

The klfimpl package¹

Philippe Faist philippe.faist@bluewin.ch

August 20, 2020

¹ This document corresponds to klfimpl v0.2, dated 2020/08/20. It is part of the klfengine tool, see <https://github.com/klatexformula/klfengine>.

klfimpl—LaTeX helper code for the implementation of the klfengine (the engine of KLatexFormula 5).

1	Introduction	1
2	Implementation	2
2.1	Some general declarations	2
2.2	Package options and settings	4
2.3	Basic/common implementation macros	6
2.4	Main implementation routine	7
2.5	Background: color, frame, and/or custom elements	10
2.6	Vertical bounding box adjustments (bbox, Xheight, baseline)	13
2.7	Baseline rule	14
2.8	Dump meta-info on standard output	14

■ 1 Introduction

KLatexFormula 5 will ship with a new engine, *klfengine*. This implementation converts a LaTeX equation to different formats such as PDF or PNG. While *KLatexFormula* up to version 4 required a few runs of *gs* and other manual EPS fixes after running *latex* and *dvips*, the new workflow aims at directly doing as much as possible from the latex end of things—get the correct paper size, correct background color, page margins, and so on—therefore cleaning up the workflow and speeding up the whole compilation.

This package provides an environment `\begin{klfcontent} . . . \end{klfcontent}` which typesets the given content in a box, measures the box dimensions, then resizes the page to fit the box as requested including margins and/or alignment to some fixed size, and renders the whole thing.

This package expects that there is a single such environment in the entire document and no other content.

■ 2 Implementation

2.1 Some general declarations

The box in which the LaTeX content will be typeset.

```
1 \newbox\klf@eqnbox
```

Some dimensions etc. that we will track. The main box dimensions (width, height, depth, total height = height + depth):

```
2 \newdimen\klf@w
3 \newdimen\klf@h
4 \newdimen\klf@d
5 \newdimen\klf@th
```

The paper size (width/height):

```
6 \newdimen\klf@ppw
7 \newdimen\klf@pph
```

Any offset that should be used to display the box (useful if margins are requested, or a fixed page width and/or height are requested):

```
8 \newdimen\klf@hshift
9 \newdimen\klf@vshift
```

And record some font dimensions:

```
10 \newdimen\klf@em
11 \newdimen\klf@ex
12 \newdimen\klf@capxhgt
```

We will compute any user input dimensions as proper dimension *inside* the equation box to make sure font settings are taken into account correctly. The dimensions will be stored here.

```
13 \newdimen\klf@dim@fixedwidth
14 \newdimen\klf@dim@fixedheight
15 \newdimen\klf@dim@topmargin
16 \newdimen\klf@dim@rightmargin
17 \newdimen\klf@dim@bottommargin
18 \newdimen\klf@dim@leftmargin
```

Remember which engine we're running under. To speed up things and to avoid problems with like `iftex` package not existing, we simply require the caller to tell us via a package option what latex engine is being run (latex with DVI output, pdf`latex`, x`latex`, or l`u`a`l`a`t`e`x`).

```
19 \newif\ifklf@ltxengine@latex \klf@ltxengine@latexfalse
20 \newif\ifklf@ltxengine@pdflatex \klf@ltxengine@pdflatexfalse
```

```

21 \newif\ifklf@ltxengine@xelatex \klf@ltxengine@xelatexfalse
22 \newif\ifklf@ltxengine@lualatex \klf@ltxengine@lualatexfalse

```

If, for some reason, this package is called in a different context where we wouldn't want the layout to be set to zero by default, then there is a package option for this (`keeplayoutsizes`). This is the corresponding `\newif` flag:

```

23 \newif\ifklf@keeplayoutsizes
24 \klf@keeplayoutsizesfalse

