

Klassen `ucph-revy`*

Kristoffer Levin Hansen
kristoffer@levinhansen.dk

4. april 2025

Resumé

En klasse, som kommer med kommandoerne til at typesætte et manuskript i den stil, som bruges blandt SaTyR-revyerne ved Science på Københavns Universitet. Tidligere kendt som `revy.sty`.

Indhold

1	Introduktion	1	2.5.1	sketch	25
			2.5.2	song	27
2	Brugsvejledning	3	2.5.3	Fælles for sketch og song	29
2.1	Invokering og valgmuligheder	3	2.6	Assorterede emner	31
2.2	Registermærkninger	5	2.6.1	SATyR	31
2.3	Infoblokken	16	2.6.2	.aux-filen	31
2.3.1	Sidehoveder	20	A	Komplet eksempel	36
2.4	Tildelingslister	22	B	Eksempel på registerindeks	38
2.5	Brødtekst	25			

1 Introduktion

I umindelige tider har studenterrevyerne på KU Science brugt den samme stil til at typesætte deres manuskripter. Den har indtil nu boet i filen `revy.sty`, som er blevet det rundt mellem revyster og revytter i andægtig stilhed. Den her klasse bibeholder funtionaliteten fra `revy.sty`, men implementerer den nu i dens korrekte plads i L^AT_EX-hierakiet, som en klasse^[4], og med et mere beskrivende navn, så den kan udgives i en pakkesamling.

Klassen kan sætte både talt dialog og sangtekster, med rolleangivelse, og kommer med en bestemt stil til titler, sidehoveder og ting som rolle- og rekvisitlister.

Et eksempel på, hvordan manuskripter skrevet i vores stil ser ud, er gengivet i figur 1. Den er tiltænkt til brug i en revy, hvor hvert nummer lever sit eget liv indtil ret sent i processen. Derfor er hvert nummer sit eget dokument, med egen titel og andre informationer, og samlingen til et færdigt T_EX-hæfte er ikke inden for klassens emnefelt.

*This document corresponds to `ucph-revy` v1.2.0, dated 2025/03/26.

<p>FysikRevy™ 2019</p> <p>Eksempel</p> <p>skrevet af en eksempelsmed</p> <p>Melodi: Queen: „Bohemian Rhapsody“ (https://youtu.be/fJ9rUzIMcZQ)</p> <table border="1"> <tr> <td>Status: eksempel</td> <td>Version 1.0</td> </tr> <tr> <td>TeX-ansvarlig: Dig</td> <td>4. april 2025</td> </tr> <tr> <td>(1 minut, 47 sekunder)</td> <td>2 sider</td> </tr> </table> <p>Roller:</p> <p>PH (Jophiel) Per Hedegaard I Vred instruktør A9 (A–ni) Anine</p> <hr/> <p>Instruktører:</p> <p>Instruktør Raben</p> <hr/> <p>Rekvisitter:</p> <p>Hedegaard-hår (Person, der skaffer) Didgeridoo (Anine)</p> <hr/> <p><i>Lys op</i></p> <p>PH: Hej venner. I den her sketch, der spiller jeg Per Hedegaard. Det kan man se ud fra farven på mit hår. Men bare rolig, jeg er ikke Per Hedegaard i virkeligheden, der er ikke nogen regneøvelser, I har glemt at lave.</p> <p>Se, hvis jeg tager håret af (<i>fjerner parykken</i>), så er det helt tydeligt, at jeg ikke er den rigtige Hedegaard.</p> <p>I (<i>Stormer ind på scenen</i>): NEEEEEEJ, for helvede. Du må ikke pille ved den fjerde væg. Det er simpelthen så uprofessionelt, det der!</p> <p>PH: Se, venner, det var meningen, at der skulle have været en anden replik der, men fordi der ikke er sat nogen på den rolle i rollelisten, så kunne vi ikke høre nogen som helst kritik af min performance.</p> <p><i>I reagerer med åbenmundet vantrø.</i></p> <p>PH: Det må også være svært at følge skuepil-instruktioner, når man ikke eksisterer, kan jeg forestille mig.</p> <p>A9: Hej Jophiel.</p> <p><i>PH tager sin paryk på igen.</i></p> <p>A9: Hej, Per Hedegaard.</p> <p>A9: Var det ikke meningen, at det her skulle være en sang?</p> <p>PH: Jo, for dølen. Har du en didgeridoo med?</p> <p>A9: Didgeridoo?</p> <p>PH: Ja. Der står i rekvisitlisten, at du skaffer en.</p> <p>A9: Jamen så må det jo passe.</p>	Status: eksempel	Version 1.0	TeX-ansvarlig: Dig	4. april 2025	(1 minut, 47 sekunder)	2 sider	<p>FysikRevy™ 2019</p> <p>Version 1.0 4. april 2025 <i>Eksempel</i> Side 2/2</p> <p>PH: Ja, ellers skriver vi "Person, der skaffer," for at minde os om at skrive et navn ind.</p> <p>A9: Nå, men så må jeg jo hellere begynde at lede. (<i>Vender sig mod Bandet</i>) Bandet, har I...?</p> <p><i>Hvert bandmedlem holder sin egen didgeridoo op.</i></p> <p>A9 (<i>Til PH</i>): Den ser ud til at være i vinkel.</p> <p>PH: Sejt.</p> <p>PH (<i>Råber til Bandet</i>): Og tre! Fire!</p> <p><i>Bandet spiller "Bohemian Rhapsody" af Queen, udelukkende på didgeridoo.</i></p> <p>A9: Åh, Per Hede Er den lede Og han dumped' mig i kvant!</p> <p>PH: Per Hedegaard har er reksamenssæt til dig A9: Til mig? PH: Til dig!</p> <p><i>Episk didgeridoo-solo, mens A9 løser reksamenssæt.</i></p> <p>...</p> <p>A9 <i>rækker hånden i vejret, og bandet stopper.</i></p> <p>PH (<i>Er et stykke tid om at lægge mærke til A9, men kommer til sidst over</i>): Ja, spørgsmål?</p> <p>A9: Det her eksamenssæt er jo trivielt...?</p> <p><i>Lys ned</i></p>
Status: eksempel	Version 1.0						
TeX-ansvarlig: Dig	4. april 2025						
(1 minut, 47 sekunder)	2 sider						

Et eksempel på, hvad denne klasse kan bruges til at lave.

Vi har også lavet en programpakke, som står for at lave et samlet \TeX -hæfte ud af en samling af `.tex`-filer (som bruger den her dokumentklasse). Den er sidst set i [3].

Ifølge `revy.sty` var de originale forfattere Uffe Friis Lichtenberg, Arne John Glenstrup og Anders Komár Ravn.

