF and Online), open-source, interactive educational content, such as textbooks, assessments, and online courses. 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 see 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 supress such content and generate a reasonable printiable "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 supress 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 supress the listing of the learning outcomes.

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

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

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

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

```
14 \DeclareOption{noinstructornotes}{\instructornotestrue}
```

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* 
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
```

3

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

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

```
88 ⟨∗classXimera⟩
89 \RequirePackage{gettitlestring}
90 \RequirePackage{nameref}
91 \RequirePackage{epstopdf}
92 ⟨/classXimera⟩
```

2.3 Page setup

We want non-indented spaced-out paragraphs.

```
93 ⟨∗classXimera⟩
94 \setlength{\parindent}{0pt}
95 \setlength{\parskip}{5pt}
96 ⟨/classXimera⟩
```

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

```
 97 ⟨∗classXimera⟩
 98 \oddsidemargin 62pt
 99 \evensidemargin 62pt
100 \textwidth 345pt
101 \headheight 14pt
102 ⟨/classXimera⟩
```

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

```
103 ⟨∗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.
121 \Configure{@HEAD}{\HCode{<meta name="ximera" content="version 2.5.1" />\Hnewline}}
122 \Configure{@HEAD}{\HCode{<link href="https://ximera.osu.edu/public/stylesheets/standalone.css
123 \Configure{@HEAD}{\HCode{<script type="text/javascript" async src="https://ximera.osu.edu/pub
124
125 % OVERWRITE css in ximera-server (to be removed whenever this has been fixed in the server; 1
126 \catcode`\%=11
127 \Configure{@BODY}{\HCode{<style>
128 .activity-body pre {
129     white-space: pre;
130     background-color: lightgray;
131 }
132 .xmyoutube {
133     aspect-ratio: 16/9;
134     min-width: 75%;
```

```
135 }
136 .image-environment img {
137     width: unset;
138 }
139 </style>\Hnewline}}
140 \catcode`\%=14
141
142 ⟨/cfgXimera⟩
```

Disable certain ligatures in HTML.

```
143 ⟨∗htXimera⟩
144 \usepackage{microtype}
145 \DisableLigatures[f]{encoding=*}
146 ⟨/htXimera⟩
```

I am not sure what this does.

```
147 ⟨∗htXimera⟩
148 \NewEnviron{html}{\HCode{\BODY}}
149 ⟨/htXimera⟩
```

2.4 Structure

2.4.1 Macros

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

```
150 ⟨∗classXimera⟩
151 \everymath{\displaystyle}
152 ⟨/classXimera⟩
```

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

```
153 ⟨∗classXimera⟩
154 \let\prelim\lim
155 \renewcommand{\lim}{\displaystyle\prelim}
156 ⟨/classXimera⟩
```

2.4.2 Theorem and theorem-like environments

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

```
157 ⟨∗htXimera⟩
158 \newcommand{\ConfigureTheoremEnv}[1]{%
159 \renewenvironment{#1}[1][]{\refstepcounter{problem}%
160 \ifthenelse{\equal{##1}{}}{}{%
161   \HCode{<span class="theorem-like-title">}##1\HCode{</span>}%
162 }}{}
163 \ConfigureEnv{#1}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div class='
164 }
165 ⟨/htXimera⟩
166 ⟨classXimera⟩\theoremstyle{definition} % No italic (because this makes also text in TikZ itali
```

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

theorem (*env.*)   Theorem
```
167 ⟨classXimera⟩        \newtheorem{theorem}{Theorem}
168 ⟨htXimera⟩           \ConfigureTheoremEnv{theorem}
```

algorithm (*env.*)   Algorithm
```
169 ⟨classXimera⟩        \newtheorem{algorithm}{Algorithm}
170 ⟨htXimera⟩           \ConfigureTheoremEnv{algorithm}
```

axiom (*env.*)   Axiom
```
171 ⟨classXimera⟩        \newtheorem{axiom}{Axiom}
172 ⟨htXimera⟩           \ConfigureTheoremEnv{axiom}
```

claim (*env.*)   Claim
```
173 ⟨classXimera⟩        \newtheorem{claim}{Claim}
174 ⟨htXimera⟩           \ConfigureTheoremEnv{claim}
```

conclusion (*env.*) Conclusion

175 ⟨classXimera⟩   \newtheorem{conclusion}{Conclusion}
176 ⟨htXimera⟩   \ConfigureTheoremEnv{conclusion}

condition (*env.*) Condition

177 ⟨classXimera⟩   \newtheorem{condition}{Condition}
178 ⟨htXimera⟩   \ConfigureTheoremEnv{condition}

conjecture (*env.*) Conjecture

179 ⟨classXimera⟩   \newtheorem{conjecture}{Conjecture}
180 ⟨htXimera⟩   \ConfigureTheoremEnv{conjecture}

corollary (*env.*) Corollary

181 ⟨classXimera⟩   \newtheorem{corollary}{Corollary}
182 ⟨htXimera⟩   \ConfigureTheoremEnv{corollary}

criterion (*env.*) Criterion

183 ⟨classXimera⟩   \newtheorem{criterion}{Criterion}
184 ⟨htXimera⟩   \ConfigureTheoremEnv{criterion}

definition (*env.*) Definition

185 ⟨classXimera⟩   \newtheorem{definition}{Definition}
186 ⟨htXimera⟩   \ConfigureTheoremEnv{definition}

example (*env.*) Example

187 ⟨classXimera⟩   \newtheorem{example}{Example}
188 ⟨htXimera⟩   \ConfigureTheoremEnv{example}

explanation (*env.*) Explanation

189 ⟨classXimera⟩   \newtheorem*{explanation}{Explanation}
190 ⟨htXimera⟩   \ConfigureTheoremEnv{explanation}

fact (*env.*) Fact

191 ⟨classXimera⟩   \newtheorem{fact}{Fact}
192 ⟨htXimera⟩   \ConfigureTheoremEnv{fact}

lemma (*env.*) Lemma

193 ⟨classXimera⟩   \newtheorem{lemma}{Lemma}
194 ⟨htXimera⟩   \ConfigureTheoremEnv{lemma}

formula (*env.*) Formula

195 ⟨classXimera⟩   \newtheorem{formula}{Formula}
196 ⟨htXimera⟩   \ConfigureTheoremEnv{formula}

idea (*env.*) Idea

197 ⟨classXimera⟩   \newtheorem{idea}{Idea}
198 ⟨htXimera⟩   \ConfigureTheoremEnv{idea}

notation (*env.*) Notation

199 ⟨classXimera⟩   \newtheorem{notation}{Notation}
200 ⟨htXimera⟩   \ConfigureTheoremEnv{notation}

model (*env.*) Model

201 ⟨classXimera⟩   \newtheorem{model}{Model}
202 ⟨htXimera⟩   \ConfigureTheoremEnv{model}

observation (*env.*) Observation

203 ⟨classXimera⟩   \newtheorem{observation}{Observation}
204 ⟨htXimera⟩   \ConfigureTheoremEnv{observation}

proposition (*env.*) Proposition

205 ⟨classXimera⟩   \newtheorem{proposition}{Proposition}
206 ⟨htXimera⟩   \ConfigureTheoremEnv{proposition}

paradox (*env.*) Paradox

207 ⟨classXimera⟩   \newtheorem{paradox}{Paradox}
208 ⟨htXimera⟩   \ConfigureTheoremEnv{paradox}

procedure (*env.*) Procedure

209 ⟨classXimera⟩   \newtheorem{procedure}{Procedure}
210 ⟨htXimera⟩   \ConfigureTheoremEnv{procedure}

remark (*env.*)   Remark

211 ⟨classXimera⟩   \newtheorem{remark}{Remark}
212 ⟨htXimera⟩   \ConfigureTheoremEnv{remark}

summary (*env.*)   Summary

213 ⟨classXimera⟩   \newtheorem{summary}{Summary}
214 ⟨htXimera⟩   \ConfigureTheoremEnv{summary}

template (*env.*) Template

215 ⟨classXimera⟩   \newtheorem{template}{Template}
216 ⟨htXimera⟩   \ConfigureTheoremEnv{template}

warning (*env.*)   Warning

217 ⟨classXimera⟩   \newtheorem{warning}{Warning}
218 ⟨htXimera⟩   \ConfigureTheoremEnv{warning}

2.4.3 Enumerate fixes

Make enumerate use a letter

219 ⟨∗classXimera⟩
220 \renewcommand{\theenumi}{\textup{(\alph{enumi})}}
221 \renewcommand{\labelenumi}{\theenumi}
222 \renewcommand{\theenumii}{\textup{(\roman{enumii})}}
223 \renewcommand{\labelenumii}{\theenumii}
224 ⟨/classXimera⟩

2.4.4 Proofs

proof (*env.*) A mathematical proof environment.

225 ⟨∗classXimera⟩
226 \renewcommand{\qedsymbol}{$\blacksquare$}
227 \renewenvironment{proof}[1][\proofname]
228   {\begin{trivlist}\item[\hskip \labelsep \itshape \bfseries #1{}\hspace{2ex}]}
229 {\qed\end{trivlist}}
230 ⟨/classXimera⟩
231 ⟨∗htXimera⟩
232   % Mmm, (why) do we want/need this ...?
233   \ConfigureTheoremEnv{proof}
234 \ConfigureEnv{proof}{\ifvmode\IgnorePar\fi\EndP\HCode{<div class="proof">}}
235 \ConfigureList{trivlist}{\ifvmode\IgnorePar\fi\EndP}{}{}{}
236 }{\ifvmode\IgnorePar\fi\EndP\HCode{</div>}}{}{}
237 ⟨/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

238 ⟨∗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'.

239 \providecommand{\latexProblemContent}[1]{#1}
240 % Iterate count for problem counts.
241 \Make@Counter{Iteration@probCnt}

242 \newcommand{\hang}{% top theorem decoration
243   \begingroup%
244   \setlength{\unitlength}{.005\linewidth}% \linewidth/200
245   \begin{picture}(0,0)(1.5,0)%
246   \linethickness{1pt} \color{black!50}%
247   \put(-3,2){\line(1,0){206}}% Top line
248   \multido{\iA=2+-1,\iB=50+-10}{5}{% Top hangs

7

```
249          \color{black!\iB}%
250          \put(-3,\iA){\line(0,-1){1}}% Top left hang
251          %\put(203,\iA){\line(0,-1){1}}% Top right hang
252        }%
253      \end{picture}%
254    \endgroup%
255 }%
256 \newcommand{\hung}{% bottom theorem decoration
257   \nobreak
258   \begingroup%
259     \setlength{\unitlength}{.005\linewidth}% \linewidth/200
260     \begin{picture}(0,0)(1.5,0)%
261       \linethickness{1pt} \color{black!50}%
262       \put(60,0){\line(1,0){143}}% Bottom line
263       \multido{\iA=0+1,\iB=50+-10}{5}{% Bottom hangs
264         \color{black!\iB}%
265         %\put(-3,\iA){\line(0,1){1}}% Bottom left hang
266         \put(203,\iA){\line(0,1){1}}% Bottom right hang
267         \put(\iB,0){\line(60,0){10}}% Left fade out
268       }%
269     \end{picture}%
270   \endgroup%
271 }%
```

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.