```

The user can specify a fixed width and/or a fixed height for the resulting layout. These will be stored here, if applicable (or they will remain empty). The user input should be stored as a *macro*, not as a *dimen*, because we want a dimension given in font-specific metrics (e.g. `4.2em`) to be computed correctly relative to the equation font.

```

25 \def\klf@set@fixedwidth{}
26 \def\klf@set@fixedheight{}

```

Enable user to specify margins around the equations.

```

27 \def\klf@set@topmargin{0.1ex}
28 \def\klf@set@rightmargin{0.1ex}
29 \def\klf@set@bottommargin{0.1ex}
30 \def\klf@set@leftmargin{0.1ex}

```

Enable user to specify global scaling factors for the full resulting box. Scale will be applied to margins as well as to fixed size.

```

31 \def\klf@set@xscale{1}
32 \def\klf@set@yscale{1}

```

User can specify how the equation is aligned inside the box, in case we have a fixed width or a fixed height. The horizontal (resp. vertical) alignment coefficient is 0 for left (resp. top) alignment, 0.5 for middle alignment, and 1 for right (resp. bottom) alignment. It can be any value that interpolates between these.

```

33 \def\klf@set@xaligncoeff{0.5}
34 \def\klf@set@yaligncoeff{0.5}

```

User can specify the reference points for top and bottom alignment. The top can be aligned to the natural height of the box (`bbox`, the default) or to the height of a capital 'X' (`Xheight`). The bottom can be aligned to the natural depth of the box (`bbox`, the default) or to the baseline (`baseline`). (When you set `baseline`, make sure you have a fixed size or reasonable margins to accomodate any box depth.)

```

35 \def\klf@set@topalignment{bbox}
36 \def\klf@set@bottomalignment{bbox}

```

User can tell if they would like a line to be drawn where the baseline is, for instance as a visual reference for use in vector graphics editing software. This can be none or line.

```
37 \def\klf@set@baselineruletype{none}
```

2.2 Package options and settings

Set the \LaTeX engine.

```
38 \DeclareOption{latex}{\klf@ltxengine@latextrue}
39 \DeclareOption{pdflatex}{\klf@ltxengine@pdflatextrue}
40 \DeclareOption{xelatex}{\klf@ltxengine@xelatextrue}
41 \DeclareOption{lualatex}{\klf@ltxengine@lualatextrue}
```

Package option to inhibit resetting the page layout to zero by default.

```
42 \DeclareOption{keeplayoutsizes}{\klf@keeplayoutsizetrue}
```

Now process those options

```
43 \DeclareOption*{\PackageError{klfimpl}{Unknown option ‘\CurrentOption’}{}}
44 \ProcessOptions\relax
```

The following could have been specified as package options, but for the sake of simplicity (and to avoid having to use keyval/xkeyval and so on, the information is provided via a simple macro call.

`\klfSetFixedWidth`
`\klfSetFixedHeight`

If applicable, set the fixed width and/or fixed height of the content to typeset.

```
45 \def\klfSetFixedWidth#1{%
46   \xdef\klf@set@fixedwidth{#1}}
47 \def\klfSetFixedHeight#1{%
48   \xdef\klf@set@fixedheight{#1}}
```

`\klfSetTopMargin`
`\klfSetRightMargin`
`\klfSetBottomMargin`
`\klfSetLeftMargin`

Same for equation margins:

```
49 \def\klfSetTopMargin#1{%
50   \xdef\klf@set@topmargin{#1}}
51 \def\klfSetRightMargin#1{%
52   \xdef\klf@set@rightmargin{#1}}
53 \def\klfSetBottomMargin#1{%
54   \xdef\klf@set@bottommargin{#1}}
55 \def\klfSetLeftMargin#1{%
56   \xdef\klf@set@leftmargin{#1}}
```