Change History

v1.0.0	Tilpasninger i eksempelfil	36
General: Initial conversion to a	<code>\SaTyR</code> : <code>SaTyR</code> -makro	31
class	v1.1.0	
<code>\does</code> : Ny makro	<code>costumes</code> : Introducer	
<code>thumbindex</code> : Nyt maskineri til	Kostume-miljøet	24
registermærkninger	<code>\movetoleftside</code> : Gør tabsne	
<code>\maketitle</code> : Nyt layout af	bredere, så printere ikke skærer	
titelblokken	dem af	12
<code>sketch</code> : Brødtekst-makroer	v1.2.0	
(<code>\scene</code> , etc.) findes ikke	<code>thumbindex</code> : Håndter stub punkter	
længere uden for <code>song</code> og	planfilen.	6
<code>sketch</code> miljøerne.	Håndtér kommentarer i planfil	5
<code>song</code> : Brødtekst-makroer (<code>\scene</code> ,	Kalder ikke længere <code>fontenc</code>	5
etc.) findes ikke længere uden	<code>\planfileAllowMacros</code> : Gør	
for <code>song</code> og <code>sketch</code> miljøerne.	tilpasning af stub-kommandoer	
v1.0.1	nemmere.	6
<code>\movetoleftside</code> : Juster	<code>\revy@tabs</code> : Flyt indlæsning af	
mærkebredden	planfil til første gang en	
<code>\revy@itemname</code> : Fjernede	registermærkning bliver sat	7
unødvendige omdefinitioner.	<code>\title</code> : Valgfri kort titel	17
<code>\revy@thumbindex</code> : Den må ikke	<code>\titleeater</code> : Håndtér valgfri korte	
dividere med 0	titler i manuskriptfiler	9
v1.0.2		
General: Skift <code>fontenc</code> til <code>OT1</code>		4

2 Brugsvejledning

Et eksempel på en `.tex`-fil, som kan bruges til at producere noget nær det i figur 1, er gengivet i sin helhed i appendiks A, eller filen `Eksempel.tex`, som \LaTeX burde have produceret sammen med den fil, som du læser nu. Her gennemgår vi nogle af højdepunkterne, for at forstå hvordan klassen kan bruges.

2.1 Invokering og valgmuligheder

Klassen (ned)kaldes, ved at starte en `.tex`-fil med

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

`article` `ucph-revy` nedarver fra `article`-klassen, så alle valgmuligheder, som godtages af `article` godtages også af `ucph-revy`. Som standard vælger `ucph-revy a4paper` og `11pt`, fra `article`'s valgmuligheder, hvis `ucph-revy` ikke bliver kaldt med nogen valgmuligheder, som er i modstrid med dem.

`thumbindex` Derudover tilføjer `ucph-revy` valgmulighederne `thumbindex` og `planfile`, som `planfile=` tilføjer registermærkninger og lader forfatteren ændre hvilken fil registermærkningerne bliver bygget ud fra. Hvad det mere specifikt betyder bliver gennemgået i sektion 2.2.

Implementering

Vi håndterer valgmuligheder til klassen her, i starten af filen. Vi skal bruge den her pakke.

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

`\if@thumbindex` Kommer til at holde styr på, om `thumbindex` er givet. Vi kommer til at gøre noget med den nede i sektion 2.2.

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

Og nu kan vi definere valgmulighederne:

`thumbindex`

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

`planfile=` Tager en tekststreng, så den skal behandles helt specielt. `xstring` definerer bl.a. makroen `\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}
```

Implementering

De her kommandoer gør tekstblokken højere.

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

I eksemplet inkluderer vi fire pakker. De er ikke strengt nødvendige for at få `ucph-revy` til at fungere, men de kommer næsten med sikkerhed til at gøre dit liv lettere.

<code>\usepackage [utf8]{inputenc}</code>	
<code>\usepackage [OT1]{fontenc}</code>	Gør L ^A T _E X's håndtering af danske bogstaver mere moderne.
<code>\usepackage [danish]{babel}</code>	Fortæller L ^A T _E X en masse ting om dansk, bl.a. hvordan man bryder ord.
<code>\usepackage {hyperref}</code>	Giver mulighed for at lave links, som det til Youtube i figur 1.
<code>\urlstyle {sf}</code>	Gør, at links bliver sat med sans serif tekst i stedet for monospace.

2.2 Registermærkninger

`thumbindex` Kasserne i kanten af siderne i figur 1 er registermærkninger. De er tænkt som en hjælp med at finde rundt i manuskripter, når de bliver sat sammen af flere enkelte numre, som hver især bruger `ucph-revy`. De giver kun mening at bruge i dokumenter som er en del af en større samling, så for at sætte dem skal klassen gives valgmuligheden `thumbindex`.

Implementering

`\if@thumbindex` `\if@thumbindex` blev defineret på side 4, og fortæller bare, om `thumbindex` er blevet valgt. Hvis den ikke er, så springer vi al koden i det her afsnit over.

Vi bruger `\@firstofone` og `\@gobble` for at undgå at pakke hele kodeblokken ind i en `\if`, og så få problemer med `\if`'er inden i hinanden¹.

```
19 \if@thumbindex
```

`\revy@readwithhash` Som vi kommer til at se senere, så kan planfilen risikere at indeholde `#`'er, som vi skal genkende som kommentarer. Derfor skal vi bruge en måde at læse ind fra planfilen, som ikke giver `#` sin sædvanlige catcode, og noget at sammenligne med.

De bliver nødt til at bo her, og ikke inde i argumentet til `\@firstofone`, pga. noget med rækkefølgen af bogstavskodning og tokenisering².

```
20 \def\revy@readwithhash{
21   \catcode`\#=11
22   \read\planread to \line
23   \catcode`\#=6
24 }
25 \catcode`\#=11
26 \edef\octothorpe{#}
27 \catcode`\#=6
28 \expandafter\@firstofone
29 \else
30 \expandafter\@gobble
31 \fi{
```

`ucph-revy` får at vide hvilke numre, som er med i et manuskript, fra en ekstern tekstfil, som vi som standard går ud fra hedder `aktoversigt.plan`, og ligger i samme mappe som `.tex`-dokumentet. Hvis det ikke passer, kan det laves om, `planfile=` ved at give valgmuligheden `planfile=<filnavn>` til pakken. Så for eksempel, hvis

¹Dem får vi nok af fra den måde, koden til dokumentationen er struktureret...

²Detaljerne efterlades som en øvelse til læseren.

planfilen ligger i mappen over `.tex`-dokumentet og hedder `plan.txt`, kan pakken kaldes således:

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

Et eksempel på en sådan planfil kunne være:

```
# Den bedste FysikRevy siden Jordens undergang
```

```
Akt 1
sange/introsang.tex
sketches/Eksempel.tex
sketches/studiepolitisk_indlæg.tex
```

```
Akt 2
sketches/noget_om_kvant.tex
sange/sang_til_en_forelæser.tex
```

```
Ekstranumre
sange/under_bæltstedet.tex
```

Her behandler vi linjer, der stater med `#` som kommentarer, og ignorerer dem. Tomme linjer bliver også ignoreret. Linjer, der ender på `.tex` tages som navne på filer i registeret, som bliver behandlet videre. Andre linjer antages at være aktoverskrifter.

Det er også muligt at angive et udvalg af info-makroerne, som bliver detaljeret i afsnit 2.3 på en enkelt linje, som en stub. En sådan har betydning for FysikRevyTeX[3], men bliver bare ignoreret her i `ucph-revy`.

`\planfileAllowedMacros` De tilladte info-makroer opbevares i toks'en `\planfileAllowedMacros`, der som standard indeholder `\title`, `\eta`, `\status`, `\melody`, `\revyname`, `\revyyear` og `\category` (der ikke er en af dem fra afsnit 2.3).

Implementering

```
32 \newtoks\planfileAllowedMacros
33 \planfileAllowedMacros={
34   \title\eta\status\melody\revyname\revyyear\category
35 }
```

`\planfileAllowMacros` Yderligere makroer kan tilføjes til den her liste med `\planfileAllowMacros`. Den kan gives flere makroer i samme argument, som fx:

```
\planfileAllowMacros{ \author \responsible }
```

Implementering

```
36 \def\planfileAllowMacros#1{
37   \toks256={#1}
38   \edef\act{
39     \noexpand\planfileAllowedMacros={
```

```

40     \the\planfileAllowedMacros \the\toks256
41   }
42 }
43 \act
44 }

```

Implementering

`\revy@tabs` Koden, som læser og forstår planfilen og de andre `.tex`-filer bor i den her makro. Den kommer til at blive kaldt midt i dokument-flowet, så vi er nødt til at være forsigtige med at afslutte linjer, som ellers kunne lave ekstra mellemrum, med %.

