

The `ucph-revy` class^{*}

Kristoffer Levin Hansen
kristoffer@levinhansen.dk

September 9, 2023

Abstract

A \LaTeX class for typesetting scripts in the style used for student theatrical productions at the University of Copenhagen’s (ucph) science faculties. Formerly known to some as `revy.sty`.

Contents

1	Introduction	1
2	Usage	3
2.1	Invoking and arguments	3
2.2	Thumb index	4
2.3	The infoblock	13
2.3.1	Page headers	17
2.4	Assignment lists	18
2.5	Body text	21
2.5.1	<code>sketch</code>	21
2.5.2	<code>song</code>	23
2.5.3	Both <code>sketch</code> and <code>song</code>	25
2.6	The <code>.aux</code> file	27
A	Complete example	31
B	Example of a thumb index page	34

1 Introduction

A style of typesetting scripts in \LaTeX , originally developed for DIKURevy, the students’ revue at DIKU, the Computer Science Institute at ucph, the University of Copenhagen. It has spread to adjacent revues at ucph Science as they have sprung into existence. Once known to its users as the package `revy.sty`, that package has been transplanted into this class, at the proper place in the \LaTeX hierarchy[3], practically in its entirety.

^{*}This document corresponds to `ucph-revy` v1.0.0, dated 2023/05/23.

Figure 1

The Meaning of Liff 3001

Example

written by an examplesmith

Melody: Monty Python: “Always look on the bright side of life”
(<https://youtu.be/SJUhlRoBL8M>)

Status: example TeX–responsible Probably you (1 minute, 47 seconds)	Version 1.0 September 9, 2023 2 pages
---	---

Roles:

KA	(Graham)	Arthur, king of all the Britons
P	(Michael)	Peasant
RC	(John)	Roman Centurion
R1...?		Several Roman Soldiers
CC	(Eric)	Crucified Criminal
B	(Graham)	Brian

Props:

Large sign
Field of crosses

3

Act 1

Lights up on a muddy field with a castle in the background.

KA (*As he enters*): Hello!

KA: Now, by rights, you should all be kneeling, given that I am Arthur, king of all the Britons. However, it has been brought to my attention that the audience seating is not laid out to allow for that. We had some considerable trouble during dress rehearsals. So I will excuse you from that requirement.

Now, has anyone seen a particularly fancy cup? Perhaps out in the bar? You see, I am on a quest given to me by God, to find the holy... (*Interrupted, as he steps in a hole.*) Oh dear...

P (*Shrieking, as he rears up from the ground*): Oi! What do you think you’re doing!

P: You broke my sign!

KA: Sign? What sign?

P: Well, look.

P raises up a sign that reads

ROMANES EUNT DOM[]

The last bit of the sign has a foot-shaped hole in it.

P: You can’t read it anymore!

KA: You certainly can! It says “Romans go home.”

RC (*Appears from offstage*): No it doesn’t.

P & KA are startled.

Act 1

Several Roman soldiers appear, and drag KA off stage, while KA attempts to protest.

KA (*As he is dragged off*): I am not him! I am Arthur, King of the Britons!

The scene changes to a field of crosses, with several crucified criminals and B.

CC is in the middle of singing to B.

CC: ...be silly chumps,
Just purse you’re lips and whistle,
That’s the thing.

And, always look on the bright side of life,
Always look on the right side of life,
For life is quite absurd,
An. death’s the final word...

RC (*Shouting as he walks on stage*): Right, you lanky reprobate! Don’t think that the mighty Roman bureaucracy doesn’t keep track of its inmates

KA is dragged on stage by two Roman soldiers behind RA.

RA: Look, we have your designated cross ready for you right... (*points to B’s cross, then falters when he notices B.*)

Everyone looks back and forth between B and KA.

End scene

3

Act 1

Several Roman soldiers appear, and drag KA off stage, while KA attempts to protest.

KA (*As he is dragged off*): I am not him! I am Arthur, King of the Britons!

The scene changes to a field of crosses, with several crucified criminals and B.

CC is in the middle of singing to B.

CC: ...be silly chumps,
Just purse you’re lips and whistle,
That’s the thing.

And, always look on the bright side of life,
Always look on the right side of life,
For life is quite absurd,
An. death’s the final word...

RC (*Shouting as he walks on stage*): Right, you lanky reprobate! Don’t think that the mighty Roman bureaucracy doesn’t keep track of its inmates

KA is dragged on stage by two Roman soldiers behind RA.

RA: Look, we have your designated cross ready for you right... (*points to B’s cross, then falters when he notices B.*)

Everyone looks back and forth between B and KA.

End scene

This class features the ability to set lines of dialogue and song lyrics with a clearly associated indication of the speaker or singer, along with stage directions. The style is intended for a revue, which is formed out of a number of little pieces, so there commands for typesetting a cast list and prop list, along with relevant information like the time to perform, or the name of a larger piece, which are intended to be typeset along with the title.

Figure 1 shown an example of how all this looks when typeset.

If you're looking for the script package that's designed to help with typesetting and compositing a full manuscript, it was last seen at <https://github.com/fysikrevy/fysikrevytex>.

According to `revy.sty`, its origianl authors were Uffe Friis Lichtenberg, Arne John Glenstrup and Anders Komár Ravn.

Change History

v1.0.0		
General: Initial conversion to a		sketch: The body text macros
class	1	(<code>\scene</code> , etc.) are no longer
New machinery for thumb		available outside the song and
indexes	4	sketch environments. 21
<code>\does</code> : New macro	26	song: The body text macros
<code>\maketitle</code> : New layout of the tile		(<code>\scene</code> , etc.) are no longer
block	13	available outside the song and
		sketch environments. 23

2 Usage

An example of a `.tex` file that will typeset something like figure 1 is presented in appendix A, or in the file `Example.tex`, which should have been produced by L^AT_EX alongside the documentation file that you are reading. In the present section, we'll hit some of the highlights of that file, to understand how it works.

2.1 Invoking and arguments

First, to use the `ucph-revy` class in a `.tex` file, that file must start with

```
\documentclass{ucph-revy}
```

article `ucph-revy` inherits from the `article` class, and accepts all its options. It sets `a4paper` and `11pt` by default, if not given any contrary options.

thumbindex In addition, `ucph-revy` add the options `thumbindex` and `planfile`, which activates the construction of thumb indexes in the file and allows setting the file that the thumb index is built from. We'll go through exactly how this works in section 2.2.

Implementation

We handle class options here, at the start of the file. To do so, we need this package.

```
1 \RequirePackage{xstring}
```

`\if@thumbindex` Keeps track of `thumbindex`. We will do stuff with it when we get to section 2.2.

```
2 \newif\if@thumbindex\@thumbindexfalse
```

And now we're ready to define the arguments:

`thumbindex`

```
3 \DeclareOption{thumbindex}{\@thumbindextrue}
```

`planfile=` Takes a text string, and so needs particular treatment. `xstring` defines, among other things, the macro `\IfBeginWith`

```
4 \newcommand{\planfile}{aktoversigt.plan}
5 \DeclareOption*{
6   \IfBeginWith{\CurrentOption}{planfile=}{
7     \def\planfile#1=#2{#2}
8     \edef\planfile{\expandafter\planfile\CurrentOption}
9   }{
10    \PassOptionsToClass{\CurrentOption}{article}
11  }
12 }
13 \ProcessOptions\relax
14 \LoadClass[a4paper,11pt]{article}
```

Implementation

These commands make the text block taller.

```
15 \setlength{\topmargin}{0cm}
16 \setlength{\voffset}{-1cm}
17 \setlength{\textheight}{\paperheight}
18 \addtolength{\textheight}{-4cm}
```

In the example, we've included three packages that aren't strictly necessary for `ucph-revy` to function, but will probably make your life easier.