```
272 \MakeCounter{problem}
273 \newcommand{\problemNumber}{
274 % First we determine if we have a counter for this question depth level.
275 \ifcsname c@depth\Roman{problem@Depth}Count\endcsname% Check to see if counter exists
276 %If so, do nothing.
277 \else
278 %If not, create it.
279 \expandafter\newcounter{depth\Roman{problem@Depth}Count}
280 \expandafter\setcounter{depth\Roman{problem@Depth}Count}{0}
281 \fi
282
283 \expandafter\stepcounter{depth\Roman{problem@Depth}Count}
284 \arabic{depthICount}% The first problem depth, what use to be |\theproblem|.
285
286 \forloop{Iteration@probCnt}{2}{\arabic{Iteration@probCnt} < \numexpr \value{problem@Depth} +
287     .\expandafter\arabic{depth\Roman{Iteration@probCnt}Count}% Get the problem number of the
288 }
289 %\@ifpackageloaded{shuffle}{<true>}{<false>}% Check if Shuffle has been added. If so, add spe
290 %\ifhandout % Currently handout mode doesn't allow hints. Putting this code in place in case
291 % \theproblem
292 %\else
293 % \theproblem
294 %\fi
295 }
296
297
298 %%%%% Configure various problem environment commands
299 \Make@Counter{problem@Depth}
300
301
302
303 %%%% Configure environments start content
304
305 \newcommand{\problemEnvironmentStart}[2]{%
306 % This takes in 2 arguments.
307 % The first is optional and is the old optional argument from existing environments.
308 % This is passed down to the associated problem environment name in case you want a global va
```

```latex
309 % The second argument is mandatory and is the name of the 'problem' environment,
310 % such as problem, question, exercise, etc.
311 % It then configures everything needed at the start of that environment.
312
313 \stepcounter{problem@Depth}% Started a problem, so we've sunk another problem layer.
314 \def\spaceatend{#1}%
315 \begin{trivlist}%
316 \item%
317   [%
318     \hskip\labelsep\sffamily\bfseries
319     #2 \problemNumber% Determine the correct number of the problem, and the format of that nu
320 ]%
321 \slshape
322 }
323
324
325
326 %%%% Configure environments end content
327
328 \newcommand{\problemEnvironmentEnd}{%This configures all the end content for a problem.
329 %
330 % First we need to see if we've dropped fully out of a depth level,
331 % so we can reset that counter back to zero for the next time we enter that depth level.
332 \stepcounter{problem@Depth}
333 \ifcsname c@depth\Roman{problem@Depth}Count\endcsname
334  \expandafter\ifnum\expandafter\value{depth\Roman{problem@Depth}Count}>0
335   \expandafter\setcounter{depth\Roman{problem@Depth}Count}{0}
336  \fi
337 \fi
338
339 \addtocounter{problem@Depth}{-2}% Exited a problem so we've exited a problem layer. Need -2 b
340
341 % 202410: no longer \par\addvspace{.5ex}\nobreak\noindent\hung %% line at the bottom
342
343 \ifhandout
344  \ifnewpage
345   \newpage
346  \fi
347 \fi
348 \end{trivlist}
349 }
350
351
352
353 %%%% Now populate the old environment names
354 %
355 % Old environments were "problem", "exercise", "exploration", and "question".
356 % Note that you can add content to the start/end code on top of these base code pieces if you
357 %
358 % These definitions will be overwritten in ximera.4ht !
359
360
361 \newenvironment{problem}[1][2in]%
362 {%Env start code
363 \problemEnvironmentStart{#1}{Problem}
364 }
365 {%Env end code
366 \problemEnvironmentEnd
367 }
368
369 \newenvironment{exercise}[1][2in]%
370 {%Env start code
371 \problemEnvironmentStart{#1}{Exercise}
```

```
372 }
373 {%Env end code
374 \problemEnvironmentEnd
375 }
376
377 \newenvironment{exploration}[1][2in]%
378 {%Env start code
379 \problemEnvironmentStart{#1}{Exploration}
380 }
381 {%Env end code
382 \problemEnvironmentEnd
383 }
384
385 \newenvironment{question}[1][2in]%
386 {%Env start code
387 \problemEnvironmentStart{#1}{Question}
388 }
389 {%Env end code
390 \problemEnvironmentEnd
391 }
392 ⟨/classXimera⟩
```

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

```
393 ⟨∗htXimera⟩
394 \newcounter{identification}
395 \setcounter{identification}{0}
396
397 % 2024: should perhaps better have been called \ConfigureProblemEnv ...??
398 \newcommand{\ConfigureQuestionEnv}[2]{%
399 % refstepcounter ensures that labels get updated within these environments
400 \renewenvironment{#1}{\refstepcounter{problem}}{}%
401 \ConfigureEnv{#1}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div role="a
402 }
403
404 \ConfigureQuestionEnv{problem}{problem}
405 \ConfigureQuestionEnv{exercise}{exercise}
406 \ConfigureQuestionEnv{question}{question}
407 \ConfigureQuestionEnv{exploration}{exploration}
408
409 \ifdefined\xmNotHintAsExpandable
410   \ConfigureQuestionEnv{hint}{hint}    % 2024: hint is no longer a 'question-environment'.
411 \fi
412 %%%%\ConfigureQuestionEnv{shuffle}{shuffle}
413 ⟨/htXimera⟩
```

2.4.6   Hints

hint (*env.*) Hint environments can be embedded inside problems.

```
414 ⟨∗classXimera⟩
```

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

```
415 \newcounter{hintLevel}
416 \setcounter{hintLevel}{0}
```

Create an empty shell to renew

```
417 \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.

```
418 \renewenvironment{hint}
419 {
420  \ifhandout
```

```
421    \setbox0\vbox\bgroup
422    \else
423    \begin{trivlist}\item[\hskip \labelsep\small\slshape\bfseries Hint:\hspace{2ex}]
424    \small\slshape
425    \fi
```

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

```
426    \stepcounter{hintLevel}
427    }
428    {
429    \ifhandout
430     \egroup\ignorespacesafterend
431    \else
432     \end{trivlist}
433    \fi
```

Detract from hint level counter to track hint nested level

```
434    \addtocounter{hintLevel}{-1}
435    }
436
437    \ifhints
438    \renewenvironment{hint}{
439    \begin{trivlist}\item[\hskip \labelsep\small\slshape\bfseries Hint:\hspace{2ex}]
440    \small\slshape}
441    {\end{trivlist}}
442    \fi
443
444    ⟨/classXimera⟩
```

2.4.7 Solution

solution (*env.*) The solution to a problem.

```
445    ⟨*classXimera⟩
446    %% solution environment
447    \ifhandout % what follows is handout behavior
448    \newenvironment{solution}%
449            {%
450        \setbox0\vbox\bgroup
451            }
452                    {%
453        \egroup
454            }
455    \else
456    \newenvironment{solution}%
457            {%
458        \begin{trivlist}
459        \item[\hskip \labelsep\bfseries Solution:\hspace{2ex}]
460            }
461            % %% line at the bottom}
462            {
463        \end{trivlist}
464        % (202410: no longer \par\addvspace{.5ex}\nobreak\noindent\hung
465            }
466    \fi
467
468
469
470    ⟨/classXimera⟩
```

2.4.8 Code listing environments

code (*env.*) A code answer environment You cannot use Environ with the fancyvrb/listings package

if you want nested environments.

```
471 ⟨*classXimera⟩
472 \DefineVerbatimEnvironment{code}{Verbatim}{numbers=left,frame=lines,label=Code,labelposition=
473 ⟨/classXimera⟩
```

python (*env.*) A python answer environment You cannot use Environ with the fancyvrb/listings package
if you want nested environments

```
474 ⟨*classXimera⟩
475 \DefineVerbatimEnvironment{python}{Verbatim}{numbers=left,frame=lines,label=Python,labelposit
476 ⟨/classXimera⟩
```

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

```
477 ⟨*classXimera⟩
478 \DefineVerbatimEnvironment{javascriptCode}{Verbatim}{numbers=left,frame=lines,label=JavaScri
479 ⟨/classXimera⟩
480 ⟨*cfgXimera⟩
481 \renewenvironment{javascriptCode}{\NoFonts}{\EndNoFonts}
482 \ScriptEnv{javascriptCode}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<d
483 ⟨/cfgXimera⟩
```

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

```
484 %%%<*cfgXimera>
485 %%\ConfigureEnv{verbatim}{\ifvmode\IgnorePar\fi\EndP\HCode{<pre style="white-space: pre; back
486 %%\ConfigureEnv{lstlisting}{\ifvmode\IgnorePar\fi\EndP\HCode{<pre>}}{\ifvmode\IgnorePar\fi\En
487 %%%</cfgXimera>
488 %%
```