```

45 \def\revy@tabs{%
46   \begingroup

```

`\stubCommand` Helt konkret, så så når vi hen til at kunne ignorere makroerne i `\planfileAllowedMacros` ved at sætte dem alle sammen til at være en makro, vi kan genkende.

```

\markStubCommands
  \swap 47   \def\stubCommand{}%
\skipFiFi
\nothing 48   \def\swap##1##2{##2##1}%
          49   \def\skipFiFi##1\nothing{\fi\fi}%
          50   \let\nothing=\relax
          51   \def\markStubCommands##1{%
          52     \ifx@undefined##1 \else
          53       \ifx\relax##1%
          54         \skipFiFi
          55         \fi
          56         \fi
          57       \let##1=\stubCommand
          58     \swap\markStubCommands\nothing
          59   }
          60   \edef\act{\noexpand\markStubCommands\the\planfileAllowedMacros\relax}%
          61   \act

```

Konstruktionen, hvor `\act` bliver defineret, og så kaldt med det samme, er for at styre, hvor dybt `TeX` udvider vores makroer[1].

`\findStubCommands` Vi skal så kunne genkende den makro senere.

```

\ifstub 62   \newif\ifstub \stubbfalse
          63   \def\findStubCommands##1{%
          64     \ifx\stubCommand##1%
          65       \stubbletrue
          66       \fi
          67     \ifx\relax##1 \else
          68       \expandafter\findStubCommands
          69     \fi
          70   }%

```

`\expandFiHere` Vi får brug for den her makro, som du måske genkender som `\hop`, hvis du har læst [2].

```

71   \def\expandFiHere##1\fi{\fi ##1}%

```

Vi får brug for at vide planfilens sti

```
72     \def\patheat##1/##2{%
73         \ifx\relax##2\else
74             \expandFiHere##1/\patheat##2%
75         \fi
76     }%
77     \edef\planpath{\expandafter\patheat\planfile/\relax}%
```

Mappenavnene (`sange`, `sketches`) er ikke nødvendige for at registermærkningerne kan fungere, men de har betydning for `FysikRevyTeX` [3]. `.tex`-filerne kan placeres hvor som helst, `TeX` kan læse fra.

Implementering

```
78     \newread\planread
79     \openin\planread=\planfile
```

Ud fra sådan en planfil kan klassen generere et register**indeks** som ligner det i figur 2, eller register**mærkninger** som dem i figur 1. De titler, som er vist i figur 2, finder `ucph-revy` ved at læse `\title`-makroen i de filer, som er angivet i planfilen.

Figur 2

Introsang	Akt 1	1
Eksempel	Akt 1	2
Studiepolitisk indlæg	Akt 1	3
Noget om kvant	Akt 2	4
Sang til en forelæser	Akt 2	5
Under bæltstedet	Eks.	6

Et eksempel på et indeks til registermærkninger. I brug er det tænkt til at fylde hele siden i højden, så mærkningernes placering matcher registermærkningerne (som i figur 1).

For at sætte et registermærke i et enkelt dokument, altså som i figur 1, skal `ucph-revy` vide hvor den aktuelle fil er i rækkefølgen. For at finde ud af det, læser den hver fil, som er oplyst i planfilen, og sammenligner indholdet af deres `\title`-makroer med den seneste `\title`, som `TeX` har set.

Derfor er det nødvendigt, for at `ucph-revy` kan sætte registermærkerne, at der er angivet en `\title`. Det er altid den sidste `\title` i en fil, som `ucph-revy` sammenligner med den `\title`, som `TeX` kender. Det kan gå galt, hvis der er flere `\title`'r i en fil.

Hvis der er flere af filerne i planfilen, som har den `\title` som `ucph-revy` leder efter, vil `ucph-revy` sætte registermærker for alle de tilsvarende pladser i rækkefølgen.

```

\revy@tabs I registermærkningens verden sorterer vi bare de valgfri korte titler (se s. 17) fra,
\skiptopt når vi leder efter \title definitioner. Til det bruger vi følgende makroer:
\titleeater 80                                     %[
81           \def\skiptopt##1]##2{\titleeater{##2}}%
82           \def\titleeater##1##2\title##3{%
83             \ifx\done##3%
84               ##1%
85             \else
86               \ifx##3[%
87                 \expandFiHere\expandFiHere\skiptopt
88               \else
89                 \expandFiHere\expandFiHere\titleeater{##3}%
90               \fi
91             \fi
92           }%

For at forsøge at afværge problemer, der kan opstå, hvis der er linjeskift midt
i en \title i den eksterne fil, slår vi TEX's interne håndtering af linjeskift fra
93       \endlinechar=-1
94       \newread\tabtex

I sidste ende kommer \revy@tabs til at omdefinere sig selv, så den burde, for
hver overskrift i planfilen, indeholde sekvensen

\undertab <tæller ved start> \text <overskrift> \stop,

for hvert filnavn i planfilen indeholde sekvensen

\overtab <tæller> \text <titel i fil> \stop,

og som afslutning sekvensen

\undertab <antal filer> \text \done \stop.

\tabtoks Undervejs bygger vi de her sekvenser op i en toks.
95       \newtoks\tabtoks

\tabscount Og vi holder tal på hvor mange mærker, vi skal lave
96       \newcount\tabscount \tabscount=0

\planfile Vi \loop over nu over hver linje i \planfile, og tjekker om der overhovedet er
flere linjer,
97       \loop\ifeof\planread
98         \closein\planread
99         \closein\tabtex
100      \else

```

om vi er midt i at læse en `.tex`-fil,

```
101     \ifeof\tabtex
102     \let\readline\empty
103     \revy@readwithhash
```

om vi kigger på en linje med stub-kommandoer (som skal ignoreres),

```
104     \stufalse
105     \expandafter\findStubCommands\line\relax
106     \ifstub \else
```

om vi kigger på en tom linje (som skal ignoreres),

```
107     \IfStrEq{\line}{}{}%
```

om vi kigger på en kommentar (som skal ignoreres)

```
108     \IfBeginWith{\line}{\octothorpe}{}{}%
```

og om linjen slutter på `.tex`. Hvis den gør, må vi have fundet den næste `.tex`-fil, der skal tjekkes.

```
109     \IfEndWith{\line}{.tex}{}%
110     \advance\tabscount by 1 %
```

`\set@curr@file` tager sig af problemer med catcode'er i filnavne.

```
111     \set@curr@file{\line}%
112     \openin\tabtex={\planpath\@curr@file}%
113     }{%
```

Ellers må vi have fundet en aktoverskrift, og vi gemmer den tilsvarende sekvens i

`\tabtoks \tabtoks`, som beskrevet ovenfor.

```
114     \edef\act{\noexpand\tabtoks={%
115     \the\tabtoks
116     \noexpand\undertab
117     \number\tabscount
118     \noexpand\text
119     \line
120     \noexpand\stop
121     }}\act
122     }%
123     }%
124     }%
125     \fi
126     \else
```

Hvis vi *er* midt i at læse en `.tex`-fil, så fortsætter vi med det (det her med, at vi skelner mellem hvilken fil vi er ved at læse med en `\ifeof` betyder, at der ikke er noget eksplicit indre `\loop`, der læser `.tex`-filen).

```
127     \read\tabtex to \subline
128     \edef\readline{%
129     \expandafter\titleeater
130     \expandafter\readline\subline\title\done
131     }%
132     \ifeof\tabtex
133     \edef\act{\noexpand\tabtoks={%
134     \the\tabtoks
135     \noexpand\overtab
136     \number\tabscount
137     \noexpand\text
```

```

138         \readline
139         \noexpand\stop
140     }}\act
141     \fi
142 \fi
143 \repeat

```

Efter, vi har læst hele planfilen, kan vi gemme den afsluttende sekvens i `\tabtoks \tabtoks`, før vi gemmer hele sekvensen i `\revy@tabs`, hvor den er tilgængelig for resten af koden i klassen.