<code>\usepackage [utf8]{inputenc}</code>	
<code>\usepackage [T1]{fontenc}</code>	Modernize L ^A T _E X's handling of (particularly special) characters.
<code>\usepackage {hyperref}</code>	Enables the creation of hyperlinks, like the one in figure 1.
<code>\urlstyle {sf}</code>	Demands that hyperlinks be set in sans serif, rather than monospace

2.2 Thumb index

To help organizing a compiled manuscript, which might consist of a collection of many `.tex` files, we provide a facility for creating a thumb index in the typeset margins. That's the gray and black boxes on the edge of the pages in figure 1. It is off by default, since it only really makes sense in a compilation of documents,

`thumbindex` but is enabled by giving the option `thumbindex` to the class.

```

\if@thumbindex    \if@thumbindex was defined on page 3, and just tells us if the option
thumbindex was given. If it wasn't, well just skip all the code in this section.
19 \if@thumbindex
20   \expandafter\@firstofone
21 \else
22   \expandafter\@gobble
23 \fi{

```

The structure of the consolidated manuscript is defined in a separate text file, which contains the filenames of the `.tex` files that make up the manuscript, and divide the show into sections with their own headings. By default, we assume that the plan file is in the same folder as the current `.tex` file, and named `planfile=aktoversigt.plan`. This may be changed by passing `planfile=<filename>` as a parameter to the class. Calling both these options might look like so:

```
\documentclass[thumbindex,planfile=../plan.txt]{ucph-revy}
```

An example of such a plan file might be:

```

Act 1
songs/opener.tex
sketches/Example.tex
sketches/anarchocommunist_commune.tex

Act 2
songs/glitzy_dance_act.tex
sketches/crossdressing.tex

Act 3
sketches/stop_police.tex

```

Note that the folder names (`songs`, `sketches`) are merely for illustration. The `.tex` files can be placed at any location that `TEX` can read.

From this plan file, the class can generate a thumb mark for each of these individual files, as demonstrated in figure 1, which shows the location of that file in the larger structure of the piece. It can also generate a thumb index, as demonstrated in figure 2, which outlines the location of every part of the piece, and can serve as an aide to navigation.

The titles in the thumb index demonstrated in figure 2 are taken from the content of the `\title` macro in the files listed in the plan file. Also, the routine for setting the thumb mark for an individual file determine *which* file it is in by comparing the content of the `\title` macro in the files in the plan file with the title that `TEX` knows about in the current file.

Therefore, it is necessary for the thumb index to work that a `\title` is declared in each of the files that are listed in the plan file.

If there are more than one `\title` invocations in a file, the thumb index routine will use the last one. If several of the files in the plan file have the same name, those files will end up with several thumb marks.

Figure 2

Opener	Act 1	1
Example	Act 1	2
Anarchocommunist Commune	Act 1	3
Non-sequitor Dance Act	Act 2	4
Find a Reason to Dress as Ladies	Act 2	5
Stop, Police	Act 3	6

An example thumb index. This example is compressed vertically. It is intended to be nearly the height of the page, to match the thumb markings (such as the ones in figure 1).

Implementation

For this comparing of titles to work, we depend of the text representation not changing between the `\title` declaration and the time when the thumb index routine runs. It might, though, if `fontenc` is called. To quickly knock that problem on the head,¹ we'll just call `fontenc` ourselves.

```
24 \RequirePackage[T1]{fontenc}
```

The code that reads and comprehends the plan file and the listed `.tex` files lives in this group. The ultimate effect ought to be defining the macro `\revy@tabs`, which will contain, for each heading in the plan file, the sequence

```
\undertab <counter at start> \text <hading> \stop,
```

for each file name in the plan file, the sequence

```
\overtab <counter> \text <title in file> \stop,
```

and finally the sequence

```
\undertab <no. of files> \text \done \stop.
```

As we go, these sequences will be built up in a `toks`. The construction where `\act` is defined, and then immediately called, controls how deeply `TEX` will expand our macros[1].

```
25 \begingroup
26 \count256=0
27 \endlinechar=-1
28 \newtoks\tabtoks
29 \newread\tabtex
30 \newread\planread
31 \openin\planread=\planfile
```

¹In a way that hasn't come back to bite *me* yet.

```

32   \def\expandFiHere#1\fi{\fi #1}
33   \def\patheat#1/#2{%
34     \ifx\relax#2\else
35       \expandFiHere#1/\patheat#2%
36     \fi
37   }
38   \edef\planpath{\expandafter\patheat\planfile/\relax}
39   \def\titleeater#1#2\title#3{%
40     \ifx\done#3%
41       #1%
42     \else
43       \expandFiHere
44       \titleeater{#3}%
45     \fi
46   }

```

In this loop, we read the plan file line by line. If the line ends in `.tex`, we take it as a file name, and open that file, looking for `\title` commands, which hopefully do not span across any line breaks.

```

47   \loop\ifeof\planread
48     \closein\planread
49   \else
50     \ifeof\tabtex
51       \let\readline\empty
52       \read\planread to \line
53       \expandafter\ifx\expandafter\par\line\else
54         \IfEndWith{\line}{.tex}{
55           \advance\count256 by 1

```

`\set@curr@file` beats down issues with catcodes in filenames.

```

56       \set@curr@file{\line}
57       \openin\tabtex={\planpath\@curr@file}
58     }{%
59       \edef\act{\noexpand\tabtoks={%
60         \the\tabtoks
61         \noexpand\undertab
62         \number\count256
63         \noexpand\text
64         \line
65         \noexpand\stop
66       }}\act
67     }
68   \fi
69   \else
70     \read\tabtex to \subline
71     \edef\readline{%
72       \expandafter\titleeater
73       \expandafter\readline\subline\title\done
74     }
75     \ifeof\tabtex
76       \edef\act{\noexpand\tabtoks={%
77         \the\tabtoks
78         \noexpand\overtab
79         \number\count256

```

```

80          \noexpand\text
81          \readline
82          \noexpand\stop
83      }}\act
84      \fi
85      \fi
86      \repeat
87      \closein\tabtex
    The final sequence:
88      \edef\act{\noexpand\tabtoks={%
89          \the\tabtoks
90          \noexpand\undertab
91          \number\count256
92          \noexpand\text
93          \noexpand\done
94          \noexpand\stop
95      }}\act

```

`\revy@tabs` And then, the money shot.

```

96      \xdef\revy@tabs{\the\tabtoks}
97  \endgroup

```

The thumbindex hasn't been enabled in the example file in appendix A, since there's nothing to index. However, if you were to place the example plan file above in a file along side `Example.tex` named `aktoversigt.plan`, and replace the class invocation at the top of `Example.tex` with

```
\documentclass[thumbindex]{ucph-revy}
```

then the typeset result should have a thumb marking like the one in figure 1. Nonexistent files are simply ignored when constructing the thumb index.

You might also like to have a page with an index of the thumb markings, something similar to figure 2, but scaled to a full page. Appendix B hold a complete example of one way to make a page like that. If you have access to the package's source files, you can also extract that code as the file `Thumbindex.tex`, by running the file `ucph-revy-ex-thumb.ins` through L^AT_EX.

Implementation

TikZ is not a drawing program^[4], but we'll use it to draw our thumbtabs regardless.

```
98 \RequirePackage{tikz}
```

`\revy@thumbindex` This macro contains the procedures that typeset the thumb index. Its argument is expanded immediately before the thumb tabs are actually typeset, allowing you to modify the macro's internal state at that point. For this purpose, you'll want to pay special attention to the macros `\filter`, `\movetoleftside` and `\writetitlestrue`, which will be defined internally in this macro, and alter which parts of the thumbindex are typeset.