2.4.9 Dialogues

dialogue (*env.*) A dialogue between people.

```
489 ⟨*classXimera⟩
490 \newenvironment{dialogue}{%
491     \renewcommand\descriptionlabel[1]{\hspace{\labelsep}\textbf{##1:}}
492     \begin{description}%
493 }{%
494     \end{description}%
495 }
496 ⟨/classXimera⟩
```

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

```
497 ⟨*htXimera⟩
498 \renewenvironment{dialogue}{\begin{description}}{\end{description}}
499
500 \ConfigureList{dialogue}%
501     {\EndP\HCode{<dl \a:LRdir class="dialogue">}%
502         \PushMacro\end:itm
503 \global\let\end:itm=\empty}
504     {\PopMacro\end:itm \global\let\end:itm \end:itm
505 \EndP\HCode{</dd></dl>}\ShowPar}
506     {\end:itm \global\def\end:itm{\EndP\Tg</dd>}\HCode{<dt
507         class="actor">}\bgroup \bf}
508     {\egroup\EndP\HCode{</dt><dd\Hnewline class="speech">}}
509 ⟨/htXimera⟩
```

2.4.10 Instructor notes

```
510 ⟨*classXimera⟩
511
512 %% instructor intro/instructor notes
513 %%
514 \ifhandout % what follows is handout behavior
515 \ifinstructornotes
516 \newenvironment{instructorIntro}%
```

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

```
580        {%
581          \egroup
582        }
583                         \fi
584 \else% for handout, so what follows is default
585 \ifinstructornotes
586 \newenvironment{instructorNotes}%
587          {%
588      \setbox0\vbox\bgroup
589          }
590          {%
591      \egroup
592          }
593          \else
594          \newenvironment{instructorNotes}%
595              {%
596          \begin{trivlist}
597          \item[\hskip \labelsep\bfseries Instructor Notes:\hspace{2ex}]
598              }
599              % %% line at the bottom}
600              {
601          \end{trivlist}
602          \par\addvspace{.5ex}\nobreak\noindent\hung
603              }
604                  \fi
605                      \fi
606
607 ⟨/classXimera⟩
```

2.4.11   Only in HTML or PDF

Ximera provides several techniques to display some content only in the PDF, or only online. The `prompt` environment can be used to hide the data-entry part of a problem from the PDF: it's contents only get displayed online.

The lower level commands `\pdfOnly` and `\htmlOnly` also limit the output to either PDF or online, similarly to the environments `onlyPdf` and `onlyHtml`.

If `\xmPrintHtmlOnlyAlsoInPdf` is set, the online/html only things are printed in the PDF anyway (e.g. for review).

Unfortunately it is not possible in LATEX to have a command and an environment with the same name. We opted for the above (confusing. . .) names.

For backward compatibility, the deprecated environment `onlineOnly` is identical to `onlyHtml`.

For more advanced usage also commands `\ifonline` and `ifonlineTF` are provided.

The technique used to distinguish between the PDF-version and the online HTML-version is always the existence of the TeX4ht macro `\HCode`. Older distinctions such as `\ifxake`, `ifhandout` or `\iftikzexport` should no longer be used for this purpose.

`prompt` (*env.*)   The prompt part for mathmode

```
608 ⟨∗classXimera⟩
609 \ifxake
610          \newenvironment{prompt}{}{}
611 \else
612 \ifhandout
613 \NewEnviron{prompt}{}
614      % Breaks when put in mathmode ?
615      % \newenvironment{prompt}{\suppress}{\endsuppress}
616 \else
617 \newenvironment{prompt}{\bgroup\color{gray!50!black}}{\egroup}
618 \fi
619 \fi
```

`onlyHtml` (*env.*) Only display online
`onlyPdf` (*env.*) Only display in the PDF
`onlineOnly` (*env.*) Only display online (deprecated: use onlyHtml instead)

```
620 \ifdefined\HCode
621 \newenvironment{onlyPdf}{\setbox0\vbox\bgroup}{\egroup}
622 \newenvironment{onlyHtml}{\bgroup}{\egroup}
623 \newenvironment{onlineOnly}{\bgroup}{\egroup}
624 \else
625 \newenvironment{onlyPdf}{\bgroup}{\egroup}
626 \ifdefined\xmPrintHtmlOnlyAlsoInPdf
627  \newenvironment{onlyHtml}{\bgroup\color{red!50!black}}{\egroup}
628  \newenvironment{onlineOnly}{\bgroup\color{red!50!black}}{\egroup}
629 \else
630  \newenvironment{onlyHtml}{\setbox0\vbox\bgroup}{\egroup}
631  \newenvironment{onlineOnly}{\setbox0\vbox\bgroup}{\egroup}
632 \fi
633 \fi
634
```

\htmlOnly   Only display online
\pdfOnly   Only display in the PDF

```
635
636 \ifdefined\HCode
637 \newcommand{\pdfOnly}[1]{}
638 \newcommand{\htmlOnly}[1]{#1}
639 \else
640 \ifdefined\xmPrintHtmlOnlyAlsoInPdf
641  \newcommand{\pdfOnly}[1]{#1}
642  \newcommand{\htmlOnly}[1]{\bgroup\color{red!50!black}#1\egroup}
643 \else
644  \newcommand{\pdfOnly}[1]{#1}
645  \newcommand{\htmlOnly}[1]{}
646 \fi
647 \fi
648
```

\ifonline   Only execute online (ie in HTML version)
\ifonlineTF   Different output online vs PDF

```
649 % An alternatife for \pdfOnly/\begin{htmlOnly} :
650 % Usage: Hello \ifonlineTF{online reader}{PDF reader}
651 \providecommand{\ifonlineTF}[2]{\htmlOnly{#1}\pdfOnly{#2}}
652 \newif{\ifonline}
653 \ifdefined\HCode
654 \onlinetrue
655 \else
656 \onlinefalse
657 \fi
658 ⟨/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.

```
659 %\iftikzexport\else\RequirePackage[framemethod=TikZ]{mdframed}\fi
```

foldable (env.)   Does it fold?

```
660 ⟨*classXimera⟩
661
662 \colorlet{textColor}{black} % since textColor is referenced below
663 \colorlet{background}{white} % since background is referenced below
664
665 % The core environments. Find results in 4ht file.
666 %% pretty-foldable
667 %\iftikzexport
668 \newenvironment{foldable}{%
669 }{%
```

```
670 }
671 %\else
672 %\renewmdenv[
673 %  font=\upshape,
674 %  outerlinewidth=3,
675 %  topline=false,
676 %  bottomline=false,
677 %  leftline=true,
678 %  rightline=false,
679 %  leftmargin=0,
680 %  innertopmargin=0pt,
681 %  innerbottommargin=0pt,
682 %  skipbelow=\baselineskip,
683 %  linecolor=textColor!20!white,
684 %  fontcolor=textColor,
685 %  backgroundcolor=background
686 %]{foldable}%
687 %\fi
688
689 %% pretty-expandable
690 %\iftikzexport
691 %% Overwritten in .4ht, but probably also in accordion!
692 \ifdefined\xmNotExpandableAsAccordion
693 \newenvironment{expandable}{}{}
694 \else
695 \newenvironment{expandable}[2]{}{}
696 \fi
697 %\else
698 %\newmdenv[
699 %  font=\upshape,
700 %  outerlinewidth=3,
701 %  topline=false,
702 %  bottomline=false,
703 %  leftline=true,
704 %  rightline=false,
705 %  leftmargin=0,
706 %  innertopmargin=0pt,
707 %  innerbottommargin=0pt,
708 %  skipbelow=\baselineskip,
709 %  linecolor=black,
710 %]{expandable}%
711 %\fi
712
713 \newcommand{\unfoldable}[1]{#1}
714
715 ⟨/classXimera⟩
```

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

```
716 ⟨*htXimera⟩
717 \renewenvironment{foldable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<
718
719 \ifdefined\xmNotExpandableAsAccordion
720 \renewenvironment{expandable}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode
721 \fi
722
723 \renewcommand{\unfoldable}[1]{\HCode{<span class="unfoldable">}#1\HCode{</span>}}
724 ⟨/htXimera⟩
```

2.4.13 Leashes

leash (*env.*) Put content inside a scrollable box.

```
725 ⟨*classXimera⟩
726
727 \newenvironment{leash}[1]{%
```

```
728 }{%
729 }
730
731
732 ⟨/classXimera⟩

733 ⟨∗htXimera⟩
734 \renewenvironment{leash}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div style="overflow: auto; he
735 ⟨/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.

```
736 ⟨∗classXimera⟩
737 \newcommand{\license}{\excludecomment}
738 ⟨/classXimera⟩
```

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

```
739 ⟨∗classXimera⟩
740 \newcommand{\acknowledgement}{\excludecomment}
741 ⟨/classXimera⟩
```

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

```
742 ⟨∗classXimera⟩
743 \renewcommand{\tag}{\excludecomment}
744 ⟨/classXimera⟩
```

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

```
745 ⟨∗htXourse⟩
746 % Mark this as a xourse file
747 \Configure{@HEAD}{\HCode{<meta name="description" content="xourse" />\Hnewline}}
748 ⟨/htXourse⟩
```

2.5.2 Abstract

abstract (*env.*) Every activity should include a short abstract.

```
749 ⟨∗classXimera⟩
750 \let\abstract\relax
751 \let\endabstract\relax
752 % Use of environ package, may want to find a better way.
753 % see the messing around with \theabstract in title.dtx ... Is this really needed/wanted?
754 \NewEnviron{abstract}{\protected@xdef\theabstract{\BODY}}
755 ⟨/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.