```

144     \edef\act{\noexpand\tabtoks={%
145         \the\tabtoks
146         \noexpand\undertab
147         \number\tabscount
148         \noexpand\text
149         \noexpand\done
150         \noexpand\stop
151     }}\act
152     \xdef\revy@tabs{\the\tabtoks}%
153 \endgroup

```

Nu, hvor vi har givet `\revy@tabs` det ønskede indhold, kalder vi den som det sidste i den her makro, så det her kald til `\revy@tabs` giver det samme resultat som alle senere kald.

```

154     \revy@tabs
155 }

```

`TikZ` er ikke et tegneprogram[5], men vi bruger den alligevel til at tegne registermærkningerne.

```

156 \RequirePackage{tikz}

```

`\revy@thumbindex` Den her makro indeholder alle procedurene, som typesætter registermærkningerne. Argumentet bliver ekspanderet lige før typesætningen rent faktisk finder sted, så makroens interne tilstand kan justeres. Læg særligt mærke til makroerne `\filter`, `\movetoleftside` og `\writetitlestrue`, som bliver defineret internt i makroen, og som lave om på hvilke dele af registermærkerne, som bliver typesat. Makroerne, som bliver defineret på side 14 og frem, demonstrerer hvordan de bruges.

Vi kommer til at se den fremgangsmåde, at der bliver lavet definitioner af `\undertab` og `\overtab`, som vi bagte ind i `\revy@tabs` i den forrige kodeblok, der gør det arbejde, vi ønsker at få lavet, når `\revy@tabs` bliver ekspanderet.

```

157 \def\revy@thumbindex#1{%
158     \begingroup

```

`\expandFiHere` Den her skal også defineres i den her gruppe

```

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

```

Find ud af hvor højt et enkelt mærke skal være.

```

160     \newdimen\tabheight
161     \tabheight=\paperheight
162     \advance\tabheight by -1cm%
163     \def\undertab##1\text##2\stop{
164         \ifx\done##2%
165         \divide\tabheight by \ifnum0=##1 1 \else ##1\fi%

```

```

166     \fi
167   }
168   \def\overtab##1\text##2\stop{}%
169   \revy@tabs

```

`\movetoleftside` De her definitioner indeholder forskellen på at vende registermærkningerne mod højre eller mod venstre.

```

170   \def\overtableft{-.75cm}
171   \def\overtabright{1.0cm}
172   \def\undertabright{\overtabright}
173   \def\undertableft{\overtableft - 1cm}
174   \def\underangle{|-}
175   \def\underpos{.25}
176   \def\undersign{}
177   \def\underlabeloffset{\undersign.5cm}
178   \def\textanchorcorner{east}
179   \def\textoffset{\undertableft - .5cm}
180   \def\textside{right}
181   \def\movetoleftside{
182
183     \def\overtableft{-2.3cm}
184     \def\overtabright{.2cm}
185     \def\undertabright{\overtabright + 1cm}
186     \def\undertableft{\overtableft}
187     \def\underangle{-|}
188     \def\undersign{-}
189     \def\underpos{.75}
190     \def\textanchorcorner{west}
191     \def\textoffset{\undertabright + .5cm}
192     \def\textside{left}
193   }

```

`\ifwritetitles`

```

193   \newif\ifwritetitles\writetitlesfalse

```

Her kobler vi mening på de tokens, som vi stoppede ind i `\revy@tabs` tidligere.

```

194   \def\printovertab##1##2{
195     \fill ( 0, -##1\tabheight )
196     node[ text=white, anchor=east ]{\bfseries##1}
197     +( \overtableft, -.5\tabheight + 1mm )
198     rectangle +( \overtabright, .5\tabheight - 1mm )
199     +( \textoffset, 0 )
200     \ifwritetitles
201     node[ anchor=\textanchorcorner ]{\small\bfseries##2}
202     \fi
203   ;
204   }%
205   \def\printundertab##1##2##3{%
206     \path ( 0, -##1\tabheight ) +( \undertabright, -.5\tabheight - 1mm )
207     coordinate (topright);
208     \path ( 0, -##2\tabheight ) +( \undertableft, -.5\tabheight + 1mm )
209     coordinate (bottomleft);
210     \fill[ color=black!30 ] (topright) rectangle (bottomleft);
211     \path (bottomleft)

```

```

212     \underangle node[ xshift=\underlabeloffset,
213                       pos=\underpos,
214                       rotate=\undersign90
215                       ]{\Large##3}
216     (topright);
217   }%
218   \def\overtab##1\text##2\stop{%
219     \printovertab{##1}{##2}
220   }
221   \def\undertab##1\text##2\stop{
222     \ifx\done##2\else
223       \let\undertab\undertabspan
224       \expandFiHere
225       \undertab##1\text##2\stop
226     \fi
227   }
228   \def\undertabspan##1\text##2\stop##3\undertab##4\text##5\stop{%
229     \printundertab{##1}{##4}{##2}%
230     ##3%
231     \ifx\done##5\else
232       \expandFiHere
233       \undertab##4\text##5\stop
234     \fi
235   }%

```

\filtername

```

236   \def\filtername##1{
237     \def\doWhenMatch{
238       \def\overtab####1\text####2\stop{
239         \IfStrEq{##1}{####2}{
240           \doWhenMatch
241           \printovertab{####1}{####2}
242         }{}
243       }
244       \def\undertabspan####1\text####2\stop##3\undertab####4\text####5\stop{
245         \def\doWhenMatch{
246           \printundertab{####1}{####4}{####2}
247           \def\doWhenMatch{
248             }
249           ####3
250           \ifx\done####5\else
251             \expandFiHere
252             \undertab####4\text####5\stop
253           \fi
254         }
255       }

```

Som lovet, argumentet til makroen, lige før registermærkningerne bliver typesat.

```

256     #1
257     \begin{tikzpicture}
258     \revy@tabs

```

```

259     \path (0, 0) -- (0, -\paperheight + .5cm);
260     \end{tikzpicture}
261   \endgroup
262 }

```

Og således slutter `\revy@thumbindex`.

I resten af det her afsnit følger de makroer, som kan bruges til at tegne registermærkningerne.

`\rectothumbtabfor` `\rectothumbtabfor{<titel>}`

Sætter et registermærke, som er orienteret mod højre, altså egnet til en recto-side, for filen (eller filerne), hvis titel matcher dens argument. Det giver mest mening hvis den får titlen på det aktuelle dokument, som er gemt i `\@title`. Det er den her makro, som bliver brugt til at sætte registermærkerne i eksemplet i figur 1.

Implementering

```

263 \newcommand{\rectothumbtabfor}[1]{
264   \revy@thumbindex{
265     \filtername{#1}
266   }
267 }

```

`\rectothumbindexwithtitles` `\rectothumbindexwithtitles`

Sætter registerindekset, som i figur 2. Læg mærke til, at eksemplet i figur 2 er trykket sammen i højden. Som standard skalerer den her makro indekset efter `\paperheight`, ligesom mærkerne i figur 1 er.

Implementering

```

268 \newcommand{\rectothumbindexwithtitles}{
269   \revy@thumbindex{
270     \writetitlestrue
271   }
272 }

```

`\rectothumbindex` `\rectothumbindex`

Sætter alle mærkerne, ligesom den forrige makro, men uden titelnavnene.

Implementering

```

273 \newcommand{\rectothumbindex}{
274   \revy@thumbindex{}
275 }

```

`\rectothumbtabwithtitlefor` `\rectothumbtabwithtitlefor{<titel>}`

Sætter kun registermærket for indgangen(e) med samme titel som parameteren, og skriver dens titel. Det svæver lidt i det uhvisse, hvad det gavner, at gøre sådan, men muligheden er med for fuldstændighedens skyld.

Implementering