The approach that we will see used, as we go through the macro, is to define `\undertab` and `\overtab` in such a way that they perform the tasks that we want to see performed, when we expand `\revy@tabs`.

```

99   \def\revy@thumbindex#1{%
100   \begingroup

```

`\expandFiHere` You may recognize this one as `\hop` if you've read [2].

```

101 \def\expandFiHere##1\fi{\fi ##1}

```

Find the height of a single tab.

```

102   \newdimen\tabheight
103   \tabheight=\paperheight
104   \advance\tabheight by -1cm%
105   \def\undertab##1\text##2\stop{
106   \ifx\done##2%
107   \divide\tabheight by ##1%
108   \fi
109   }
110   \def\overtab##1\text##2\stop{}%
111   \revy@tabs

```

`\movetoleftside` These definitions contain the differences between right and left facing thumb tabs.

```

112   \def\overtableft{-1cm}
113   \def\overtabright{1cm}
114   \def\undertabright{\overtabright}
115   \def\undertableft{\overtableft - 1cm}
116   \def\underangle{|-}
117   \def\underpos{.25}
118   \def\undersign{}
119   \def\underlabeloffset{\undersign.5cm}
120   \def\textanchorcorner{east}
121   \def\textoffset{\undertableft - .5cm}
122   \def\textside{right}
123   \def\movetoleftside{
124   \def\overtableft{-1.8cm}
125   \def\overtabright{.2cm}
126   \def\undertabright{\overtabright + 1cm}
127   \def\undertableft{\overtableft}
128   \def\underangle{-|}
129   \def\undersign{-}
130   \def\underpos{.75}
131   \def\textanchorcorner{west}
132   \def\textoffset{\undertabright + .5cm}
133   \def\textside{left}
134   }

```

`\ifwritetitles`

```

135 \newif\ifwritetitles\writetitlesfalse

```

Here, we attach meaning to the tokens we stuffed into `\revy@tabtoks` previously.

```

136   \def\printovertab##1##2{
137   \fill ( 0, -##1\tabheight )
138   node[ text=white, anchor=east ]{\bfseries##1}

```

```

139      +( \overtableft, -.5\tabheight + 1mm )
140      rectangle +( \overtabright, .5\tabheight - 1mm )
141      +( \textoffset, 0 )
142      \ifwritetitles
143      node[ anchor=\textanchorcorner ]{\small\bfseries##2}
144      \fi
145      ;
146  }%

147  \def\printundertab##1##2##3{%
148      \path ( 0, -##1\tabheight ) +( \undertabright, -.5\tabheight - 1mm )
149      coordinate (topright);
150      \path ( 0, -##2\tabheight ) +( \undertableft, -.5\tabheight + 1mm )
151      coordinate (bottomleft);
152      \fill[ color=black!30 ] (topright) rectangle (bottomleft);
153      \path (bottomleft)
154      \underangle node[ xshift=\underlabeloffset,
155      pos=\underpos,
156      rotate=\undersign90
157      ]{\Large##3}
158      (topright);
159  }%

160  \def\overtab##1\text##2\stop{%
161      \printovertab{##1}{##2}
162  }

163  \def\undertab##1\text##2\stop{
164      \ifx\done##2\else
165          \let\undertab\undertabspan
166          \expandFiHere
167          \undertab##1\text##2\stop
168      \fi
169  }

170  \def\undertabspan##1\text##2\stop##3\undertab##4\text##5\stop{%
171      \printundertab{##1}{##4}{##2}%
172      ##3%
173      \ifx\done##5\else
174          \expandFiHere
175          \undertab##4\text##5\stop
176      \fi
177  }%

\filtername

178  \def\filtername##1{
179      \def\doWhenMatch{}
180      \def\overtab####1\text####2\stop{
181          \IfStrEq{##1}{####2}{
182              \doWhenMatch
183              \printovertab{####1}{####2}
184          }{}
185      }

186      \def\undertabspan####1\text####2\stop####3\undertab####4\text####5\stop{
187          \def\doWhenMatch{
188              \printundertab{####1}{####4}{####2}

```

```

189         \def\doWhenMatch{}
190     }
191     ####3
192     \ifx\done####5\else
193         \expandFiHere
194         \undertab####4\text####5\stop
195     \fi
196 }
197 }

```

As promised, the macro argument, just before the thumbindex is actually typeset.

```

198     #1
199     \begin{tikzpicture}
200         \revy@tabs
201         \path (0, 0) -- (0, -\paperheight + .5cm);
202     \end{tikzpicture}
203 \endgroup
204 }

```

And thus concludes `\revy@thumbindex`.

In the remainder of this section, we list the macros that may be used to typeset the thumb indices.

`\rectothumbtabfor` `\rectothumbtabfor{<title>}`

Sets a thumb tab, oriented to the right, so suitable for a recto page, that corresponds to the file (or files) whose discovered title matches the argument. The most sensible argument to give it is the title of the current document, which can be found in `\@title`. This is the macro that typeset the thumb tabs in figure 1.

Implementation

```

205 \newcommand{\rectothumbtabfor}[1]{
206     \revy@thumbindex{
207         \filtername{#1}
208     }
209 }

```

`\rectothumbindexwithtitles` `\rectothumbindexwithtitles`

Sets the thumbs index, as was demonstrated in figure 2. Do note that the example in figure 2 has been squeezed vertically. By default, the index is scaled vertically after `\paperheight`, like the thumb tabs in figure 1 were.

Implementation

```

210 \newcommand{\rectothumbindexwithtitles}{
211     \revy@thumbindex{
212         \writetitlestrue
213     }
214 }

```

`\rectothumbindex` `\rectothumbindex`
 Sets all the thumb marks, just like the previous macro, but without writing out the titles.

Implementation

```
215 \newcommand{\rectothumbindex}{
216   \revy@thumbindex{ }
217 }
```

`\rectothumbtabwithtitlefor` `\rectothumbtabwithtitlefor{<title>}`
 Sets only the thumb tab for the entr(y/ies) with the title given in the parameter. It's unclear why you'd want to, but the option is included, for completeness.

Implementation

```
218 \newcommand{\rectothumbtabwithtitlefor}[1]{
219   \revy@thumbindex{
220     \filtername{#1}
221     \writetitlestrue
222   }
223 }
```

`\versothumbtabfor` These macros also come in left oriented versions, suitable for the margin of verso pages.
`\versothumbindexwithtitles`

`\versothumbindex` `\versothumbtabfor{<titel>}`
`\versothumbtabwithtitlefor` `\versothumbindexwithtitle`
 `\versothumbindex`
 `\versothumbtabwithtitlefor{<titel>}`

Implementation

```
224 \newcommand{\versothumbindex}{
225   \revy@thumbindex{
226     \movetoleftside
227   }
228 }
229 \newcommand{\versothumbindexwithtitles}{
230   \revy@thumbindex{
231     \movetoleftside
232     \writetitlestrue
233   }
234 }
235 \newcommand{\versothumbtabfor}[1]{
236   \revy@thumbindex{
237     \movetoleftside
238     \filtername{#1}
239   }
```

```

240 }
241 \newcommand{\versothumbtabwithtitlefor}[1]{
242   \revy@thumbindex{
243     \movetoleftside
244     \filtername{#1}
245     \writetitlestrue
246   }
247 }

```

Implementation

And thus concludes the conditional group for `\if@thumbindex`.

```

248 }

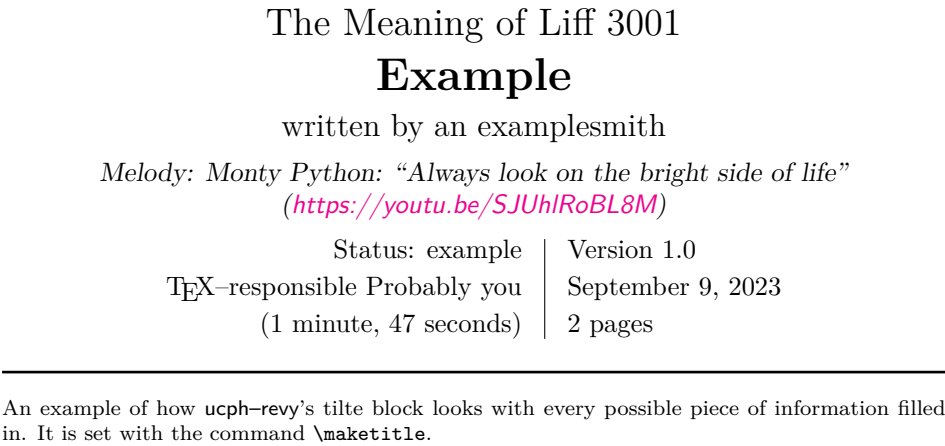
```

2.3 The infoblock

`ucph-revy`'s modified title block contains some additional information that is useful in organizing a show. Figure 3 shows an example of how the title block looks with every possible piece of information included.