```
756 ⟨∗cfgXimera⟩
757 \ifvmode\IgnorePar\fi\EndP
758 \ConfigureEnv{abstract}{\ifvmode\IgnorePar\fi\EndP\HCode{\Hnewline<div class="abstract">}\par
759 ⟨/cfgXimera⟩
```

```
760 ⟨∗htXimera⟩
761 \RenewEnviron{abstract}{\BODY}
762 ⟨∗htXimera⟩
```

17

2.5.3 Titles and authors

2.5.4 Authors

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

```
763 ⟨∗classXimera⟩
764 \let\@emptyauthor\@author
765 \def\author#1{\gdef\@author{#1}}
766 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}
767 ⟨/classXimera⟩
```

Include author name in meta tags

```
768 ⟨∗htXimera⟩
769 \Configure{@HEAD}{\HCode{<meta name="author" content="}\@author\HCode{" />\Hnewline}}
770 ⟨/htXimera⟩
```

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

```
771 ⟨htXimera | classXimera⟩\def\and{and }
```

2.5.5 Title

\title Activities have titles.

```
772 ⟨∗classXimera⟩
773 \let\title\relax
774 \newcommand{\title}[1][]{{\protected@xdef\@pretitle{#1}}\protected@xdef\@title}
775
776 \title{}
777
778 \newcounter{titlenumber}
779 \renewcommand{\thetitlenumber}{\arabic{titlenumber}}
780 %\renewcommand{\thesection}{\arabic{titlenumber}} %% Makes section numbers work
781 \setcounter{titlenumber}{0}
782
783 \newpagestyle{main}{
784 \sethead[\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}][][] % even
785 {}{}{\textsl{\ifnumbers\thetitlenumber\hspace{1em}\fi\@title}} % odd
786 \setfoot[\thepage][][] % even
787 {}{}{\thepage} % odd
788 }
789 \pagestyle{main}
```

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

```
790 \renewcommand\maketitle{%
791   \addtocounter{titlenumber}{1}%
792   {\flushleft\large\bfseries \@pretitle\par\vspace{-1em}}
793   {\flushleft\LARGE\bfseries {\ifnumbers\thetitlenumber\fi}{\ifnumbers\hspace{1em}\else\hspa
794   \phantomsection%
795   \ifnumbers\addcontentsline{toc}{section}{\thetitlenumber~\@title}\else\addcontentsline{toc}
796   \vskip .6em\noindent\textit\theabstract\setcounter{problem}{0}\setcounter{section}{0}\setco
797   \ifnooutcomes\else\let\thefootnote\relax\footnote{Learning outcomes: \theoutcomes}\fi
798   \ifnoauthor\else\let\thefootnote\relax\footnote{Author(s):~\@author}\fi
799   \aftergroup\@afterindentfalse
800   \aftergroup\@afterheading}
801
802 \ifnumbers
803 \setcounter{secnumdepth}{2}
804 \renewcommand{\thesection}{\arabic{titlenumber}.\arabic{section}}
805 \renewcommand{\thesubsection}{\arabic{titlenumber}.\arabic{section}.\arabic{subsection}}
806 \else
807 \setcounter{secnumdepth}{-2}
808 \fi
809
810 \def\activitystyle{}
```

```
811 \newcounter{sectiontitlenumber}
812 \setcounter{secnumdepth}{2}
813 \setcounter{tocdepth}{2}
814 \newcommand\chapterstyle{%
815   \def\activitystyle{activity-chapter}
816   \def\maketitle{%
817     \addtocounter{titlenumber}{1}%
818                   {\flushleft\small\sffamily\bfseries\@pretitle\par\vspace{-1.5em}}%
819                   {\flushleft\LARGE\sffamily\bfseries\thetitlenumber\hspace{1em}\@title \pa
820                   {\vskip .6em\noindent\textit\theabstract\setcounter{problem}{0}\setcounte
821                   \par\vspace{2em}
822                   \phantomsection\addcontentsline{toc}{section}{\textbf{\thetitlenumber\hsp
823 }}
824
825
826 \newcommand\sectionstyle{%
827   \def\activitystyle{activity-section}
828   \def\maketitle{%
829     \addtocounter{section}{1}
830     \setcounter{sectiontitlenumber}{\value{section}}
831     {\flushleft\small\sffamily\bfseries\@pretitle\par\vspace{-1.5em}}%
832     {\flushleft\Large\sffamily\bfseries\thetitlenumber.\thesectiontitlenumber\hspace{1em}\@ti
833     {\vskip .6em\noindent\textit\theabstract\setcounter{subsection}{0}}%
834     \par\vspace{2em}
835     \phantomsection\addcontentsline{toc}{section}{\thetitlenumber.\thesectiontitlenumber\hspa
836   \renewcommand\section{\@startsection{subsection}{2}{\z@}%
837                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
838                                     {1.5ex \@plus .2ex}%
839                                     {\normalfont\large\bfseries}}
840
841   \renewcommand\subsection{\@startsection{subsubsection}{3}{\z@}%
842                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
843                                     {1.5ex \@plus .2ex}%
844                                     {\normalfont\normalsize\bfseries}}
845
846 }}
847
848
849 \iftikzexport%% allows xake to handle \chapterstyle and \sectionstye
850 \renewcommand\chapterstyle{\def\activitystyle{chapter}}
851 \renewcommand\sectionstyle{\def\activitystyle{section}}
852 \else
853 \fi
854
855 ⟨/classXimera⟩
```

Eliminate some formatting that we'll handle later with CSS

```
856 ⟨∗htXimera⟩
857 \renewcommand{\maketitle}{}
858 ⟨/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.

```
859 ⟨∗classXimera⟩
860 \def\theoutcomes{}
861
862 \ifdefined\HCode%
863   \newcommand{\outcome}[1]{}
864 \else%
865   \newwrite\outcomefile
866   \immediate\openout\outcomefile=\jobname.oc
867
```

19

```
868    \newcommand{\outcome}[1]{\edef\theoutcomes{\theoutcomes #1~}%
869    \immediate\write\outcomefile{\unexpanded{\outcome}{#1}}}
870    \fi%
871 ⟨/classXimera⟩
```

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

```
872 ⟨∗cfgXimera⟩
873 \renewcommand{\outcome}[1]{
874    \Configure{@HEAD}{\HCode{<meta name="learning-outcome" content="#1"/>\Hnewline}}
875 }
876 % Sometimes there are no outcomes at all
877 \IfFileExists{\jobname.oc}{\input{\jobname.oc}}{}
878
879 \renewcommand{\outcome}[1]{%
880    \HCode{<span class="learning-outcome">#1</span>}
881 }
882 ⟨/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.

```
883 ⟨∗htXimera⟩
884 \let\oldlabel\label
885 \renewcommand{\label}[1]{\oldlabel{#1}\HCode{<a class="ximera-label" id="#1"></a>}}
886 ⟨/htXimera⟩
```

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

```
887 ⟨∗htXimera⟩
888 \renewcommand{\ref}[1]{\HCode{<a class="reference" href="\##1">#1</a>}}
889 ⟨/htXimera⟩
```

2.6   Images

2.6.1   Images

image (*env.*) Place images inside an `image` environment. On paper, this centers the image. On the
\xmDefaultGraphicsPath web, this provides additional benefits. Base graphicspath, deafult '/xmPictures'. Can only be changed BEFORE loading ximera.cls!

```
890 ⟨∗classXimera⟩
891 % Provide a default graphicspath
892 % (somewhat tricky: an activity can be included in a xourse in a wildly different path !)
893 % Suggested convention: put all images in i /pictures folder in the root of your project
894 \providecommand{\xmDefaultGraphicsPath}{/xmPictures}
895 \graphicspath{  %% When looking for images,
896 {./}             %% look here first,
897 {.\xmDefaultGraphicsPath/}   %% then look for a pictures folder,
898 {..\xmDefaultGraphicsPath/}   %% then look for a pictures folder,
899 {../..\xmDefaultGraphicsPath/}   %% then look for a pictures folder,
900 {../../..\xmDefaultGraphicsPath/}   %% then look for a pictures folder,
901 }
902 %\newenvironment{image}[1][]{\begin{center}}{\end{center}}
903 \NewEnviron{image}[1][3in]{%
904    \begin{center}\resizebox{#1}{!}{\BODY}\end{center}% resize and center
905 }
906 ⟨/classXimera⟩
```

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

```
907 ⟨∗classXimera⟩
908 \newcommand{\alt}[1]{}
909 ⟨/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.

```
910 ⟨*htXimera⟩
911 \newcounter{imagealt}
912 \setcounter{imagealt}{0}
913 \renewenvironment{image}[1][]{\stepcounter{imagealt}%
914   \ifvmode \IgnorePar\fi \EndP%
915   \HCode{<div class="image-environment" role="img" aria-labelledby="image-alt-\arabic{imageal
916 }{\HCode{</div>}}
917 \renewcommand{\alt}[1]{\HCode{<div style="display: none;" id="image-alt-\arabic{imagealt}">}
918 ⟨/htXimera⟩
919 ⟨*cfgXimera⟩
920 %%  Although we accept many formats, SVG is preferred on the web.
921 %%  Since we have a different mechanism for producing |alt| text, we
922 %%  want to ignore tex4ht's own method fo producing alt text.
923 %%  2024: is now in TeX4ht ...
924 % \DeclareGraphicsExtensions{.jpg,.png,.gif,.svg}
925 % \Configure{graphics*}
926 % {svg}{
927 %   {\Configure{Needs}{File: \Gin@base.svg}\Needs{}}
928 %   \Picture[]{\csname Gin@base\endcsname.svg \csname a:Gin-dim\endcsname}%
929 % }
930 ⟨/cfgXimera⟩
```

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

```
931 ⟨*cfgXimera⟩
932 \ifcsname ifstandalone\endcsname
933   \ifstandalone
934     \renewcommand\includegraphics[2][]{}
935   \fi
936 ⟨/cfgXimera⟩
```

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

```
937 ⟨*htXimera⟩
938 \providecommand{\pgfsyspdfmark}[3]{}
939 ⟨/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`.