```
276 \newcommand{\rectothumbtabwithtitlefor}[1]{
277   \revy@thumbindex{
278     \filtername{#1}
279     \writetitlestrue
280   }
281 }
```

`\versothumbtabfor` Alle disse makroer kommer også i en venstreorienteret udgave, som egner sig til
`\versothumbindexwithtitles` marginen på en verso-side:

```
\versothumbindex      \versothumbtabfor{\langle titel \rangle}
\versothumbtabwithtitlefor
\versothumbindexwithtitle
\versothumbindex
\versothumbtabwithtitlefor{\langle titel \rangle}
```

Implementering

```
282 \newcommand{\versothumbindex}{
283   \revy@thumbindex{
284     \movetoleftside
285   }
286 }
287 \newcommand{\versothumbindexwithtitles}{
288   \revy@thumbindex{
289     \movetoleftside
290     \writetitlestrue
291   }
292 }
293 \newcommand{\versothumbtabfor}[1]{
294   \revy@thumbindex{
295     \movetoleftside
296     \filtername{#1}
297   }
298 }
299 \newcommand{\versothumbtabwithtitlefor}[1]{
300   \revy@thumbindex{
301     \movetoleftside
302     \filtername{#1}
303     \writetitlestrue
304   }
305 }
```

Implementering

Og således afsluttes den betingede gruppe for `\if@thumbindex`.
306 }

I eksemplet i appendiks A er `thumbindex` ikke taget med, fordi det kun rigtig giver mening at lave registermærkninger i en samling af manuskripter. Men hvis der ligger en fil i samme mappe som eksempelfilen, som hedder `aktoversigt.plan`, hvor indholdet er eksemplet fra før, og hvis invokeringen i eksempelfilen skiftes ud med

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

så burde `.tex`-filen kompilere til et resultat, som har registermærkninger ligesom i eksemplet i figur 1. Manglende filer bliver bare ignoreret.

Det lyder måske som en god idé at have en side med et register, som ligner det i figur 2, men som passer i størrelsen til de mærkninger, som bliver sat i den enkelte manuskriptfil. Et eksempel på hvordan man kunne lave en side med sådan et registerindeks findes i appendiks B. Hvis du har adgang til klassens kildefiler kan du også trække den kode ud i filen `Thumbindex.tex`, ved at køre filen `upch-revy-ex-thumb.ins` igennem L^AT_EX.

2.3 Infoblokken

Lige efter titlen kan vi sætte en masse information, som er brugbar i arbejdet med at organisere en revy. Figur 3 viser et eksempel på, hvordan titlen ser ud med alle informationer fyldt ud.

Figur 3

```
\maketitle
```

FysikRevy™ 2019

**Et eksempel til at illustrere brugen af
ucph-revy**

skrevet af en eksempelsmed

Melodi: Queen: „Bohemian Rhapsody“ (<https://youtu.be/fJ9rUzIMcZQ>)

Status: eksempel	Version 1.0
T _E X-ansvarlig: Dig	4. april 2025
(1 minut, 47 sekunder)	2 sider

Et eksempel på, hvordan titlen er ud, når alle de informationer, som kan sættes sammen med den, er fyldt ud. Det bliver sat med kommandoen `\maketitle`.

I `.tex`-filen fortæller vi `ucph-revy` om de her ting med en række makroer i preamble'en. For eksemplets skyld er alle makroerne vist i eksemplet, men de kan, untagen versionsnummeret, sagtens udelades i et `ucph-revy`-dokument, hvis det ikke giver mening at have dem med. Makroerne var:

```
\version \version{1.0}
```

Implementering

```
307 \def\version#1{\def\@version{#1}}
```

`\version` er den eneste af de her kommandoer, som ikke kan udelades, fordi versionskontrol er vigtigt³.

Implementering

Versionsnummeret gøres påkrævet af de her kommandoer:

```
308 \def\@version{\@ifundefined{the@version}
309 {\typein[\versionsnr]{Indtast revytekstens versionsnummer: }%
310 \global\let\the@version=\versionsnr}{}\the@version}
```

```
\revyname \revyname{FysikRevy\texttrademark}
\revyyear \revyyear{2019}
\title \title[Eksempel]{Et eksempel til at illustrere brugen af ucph--revy}
```

Implementering

```
311 \def\revyname#1{\def\@revyname{#1}}
312 \def\revyyear#1{\def\@revyyear{#1}}
```

Her demonstreres brugen af den valgfri parameter til `\title`, som er en kort udgave af titlen, som bliver brugt i sidehovedet, og kan bruges i indholdsoversigter. I eksemplet i appendix A er den valgfri parameter udeladt, fordi titlen der ikke er uforholdsmæssigt lang.

Implementering

```
313 \let\revy@oldtitle=\title
314 \renewcommand{\title}[2][\relax]{%
315 \ifx\relax#1 \shorttitle{\@title}\else\shorttitle{#1}\fi
316 \revy@oldtitle{#2}%
317 }
```

`\shorttitle` Den korte titel kan også sættes med makroen `\shorttitle{<kort titel>}`.

Implementering

```
318 \newcommand{\shorttitle}[1]{\def\theshorttitle{#1}}
319 \shorttitle{\@title}
```

Hvis `ucph-revy` ikke får at vide, hvad revyens navn og år skal være, og hvad titlen er, går den som standard ud fra, at det er „DIKUrevy“, „1973“ og „En sketch“.

³Måske synes du, at der findes bedre måder, at lave versionskontrol på her i det futuristiske år 2023, men så kan du se det her påmindelse om rent faktiske at bruge dem.

Implementering

```
320 \def\@revyyear{1973}
321 \def\@revyname{DIKUrevy}
322 \def\@title{En sketch}
```

```
\author De resterende bliver ikke sat, hvis de ikke bliver defineret.
\status \author{en eksempelsmed}
\eta \status{eksempel}
\responsible \eta{$1$ minut, $47$ sekunder}
\melody \responsible{Dig}
\melody{Queen: "Bohemian Rhapsody" (\url{https://youtu.be/fJ9rUzIMcZQ})}
```

Implementering

For at holde styr på, om de her ting er defineret, definerer vi følgende `\if-` makroer, og sætter dem til falsk som udgangspunkt:

```
323 \newif\if@author\@authorfalse
324 \newif\if@status\@statusfalse
325 \newif\if@eta\@etafalse
326 \newif\if@responsible\@responsiblefalse
327 \newif\if@melody\@melodyfalse
```

Og så makroerne, som sætter værdierne:

```
328 \def\author#1{\def\@author{#1}\@authortrue}
329 \def\status#1{\def\@status{#1}\@statustrue}
330 \def\eta#1{\def\@eta{#1}\@etattrue}
331 \def\responsible#1{\def\@responsible{#1}\@responsibletrue}
332 \def\melody#1{\def\@melody{#1}\@melodytrue}
```

Vi sørger for, at de interne makroer ikke er undefinerede:

```
333 \def\@author{}
334 \def\@status{}
335 \def\@eta{}
336 \def\@responsible{}
337 \def\@melody{}

---


```

`\auteurs` Og som bonus har vi bibeholdt `\auteurs`, som er et alias til `\author`, for folk med fine fornemmelser.

Implementering

```
338 \let\auteurs\author
```

```
\writtenbyname Mærkaterne, som ucph-revy bruger i infoblokken, altså „skrevet af“, „Melodi:“,
\melodyname etc., kan laves om med makroerne
\responsibletext \writtenbyname{tekst}
\statustext \melodyname{tekst}
```