`\klfSetXScale`
`\klfSetYScale`
`\klfSetScale`

Specify horizontal and vertical scaling factors. Values are a multiplying factor for dimensions: the value 1 means original size, 0.5 half size, 2 double size. If a value different than 1 was specified, the graphics package is loaded. *NOTE: This*

has to be the exact expression 1, not 1.0 or 1.. The \klfSetScale macro is a convenience macro that sets both horizontal and vertical scaling factors to the same value.

```

57 \def\klfSetXScale#1{%
58   \xdef\klf@set@xscale{#1}%
59   \ifx\klf@set@xscale\klf@macro@one
60   \else
61     \RequirePackage{graphics}%
62   \fi
63 }
64 \def\klfSetYScale#1{%
65   \xdef\klf@set@yscale{#1}%
66   \ifx\klf@set@yscale\klf@macro@one
67   \else
68     \RequirePackage{graphics}%
69   \fi
70 }
71 \def\klfSetScale#1{%
72   \xdef\klf@set@xscale{#1}%
73   \xdef\klf@set@yscale{#1}%
74   \ifx\klf@set@xscale\klf@macro@one
75   \else
76     \RequirePackage{graphics}%
77   \fi
78 }
79 \def\klf@macro@one{1}

```

\klfSetXAlignCoeff
\klfSetYAlignCoeff

```

80 \def\klfSetXAlignCoeff#1{%
81   \xdef\klf@set@xaligncoeff{#1}%
82 }
83 \def\klfSetYAlignCoeff#1{%
84   \xdef\klf@set@yaligncoeff{#1}%
85 }

```

\klfSetTopAlignment
\klfSetBottomAlignment

Macros for the user to set top and bottom alignment. See earlier for explanation of possible values. The possible values will be used to invoke a macro named e.g. \klf@correctboxheight@@<top-alignment-name> and \klf@correctboxdepth@@<bottom-alignment-name>, see below in {klfcontent} environment.

```

86 \def\klfSetTopAlignment#1{%
87   \xdef\klf@set@topalignment{#1}}
88 \def\klfSetBottomAlignment#1{%
89   \xdef\klf@set@bottomalignment{#1}}

```

\klfSetBaselineRuleType

Set which kind of baseline rule we would like, if any.

```

90 \def\klfSetBaselineRuleType#1{%
91   \xdef\klf@set@baselineruletype{#1}}

```

2.3 Basic/common implementation macros

First of all, a simple macro to reset all LaTeX layout dimensions.

```

92 \def\klf@ZeroLayoutSizes{%
93   \oddsidemargin=\z@\relax
94   \evensidemargin=\z@\relax
95   \topmargin=\z@\relax
96   \voffset=-1in\relax
97   \hoffset=-1in\relax
98   \headsep=\z@\relax
99   \headheight=\z@\relax
100  \marginparsep=\z@\relax
101  \footskip=\z@\relax
102  \parindent=\z@\relax
103  \parskip=\z@\relax
104  \topskip=\z@\relax
105 }

```

`\klf@ZeroDisplaySkips` And define a routine that sets all the display-related skips to zero so that we can use this inside a `\vbox`.

```

106 \def\klf@ZeroDisplaySkips{%
107   \abovedisplayskip=\z@\relax
108   \belowdisplayskip=\z@\relax
109   \abovedisplayshortskip=\z@\relax
110   \belowdisplayshortskip=\z@\relax
111 }

```

By default, reset all these dimensions right away, unless the `keeplayoutsizes` package option was provided.

```

112 \ifklf@keeplayoutsizes
113 \else
114   \klf@ZeroLayoutSizes
115   \klf@ZeroDisplaySkips
116 \fi

```

`\klfSetPaperSize` Change the paper size. For pdf_latex and xe/luatex this can be called after `\begin{document}` but for latex with traditional dvi output this must be issued in the preamble.

```

117 \def\klfSetPaperSize#1#2{%
118   \@tempdima=#1\relax
119   \@tempdimb=#2\relax
120   \klf@SetPaperSize@FromDims\@tempdima\@tempdimb