```
940 ⟨*classXimera⟩
941 % everything skipped, assumle TeX4ht does the jjb now
942 \ifdefined\reallyneverever
943
944 \ifdefined\HCode
945   \tikzexporttrue
946 \fi
947
948 \iftikzexport
949   \usetikzlibrary{external}
950
951   \ifdefined\HCode
952     % in htlatex, just include the svg files
953     \def\pgfsys@imagesuffixlist{.svg}
954
955     \tikzexternalize[prefix=./,mode=graphics if exists]
956   \else
```

```
957    % in pdflatex, actually generate the svg files
958    \tikzset{
959      /tikz/external/system call={
960        pdflatex \tikzexternalcheckshellescape
961        -halt-on-error -interaction=batchmode
962        -jobname "\image" "\\PassOptionsToClass{tikzexport}{ximera}\texsource";
963        mutool draw -F svg \image.pdf > \image.svg ;      % mutool adds "1" to filename ????
964        mutool draw -o \image.svg \image.pdf ;
965        mutool draw -r 150 -c rgbalpha -o \image.png \image.pdf ;
966        ebb -x \image.png
967      }
968    }
969    \tikzexternalize[optimize=false,prefix=./]
970  \fi
971
972  \fi
973 \fi
974 ⟨/classXimera⟩
```

2.6.3 XKCD

\xkcd   Reference an XKCD cartoon.

```
975 ⟨∗classXimera⟩
976 \newcommand{\xkcd}[1]{#1}
977 ⟨/classXimera⟩
```

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

```
978 ⟨∗htXimera⟩
979 \renewcommand{\xkcd}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<img src="https://imgs.xkcd.com/c
980 ⟨/htXimera⟩
```

2.7   Links

We put hyperref after all other packages becuase that is better.

```
981 ⟨∗classXimera⟩
982 % Don't use hyperref when using Tex4ht
983 \ifdefined\HCode
984 \RequirePackage{hyperref}
985 \else
986 \RequirePackage[pdfpagelabels,colorlinks=true,allcolors=blue!30!black]{hyperref}
987 \pdfstringdefDisableCommands{\def\hskip{}}%% quiets warning
988 \fi
989 ⟨/classXimera⟩
```

2.8   Interactives

2.8.1   Including widgets

\includeinteractive   Cognate to `includegraphics` but instead of a graphics file, accepts a `.js` file which will be loaded as an interactive widget.

```
990 ⟨∗classXimera⟩
991 \define@key{interactive}{id}{\def\interactive@id{#1}}
992 \setkeys{interactive}{id=}
993 \newcommand{\includeinteractive}[2][]{
994 \setkeys*{interactive}{#1}%
995 \ifthenelse{\equal{\interactive@id}{}}{}{\recordvariable{\interactive@id}}
996 Interactive
997 }
998 ⟨/classXimera⟩

999 ⟨∗htXimera⟩
1000 \renewcommand{\includeinteractive}[2][]{\stepcounter{identification}\ifvmode \IgnorePar\fi \I
1001 ⟨/htXimera⟩
```

2.8.2 Google Sheet

`\googleSheet` googleSheet command. Requires id, width, and height as arguments. optional arguments are gid for sheet ID and range for cell range. command definition

```
1002 ⟨∗classXimera⟩
1003 % Google Spreadsheet link (read only)
1004 \newcommand{\googleSheet}[5]{%
1005   Google Spreadsheet link: \url{https://docs.google.com/spreadsheets/d/#1}%
1006 }
1007 ⟨/classXimera⟩

1008 ⟨∗htXimera⟩
1009 \renewcommand{\googleSheet}[5]{%
1010   \ifthenelse{\equal{#4}{}}%
1011     {\HCode{<iframe width="#2px" height="#3px" src="https://docs.google.com/spreadsheets/d/#1
1012     {\ifthenelse{\equal{#5}{}}%
1013       {\HCode{<iframe width="#2px" height="#3px" src="https://docs.google.com/spreadsheets/d
1014       {\HCode{<iframe width="#2px" height="#3px" src="https://docs.google.com/spreadsheets/d
1015     }%
1016   }%
1017 ⟨/htXimera⟩
```

2.8.3 Geogebra

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

```
1018 ⟨∗classXimera⟩
1019 %Geogebra link
1020 \newcommand{\geogebra}[3]{GeoGebra link: \url{https://www.geogebra.org/m/#1}}
1021 ⟨/classXimera⟩
```

Define keys for answer geogebra key=value pairs.

```
1022 ⟨∗htXimera⟩
1023 \define@key{geogebra}{rc}[true]{\def\geo@rc{#1}}
1024 \define@key{geogebra}{sdz}[true]{\def\geo@sdz{#1}}
1025 \define@key{geogebra}{smb}[true]{\def\geo@smb{#1}}
1026 \define@key{geogebra}{stb}[true]{\def\geo@stb{#1}}
1027 \define@key{geogebra}{stbh}[true]{\def\geo@stbh{#1}}
1028 \define@key{geogebra}{ld}[true]{\def\geo@ld{#1}}
1029 \define@key{geogebra}{sri}[true]{\def\geo@sri{#1}}
1030 %set default key values
1031 \setkeys{geogebra}{rc=false,sdz=false,smb=false,stb=false,stbh=false,ld=false,sri=false}
1032 %command definition
1033 \renewcommand{\geogebra}[4][]{%
1034   \setkeys{geogebra}{#1}% Set new keys
1035   \HCode{<iframe scrolling="no" src="https://www.geogebra.org/material/iframe/id/#2/width/#3/
1036 ⟨/htXimera⟩
```

2.8.4 Desmos

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

```
1037 ⟨∗classXimera⟩
1038 \newcommand{\desmos}[3]{Desmos link: \url{https://www.desmos.com/calculator/#1}}
1039 \newcommand{\desmosThreeD}[3]{Desmos3D link: \url{https://www.desmos.com/3d/#1}}
1040 ⟨/classXimera⟩

1041 ⟨∗htXimera⟩
1042 \catcode`\%=11
1043 \renewcommand{\desmos}[3]{\HCode{<iframe src="https://www.desmos.com/calculator/#1" width="10
1044 \catcode`\%=14
1045 \renewcommand{\desmosThreeD}[3]{\HCode{<iframe src="https://www.desmos.com/3d/#1" width="#2px
1046 ⟨/htXimera⟩
```

2.8.5 Graphs

\graph An embedded graph (in math mode).

```
1047 ⟨∗classXimera⟩
1048 \newcommand{\graph}[2][]{\text{Graph of $#2$}}
1049 ⟨/classXimera⟩

1050 ⟨∗htXimera⟩
1051 \renewcommand{\graph}[2][]{\HCode{<div class="graph" data-options="#1">}#2\HCode{</div>}}
1052 ⟨/htXimera⟩
```

2.8.6 Video

\youtube Youtube command. Requires id.

```
1053 ⟨∗classXimera⟩
1054 \newcommand{\youtube}[1]{YouTube link: \url{https://www.youtube.com/watch?v=#1}}
1055 ⟨/classXimera⟩

1056 ⟨∗htXimera⟩
1057 %% \renewcommand{\youtube}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="video youtube-p]
1058 % Fixes no-youtube-when-no-cookies-accepted. Class xmyoutube allows for css customization.
1059 \renewcommand{\youtube}[1]{\ifvmode \IgnorePar\fi \EndP\HCode{<iframe class="xmyoutube" src="
1060
1061 ⟨/htXimera⟩
```

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

```
1062 ⟨∗htXourse⟩
1063 \renewcommand\youtube[1]{%
1064 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="youtube" href="https://www.youtube.com/watch?v=#
1065 }
1066 ⟨/htXourse⟩
```

2.8.7 JavaScript

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

```
1067 ⟨∗classXimera⟩
1068 \DefineVerbatimEnvironment{javascript}{Verbatim}{numbers=left,frame=lines,label=JavaScript,la
1069 ⟨/classXimera⟩

1070 ⟨∗htXimera⟩
1071 % for programming javascript
1072 \renewenvironment{javascript}{\NoFonts}{\EndNoFonts}
1073 \ScriptEnv{javascript}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div cl
1074 ⟨/htXimera⟩
```

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

```
1075 ⟨∗classXimera⟩
1076 \def\js#1{\mbox{\texttt{\detokenize{#1}}}}
1077 ⟨/classXimera⟩

1078 ⟨∗htXimera⟩
1079 \def\js#1{\stepcounter{identification}\HCode{<span class="inline-javascript" id="javascript\a
1080 ⟨/htXimera⟩
```

2.9 SageMath support

Load SageTeX if it exists.

```
1081 ⟨∗classXimera⟩
1082 \IfFileExists{sagetex.sty}{\RequirePackage{sagetex}}{}
1083 ⟨/classXimera⟩
```

sageCell (env.) Create an interactive SageMath widget.

```
1084 ⟨∗classXimera⟩
1085 \DefineVerbatimEnvironment{sageCell}{Verbatim}{numbers=left,frame=lines,label=SAGE,labelposit
1086 ⟨/classXimera⟩
```

```
1087 ⟨∗htXimera⟩
1088 \renewenvironment{sageCell}{\NoFonts}{\EndNoFonts}
1089 \ScriptEnv{sageCell}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sage"><script type="text,
1090 ⟨/htXimera⟩
```

sageOutput (*env.*)   Execute SageMath code and output the result.

```
1091 ⟨∗classXimera⟩
1092 \DefineVerbatimEnvironment{sageOutput}{Verbatim}{numbers=left,frame=lines,label=SAGE-Output,]
1093 ⟨/classXimera⟩
```

```
1094 ⟨∗htXimera⟩
1095 \renewenvironment{sageOutput}{\NoFonts}{\EndNoFonts}
1096 \ScriptEnv{sageOutput}{\ifvmode \IgnorePar\fi \EndP\HCode{<div class="sageOutput"><script ty]
1097 ⟨/htXimera⟩
```

sageSilent (*env.*)   Execute SageMath code without outputting the result.