```
\responsibletext{\langle tekst\rangle}
```

og

```
\statustext{\langle tekst\rangle}
```

De fleste af dem tager bare den nye mærkat som argument, så fx „`\TeX`–ansvarlig:“ kan laves om til „Kæmpehare:“ ved at sige

```
\responsibletext{Kæmpehare:}
```

Implementering

```
339 \def\writtenbyname#1{\def\@writtenbyname{#1}}
340 \def\melodyname#1{\def\@melodyname{#1}}
341 \def\responsibletext#1{\def\@responsibletext{#1}}
342 \def\statustext#1{\def\@statustext{#1}}
343 \def\@writtenbyname{skrevet af}
344 \def\@melodyname{Melodi:}
345 \def\@responsibletext{\TeX--ansvarlig:}
346 \def\@statustext{Status:}
```

`\pagessum` Men sidetallet er lidt mere kompliceret, så her har vi brug for en makro, som tager 1 argument (et *tal*), og sætter ordet ‘side’ i det rigtige tal efter det. Så `\pagessum{0}` giver „0 sider“, mens `\pagessum{1}` giver „1 side“. En ny makro, som hedder `\pagessum`, men gør noget andet, kan defineres med `\renewcommand`.

I udgangspunktet er definitionen

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

Implementering

`\maketitle` omskrives, så den laver en titelblok som den i figur 3.

```
348 \def\maketitle{\thispagestyle{empty}
349 \vspace*{-\headheight}\vspace*{-\headsep}
350 \centering{\Large\@revyname{} \@revyyear}\vspace{5pt}\
351 {\LARGE \bf \@title}\vspace{5pt}\
352 \if@author{\large\@writtenbyname{} \@author}\vspace{5pt}\
353 \if@melody{\sl\@melodyname{} \@melody}\vspace{5pt}\
354 \if@leftfields
355 \begin{varwidth}[t]{\bigheaderwidth}\raggedleft
356 \if@status{\@statustext{} \@status}\vspace{2pt}\
357 \if@responsible{\@responsibletext{} \@responsible}\vspace{2pt}\
358 \if@eta{\@eta}\
359 \end{varwidth}%
```

Det eksplicitte mellemrum her holder formatteringen konsistent.

```
360 \hspace{.5em}\vrule{\hspace{.5em}}
361 \begin{varwidth}[t]{\smallheaderwidth}\raggedright
362 Version \@version\vspace{2pt}\
363 \today\vspace{2pt}\
364 \pagessum{\getpagerefname{lastpage}}
365 \end{varwidth}
```

```

366 \else
367   Version \@version
368   \ \hspace{.5em}\vrule\hspace{.5em}
369   \today
370   \ \hspace{.5em}\vrule\hspace{.5em}
371   \pagessum{\getpagerefnnumber{lastpage}}
372 \fi
373 \vskip 5pt }}

```

`\varwidth` (*env.*) Miljøet `\varwidth` styrer placeringen af dobbeltkolonnedelen af titelblokken. Det kommer fra

```
374 \RequirePackage{varwidth}
```

`\getpagerefnnumber` er en version af `\pageref` som kan udvides. Den kommer fra

```
375 \RequirePackage{refcount}
```

`\if@leftfields` Den her makro fungerer som⁴ `\if@responsible` \vee `\if@status` \vee `\if@eta`. T_EX's `\if`-mekanisme understøtter ikke som sådan en \vee -operator, så vi bruger tallene 0 og 1 som stedfortrædere for boolske værdier, og bruger talsammenligninger til at definere

```

376 \def\if@leftfields{
377   \def\@responsiblenum{\ifx\iftrue\if@responsible 1 \else 0 \fi}
378   \def\@statusnum{\ifx\iftrue\if@status 1 \else 0 \fi}
379   \def\@etanum{\ifx\iftrue\if@eta 1 \else 0 \fi}

380   \ifnum1=\ifnum1=\@responsiblenum 1
381     \else \ifnum1=\@statusnum 1
382     \else \ifnum1=\@etanum 1
383     \else 0
384   \fi\fi\fi
385 }

```

`\smallheaderwidth` Og disse længder blev brugt.

```

\bigheaderwidth 386 \newlength{\smallheaderwidth}
387 \setlength{\smallheaderwidth}{22ex}
388 \newlength{\bigheaderwidth}
389 \setlength{\bigheaderwidth}{\textwidth}
390 \addtolength{\bigheaderwidth}{-\smallheaderwidth}
391 \addtolength{\bigheaderwidth}{-1.5em}

```

2.3.1 Sidehoveder

Klassen definerer også sine egne sidehoveder, illustreret i figur 4, som indeholder nogle af de samme informationer.

De her sidehoveder hører til en ny `pagestyle`, som hedder `revyheadings`.

Implementering

⁴her er \vee logisk 'eller'.

Et eksempel på sidehovederne, som `ucph-revy` definerer.

`\ps@revyheadings`

```

392 \gdef\ps@revyheadings{
393   \def\@oddhead{\vbox to Opt{\vss
394     \hbox to \textwidth{\hfil\rectoheaderthumbtab}
395     \hbox{\rm\strut\@revyname{} \@revyyear}
396     \hbox to \textwidth{Version \@version\quad\today
397       \hfil {\large\sl\theshorttitle}\hfil
398       Side \rm\thepage/\pageref{lastpage}}
399     \hrule}}
400   \def\@evenhead{\vbox to Opt{\vss
401     \hbox{\versoheaderthumbtab}
402     \hbox to \textwidth{\hfil\rm\strut\@revyname{} \@revyyear}
403     \hbox to \textwidth{Side \rm\thepage/\pageref{lastpage}
404       \hfil {\large\sl\theshorttitle}\hfil
405       Version \@version\quad\today}
406     \hrule}}
407   \def\@oddfoot{}\def\@evenfoot{}

```

`\rectoheaderthumbtab` Her er `\rectoheaderthumbtab` og `\versoheaderthumbtab` makroer, som placerer registermærkningerne korrekt på siden i forhold til sidehovedet, og kalder kommandoen, som tegner dem, så længe `ucph-revy` er blevet kaldt med valgmuligheden `thumbindex`, se sektion 2.2. Hvis ikke, så gør de ikke noget.

```

408 \newcommand{\rectoheaderthumbtab}{%
409   \if@thumbindex
410     \rectothumbtabfor{\@title}%
411     \vspace*{ -\paperheight }\vspace*{ -\voffset }\vspace*{ 0.32in }%
412     \hspace{-\paperwidth }\hspace{ 1in }\hspace{ \oddsidemargin }%
413     \hspace{ \hoffset }\hspace{ \textwidth }\hspace{-0.6cm}%
414   \fi
415 }
416 \newcommand{\versoheaderthumbtab}{%
417   \if@thumbindex
418     \hspace{ -2.5in }\hspace{ -\evensidemargin }%
419     \hspace{ -\hoffset }\hspace{ -1in }%
420     \versothumbtabfor{\@title}%
421     \vspace*{ -\paperheight }\vspace*{ -\voffset }\vspace*{ 0.32in }%
422   \fi
423 }

```

Den nye `pagestyle` er aktiveret som standard.

Implementering

```
424 \pagestyle{revyheadings}
```

`\ps@empty` Men vi omdefinerer også `pagestyle`'en „empty“ til at være

```
425 \gdef\ps@empty{
426   \def\@oddhead{\vbox to 0pt{\vss
427     \hbox to \textwidth {\hfil\rectoheaderthumbtab }
428     \vbox to 2em{}}}
429   \def\@evenhead{\vbox to 0pt{\vss\hbox{\versoheaderthumbtab}}}
430   \def\@oddfoot{ }\def\@evenfoot{}}
```

2.4 Tildelingslister

Rolle- og rekvisitlisterne i starten af eksemplet, som også er gengivet i figur 5, produceres af følgende linjer:

```
\begin{roles}
  \role{PH}[Jophiel] Per Hedegaard
  \role{I} Vred instruktør
  \role{A9}[A--ni] Anine
\end{roles}
\begin{instructors}
  \instructor[Instruktør] Raben
\end{instructors}
\begin{props}
  \prop{Hedegaard--hår}[Person, der skaffer]
  \prop{Didgeridoo}[Anine]
\end{props}
```

Figur 5

Roller:

PH (Jophiel)	Per Hedegaard
I	Vred instruktør
A9 (A--ni)	Anine

Instruktører:

Instruktør	Raben
------------	-------

Rekvisitter:

Hedegaard-hår (Person, der skaffer)	
Didgeridoo (Anine)	Et traditionelt Australsk blæseinstrument

Eksempel på lister, som kan sættes med `ucph-revy` til at uddele, for eksempel, roller og ansvar for rekvisitter.

`revy@list` (*env.*) Begge de miljøer blev udledt fra det her miljø.