```

```

121 }
122 \def\klf@SetPaperSize@FromDims#1#2{%
123   \global\textwidth=#1\relax
124   \global\textheight=#2\relax
125   \global\hsize=#1\relax
126   \global\vsizer=#2\relax
127   \global\paperwidth=#1\relax
128   \global\paperheight=#2\relax
129   \ifklf@ltxengine@pdflatex
130     \global\pdfpagewidth=#1\relax
131     \global\pdfpageheight=#2\relax
132   \fi
133   \ifklf@ltxengine@xelatex
134     \global\pdfpagewidth=#1\relax
135     \global\pdfpageheight=#2\relax
136   \fi
137   \ifklf@ltxengine@lualatex
138     \global\pagewidth=#1\relax
139     \global\pageheight=#2\relax
140   \fi
141 }

```

2.4 Main implementation routine

klfcontent The argument should be a box command (e.g., `\hbox`, `\vbox`, `\vtop`, `\vcenter`). Example usage: `\begin{klfcontent}\vcenter{\langle initialization code \rangle}...` or `\begin{klfcontent}\hbox to 1cm{\langle init. code \rangle}...`

```

142 \def\klfcontent#1#2{%
143   \unskip
144   \samepage
145   \setbox\klf@eqnbox=#1\bgroup
146     \klf@ZeroDisplaySkips%
147     #2%
148     \global\klf@em=1em\relax
149     \global\klf@ex=1ex\relax
150     \setbox0=\hbox{X}%
151     \global\klf@capxhgt=\ht0%
152     \ifx\klf@set@fixedwidth\@empty\else
153       \global\klf@dim@fixedwidth=\klf@set@fixedwidth\relax
154     \fi
155     \ifx\klf@set@fixedheight\@empty\else
156       \global\klf@dim@fixedheight=\klf@set@fixedheight\relax
157     \fi
158     \global\klf@dim@topmargin=\klf@set@topmargin\relax
159     \global\klf@dim@rightmargin=\klf@set@rightmargin\relax
160     \global\klf@dim@bottommargin=\klf@set@bottommargin\relax
161     \global\klf@dim@leftmargin=\klf@set@leftmargin\relax
162 }

```

```

163 \def\endklfcontent{%
164   \egroup

```

Now we record the box dimensions.

```

165   \klf@w=\wd\klf@eqnbox\relax
166   \klf@h=\ht\klf@eqnbox\relax
167   \klf@d=\dp\klf@eqnbox\relax

```

If we shouldn't align to the bounding box, correct the height and the depth of the box to whatever is requested by the corresponding top/bottom alignment options.

```

168   \csname klf@correctboxheight@\klf@set@topalignment\endcsname
169   \csname klf@correctboxdepth@\klf@set@bottomalignment\endcsname

170   \klf@th=\klf@h\relax
171   \advance \klf@th \klf@d \relax

```

Determine page size and offsets. Take into account any possible fixed paper width or height and any margins.

```

172   \ifx\klf@set@fixedwidth\@empty%
173     \klf@ppw=\klf@w\relax
174     \advance \klf@ppw \klf@dim@leftmargin \relax
175     \advance \klf@ppw \klf@dim@rightmargin \relax
176     \klf@hshift=\klf@dim@leftmargin\relax
177   \else%
178     \klf@ppw=\klf@dim@fixedwidth\relax
179     \klf@hshift=\klf@set@xaligncoeff\klf@ppw\relax
180     \advance \klf@hshift -\klf@set@xaligncoeff\klf@w\relax
181     \advance \klf@hshift -\klf@set@xaligncoeff\klf@dim@rightmargin\relax
182     \advance \klf@hshift -\klf@set@xaligncoeff\klf@dim@leftmargin\relax
183     \advance \klf@hshift \klf@dim@leftmargin\relax
184   \fi

185   \ifx\klf@set@fixedheight\@empty%
186     \klf@pph=\klf@th\relax
187     \advance \klf@pph \klf@dim@topmargin \relax
188     \advance \klf@pph \klf@dim@bottommargin \relax
189     \klf@vshift=\klf@dim@topmargin\relax
190   \else%
191     \klf@pph=\klf@dim@fixedheight\relax
192     \klf@vshift=\klf@set@yaligncoeff\klf@pph\relax
193     \advance \klf@vshift -\klf@set@yaligncoeff\klf@th\relax
194     \advance \klf@vshift -\klf@set@yaligncoeff\klf@dim@bottommargin\relax
195     \advance \klf@vshift -\klf@set@yaligncoeff\klf@dim@topmargin\relax
196     \advance \klf@vshift \klf@dim@topmargin\relax
197   \fi