```
1098 ⟨∗htXimera⟩
1099 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1100 \ifdefined\sagesilent
1101   \renewenvironment{sagesilent}{\NoFonts}{\EndNoFonts}
1102 \fi
1103 \ScriptEnv{sagesilent}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="text/sagemath">}\Html
1104 ⟨/htXimera⟩
```

2.10   Answerables

2.10.1   Answers

\answer A math answer

```
1105 ⟨∗classXimera⟩
1106
1107 \ifdefined\HCode
1108 \newcommand{\recordvariable}[1]{}
1109 \else
1110 \newwrite\idfile
1111 \immediate\openout\idfile=\jobname.ids
1112 \newcommand{\recordvariable}[1]{\ifthenelse{\equal{#1}{}}{}{\immediate\write\idfile{var #1;}
1113 \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

```
1114 \define@key{answer}{given}[true]{\def\ans@given{#1}}
```

Used for setting numeric answer tolerance for online student input.

```
1115 \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.

```
1116 \define@key{answer}{validator}{}
```

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

```
1117 \define@key{answer}{id}{\def\ans@id{#1}}
```

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

```
1118 \define@key{answer}{format}{}
```

Used to hide the answer input box on the web.

```
1119 \define@key{answer}{onlinenoinput}[false]{}
```

Used to add a 'show answer' button to the answer blank.

```
1120 \define@key{answer}{onlineshowanswerbutton}[false]{}
```

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

```
1121 \setkeys{answer}{id=,given=false,onlinenoinput=false,onlineshowanswerbutton=false}
```

Basic code for \answer.

```
1122
1123 % Options for handout
1124 \newcommand{\answerFormatLength}{2cm}
1125
1126 \newcommand{\answerFormatDots}[1]{\ldots\ldots}
1127 \newcommand{\answerFormatLine}[1]{\protect\rule{\answerFormatLength}{0.4pt}}
1128 \newcommand{\answerFormatFlexibleLine}[1]{\protect\rule{\widthof{$#1$}*2}{0.4pt}}
1129 \newcommand{\answerFormatFlexibleBox}[1]{\fbox{\scalebox{2}{\phantom{$#1$}}}}
1130
1131 % options for default (i.e with answers filled in)
1132 \newcommand{\answerFormatPlain}[1]{\ensuremath{#1}}
1133 \newcommand{\answerFormatBlue}[1]{\color{blue}\ensuremath{#1}}
1134 \newcommand{\answerFormatBoxed}[1]{\fbox{\ensuremath{#1}}}
1135 \newcommand{\answerFormatBoxedGiven}[1]{\underset{\scriptstyle\mathrm{given}}{\fbox{\ensurema
1136
1137 % defaults for handout and default mode, and for \answer[given]
1138 \let\handoutAnswerFormat\answerFormatDots
1139 \let\defaultAnswerFormat\answerFormatBlue
1140 \let\givenAnswerFormat\answerFormatBoxedGiven
1141
1142 \newcommand{\answer}[2][]{%
1143  \ifmmode%
1144   \setkeys{answer}{#1}%
1145   \recordvariable{\ans@id}
1146   \ifthenelse{\boolean{\ans@given}}
1147    {% Start then statement
1148    \ifhandout
1149     #2
1150    \else
1151     \givenAnswerFormat{#2} %% in case the argument helps formatting
1152    \fi
1153    }% End then statement
1154    {% Start else statement
1155    \ifhandout
1156     \handoutAnswerFormat{#2} %% in case the argument helps formatting
1157    \else% show answer in box outside handout mode
1158     \defaultAnswerFormat{#2} %% in case the argument helps formatting
1159    \fi
1160    }% End else statement
1161   \else%
1162   \GenericError{\space\space\space\space}% Throw an error based on... something? -- Jason
1163    {Attempt to use \@backslashchar answer outside of math mode}
1164    {See https://github.com/ximeraProject/ximeraLatex for explanation.}
1165    {Need to use either inline or display math.}%
1166   \fi
1167 }
1168 ⟨/classXimera⟩
```

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

```
1169 ⟨*htXimera⟩
1170 \renewcommand{\answer}[2][false]{\HCode{<span class="answer respondable">}#2\HCode{</span>}}
1171
1172 \def\validator[#1]{\stepcounter{identification}\HCode{<div class="validator" id="validator\a
1173 \def\endvalidator{\HCode{</div>}}
1174
1175 ⟨/htXimera⟩
```

2.10.2 Multiple choice and the like

multipleChoice (*env.*) Multiple choice

```
1176 ⟨*classXimera⟩
```

```
1177 % Jim: Originally this was \renewcommand{\theenumi}{$(\mathrm{\alph{enumi}})$}
1178 % but that breaks tex4ht because mathmode can only be processed by mathjax.
1179 % so now I made this just italicized.
```

2.10.3   Options

```
1180 \define@key{choice}{value}[]{\def\choice@value{#1}}
```

This flags the answer as the correct answer
```
1181 \define@boolkey{choice}{correct}[true]{\def\choice@correct{#1}}
```

Use an ID to refer to the choice.
```
1182 \define@key{multipleChoice}{id}{\def\mc@id{#1}}
```

\otherchoice outputs the item if correct and nothing if incorrect.
```
1183 \define@key{otherchoice}{value}[]{\def\otherchoice@value{#1}}
1184 \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".
```
1185 \setkeys{choice}{correct=false,value=}
```

Defaults for multipleChoice pairs. Default to no id? – Jason
```
1186 \setkeys{multipleChoice}{id=}
```

Defaults for otherchoice pairs. Default "otherchoice" to behave like "choice" for error checking.
```
1187 \setkeys{otherchoice}{correct=false,value=}
1188 ⟨/classXimera⟩
```

2.10.4   Choices

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

```
1189 ⟨*classXimera⟩
1190 \newcommand{\choice}[2][]{%
1191 \setkeys{choice}{#1}%
1192 \item{#2}
1193 \ifthenelse{\boolean{\choice@correct}}
1194     {% Begin then result
1195     \ifhandout% if it's a handout do nothing.
1196     \else% otherwise place a checkmark when you select the "correct choice"... maybe? -- Jaso
1197         \,\checkmark\,\setkeys{choice}{correct=false}
1198     \fi
1199     }% End then result
1200     {}% Begin/End else result.
1201 }
1202
1203 %Define an expandable version of choice Not really meant to be used outside this package (use
1204 % Is there a reason we can't just always use this as default? -- Jason
1205 \newcommand{\choiceEXP}[2][]{%
1206 \expandafter\setkeys\expandafter{choice}{#1}%
1207 \item{#2}
1208 \ifthenelse{\boolean{\choice@correct}}
1209 {% Begin then result
1210 \ifhandout
1211 \else
1212  \,\checkmark\,\setkeys{choice}{correct=false}
1213 \fi
1214 }% End then result
1215 {}% Begin/End else result.
1216 } %% note all the {} are needed in case the choice has [] in it.
1217
1218 % \otherchoice is the \choice used in wordChoice command.
1219 \newcommand{\otherchoice}[2][]{%
1220 \ignorespaces%
1221 \setkeys{otherchoice}{#1}%
1222 \ifthenelse{\boolean{\otherchoice@correct}}%
```

```
1223 {% Start then result
1224 #2\ignorespaces\setkeys{otherchoice}{correct=false}\ignorespaces%
1225 }% End then result
1226 {}% Start/End else result
1227 \ignorespaces%
1228 }%
1229 \newcommand{\inlinechoice}[2][]{%
1230 \setkeys{choice}{#1}%
1231 \iffirstinlinechoice
1232 (\hspace{-.25em}
1233 \firstinlinechoicefalse
1234 \else
1235 /
1236 \fi
1237 #2
1238 \ifthenelse{\boolean{\choice@correct}}%
1239 {% Start then result
1240 \ifhandout\else\checkmark\ignorespaces\setkeys{choice}{correct=false}\ignorespaces\fi%
1241 }% End then result
1242 {}% Start/End else result
1243 \hspace{-.25em}\ignorespaces%
1244 }
1245
1246 ⟨/classXimera⟩
```

On the HTML side, \choice emits <span>s.

```
1247 ⟨*htXimera⟩
1248 \newcounter{choiceId}
1249 \renewcommand{\choice}[2][]{%
1250 \setkeys{choice}{correct=false}%
1251 \setkeys{choice}{#1}%
1252 \stepcounter{choiceId}\IgnorePar%
1253 \HCode{<span class="choice }%
1254 \ifthenelse{\boolean{\choice@correct}}{\HCode{correct}}{}
1255 \HCode{" }
1256 \ifthenelse{\equal{\choice@value}{}}{}{\HCode{data-value="\choice@value" }}
1257 \HCode{id="choice\arabic{choiceId}">}%
1258 #2\HCode{</span>}}
1259 \let\inlinechoice\choice
1260 ⟨/htXimera⟩
```

2.10.5 Environment(s)

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

```
1261 ⟨*classXimera⟩
1262 \newenvironment{multipleChoice}[1][]
1263 {% Environment Start Code
1264 \setkeys{multipleChoice}{#1}%
1265 \recordvariable{\mc@id}%
1266 \begin{trivlist}
1267  \item[\hskip \labelsep\small\bfseries Multiple Choice:]\hfil
1268  \begin{enumerate}
1269 }% Note this means that \item has to be the first line after \begin{multipleChoice}.
1270 {% Environment End Code
1271  \end{enumerate}
1272 \end{trivlist}
1273 }
1274
1275 %multipleChoice@ is for internal use only! (used in wordChoice)
1276 %this is simply a wrapper for the sole showing (other)choice.
1277 \newenvironment{multipleChoice@}[1][]{}{()}
1278 ⟨/classXimera⟩
```

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

```
1279 ⟨∗htXimera⟩
1280 \renewenvironment{multipleChoice}[1][]
1281 {\setkeys{multipleChoice}{#1}%
1282 \stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div class="multiple-choice"
1283 \ifthenelse{\equal{\mc@id}{}}{}{\HCode{data-id="\mc@id" }}%
1284 \HCode{id="problem\arabic{identification}">}%
1285 }{\HCode{</div>}\IgnoreIndent}
1286 \ConfigureEnv{multipleChoice}{}{}{}{}
1287 ⟨/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.