Figure 3

`\maketitle`



We give `ucph-revy` this information with a series of preamble commands. For the sake of the example, the example file includes all of these commands, but they are not required for a `ucph-revy` document to compile. If they don't make sense to include in a document, just leave them out. The macros were:

```

\version \version{1.0}

```

Implementation

```

249 \def\version#1{\def\@version{#1}}

```

`\version` is the only one of these commands that may not be omitted, because version control is important².

Implementation

The version number is made a requirement by these commands:

```
250 \def\@version{\@ifundefined{the@version}
251 {\typein[\versionsnr]{Indast revytekstens versionsnummer: }%
252 \global\let\the@version=\versionsnr}{}}\the@version
```

```
\revyname \revyname{The Meaning of Liff}
\revyyear \revyyear{3001}
\title \title{Example}
\author{an examplesmith}
```

Implementation

```
253 \def\revyname#1{\def\@revyname{#1}}
254 \def\revyyear#1{\def\@revyyear{#1}}
\title and \author remain unaltered.
```

If `ucph-revy` isn't given a revue name, year or a title, it will use the defaults of "DIKUrevy", "1973" and "En sketch".

Implementation

```
255 \def\@revyyear{1973}
256 \def\@revyname{DIKUrevy}
257 \def\@title{En sketch}
```

```
\author The rest aren't typeset if they are not defined.
\status \status{example}
\eta \eta{$1$ minute, $47$ seconds}
\responsible \responsible{Probably you}
\melody \melody{Monty Python: ``Always look on the bright side of life''
(\url{https://youtu.be/SJUhlRoBL8M})}
```

Implementation

These new `\if` macros keep track of which of these things have been defined.

```
258 \newif\if@author\@authorfalse
259 \newif\if@status\@statusfalse
260 \newif\if@eta\@etafalse
261 \newif\if@responsible\@responsiblefalse
262 \newif\if@melody\@melodyfalse
```

²You might be of the opinion that there are better ways of doing version control in the far future year 2023. In which case, you may take this as a reminder to actually use one of them.

And these macros define them:

```
263 \def\author#1{\def\@author{#1}\@authortrue}
264 \def\status#1{\def\@status{#1}\@statustrue}
265 \def\eta#1{\def\@eta{#1}\@etatrue}
266 \def\responsible#1{\def\@responsible{#1}\@responsibletrue}
267 \def\melody#1{\def\@melody{#1}\@melodytrue}
```

Ensuring that the internal macros aren't undefined:

```
268 \def\@author{}
269 \def\@status{}
270 \def\@eta{}
271 \def\@responsible{}
272 \def\@melody{}

```

`\auteurs` And as a bonus, we've retained `\auteurs` as an alias of `\author`, for fancy folk.

Implementation

```
273 \let\auteurs\author
```

```
\writtenbyname \writtenbyname{written by}
\melodyname \melodyname{Melody:}
\responsibletext \responsibletext{\TeX--responsible}
\statustext
```

The labels on some of the bits of info are in danish by default: “skrevet af”, “Melodi:” and “TeX-ansvarlig:”, respectively. These commands overwrite those defaults. There is also a command for changing the “Status” label, which happens not to be necessary in english:

```
\statustext {(text)}
```

Implementation

```
274 \def\writtenbyname#1{\def\@writtenbyname{#1}}
275 \def\melodyname#1{\def\@melodyname{#1}}
276 \def\responsibletext#1{\def\@responsibletext{#1}}
277 \def\statustext#1{\def\@statustext{#1}}
278 \def\@writtenbyname{skrevet af}
279 \def\@melodyname{Melodi:}
280 \def\@responsibletext{\TeX--ansvarlig:}
281 \def\@statustext{Status:}
```

`\pagessum` The page count is a touch more complicated, and requires a macro that takes 1 argument (a *number*), and evaluates to the grammatically correct phrase for that page count. Thus `\pagessum{0}` becomes “0 sider” (the plural), while `\pagessum{1}` becomes “1 side” (the singular).

Implementation

The default macro, for danish, is

```
282 \newcommand{\pagessum}[1]{#1 side\ifnum1=#1\else r\fi}
```

In the example, such a macro is given for english in the line

```
\renewcommand{\pagessum}[1]{#1 page\ifnum#1=1\else s\fi}
```

Implementation

`\maketitle` is rewritten to create a title segment like what was shown in figure 3.

```
283 \def\maketitle{\thispagestyle{empty}
284   \vspace*{-\headheight}\vspace*{-\headsep}
285   \centering{\Large\@revname{} \@revyear}\vspace{5pt}\\
286   {\LARGE \bf \@title}\vspace{5pt}\\
287   \if@author{\large\@writtenbyname{} \@author}\vspace{5pt}\\\fi
288   \if@melody{\sl\@melodyname{} \@melody}\vspace{5pt}\\\fi

289   \if@leftfields
290     \begin{varwidth}[t]{\bigheaderwidth}\raggedleft
291       \if@status{\@statustext{} \@status}\vspace{2pt}\\\fi
292       \if@responsible{\@responsibletext{} \@responsible}\vspace{2pt}\\\fi
293       \if@eta{(\@eta)}\\\fi
294     \end{varwidth}%
```

The explicit space here keeps the formatting regular.

```
295     \ \hspace{.5em}\vrule{}\hspace{.5em}
296     \begin{varwidth}[t]{\smallheaderwidth}\raggedright
297       Version \@version\vspace{2pt}\\
298       \today\vspace{2pt}\\
299       \pagessum{\getpagerefnumber{lastpage}}
300     \end{varwidth}

301   \else
302     Version \@version
303     \ \hspace{.5em}\vrule\hspace{.5em}
304     \today
305     \ \hspace{.5em}\vrule\hspace{.5em}
306     \pagessum{\getpagerefnumber{lastpage}}
307   \fi
308   \vskip 5pt }
```

`varwidth` (*env.*) The environment `varwidth` controls the placement of the two-column part of the title block. It comes from

```
309 \RequirePackage{varwidth}
    \getpagerefnumber is an expandable version of \pageref from
310 \RequirePackage{refcount}
```

`\if@leftfields` This macro is defined to effectively be³ `\if@responsible ∨ \if@status ∨ \if@eta`. In the absence of a real \vee -operator for T_EX's `\if`, it uses 1's and 0's as standins for boolean values.

```
311 \def\if@leftfields{
312   \def\@responsiblenum{\ifx\iftrue\if@responsible 1 \else 0 \fi}
313   \def\@statusnum{\ifx\iftrue\if@status 1 \else 0 \fi}
314   \def\@etanum{\ifx\iftrue\if@eta 1 \else 0 \fi}
```

³ \vee being logical OR.


```

315 \ifnum1=\ifnum1=\@responsiblenum 1
316 \else \ifnum1=\@statusnum 1
317 \else \ifnum1=\@etanum 1
318 \else 0
319 \fi\fi\fi
320 }

```

`\smallheaderwidth` And these lengths were used.

```

\bigheaderwidth 321 \newlength{\smallheaderwidth}
322 \setlength{\smallheaderwidth}{22ex}
323 \newlength{\bigheaderwidth}
324 \setlength{\bigheaderwidth}{\textwidth}
325 \addtolength{\bigheaderwidth}{-\smallheaderwidth}
326 \addtolength{\bigheaderwidth}{-1.5em}

```

2.3.1 Page headers

The class also defines its own header style, which is illustrated in figure 4, and contains some of the same information.

Figure 4

The Meaning of Liff 3001		
Version 1.0 September 9, 2023	<i>Example</i>	Side 2/2

An example of the headers that `ucph-revy` define.