```

No scale has been applied yet. Call the rendering routine that will take into account scaling factors as necessary.


```
198 \klf@RenderContentBox
```

Finally dump all meta-info to the standard output to provide additional information back to *klatexformula*.

```
199 \klfDumpMetaInfo
200 \ignorespaces
201 }
```

`\klf@RenderContentBox` The `\klf@RenderContentBox` macro sets the paper size (if the current latex engine allows this at this point), and displays the box accordingly.

If we're using a pdf-based engine (`pdflatex`, `xelatex` or `lualatex`), then we set the page size immediately (scaled correctly). If we're using the latex (\LaTeX \rightarrow DVI) engine, then we cannot set the page size at this point, because the page size needs to be set in the preamble. In this case, we don't do anything now, but *klatexformula* performs a second pass where the correct exact page size is set in preamble (via meta-data dumped by `\klfDumpMetaInfo`).

```
202 \newbox\klf@final@box
203 \def\klf@RenderContentBox{%
204   \ifklf@ltxengine@latex% tough luck
205   \else
206     \@tempdima=\klf@ppw
207     \@tempdima=\klf@set@xscale\@tempdima\relax
208     \@tempdimb=\klf@pph
209     \@tempdimb=\klf@set@yscale\@tempdimb\relax
210     \klf@SetPaperSize@FromDims\@tempdima\@tempdimb%
211   \fi
```

To render the contents, we check whether any scaling is applied. If so, we call `\klf@do@scale` which wraps the argument in an appropriate `\scalebox` (provided by the graphics package). (Otherwise we simply render the box without any `\scalebox`.)

```
212 \let\klf@next\@firstofone
213 \ifx\klf@set@xscale\klf@macro@one\else
214   \let\klf@next\klf@do@scale\fi
215 \ifx\klf@set@yscale\klf@macro@one\else
216   \let\klf@next\klf@do@scale\fi
217 \nobreak
218 \hsize=\klf@ppw
219 \setbox\klf@final@box=\vbox{\hbox to \z@{%
220   \klf@next{%
```

Here, we actually render the box contents. There are three items to draw: (1) the background, (2) the baseline rule, if any, and (3) the actual equation box.

Begin with the background. The code in `\klf@DrawBackground` is designed so that it takes no horizontal or vertical space.

```
221 \klf@DrawBackground
```

Draw the baseline rule and contents.

```

222     \vbox to \z@{%
223         \hrule \@height\z@\nobreak
224         \vskip \klf@vshift\relax\nobreak
225         \hbox{\vrule \@width\z@ \relax
226             \raise \klf@d \hbox to \z@{%
227                 \csname klf@baseline@rule@@\klf@set@baselineruletype\endcsname
228             }%
229             \hskip \klf@hshift\relax
230             \raise \klf@d \box\klf@eqnbox
231         }%
232     }%
233 }%
234 }}%
235 \c@page=\z@
236 \shipout\box\klf@final@box
237 }
238 \def\klf@do@scale#1{%
239     \scalebox{\klf@set@xscale}{\klf@set@yscale}{#1}%
240 }

```

2.5 Background: color, frame, and/or custom elements

Enable the user to specify a custom background, and even draw stuff on it if they like. I'm not sure what the best API is to let the user draw what they like.

`\klfSetBackgroundColor` Let the user set a simple color, with a custom opacity (semitransparency is provided by the pgf package, so included it if necessary).