```
1288 ⟨∗classXimera⟩
1289 \newcommand{\wordChoice}[1]{%
1290 \let\choicetemp\choice% Assign a "choicetemp" command to duplicate choice.
1291 \ifwordchoicegiven% If wordchoice option is on, we need to juggle around some definitions.
1292 \let\choice\otherchoice%
1293 %\begin{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1294 #1
1295 %\end{multipleChoice@}% -unnecessary (REMOVE THIS LINE IF THE YEAR IS 2019 or Beyond)
1296 \else% If it isn't the regular "choice" command should work.
1297 \let\choice\inlinechoice%
1298 \begin{multipleChoice@}%
1299 #1%
1300 \end{multipleChoice@}%
1301 \fi%
1302 \let\choice\choicetemp% Now that choicetmp has been manipulated to what we want, replace cho
1303 }%
1304
1305
1306 ⟨/classXimera⟩
```

This is actually just word choice

```
1307 ⟨∗htXimera⟩
1308 \renewenvironment{multipleChoice@}{\refstepcounter{problem}}{}%
1309 \ConfigureEnv{multipleChoice@}{\stepcounter{identification}\IgnorePar\HCode{<span class="word
1310 ⟨/htXimera⟩
```

2.12   Select all

selectAll (env.) A multiple-multiple choice question

```
1311 ⟨∗classXimera⟩
1312 \newenvironment{selectAll}[1][]
1313 {\begin{trivlist}\item[\hskip \labelsep\small\bfseries Select All Correct Answers:]\hfil\beg
1314   {\end{enumerate}\end{trivlist}}
1315 ⟨/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.

```
1316 ⟨∗htXimera⟩
1317 \renewenvironment{selectAll}{\refstepcounter{problem}}{}%
1318 \ConfigureEnv{selectAll}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<div
1319 ⟨/htXimera⟩
```

29

2.12.1 Free response

freeResponse (*env.*) A freeform input box.

```
1320 ⟨∗classXimera⟩
1321 \newboolean{given} %% required for freeResponse
1322 \setboolean{given}{true} %% could be replaced by a key=value pair later if needed
1323
1324 \ifhandout
1325 \newenvironment{freeResponse}[1][false]%
1326  {%
1327  \def\givenatend{\boolean{#1}}
1328  \ifthenelse{\boolean{#1}}
1329   {% Begin then result
1330   \begin{trivlist}
1331    \item
1332   }% End then result
1333   {% Begin else result
1334   \setbox0\vbox\bgroup
1335   }% End else result
1336 %   {}% Don't think this is doing anything? -- Jason
1337   }
1338   {%
1339   \ifthenelse{\givenatend}
1340    {% Begin then result
1341    \end{trivlist}
1342   }% End then result
1343   {% Begin else result
1344   \egroup
1345   }% End else result
1346 %    {}% Don't think this is doing anything? -- Jason
1347   }
1348 \else
1349  \newenvironment{freeResponse}[1][false]%
1350   {% Environment Beginning Code
1351    \ifthenelse{\boolean{#1}}%% Could probably change this with just putting the (given) in t
1352     {% Begin then result
1353     \begin{trivlist}
1354      \item[\hskip \labelsep\bfseries Free Response (Given):\hspace{2ex}]
1355     }% End then result
1356    {% Begin else result
1357    \begin{trivlist}
1358     \item[\hskip \labelsep\bfseries Free Response:\hspace{2ex}]
1359    }% End else result
1360   }
1361   {% Environment Ending Code
1362    \end{trivlist}
1363   }
1364 \fi
1365
1366 ⟨/classXimera⟩
1367 ⟨∗htXimera⟩
1368
1369 \renewenvironment{freeResponse}{\refstepcounter{problem}}{}%
1370 \ConfigureEnv{freeResponse}{\stepcounter{identification}\ifvmode \IgnorePar\fi \EndP\HCode{<
1371
1372 ⟨/htXimera⟩
```

2.12.2 Feedback

feedback (*env.*) An initially hidden environment that uncovers itself at an appropriate time. New Validator rewrite code added by Jason Nowell. Original code orovided 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.

```
1373 ⟨*classXimera⟩
1374 \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.

```
1375 \newenvironment{validator}[1][]{
1376 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter"
1377 \mbox{\texttt{\detokenize\expandafter{\PH@Command}}}% Now expand PH@Command once and then de
1378 }{}
```

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

```
1379 \ifhandout%
1380 \newenvironment{feedback}
1381            {%
1382    \setbox0\vbox\bgroup
1383        }
1384            {%
1385    \egroup
1386        }
```

If this isn't a handout, then we want to display the Feedback by using a label, positioned and formated 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.

```
1387 \else
1388 \newenvironment{feedback}[1][attempt]{
1389
1390 \def\PH@Command{#1}% Use PH@Command to hold the content and be a target for "\expandafter"
1391
1392 \begin{trivlist}% Begin the trivlist to use formating of the "Feedback" label.
1393 \item[\hskip \labelsep\small\slshape\bfseries Feedback% Format the "Feedback" label. Don't
1394 (\texttt{\detokenize\expandafter{\PH@Command}}):% Format (and detokenize) the condition for
1395 \hspace{2ex}]\small\slshape% Insert some space before the actual feedback given.
1396 }{
1397 \end{trivlist}
1398 }
1399
1400 \fi
1401 ⟨/classXimera⟩
```

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

```
1402 ⟨*htXimera⟩
1403 \def\feedback{\@ifnextchar[{\@feedbackcode}{\@feedbackattempt}}
1404 \def\@feedbackattempt{\@feedbackcode[attempt]}
1405 \def\@feedbackcode[#1]{\stepcounter{identification}%
1406 \ifvmode \IgnorePar\fi \EndP%
1407 \ifthenelse{\equal{#1}{attempt}}{\HCode{<div class="feedback" data-feedback="attempt" id="fee
1408 {\ifthenelse{\equal{#1}{correct}}{\HCode{<div class="feedback" data-feedback="correct" id="fe
1409 {\HCode{<div class="feedback" data-feedback="script" id="feedback\arabic{identification}"><sc
1410 \def\endfeedback{\HCode{</div>}\IgnoreIndent}
1411 ⟨/htXimera⟩
```

2.12.3 Ungraded activities

ungraded (*env.*) 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.

1412 ⟨∗classXimera⟩
1413 `\newenvironment{ungraded}{}{}`
1414 ⟨/classXimera⟩

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

1415 ⟨∗htXimera⟩
1416 `\renewenvironment{ungraded}{%`
1417 `\ifvmode \IgnorePar\fi \EndP\HCode{<div class="ungraded">}\IgnoreIndent%`
1418 `}{`
1419 `\ifvmode \IgnorePar\fi \EndP\HCode{</div>}\IgnoreIndent%`
1420 `}`
1421 ⟨/htXimera⟩

2.13 Support for the web

2.13.1 MathJax support

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

First, create the `.jax` file. Redefine newcommand appropriately.

1422 ⟨∗classXimera⟩
1423 `%% Pre-202412: .jax file written in non-\HCode, and in a next run inserted by ximera.cfg in `
1424 `%% Post-202501: .mjax file written only in \HCode, and in luaxake post-processing inserted i`
1425 `%%    ( used luaxake rather than sed ...)`
1426 `\newwrite\myfile`
1427 `\ifdefined\HCode`
1428 `\immediate\openout\myfile=\jobname.xmjax`
1429
1430 `%% From |only.dtx| we must also create |prompt| on the MathJax side.`
1431 `\immediate\write\myfile{\unexpanded{\newenvironment}{prompt}{}{}}`
1432
1433 `%% Write all newcommands to .xmjax file, that will be included in the .html via luaxake`
1434 `\let\@oldargdef\@argdef`
1435 `\long\def\@argdef#1[#2]#3{%`
1436 `\immediate\write\myfile{\unexpanded{\newcommand}{\unexpanded{#1}}[\unexpanded{#2}]{\unexpande`
1437 `\@oldargdef#1[#2]{#3}%`
1438 `}`
1439
1440 `%% Same for \DeclareMathOperator`
1441 `\let\@OldDeclareMathOperator\DeclareMathOperator`
1442 `\renewcommand{\DeclareMathOperator}[2]{\@OldDeclareMathOperator{#1}{#2}\immediate\write\myfil`
1443
1444 `\fi`
1445
1446
1447 ⟨/classXimera⟩

Include the jax'ed newcommands (pre-202412 versions ....)

1448 ⟨∗cfgXimera⟩
1449
1450 `% 202501: removed sed-manipulation of .jax file; see luaxake now`
1451
1452 `\Configure{BVerbatimInput}{}{}{}{}`
1453
1454 `\Configure{verbatiminput}{}{}{}{}`
1455
1456 `% Instead of a nonbreaking space, use a standard space`
1457 `\makeatletter`
1458 `\def\FV@Space{\space}`
1459 `\makeatother`
1460
1461 `% Include the (problem-?) .ids in a text/javascript script right at the beginning of the body`
1462 `\Configure{BODY}{%`
1463 `\HCode{<body>\Hnewline}%`

```
1464 \Tg<div class="preamble">%
1465 %% 202501: removed .jax inclusion  (see luaxake)
1466
1467 %% Include the .ids file
1468 \IfFileExists{\jobname.ids}{\HCode{<script type="text/javascript">\Hnewline}%
1469 \BVerbatimInput{\jobname.ids}%
1470 \HCode{</script>\Hnewline}%
1471 }{}
1472 \Tg</div>%
1473 }{%
1474 \ifvmode\IgnorePar\fi\EndP\HCode{</body>\Hnewline}%
1475 }
1476
1477 % 202501: removed 'prevent spaces as in  "\begin {align}": this is done in luaxake now
1478
1479 % This is a fix for the LAODE book, which uses matlabEquation as if it were an equation
1480 \ScriptEnv{matlabEquation}{\ifvmode \IgnorePar\fi \EndP\HCode{<script type="math/tex; mode=d
1481
1482 ⟨/cfgXimera⟩
```

2.13.2 Semantic HTML

\textbf Using \textbf emits a `<strong>` tag.