These headers belong ot a new `pagestyle` named `revyheadings`

Implementation

```

\ps@revyheadings
327 \gdef\ps@revyheadings{
328   \def\@oddhead{\vbox to 0pt{\vss
329     \hbox to \textwidth{\hfil\rectoheaderthumtab}
330     \hbox{\rm\strut\@revynum{} \@revyyear}
331     \hbox to \textwidth{Version \@version\quad\today
332       \hfil {\large\sl\@title}\hfil
333       Side \rm\thepage/\pageref{lastpage}}
334     \hrule}}
335   \def\@evenhead{\vbox to 0pt{\vss
336     \hbox{\versoheaderthumtab}
337     \hbox to \textwidth{\hfil\rm\strut\@revynum{} \@revyyear}
338     \hbox to \textwidth{Side \rm\thepage/\pageref{lastpage}
339       \hfil {\large\sl\@title}\hfil
340       Version \@version\quad\today}
341     \hrule}}
342   \def\@oddfoot{}\def\@evenfoot{}

```

`\rectoheaderthumbtab` Here, the `\rectoheaderthumbtab` and `\versoheaderthumbtab` are macros that place the thumb markings correctly in relation to the page header, and call the drawing macro, so long as `ucph-revy` has been given the option `thumbindex`, see section 2.2. If not, they do nothing.

```

343 \newcommand{\rectoheaderthumbtab}{%
344   \if@thumbindex
345     \rectothumbtabfor{@title}%
346     \vspace*{ -\paperheight }\vspace*{ -\voffset }\vspace*{ 0.32in }%
347     \hspace{-\paperwidth }\hspace{ 1in }\hspace{ \oddsidemargin }%
348     \hspace{ \hoffset }\hspace{ \textwidth }\hspace{-.6cm}%
349   \fi
350 }
351 \newcommand{\versoheaderthumbtab}{%
352   \if@thumbindex
353     \hspace{ -2.5in }\hspace{ -\evensidemargin }%
354     \hspace{ -\hoffset }\hspace{ -1in }%
355     \versothumbtabfor{@title}%
356     \vspace*{ -\paperheight }\vspace*{ -\voffset }\vspace*{ 0.32in }%
357   \fi
358 }

```

The new `pagestyle` is activated by default.

Implementation

```

359 \pagestyle{revyheadings}

\ps@empty But we also redefine the pagestyle “empty” to be
360 \gdef\ps@empty{
361   \def\@oddhead{\vbox to 0pt{\vss
362     \hbox to \textwidth {\hfil\rectoheaderthumbtab }
363     \vbox to 2em{}}}
364   \def\@evenhead{\vbox to 0pt{\vss\hbox{\versoheaderthumbtab}}}
365   \def\@oddfoot{}\def\@evenfoot{}

```

2.4 Assignment lists

We provide a number of environments for setting information about the material in a file in lists. In the example file, we produce something like the lists shown in figure 5, with the passage

```

\rolename{Roles:}
\begin{roles}
  \role{KA}[Graham] Arthur, king of all the Britons
  \role{P}[Michael] Peasant
  \role{RC}[John] Roman Centurion
  % [...]
\end{roles}
\propname{Props:}
\begin{props}
  \prop{Large sign}
  \prop{Field of crosses}

```

`\end{props}`

Figure 5

Roles:

KA (Graham)	Arthur, king of all the Britons
P (Michael)	Peasant
RC (John)	Roman Centurion

Props:

Large sign
Field of crosses

Examples of the assignment lists that can be typeset with `ucph-revy`, for noting cast lists, among other things.

Implementation

`revy@list` (*env.*) Both those environments are derived from this environment.

```
366 \newenvironment{revy@list}[1]%
367 {{\smallskip\noindent\Large\bf#1}
368  \begin{list}{\{
369    \labelwidth 8em
370    \leftmargin 10em
371    \rightmargin 0em
372    \labelsep 1em
373    \listparindent 0em
374    \topsep 1ex
375    \partopsep 0ex
376    \parsep 1ex
377    \itemsep -1ex\relax
378  }%
379  }%
380  {\end{list}\smallskip\hfil\rule{6cm}{0.1mm}\medskip\par}
```

The argument that this environment takes is the title for the list.

`\revy@itemname` As the `revy@list` environment is derived from the `list` environment, it expects entries in its list to ultimately be called with `\item`. However, for the environments presented to the end user, we like to provide item commands that more closely match the intended use. This macro bundles the process of creating those commands.

```
381 \def\revy@itemname#1{%
382   \def\@itemwithout##1{\item [{\bf ##1} \hfill] }
383   \def\@itemwith##1[##2]{\item [{\bf ##1} (##2) \hfill] }
```

```

384 \expandafter\def\csname #1\endcsname##1{%
385   \ifnextchar [{\@itemwith{##1}}{\@itemwithout{##1}}}%]
386 }
387 }

```

The argument is the name of the macro that will be created for setting items on our lists.

Some examples of how these are used will follow.

roles (*env.*) In the role list, each role is set with the macro

`\role{<abr.>}[<actor>]<description>`

Implementation

The roles list environment is here derived from `revy@list`.

```

388 \newenvironment{roles}{
389   \begin{revy@list}{\@rolename}
390   \revy@itemname{role}
391 }{
392   \end{revy@list}
393 }

```

props (*env.*) Beyond the roles environment, `ucph-revy` defines these three additional environments, for typesetting lists that may be useful to a production. Of these, the **mics** (*env.*) examples in figures 1 and 5 used **props**. **instructors** (*env.*)

Within these environments, **props**, **mics** and **instructors**, the macros for setting a list item are

`\prop{<prop>}[<responsible>]<description>`,

`\mic{<abr.>}[<actor>]<microphon>`

or

`\instructor[<title>]<name>`,

respectively.

Implementation

Their definitions run along the same lines as the **roles** environment.

```

394 \newenvironment{props}{
395   \begin{revy@list}{\@propname}
396   \revy@itemname{prop}
397 }{
398   \end{revy@list}
399 }
400 \newenvironment{mics}{
401   \begin{revy@list}{\@micname}
402   \revy@itemname{mic}
403 }{
404   \end{revy@list}
405 }

```

However, the structure of the item macro in `instructors` doesn't fit the pattern, and so needs its own particular implementation.

```

406 \newenvironment{instructors}{
407   \begin{revy@list}{\@instructorname}
408   \def\@instructorwith[##1]{\item [{\bf ##1} \hfill] }
409   \def\@instructorwithout{\item}
410   \def\instructor{%
411     \@ifnextchar [{\@instructorwith}{\@instructorwithout}]
412   }
413 }{
414   \end{revy@list}
415 }

```

<code>\rolename</code> <code>\propname</code> <code>\micname</code> <code>\instructorname</code>	<p>The default headings for these environments—“Roller:”, “Rekvisitter:”, “Mikrofoner” and “Instruktører:”—may be changed with the macros</p> <pre> \rolename{\text}, \propname{\text}, \micname{\text} og \instructorname{\text} </pre>
---	--

Implementation

```

416 \def\rolename#1{\def\@rolename{#1}}
417 \def\propname#1{\def\@propname{#1}}
418 \def\micname#1{\def\@micname{#1}}
419 \def\instructorname#1{\def\@instructorname{#1}}

```

The macros that store these names are

```

420 \def\@rolename{Roller:}
421 \def\@propname{Rekvisitter:}
422 \def\@micname{Mikrofoner:}
423 \def\@instructorname{Instruktører:}

```

2.5 Body text

2.5.1 sketch

`sketch (env.)` To typeset spoken lines, as in the example shown in figure 6, use the `sketch` environment.

Implementation

`sketch`, like the assignment list environments, is derived from the `list` environment. The environment `revy@bodycommon` introduces macros that are shared between the `sketch` and `song` environments. It is defined on page 26.

```

424 \newenvironment{sketch}{%

```

⋮

RC (*To KA*): Did you write th... wait a minute. (*Off stage*) It's him! (*Back at KA*) How did you get here?

KA *attempts to sidle away.*

RC: Get him!

Several Roman soldiers appear, and drag KA off stage, while KA attempts to protest.