`\klfSetBackgroundColorOpacity`

```

241 \newcommand\klfSetBackgroundColor[1]{%
242     \klfEnsureColorPackageLoaded
243     \definecolor{klfbgcolor}{RGB}{#1}%
244     \def\klf@set@bgcoloropacity{1}%
245 }
246 \def\klf@set@bgcoloropacity{0}
247 \newcommand\klfSetBackgroundColorOpacity[1]{%
248     \edef\klf@set@bgcoloropacity{#1}%
249     \ifdim#1\p@=\z@\relax
250     \else
251         \ifdim#1\p@=\p@\relax
252         \else
253             \RequirePackage{pgf}%
254         \fi
255     \fi
256 }
257 \def\klfEnsureColorPackageLoaded{%
258     \@ifpackageloaded{color}{}%
259     \@ifpackageloaded{xcolor}{}%

```

```

260     \RequirePackage{color}%
261   }%
262 }%
263 }

```

Some temporary registers.

```

264 \newdimen\klf@set@bgtmp@rectw
265 \newdimen\klf@set@bgtmp@recth

```

`\klf@DrawBackground@Color` The code to draw the background color. Draw the background color as a rectangle, with the required opacity. Of course, don't do this if the background rectangle is fully transparent.

```

266 \def\klf@DrawBackground@Color{%
267   \ifdim\klf@set@bgcoloropacity pt=\z@\relax
268   \else
269     \klf@set@bgtmp@rectw=\klf@ppw
270     \advance\klf@set@bgtmp@rectw 2\klf@set@bgcolor@bleed
271     \klf@set@bgtmp@recth=\klf@pph
272     \advance\klf@set@bgtmp@recth 2\klf@set@bgcolor@bleed
273     \if\klf@set@bgcoloropacity pt=\p@\relax
274       \let\klf@tmp@pgfsetfillopacity\@gobble
275     \else
276       \let\klf@tmp@pgfsetfillopacity\pgfsetfillopacity
277     \fi
278     \begingroup
279       \color{klfbgcolor}%
280       \klf@tmp@pgfsetfillopacity{\klf@set@bgcoloropacity}%
281       \hbox to \z@{%
282         \hskip -\klf@set@bgcolor@bleed\relax
283         \vbox to \z@{%
284           \vskip-\klf@set@bgcolor@bleed\relax
285           \rule{\klf@set@bgtmp@rectw}{\klf@set@bgtmp@recth}%
286         }%
287       }%
288       \klf@tmp@pgfsetfillopacity{1}%
289     \endgroup
290   \fi
291 }

```

`\klfSetBackgroundFrameXOffset` Optionally draw a frame around the contents as a background decoration.

```

\klfSetBackgroundFrameYOffset
\klfSetBackgroundFrameOffset
\klfSetBackgroundFrameThickness
\klfSetBackgroundFrameColor

```

```

292 \newdimen\klf@set@bgframe@xoffset
293 \klf@set@bgframe@xoffset=\z@\relax
294 \newdimen\klf@set@bgframe@yoffset
295 \klf@set@bgframe@yoffset=\z@\relax
296 \def\klfSetBackgroundFrameXOffset#1{%
297   \klf@set@bgframe@xoffset=#1\relax
298 }

```

```

299 \def\klfSetBackgroundFrameYOffset#1{%
300   \klf@set@bgframe@yoffset=#1\relax
301 }
302 \def\klfSetBackgroundFrameOffset#1{%
303   \klf@set@bgframe@xoffset=#1\relax
304   \klf@set@bgframe@yoffset=\klf@set@bgframe@xoffset\relax
305 }
306 \newdimen\klf@set@bgframe@thickness
307 \klf@set@bgframe@thickness=\z@\relax
308 \def\klfSetBackgroundFrameThickness#1{%
309   \klf@set@bgframe@thickness=#1\relax
310 }
311 \def\klf@set@bgframe@setcolor{}
312 \def\klfSetBackgroundFrameColor#1{%
313   \klfEnsureColorPackageLoaded
314   \definecolor{klffrmcolor}{RGB}{#1}%
315   \def\klf@set@bgframe@setcolor{\color{klffrmcolor}}
316 }