```

431 \newenvironment{revy@list}[1]%
432  {{{\smallskip\noindent\Large\bf#1}
433   \begin{list}{}{
434     \labelwidth 8em
435     \leftmargin 10em
436     \rightmargin 0em
437     \labelsep 1em
438     \listparindent 0em
439     \topsep 1ex
440     \partopsep 0ex
441     \parsep 1ex
442     \itemsep -1ex\relax
443   }%
444 }%
445 {\end{list}\smallskip\hfil\rule{6cm}{0.1mm}\medskip\par}

```

Argumentet, som det her miljø tager, er listens overskrift.

`\revy@itemwithout` Fordi `revy@list` er afledt fra `list`-miljøet, så forventer det, at punkter på listen bliver kaldt med `\item`. Vi, som brugere, forventer, at vi kan bruge vores egne navne for punkter på de her lister, og at de bliver sat i deres egen stil. `\revy@itemname` laver de definitioner, som er nødvendige, for at bruge de kommandoer, mens `\revy@itemwith` og `\revy@itemwithout` holder oversættelsen til en `\item`-kommando.

```

446 \def\revy@itemwithout#1{\item [{{\bf #1} \hfill} ] }
447 \def\revy@itemwith#1[#2]{\item [{{\bf #1} (#2) \hfill} ] }
448 \def\revy@itemname#1{%
449   \expandafter\def\csname #1\endcsname##1{%
450     \ifnextchar [ %
451       {\revy@itemwith{##1}}{\revy@itemwithout{##1}}%
452   }
453 }

```

Argumentet er navnet på den kommando, som kommer til at sætte punkter på listen.

Se nogle eksempler på, hvordan den bruges, lige under her.

`roles` (*env.*) I rollelisten, for eksempel, sættes hver rolletildeling med kommandoen `\role{<fk.>}[<skuespiller>](<rollebeskrivelse>)`.

Her afledes rollelistemiljøet fra `revy@list`.

```

454 \newenvironment{roles}{
455   \begin{revy@list}{\@rolename}
456   \revy@itemname{role}
457 }{
458   \end{revy@list}
459 }

```

`props` (*env.*) Ud over rollelisten findes også miljøer til lister over rekvisitter (som også ver
`costumes` (*env.*) med i eksemplet), kostumer, mikrofoner og instruktører.⁵ For de sidste miljøer er
`mics` (*env.*) kommandoerne til at lave et punkt på de lister henholdsvis
`instructors` (*env.*)

```
\prop{<rekvisit>}[<ansvarlig>]<beskrivelse>,
\costume{<kostume>}[<ansvarlig>]<beskrivelse>,
\mic{<fk.>}[<skuespiller>]<mikrofon>
```

og

```
\instructor[<titel>]<navn>.
```

Implementering

Definitionen følger de samme linjer som definitionen af `roles`-miljøet.

```
460 \newenvironment{props}{
461   \begin{revy@list}{\@propname}
462   \revy@itemname{prop}
463 }{
464   \end{revy@list}
465 }
466 \newenvironment{costumes}{
467   \begin{revy@list}{\@costumename}
468   \revy@itemname{costume}
469 }{
470   \end{revy@list}
471 }
472 \newenvironment{mics}{
473   \begin{revy@list}{\@micname}
474   \revy@itemname{mic}
475 }{
476   \end{revy@list}
477 }
```

Strukturen i listekommandoen `\instructor`, i `instructors`, er forskellig fra den, som `\revy@itemname` definerer, så vi implementerer den særskilt.

```
478 \newenvironment{instructors}{
479   \begin{revy@list}{\@instructorname}
480   \def\@instructorwith[##1]{\item [{\bf ##1} \hfill] }
481   \def\@instructorwithout{\item}
482   \def\instructor{%
483     \@ifnextchar [{\@instructorwith}{\@instructorwithout}%
484   }
485 }{
486   \end{revy@list}
487 }
```

⁵`ucph-revy` stiller ikke nogen krav til hvorvidt nogen af de her miljøer bliver brugt. Check med din revyorganisation for at høre hvad der skal være med i materiale til din revy.


```

\rolename Miljøernes standardoverskrifter—„Roller:“, „Kostumer:“, „Rekvisitter:“, „Mikro-
\costumename foner“ og „Instruktører:“—kan ændres med makroerne
\propname      \rolename{<tekst>},
\micname
\instructorname \costumename{<tekst>},
                \propname{<tekst>},
                \micname{<tekst>}

og

\instructorname{<tekst>}

```

Implementering

```

488 \def\rolename#1{\def\@rolename{#1}}
489 \def\costumename#1{\def\@costumename{#1}}
490 \def\propname#1{\def\@propname{#1}}
491 \def\micname#1{\def\@micname{#1}}
492 \def\instructorname#1{\def\@instructorname{#1}}

```

Makroerne, som opbevarer de her navne, er

```

493 \def\@rolename{Roller:}
494 \def\@costumename{Kostumer:}
495 \def\@propname{Rekvisitter:}
496 \def\@micname{Mikrofoner:}
497 \def\@instructorname{Instruktører:}

```

2.5 Brødtekst

2.5.1 sketch

`sketch` (*env.*) For at sætte talte replikker, som eksemplet vist i figur 6, bruger vi miljøet `sketch`.

Figur 6

```

      ⋮

A9 : Nå, men så må jeg jo hellere begynde at lede. (Vender sig mod Bandet)
      Bandet, har I...?

      Hvert bandmedlem holder sin egen didgeridoo op.

A9 (Til PH): Den ser ud til at være i vinkel.
PH : Sejt.

```

⋮

Et eksempel på, hvordan replikker typesættes med `sketch`-miljøet.

Implementering

`sketch` er, ligesom tildelingslisterne, implementeret ved brug af `list`-miljøet. Miljøet `revy@bodycommon` introducerer makroer, som er fælles mellem miljøerne `sketch` og `song`. Den bliver defineret på side 29.

```
498 \newenvironment{sketch}{%
499   \begin{revy@bodycommon}
500   \rm\begin{list}{}{
501     \labelwidth 2em
502     \leftmargin 3em
503     \rightmargin 0em
504     \labelsep 0.5em
505     \listparindent 2em
506     \topsep 1ex
507     \partopsep 1ex
508     \parsep 0ex
509     \itemsep 1ex\relax}%
510   \item \rule{0.2em}{0em}\vspace{-1em}\par %}
```

`\says` Inde i `sketch` sætter makroen `\says` hver enkelt replik. Den fulde syntaks er:

```
\says{<fk.>}[<instruktion>] <replik>
```

Implementering

`\says` og `\scene` (se side 30) skifter frem og tilbage mellem hver deres tilstand i miljøet. Vi piller ved `\leftskip`, for at ændre indrykningen af `\scene`.