⋮

An example of how spoken lines are typeset, using the `sketch` environment.

```

425 \begin{revy@bodycommon}
426 \rm\begin{list}{}{
427     \labelwidth 2em
428     \leftmargin 3em
429     \rightmargin 0em
430     \labelsep 0.5em
431     \listparindent 2em
432     \topsep 1ex
433     \partopsep 1ex
434     \parsep 0ex
435     \itemsep 1ex\relax}%
436 \item \rule{0.2em}{0em}\vspace{-1em}\par %}

```

`\says` Inside the `sketch` environment, the macro `\says` sets each line of dialouge. Its complete syntax is:

`\says{⟨abr.⟩}[⟨direction⟩] ⟨line⟩`

Implementation

`\says` and `\scene` (see page 26) effectively shift the environment back and forth between two modes of typesetting. We mess around with `\leftskip` to accomplish changes in indentation.

```

437 \newdimen\old@leftskip
438 \old@leftskip \leftskip
439 \newdimen\short@leftskip
440 \short@leftskip \leftskip
441 \advance\short@leftskip -1.5em

```

`\says`

```

442 \def\says##1{\ifnextchar [{\@saysas{##1}}{\@says{##1}}} %]
443 \def\@saysas##1[##2]{\rm\item [{\bf ##1 }{\it (##2)\,}\hfill:]}
444 \leftskip \old@leftskip
445 \def\@says##1{\rm\item [{\bf ##1\,}\hfill:]\leftskip \old@leftskip}

```

```
\scene
446 \def\scenef{\medskip\par\noindent\it\leftskip \short@leftskip }
```

Thus, for example, part of the first line of figure 6 gets created with the following code:

```
\says{RC}[To KA] Did you write th... wait a minute. \act{Off stage} (...)
```

Implementation

And here, the `sketch` environment is concluded.

```
447 }{%
448 \end{list}
449 \end{revy@bodycommon}
450 }
```

2.5.2 song

`song` (*env.*) The `song` environment is for song lyrics, which are set as demonstrated in figure 7.

Figure 7

```

:
:
CC:    ...be silly chumps,
      Just purse you're lips and whistle,
      That's the thing.

      And, always look on the bright side of life,
:
:
```

An example of how song lyrics are typeset, using the `song` environment.

Implementation

The `song` environment is also implemented as a derivative of `list` and uses `revy@bodycommon`.

```
451 \newenvironment{song}{%
452 \begin{revy@bodycommon}
453 \rm\begin{list}{}{}{%
454 \raggedright
455 \labelwidth 4.5em
456 \leftmargin 7em
457 \rightmargin 0em
458 \labelsep 2em
```

```

459     \listparindent -2em
460     \topsep 0ex
461     \partopsep \bigskipamount
462     \parsep .6\baselineskip
463     \itemsep -.6\baselineskip
464     \relax
465   }
466   \item[] \rule{0.2em}{0em}\vspace{-\baselineskip}\strut\par %}

```

`\sings` Within the `song` environment, the macro `\sings` sets song lyrics. Its full syntax is

`\sings{⟨abr.⟩}[⟨direction⟩] ⟨lyric⟩`

Line breaks are significant in song lyrics, and so the `song` changes how L^AT_EX reacts to carriage returns in the source file. For example, the lyric shown in figure 7 is the product, in part, of this source code:

```

\sings{CC} ...be silly chumps,
          Just purse you're lips and whistle,
          That's the thing.

```

Implementation

`\sings` and `\scene` shift back and forth between to even more different states within the `song` environment than was the case in `sketch`. Much of the magic relates to the macros `\obeycr` and `\restorecr`.

```

467   \newdimen\old@leftskip
468   \old@leftskip \leftskip
469   \newdimen\short@leftskip
470   \short@leftskip \leftskip
471   \advance\short@leftskip -5.5em

```

`\sings`

```

472 \def\sings##1{\obeycr
473   \@ifnextchar [{\@singsas{##1}}{\@sings{##1}} %}
474 \def\@singsas##1[##2]{\rm\item [\hskip\@leftsingpad{\bf ##1 }{\it
475   (##2)\,}:\hskip\@rightsingpad]\hskip-2em\leftskip \old@leftskip}
476 \def\@sings##1{\rm\item [\hskip\@leftsingpad{\bf ##1\,}:\hskip
477   \@rightsingpad]\hskip-2em\leftskip \old@leftskip}

```

`\scene`

```

478 \let\old@sings=\sings
479 \def\scene{%
480   \def\sings{\let\sings=\old@sings \medskip\sings}
481   \medskip\par\leftskip \short@leftskip \restorecr
482   \it\noindent\relax
483 }

```

`\@leftsingpad` and `\@rightsingpad` allow us to set lyrics to be left or right justified.

```

484   \newskip\@leftsingpad

```



```

485 \newskip\@rightssingpad
486 \def\flushsingsright{\@leftssingpad Opt plus 1fill\@rightssingpad Opt\relax}
487 \def\flushsingsleft{\@leftssingpad Opt\@rightssingpad Opt plus 1fill\relax}
We set left justification as the default.
488 \flushsingsleft

```

Implementation

We call `\obeycr` here, along with some macros that get L^AT_EX to be less concerned about the proper length of lines.

```

489 \tolerance10000\hfuzz21cm\obeycr
490 }{
491 \end{list}
492 \end{revy@bodycommon}
493 }

```

Implementation

`\obeycr` The macros that make the line breaking magic possible, are

```

\restorecr 494 \newif\if@newpar
495 {
496 \catcode`\^^M=13
At this point, we must be careful with carriage returns in the code itself...
497 \gdef\@carriagereturn{%
498 \ifnextchar^^M{%
499 \@newpartrue%
500 }{%
501 \ifnextchar\end{}}{%
502 \ifnextchar\sings{%
503 \if@newpar\bigskip\par\fi\@newparfalse%
504 }{%
505 \if@newpar\par\else\\\fi\@newparfalse%
506 }%
507 }%
508 }%
509 }%
510 \gdef\obeycr{%
511 \catcode`\^^M=13 %
512 \let^^M=\@carriagereturn%
513 \@gobblecr%
514 }%
515 \gdef\restorecr{\catcode`\^^M=5 }%
516 }

```

2.5.3 Both sketch and song

The `sketch` and `song` macros both provide the following macros:

Implementation

`revy@bodycommon` (*env.*) They are packaged in the environment `revy@bodycommon`.

```
517 \newenvironment{revy@bodycommon}{
```

`\role` A macro for formatting role abbreviations within the text.

Implementation

This macro has the same name as the `item` command inside the `roles` environment. But it's OK, because that command is defined inside `roles`'s internal group.

```
518 \let\role = \textbf
```

`\does` A macro for giving instructions for a particular role that aren't a spoken line, such as

```
\does{KA} attempts to sidle away.
```

Implementation

The `1em` exdent is a hardcoded length that matches how `song` and `sketch` are defined right now. I'm sure it won't jump up and bite anyone in the future...

```
519 \def\does##1{\scene \hspace{-1em}\role{##1}}
```

`\scene` `\scene` is used for stage directions, such as

```
\scene P \& KA are startled.
```

Implementation

`\scene` was implemented inside the `sketch` and `song` environments.

`\act` `\act` is for directions, such as in

```
(...) It's him! \act{Back at KA} How did you get here?
```

Implementation

```
520 \def\act##1{\textit{##1}}
```

Both were used in the example in figure 6.

And here, we end the the environment `revy@bodycommon`.

521 }{}

2.6 The .aux file

For futue use, we will also ensure that our metadata is written to the `.aux` file at the end of the document.