```
1483 ⟨∗cfgXimera⟩
1484 \Configure{textbf}{\ifvmode\ShowPar\fi\HCode{<strong>}}{\HCode{</strong>}}
1485 ⟨/cfgXimera⟩
```

\textit Using \textit or similar emits an `<em>` tag.

```
1486 ⟨∗cfgXimera⟩
1487 \Configure{textit}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1488 \Configure{emph}{\ifvmode\ShowPar\fi\HCode{<em>}}{\HCode{</em>}}
1489 ⟨/cfgXimera⟩
```

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

```
1490 ⟨∗cfgXimera⟩
1491 \Configure{texttt}{\ifvmode\ShowPar\fi\HCode{<code>}}{\HCode{</code>}}
1492 ⟨/cfgXimera⟩
```

2.14 Tools

2.14.1 Suppress

suppress (*env.*) 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`.

```
1493 ⟨∗classXimera⟩
1494 \font\dummyft@=dummy \relax
1495 \def\suppress{%
1496   \begingroup\par
1497   \parskip\z@
1498   \offinterlineskip
1499   \baselineskip=\z@skip
1500   \lineskip=\z@skip
1501   \lineskiplimit=\maxdimen
1502   \dummyft@
1503   \count@\sixt@@n
1504   \loop\ifnum\count@ >\z@
1505     \advance\count@\m@ne
1506     \textfont\count@\dummyft@
1507     \scriptfont\count@\dummyft@
1508     \scriptscriptfont\count@\dummyft@
1509   \repeat
1510   \let\selectfont\relax
1511   \let\mathversion\@gobble
```

```
1512    \let\getanddefine@fonts\@gobbletwo
1513    \tracinglostchars\z@
1514    \frenchspacing
1515    \hbadness\@M}
1516 \def\endsuppress{\par\endgroup}
1517 ⟨/classXimera⟩
```

2.14.2   The End

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

```
1518 ⟨*htXimera⟩
1519 \Hinput{ximera}
1520 ⟨/htXimera⟩

1521 ⟨*htXourse⟩
1522 \Hinput{xourse}
1523 ⟨/htXourse⟩

1524 ⟨*cfgXimera⟩
1525 \begin{document}
1526 \EndPreamble
1527 ⟨/cfgXimera⟩
```

3   xourse.cls

```
1528 ⟨*classXourse⟩
```

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

```
1529 \newif\ifnotoc
1530 \notocfalse
1531 \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.

```
1532 \newif\ifnonewpage
1533 \nonewpagefalse
1534 \DeclareOption{nonewpage}{\nonewpagetrue}

1535 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{ximera}}
1536 \ProcessOptions\relax
1537 \LoadClass{ximera}
1538 %      \begin{macrocode}
1539 ⟨/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.

```
1540 ⟨*classXourse⟩
1541 \newcommand{\skip@preamble}{%
1542    \let\document\relax\let\enddocument\relax%
1543    \newenvironment{document}{\let\input\otherinput}{}%
1544    \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:

```
1545 \let\otherinput\input
```

Store usual \maketitle as \othermaketitle

```
1546 \let\othermaketitle\maketitle
```

\maketitle In a xourse file, \maketitle is redefined to give course packet title page and toc.

```
1547 \renewcommand{\maketitle}{ %
1548 \pagestyle{empty}
1549 \begin{center}
1550 ~\\ %puts space at top of page to move title down.
1551 \vskip .25\textheight
1552 \hrulefill\\
1553 \vskip 1em
1554 \bfseries{\Huge \@title} \\
1555 \hrulefill\\
1556 \vskip 3em
1557 {\Large \@author}
1558 \vskip 2em
1559 {\large \@date}
1560 \end{center}
1561 \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.

```
1562 \ifnotoc
1563 \else
1564   \tableofcontents\clearpage
1565   \clearpage
1566 \fi
```

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

```
1567 \pagestyle{main}
```

Renew maketitle to usual definition.

```
1568 \let\maketitle\othermaketitle
```

And we finish with our redefinition of \maketitle.

```
1569 }
1570 \relax
1571 ⟨/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.

```
1572 ⟨*classXourse⟩
1573 \ifnonewpage
1574 \newcommand{\activity}[2][]{%
1575 \setkeys{activity}{#1}
1576   \renewcommand{\input}[1]{}
1577   \begingroup\skip@preamble\otherinput{#2}\endgroup\par\vspace{\topsep}
1578   \let\input\otherinput}
1579 \else
1580 \newcommand{\activity}[2][]{%
1581 \setkeys{activity}{#1}
1582   \renewcommand{\input}[1]{}
1583   \begingroup\skip@preamble\otherinput{#2}\endgroup\clearpage
1584   \let\input\otherinput}
1585 \fi
1586 \relax
1587 ⟨/classXourse⟩

1588 ⟨*htXourse⟩
1589 \renewcommand\activity[2][]{%
1590 \ifvmode \IgnorePar\fi \EndP\HCode{<a class="activity card \activitystyle" href="#2" data-op
```

```
1591 }
1592 ⟨/htXourse⟩
```

When running xake, we can just ignore activities

```
1593 ⟨∗classXourse⟩
1594 \ifxake
1595 \renewcommand\activity[2][]{}
1596 \fi
1597 ⟨/classXourse⟩
```

3.1.2 Practice activities

\practice Like \activity but not expecting a title.

```
1598 ⟨∗classXourse⟩
1599 \ifhandout
1600 \newcommand{\practice}[2][]{
1601  \setkeys{practice}{#1}%!!!!!
1602   \renewcommand{\input}[1]{}
1603   \begingroup\skip@preamble\otherinput{#2}\endgroup
1604   \let\input\otherinput}
1605 \else
1606 \newcommand{\practice}[2][]{\texttt{\detokenize{#2}}%% gives file name for practice
1607  \setkeys{practice}{#1}%!!!!!
1608   \renewcommand{\input}[1]{}
1609   \begingroup\skip@preamble\otherinput{#2}\endgroup
1610   \let\input\otherinput}
1611 \fi
1612 \relax
1613 ⟨/classXourse⟩
```

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

```
1614 ⟨∗classXourse⟩
1615 \ifxake
1616 \renewcommand\practice[2][]{}
1617 \fi
1618 ⟨/classXourse⟩
```

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

```
1619 ⟨∗htXourse⟩
1620 \renewcommand\practice[2][]{%
1621   \ifvmode\IgnorePar\fi\EndP%
1622   \HCode{<a class="activity card practice" href="#2" data-options="#1">#2</a>}%
1623   \IgnoreIndent%
1624 }
1625 ⟨/htXourse⟩
```

3.2 Sectioning

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

```
1626 ⟨∗classXourse⟩
1627 \renewcommand*\l@section{\@dottedtocline{1}{1.5em}{4.2em}}
1628 ⟨/classXourse⟩
```

\subsection The name of a subsection inside an activity.

```
1629 ⟨∗classXourse⟩
1630 \renewcommand*\l@subsection{\@dottedtocline{2}{3.8em}{4.2em}}
1631 ⟨/classXourse⟩
```

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

```
1632 ⟨∗htXourse⟩
1633 \newcounter{ximera@part}
```

```
1634 \setcounter{ximera@part}{0}
1635 \renewcommand\part[1]{%
1636 \stepcounter{ximera@part}%
1637 \ifvmode \IgnorePar\fi \EndP%
1638 %\HCode{<h1 id="part\arabic{ximera@part}" class="card part">}#1\HCode{</h1>}% makes cards dis
1639 \HCode{<h1 id="part\arabic{ximera@part}" class="card part">#1</h1>}%
1640 \IgnoreIndent%
1641 }
1642 ⟨/htXourse⟩
```

\paragraph Paragraph commands emit spans. A small heading.

```
1643 ⟨∗cfgXimera⟩
1644 \renewcommand{\paragraph}[1]{%
1645   \HCode{<span class="paragraphHead">}%
1646   #1%
1647   \HCode{</span>}\par\IgnorePar}
1648 ⟨/cfgXimera⟩
```

\subparagraph An even smaller heading.

```
1649 ⟨∗cfgXimera⟩
1650 \renewcommand{\subparagraph}[1]{%
1651   \HCode{<span class="subparagraphHead">}%
1652   #1%
1653   \HCode{</span>}\par\IgnorePar}
1654 ⟨/cfgXimera⟩
```

3.3 Grading by points

graded (*env.*) 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.

```
1655 ⟨∗classXourse⟩
1656 \newenvironment{graded}[1]{}{}
1657 ⟨/classXourse⟩
```

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

```
1658 ⟨∗htXourse⟩
1659 \renewenvironment{graded}[1]{%
1660 \ifvmode \IgnorePar\fi \EndP\HCode{<div class="graded" data-weight="#1">}\IgnoreIndent%
1661 }{
1662 \ifvmode \IgnorePar\fi \EndP\HCode{</div>}\IgnoreIndent%
1663 }
1664 ⟨/htXourse⟩
```

3.4 Logos

\logo A logo for the xourse.

```
1665 ⟨∗classXourse⟩
1666 \newcommand*{\logo}[1]{%
1667   \ifx\@onlypreamble\@notprerr
1668     \ClassError{xourse}{logo can only be used in the preamble}
1669       {Move your logo command to the preamble}
1670   \else %
1671     \IfFileExists{#1}%
1672       {\gdef\xourse@logo{#1}}%
1673       {\ClassError{xourse}{logo file does not exist}
1674         {To use logo, make sure that the referenced image file exists}}%
1675   \fi%
1676 }
1677
1678 ⟨/classXourse⟩
```

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

```
1679 ⟨∗htXourse⟩
```

```
1680 \Configure{@HEAD}{%
1681   \HCode{<meta name="og:image" content="}%
1682 \ifdefined\xourse@logo%
1683   \xourse@logo%
1684 \fi%
1685 \HCode{" />\Hnewline}}%
1686 ⟨/htXourse⟩
```