```

Code to draw the background frame:

`\klf@DrawBackground@Frame`

```

317 \def\klf@DrawBackground@Frame{%
318   \ifdim\klf@set@bgframe@thickness=\z@\relax
319   \else
320     \klf@set@bgtmp@rectw=\klf@ppw
321     \advance\klf@set@bgtmp@rectw -2\klf@set@bgframe@xoffset
322     \klf@set@bgtmp@recth=\klf@pph
323     \advance\klf@set@bgtmp@recth -2\klf@set@bgframe@yoffset
324     \hbox to \z@{%
325       \hskip \klf@set@bgframe@xoffset\relax
326       \vbox to \z@{%
327         \vskip \klf@set@bgframe@yoffset\relax
328         \begingroup
329           \fboxsep=-\klf@set@bgframe@thickness\relax
330           \fboxrule=\klf@set@bgframe@thickness\relax
331           \klf@set@bgframe@setcolor
332           \fbox{\phantom{\rule{\klf@set@bgtmp@rectw}{\klf@set@bgtmp@recth}}}%
333         \endgroup
334       }%
335     }%
336   \fi
337 }

```

`\klfAddDrawBackgroundCommands`
`\klfAddDrawBackgroundGraphics`

We also provide a generic hook, in case the user wants to draw more fancy stuff. The user can call `\klfAddDrawBackgroundCommands` to append drawing commands to the background. The origin is the top left corner of the image. The user can get dimensions, etc., via the `\klf@ppw/\klf@pph`, etc. lengths

(for now). The user code for each call to `\klfAddDrawBackgroundCommands` is wrapped in a zero-sized box located at the top left point of the image.

The command `\klfAddDrawBackgroundGraphics` is a shorthand for inserting a background graphic using the `graphicx` package's `\includegraphics[...]{...}`.

```

338 \newtoks\klf@set@bgextradrawcommands
339 \newcommand\klfAddDrawBackgroundCommands[1]{%
340   \klf@set@bgextradrawcommands=\expandafter{\the\klf@set@bgextradrawcommands
341     \hbox to \z@{\vbox to \z@{%
342       #1%
343     }}%
344   }%
345 }
346 \newcommand\klfAddDrawBackgroundGraphics[2][ ]{%
347   \RequirePackage{graphicx}%
348   \klfAddDrawBackgroundCommands{%
349     \includegraphics[#1]{#2}%
350   }%
351 }

```

`\klf@DrawBackground@CustomCommands`

Code to render the custom commands. Remember to enclose code in a zero-sized box.

```

352 \def\klf@DrawBackground@CustomCommands{%
353   \if\relax\detokenize\expandafter{\the\klf@set@bgextradrawcommands}\relax
354   \else
355     \the\klf@set@bgextradrawcommands
356   \fi
357 }

```

`\klf@DrawBackground`

Here's our internal code that renders the background. The background color will be drawn as a filled rectangle, extending on all sides with a bleed length stored in `\klf@set@bgcolor@bleed`.

```

358 \newdimen\klf@set@bgcolor@bleed
359 \klf@set@bgcolor@bleed=\p@
360 \def\klf@DrawBackground{%
361   \klf@DrawBackground@Color
362   \klf@DrawBackground@Frame
363   \klf@DrawBackground@CustomCommands
364 }

```

2.6 Vertical bounding box adjustments (`bbox`, `Xheight`, `baseline`)

Now we define the top/bottom alignment correction routines. These “fix” the height and the depth of the box (rather, their values recorded in `\klf@h` and

`\klf@d)` according to the given options.