This is also where we `label` the last page.

```

522 \let\revyinfo=\relax
523 \newtoks\@titletoks
524 \newtoks\@authortoks
525 \newtoks\@melodytoks
526 \newtoks\@revyrevynametoks
527 \newtoks\@revyrevyyeartoks
528 \let\end@document=\enddocument
529 \def\enddocument{\label{lastpage}\write\@auxout
530   {\string\@ifundefined{revyinfo}{\string\def\string
531     \revyinfo\string##1\string##2\string##3\string
532     ##4\string##5\string##6\string##7\string##8{}}{}}\@titletoks=
533   \expandafter{\@title}\@titletoks=
534   \expandafter{\@title}\@authortoks=
535   \expandafter{\@author}\@melodytoks=
536   \expandafter{\@melody}\@revyrevynametoks=
537   \expandafter{\@revyname}\@revyrevyyeartoks=
538   \expandafter{\@revyyear}\write\@auxout
539   {\string\revyinfo{\the\@titletoks}{\@version}
540     {\today}%
541     {\the\@authortoks}%
542     {\the\@melodytoks}%
543     {}%
544     {\the\@revyrevynametoks}%
545     {\the\@revyrevyyeartoks}%
546   }%
547   \end@document
548 }
```

References

- [1] Victor Eijkhout. *TeX by Topic*, chapter 14.5.1. <https://github.com/VictorEijkhout/tex-by-topic/raw/main/TeXbyTopic.pdf>, 2019.
- [2] Victor Eijkhout. *TeX by Topic*, chapter 13.8.6. <https://github.com/VictorEijkhout/tex-by-topic/raw/main/TeXbyTopic.pdf>, 2019.
- [3] Overleaf. Understanding packages and class files. https://www.overleaf.com/learn/latex/Understanding_packages_and_class_files.

[4] Till Tantau. *The TikZ and PGF Packages*. <https://mirrors.dotsrc.org/ctan/graphics/pgf/base/doc/pgfmanual.pdf>, 2023.

Implementation

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols		
\@author 263, 268, 287, 535	\@melodyname 275, 279, 288	\@saysas 442, 443
\@authorfalse 258	\@melodytoks 525, 535, 542	\@sings 473, 476
\@authortoks 524, 534, 541	\@melodytrue 267	\@singsas 473, 474
\@authortrue 263	\@micname 401, 418, 422	\@status 264, 269, 291
\@auxout 529, 538	\@newparfalse 503, 505	\@statusfalse 259
\@carriagereturn 497, 512	\@newpartrue 499	\@statusnum 313, 316
\@curr@file 57	\@oddfoot 342, 365	\@statustext 277, 281, 291
\@eta 265, 270, 293	\@oddhead 328, 361	\@statustrue 264
\@etafalse 260	\@propname 395, 417, 421	\@thumbindexfalse 2
\@etanum 314, 317	\@responsible 266, 271, 292	\@thumbindextrue 3
\@etatrue 265	\@responsiblefalse 261	\@title 257, 286, 332, 339, 345, 355, 533, 534
\@evenfoot 342, 365	\@responsiblenum 312, 315	\@titletoks 523, 532, 533, 539
\@evenhead 335, 364	\@responsibletext 276, 280, 292	\@version 249, 250, 297, 302, 331, 340, 539
\@firstofone 20	\@responsibletrue 266	\@writtenbyname 274, 278, 287
\@gobble 22	\@revyname 253, 256, 285, 330, 337, 537	A
\@gobblecr 513	\@revyrevynametoks 526, 536, 544	\act 59, 66, 76, 83, 88, 95, <u>520</u>
\@ifnextchar 385, 411, 442, 473, 498, 501, 502	\@revyrevyyear 527, 537, 545	\addtolength 18, 325, 326
\@ifundefined 250, 530	\@revyyear 254, 255, 285, 330, 337, 538	\article <u>3</u>
\@instructorname 407, 419, 423	\@rightsingpad 475, 477, 485, 486, 487	\auteurs <u>273</u>
\@instructorwith 408, 411	\@rolename 389, 416, 420	\author <u>258</u> , <u>273</u>
\@instructorwithout 409, 411	\@says 442, 445	B
\@itemwith 383, 385		\baselineskip 462, 463, 466
\@itemwithout 382, 385		\bf 286, 367, 382, 383, 408, 443, 445, 474, 476
\@leftsingpad 474, 476, 484, 486, 487		\bfseries 138, 143
\@melody 267, 272, 288, 536		\bigheaderwidth 290, <u>321</u>
\@melodyfalse 262		\bigskip 503

<code>\bigskipamount</code> ...	461	<code>\flushsingsright</code> .	486	<code>\instructor</code>	410
C		G		<code>\instructorname</code> ..	416
<code>\centering</code>	285	<code>\gdef</code>	327, 360, 497, 510, 515	<code>instructors (env.)</code> .	394
<code>\count</code>	26, 55, 62, 79, 91	<code>\getpagerefnumber</code> .	299, 306	<code>\it</code> ..	443, 446, 474, 482
<code>\curname</code>	384	<code>\global</code>	252	<code>\item</code>	382, 383, 408, 409, 436, 443, 445, 466, 474, 476
<code>\CurrentOption</code>	6, 8, 10	H		<code>\itemsep</code> .	377, 435, 463
D		<code>\hbox</code>	329, 330, 331, 336, 337, 338, 362, 364	L	
<code>\DeclareOption</code> ...	3, 5	<code>\headheight</code>	284	<code>\label</code>	529
<code>\divide</code>	107	<code>\headsep</code>	284	<code>\labelsep</code>	372, 430, 458
<code>\does</code>	519	<code>\hfil</code> ..	329, 332, 337, 339, 362, 380	<code>\labelwidth</code>	369, 427, 455
<code>\done</code> 40, 73, 93, 106, 164, 173, 192		<code>\hfill</code>	382, 383, 408, 443, 445	<code>\LARGE</code>	286
<code>\doWhenMatch</code> ..	179, 182, 187, 189	<code>\hfuzz</code>	489	<code>\Large</code> ...	157, 285, 367
E		<code>\hoffset</code>	348, 354	<code>\large</code> ...	287, 332, 339
<code>\end@document</code>	528, 547	<code>\hrule</code>	334, 341	<code>\leftmargin</code>	370, 428, 456
<code>\endcurname</code>	384	<code>\hskip</code>	474, 475, 476, 477	<code>\leftskip</code>	438, 440, 444, 445, 446, 468, 470, 475, 477, 481
<code>\enddocument</code> .	528, 529	<code>\hspace</code> ...	295, 303, 305, 347, 348, 353, 354, 519	<code>\let</code> 51, 165, 252, 273, 478, 480, 512, 518, 522, 528	
<code>\endlinechar</code>	27	I		<code>\line</code> .	52, 53, 54, 56, 64
environments:		<code>\if@author</code> ...	258, 287	<code>\listparindent</code>	373, 431, 459
<code>instructors</code> ...	394	<code>\if@eta</code>	260, 293	<code>\LoadClass</code>	14
<code>mics</code>	394	<code>\if@leftfields</code>	289, 311	<code>\loop</code>	47
<code>props</code>	394	<code>\if@melody</code> ...	262, 288	M	
<code>revy@bodycommon</code>	26, 517	<code>\if@newpar</code>	494, 503, 505	<code>\maketitle</code> ...	249, 283
<code>revy@list</code>	366	<code>\if@responsible</code> ...	261, 292, 312	<code>\medskip</code>	380, 446, 480, 481
<code>roles</code>	388	<code>\if@status</code>	259, 291, 313, 314	<code>\melody</code>	258
<code>sketch</code>	424	<code>\if@thumbindex</code> ..	2, 5, 19, 344, 352	<code>\melodyname</code>	274
<code>song</code>	451	<code>\IfBeginWith</code>	6	<code>\micname</code>	416
<code>varwidth</code>	16	<code>\IfEndWith</code>	54	<code>mics (env.)</code>	394
<code>\eta</code>	258	<code>\ifeof</code>	47, 50, 75	<code>\movetoleftside</code> ...	112, 226, 231, 237, 243
<code>\evensidemargin</code> ..	353	<code>\ifnum</code>	282, 315, 316, 317	N	
<code>\expandafter</code> ...	8, 20, 22, 38, 53, 72, 73, 384, 533, 534, 535, 536, 537, 538	<code>\IfStrEq</code>	181	<code>\newdimen</code> .	102, 437, 439, 467, 469
<code>\expandFiHere</code> ...	32, 35, 43, 101, 166, 174, 193	<code>\iftrue</code> ..	312, 313, 314	<code>\newif</code>	2, 135, 258, 259, 260, 261, 262, 494
F		<code>\ifwritetitles</code>	135, 142	<code>\newlength</code> ...	321, 323
<code>\fill</code>	137, 152	<code>\ifx</code> 34, 40, 53, 106, 164, 173, 192, 312, 313, 314			
<code>\filtername</code> 178, 207, 220, 238, 244					
<code>\flushsingsleft</code> ...	487, 488				

\newread 29, 30	\printovertab	\rm	330, 333, 337, 338,
\newskip 484, 485	 136, 161, 183		426, 443, 445,
\newtoks	28, 523, 524, 525, 526, 527	\printundertab		453, 474, 476
\noexpand 59, 61, 63, 65, 76, 78, 80, 82, 88, 90, 92, 93, 94	\ProcessOptions	... 13	\role 518, 519
\noindent 367, 446, 482	\propname 416	\rolename 416
\number 62, 79, 91	props (env.) 394	roles (env.) 388
		\ps@empty 18, 360	\rule 380, 436, 466
		\ps@revyheadings	. 327		
O				S	
\obeycr	.. 472, 489, 494	Q		\says 437, 442
\oddsidemargin	... 347	\quad 331, 340	\scene 446, 478, 519, 520
\old@leftskip 437, 438, 444, 445, 467, 468, 475, 477	R		\setcurr@file 56
\old@sings	... 478, 480	\raggedleft 290	\setlength 15, 16, 17, 322, 324
\openin 31, 57	\raggedright	. 296, 454	\short@leftskip 439, 440,
\overtab 78, 110, 160, 180	\read 52, 70		441, 446, 469, 470, 471, 481
\overtableft	112, 115, 124, 127, 139	\readline 51, 71, 73, 81	\sings	... 467, 472, 502
\overtabright 113, 114, 125, 126, 140	\rectoheaderthumbtab 329, 343, 362	sketch (env.) 424
P		\rectothumbindex	. 215	\sl 288, 332, 339
\pageref 333, 338	\rectothumbindexwithtitles 210	\small 143
\pagesum 282, 299, 306	\rectothumbtabfor 205, 345	\smallheaderwidth 296, 321
\pagestyle 359	\rectothumbtabwithtitleforsong 218	\smallskip	... 367, 380
\paperheight	17, 103, 201, 346, 356	\relax	13, 34, 38, 377, 435, 464, 482, 486, 487, 522	\status 258
\paperwidth 347	\repeat 86	\statustext 274
\par 53, 380, 436, 446, 466, 481, 503, 505	\RequirePackage	. 1, 24, 98, 309, 310	\stop	65, 82, 94, 105, 110, 160, 163, 167, 170, 175, 180, 186, 194
\parsep	.. 376, 434, 462	\responsible 258	\string 530, 531, 532, 539
\partopsep 375, 433, 461	\responsibletext	. 274	\strut	... 330, 337, 466
\PassOptionsToClass 10	\restorecr	... 481, 494	\subline 70, 73
\path 148, 150, 153, 201	revy@bodycommon	(env.) ... 26, 517	T	
\patheat 33, 35, 38	\revy@itemname	. 381, 390, 396, 402	\tabheight 102, 103, 104, 107, 137, 139, 140, 148, 150
\planfile 4, 7, 8, 31, 38	revy@list (env.)	.. 366	\tabtex 29, 50, 57, 70, 75, 87
\planpath 38, 57	\revy@tabs 6, 96, 111, 200	\tabtoks	28, 59, 60, 76, 77, 88, 89, 96
\planread 30, 31, 47, 48, 52	\revy@thumbindex 99, 206, 211, 216, 219, 225, 230, 236, 242	\text	63, 80, 92, 105, 110, 160, 163, 167, 170, 175, 180, 186, 194


```

%%
\documentclass{ucph-revy}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{hyperref}
\urlstyle{sf}

\version{1.0}
\revyname{The Meaning of Liff}
\revyyear{3001}
\title{Example}
\author{an examplesmith}
\status{example}
\eta{$1$ minute, $47$ seconds}
\responsible{Probably you}
\melody{Monty Python: ``Always look on the bright side of life''
  (\url{https://youtu.be/SJUh1RoBL8M})}

\writtenbyname{written by}
\melodyname{Melody:}
\responsibletext{\TeX--responsible}

\renewcommand{\pagessum}[1]{#1 page\ifnum#1=1\else s\fi}

\begin{document}
\maketitle

\rolename{Roles:}
\begin{roles}
  \role{KA}{Graham} Arthur, king of all the Britons
  \role{P}{Michael} Peasant
  \role{RC}{John} Roman Centurion
  \role{R1...?}{Several Roman Soldiers}
  \role{CC}{Eric} Crucified Criminal
  \role{B}{Graham} Brian
\end{roles}
\propname{Props:}
\begin{props}
  \prop{Large sign}
  \prop{Field of crosses}
\end{props}

\begin{sketch}
  \scene Lights up on a muddy field with a castle in the background.

  \says{KA}{As he enters} Hello!

  \says{KA} Now, by rights, you should all be kneeling, given that I
  am Arthur, king of all the Britons. However, it has been brought to

```


my attention that the audience seating is not laid out to allow for that. We had some considerable trouble during dress rehearsals. So I will excuse you from that requirement.

Now, has anyone seen a particularly fancy cup? Perhaps out in the bar? You see, I am on a quest given to me by God, to find the holy... \act{Interrupted, as he steps in a hole.} Oh dear...

\says{P}[Shrieking, as he rears up from the ground] Oi! What do you think you're doing!

\says{P} You broke my sign!

\says{KA} Sign? What sign?

\says{P} Well, look.

\does{P} raises up a sign that reads

\begin{center}\sc Romanes Eunt Dom[]\end{center}

The last bit of the sign has a foot--shaped hole in it.

\says{P} You can't read it anymore!

\says{KA} You certainly can! It says ``Romans go home.''

\says{RC}[Appears from offstage] No it doesn't.

\does{P \& KA} are startled.

\says{RC}[To KA] Did you write th... wait a minute. \act{Off stage} It's him! \act{Back at KA} How did you get here?

\does{KA} attempts to sidle away.

\says{RC} Get him!

\scene Several Roman soldiers appear, and drag \role{KA} off stage, while \role{KA} attempts to protest.

\says{KA}[As he is dragged off] I am not him! I am Arthur, King of the Britons!

\scene The scene changes to a field of crosses, with several crucified criminals and B.

\does{CC} is in the middle of singing to \role{B}.

\end{sketch}

\begin{song}%

```

\sings{CC} ...be silly chumps,
          Just purse you're lips and whistle,
          That's the thing.

          And, always look on the bright side of life,
          Always look on the right side of life,
          For life is quite absurd,
          An. death's the final word...

\end{song}
\begin{sketch}
\says{RC}[Shouting as he walks on stage] Right, you lanky reprobate!
Don't think that the mighty Roman bureaucracy doesn't keep track of
its inmates

\scene KA is dragged on stage by two Roman soldiers behind RA.

\says{RA} Look, we have your designated cross ready for you
right... \act{points to B'a cross, then falters when he notices
B.}

\scene Everyone looks back and forth between B and KA.

\scene End scene
\end{sketch}
\end{document}
}
\endinput
%%
%% End of file `Example.tex'.

```

B Example of a thumb index page

```

\documentclass[a4paper,11pt,thumbindex]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[danish]{babel}
\usepackage{calc}
\usepackage{tikz}

\version{1.0}
\title{Registerindeks}

\begin{document}
\thispagestyle{empty}
\newlength{\torightside}
\setlength{\torightside}{\paperwidth - 1in - \hoffset %
- \oddsidemargin + .6cm}
\newlength{\totop}
\setlength{\totop}{-\headsep - \headheight - \topmargin %

```

```

- \voffset - 1in - .5cm}

\ vbox to Opt{\vskip \totop \hbox to \torightside{\hss%
\rectothumbindexwithtitles%
}}
\ vbox to Opt{\tikz \path (0,0) -- (0, -.5\textheight)
node[rotate=90]{\parbox{\textheight}{\maketitle}}; }
\end{document}

```