The `bbox` top and bottom alignment options is the default, and leaves the box dimensions unchanged.

```
365 \def\klf@correctboxheight@@bbox{}
366 \def\klf@correctboxdepth@@bbox{}
```

The `Xheight` top alignment option sets the height of the “box” to be the height of a capital “X”. WARNING: This assumes that the box only contains a single line of text.

```
367 \def\klf@correctboxheight@@Xheight{%
368   \klf@h=\klf@capxhgt
369 }
```

The baseline bottom alignment option sets the bottom of the “box” to be the baseline, so sets the depth to zero.

```
370 \def\klf@correctboxdepth@@baseline{%
371   \klf@d=\z@
372 }
```

2.7 Baseline rule

Now we define the baseline rule types. These simply draw whatever they want, typically a simple `\vrule` of a given width (because we’re in horizontal mode).

First, we have the none rule type which simply typesets nothing.

```
373 \def\klf@baseline@rule@@none{}
```

Then we have the line rule, which draws a horizontal line throughout the box at the baseline height.

`\klfBaselineRuleLineSetup` The line is controlled by the macros `\klfBaselineRuleLineSetup` and
`\klfBaselineRuleLineThickness` `\klfBaselineRuleLineThickness`:

```
374 \def\klfBaselineRuleLineSetup{}
375 \def\klfBaselineRuleLineThickness{0.05\p@}
```

You may redefine these to style the line as appropriate. For instance, this would give you a blue baseline that is 0.2pt thick:

```
\renewcommand\klfBaselineRuleLineSetup{\color{blue}}
\renewcommand\klfBaselineRuleLineThickness{0.2pt}
```

And finally this is the code that draws the line.

```
376 \def\klf@baseline@rule@@line{%
```

```

377 \begingroup
378 \klfBaselineRuleLineSetup
379 \vrule width\klf@ppw height\z@ depth\klfBaselineRuleLineThickness\relax
380 \endgroup
381 }

```

2.8 Dump meta-info on standard output

Define the routine that communicates back to *klatexformula* meta-info about the typeset content. This is automatically called in `\end{klfcontent}`.

`\klfDumpMetaInfo` Dump meta-info on standard output to provide additional information to KLaTeXFormula. Careful, scaling factors have not been applied yet. So apply them here before displaying the quantities.

```

382 \def\klfDumpMetaInfo{%
383 \begingroup
384 \klf@em=\klf@set@xscale\klf@em\relax
385 \klf@ex=\klf@set@yscale\klf@ex\relax
386 \klf@capxhgt=\klf@set@yscale\klf@capxhgt\relax
387 \klf@ppw=\klf@set@xscale\klf@ppw\relax
388 \klf@pph=\klf@set@yscale\klf@pph\relax
389 \klf@hshift=\klf@set@xscale\klf@hshift\relax
390 \klf@vshift=\klf@set@yscale\klf@vshift\relax
391 \klf@w=\klf@set@xscale\klf@w\relax
392 \klf@h=\klf@set@yscale\klf@h\relax
393 \klf@d=\klf@set@yscale\klf@d\relax
394 \klf@th=\klf@set@yscale\klf@th\relax
395 \message{%
396 ^^J%
397 ***-KLF-META-INFO-BEGIN-***^^J%
398 EM={\the\klf@em}^^J%
399 EX={\the\klf@ex}^^J%
400 CAP_X_HEIGHT={\the\klf@capxhgt}^^J%
401 PAPER_WIDTH={\the\klf@ppw}^^J%
402 PAPER_HEIGHT={\the\klf@pph}^^J%
403 HSHIFT={\the\klf@hshift}^^J%
404 VSHIFT={\the\klf@vshift}^^J%
405 BOX_WIDTH={\the\klf@w}^^J%
406 BOX_HEIGHT={\the\klf@h}^^J%
407 BOX_DEPTH={\the\klf@d}^^J%
408 BOX_TOTALHEIGHT={\the\klf@th}^^J%
409 ^^J%
410 ***-KLF-META-INFO-END-***^^J%
411 }
412 \endgroup
413 }%

```