```
511 \newdimen\old@leftskip
512 \old@leftskip \leftskip
513 \newdimen\short@leftskip
514 \short@leftskip \leftskip
515 \advance\short@leftskip -1.5em
```

`\says`

```
516 \def\says##1{\@ifnextchar [{\@saysas{##1}}{\@says{##1}} %]
517 \def\@saysas##1[##2]{\rm\item [{\bf ##1 }{\it (##2)\,}\hfill:]}
518   \leftskip \old@leftskip}
519 \def\@says##1{\rm\item [{\bf ##1\,}\hfill:]\leftskip \old@leftskip}
```

`\scene`

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

Så for eksempel den næstsidste replik i figur 6 er kreeret med følgende kode:

```
\says{A9}[Til PH] Den ser ud til at være i vinkel.
```

Og her afslutter vi `sketch`-miljøet.

```
521 }{%
522 \end{list}
523 \end{revy@bodycommon}
524 }
```

2.5.2 song

`song` (*env.*) Miljøet `song` er til sangtekster, som sættes som eksemplet i figur 7.

Figur 7

⋮

Bandet spiller „Bohemian Rhapsody“ af Queen, udelukkende på didgeridoo.

A9: Åh, Per Hede
Er den lede
Og han dumped' mig i kvant!

PH: Per Hedegaard har er reeksamenssæt til dig

⋮

Et eksempel på, hvordan sangtekster typesættes, med miljøet `song`.

`song`-miljøet er også implementeret ud fra `list` og `revy@bodycommon`.

```
525 \newenvironment{song}{%
526 \begin{revy@bodycommon}
527 \rm\begin{list}{}{%
528 \raggedright
529 \labelwidth 4.5em
530 \leftmargin 7em
531 \rightmargin 0em
532 \labelsep 2em
533 \listparindent -2em
534 \topsep 0ex
535 \partopsep \bigskipamount
536 \parsep .6\baselineskip
537 \itemsep -.6\baselineskip
538 \relax
539 }
540 \item[] \rule{0.2em}{0em}\vspace{-\baselineskip}\strut\par %}
```

`\sings` Inde i `song`-miljøet sætter makroen `\sings` sangstrofer. Den fulde syntaks er

```
\sings{<fk.>}[<instruktion>] <sangtekst>
```

Linjeskift er vigtige i sangtekster, så `song`-miljøet ændrer, hvordan L^AT_EX reagerer på linjeskift i kildefilen. For eksempel kan **A9**'s strofer i figur 7 kreeres med denne kildekode:

```
\sings{A9} Åh, Per Hede
           Er den lede
           Og han dumped' mig i kvant!
```

Implementering

`\sings` og `\scene` skifter frem og tilbage mellem to endnu mere forskellige tilstande, end `\says` og `\scene` i `sketch`-miljøet. Læg især mærke til brugen af `\obeycr` og `\restorecr`.

```
541 \newdimen\old@leftskip
542 \old@leftskip \leftskip
543 \newdimen\short@leftskip
544 \short@leftskip \leftskip
545 \advance\short@leftskip -5.5em
```

`\sings`

```
546 \def\sings##1{\obeycr
547   \@ifnextchar [{\@singsas{##1}}{\@sings{##1}} %}
548 \def\@singsas##1[##2]{\rm\item [\hskip\@leftsingpad{\bf ##1 }{\it
549   (##2)\,}:\hskip\@rightsingpad]\hskip-2em\leftskip \old@leftskip}
550 \def\@sings##1{\rm\item [\hskip\@leftsingpad{\bf ##1}\,}:\hskip
551   \@rightsingpad]\hskip-2em\leftskip \old@leftskip}
```

`\scene`

```
552 \let\old@sings=\sings
553 \def\scene{%
554   \def\sings{\let\sings=\old@sings \medskip\sings}
555   \medskip\par\leftskip \short@leftskip \restorecr
556   \it\noindent\relax
557 }
```

`\@leftsingpad` og `\@rightsingpad` gør det muligt, at stille sangtekster til at være venstre- eller højrejusterede.

```
558 \newskip\@leftsingpad
559 \newskip\@rightsingpad
560 \def\flushsingsright{\@leftsingpad Opt plus 1fill\@rightsingpad Opt\relax}
561 \def\flushsingsleft{\@leftsingpad Opt\@rightsingpad Opt plus 1fill\relax}
```

Som standard sætter vi venstrejustering.

```
562 \flushsingsleft
```

Implementering

Vi kalder `\obeycr` her, sammen med nogle kommandoer, som fortæller L^AT_EX, at den ikke skal bekymre sig så meget om længderne på linjer.

```
563 \tolerance10000\hfuzz21cm\obeycr
564 }{
565 \end{list}
566 \end{revy@bodycommon}
567 }
```

Implementering

`\obeycr` Makroerne, som gør linjeskift-magien mulig, er

```
\restorecr 568 \newif\if@newpar
569 {
570 \catcode\^^M=13
Så nu skal vi passe på med vores linjeskift her i koden...
571 \gdef\@carriagereturn{%
572 \@ifnextchar^^M{%
573 \@newpartrue%
574 }{%
575 \@ifnextchar\end{}}{%
576 \@ifnextchar\sings{%
577 \if@newpar\bigskip\par\fi\@newparfalse%
578 }{%
579 \if@newpar\par\else\\\fi\@newparfalse%
580 }%
581 }%
582 }%
583 }%
584 \gdef\obeycr{%
585 \catcode\^^M=13 %
586 \let^^M=\@carriagereturn%
587 \@gobblecr%
588 }%
589 \gdef\restorecr{\catcode\^^M=5 }%
590 }
```

2.5.3 Fælles for sketch og song

Vi har også de følgende makroer, som kan bruges i både `song-` og `sketch-` miljøerne:

Implementering

`revy@bodycommon` (*env.*) De bliver pakket i miljøet `revy@bodycommon`.

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

`\role` En makro til at formattere rolleforkortelser i teksten.

Implementering

Den her makro har det samme navn som indgangskommandoen i `roles-` miljøet. Men det er fint, for den bliver defineret inde i `roles`'s interne `group`.

```
592 \let\role = \textbf
```

`\does` En makro til at give instruktioner til en bestemt rolle, som ikke er en talt replik, som for eksempel

```
\does{I} reagerer med åbenmundet vantro.
```

Implementering

Udtrykkningen på 1em er en hårdt indkodet længde, som passer til, hvordan `song` og `sketch` er sat op lige nu. Jeg er sikker på den ikke kommer til at hoppe op og bide nogen bagi senere...

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

`\scene` `\scene` bruges til regibemærkninger, som for eksempel

```
\scene Hvert bandmedlem holder sin egen didgeridoo op.
```

Implementering

```
\scene blev implementeret inde i sketch- og song- miljøerne.
```

`\act` `\act` er til instruktioner, som for eksempel

```
(...) \act{Vender sig mod Bandet} Bandet, har I...?
```

Implementering

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

De blev begge brugt til at lave eksemplet i figur 6.

Implementering

Og her afslutter vi miljøet `revy@bodycommon`.

```
595 }{\}
```

2.6 Assorterede emner

2.6.1 SATyR

\SaTyR Man skriver SATyRRevy ved at elske alle dyr, og derefter skrive

```
\SaTyR{}Revy
```

Implementering

```
596 \newcommand{\SaTyR}{S\hspace*{-.2ex}\raisebox{-.15em}{A}\hspace*{-.5ex}TyR}
```

Implementering

2.6.2 .aux-filen

Til fremtidigt brug sørger vi for, at vores metadata også bliver skrevet ud til .aux-filen, når dokumentet er færdigt.

Det er også her, vi sætter en label på den sidste side.

```
597 \let\revyinfo=\relax
598 \newtoks\@titletoks
599 \newtoks\@authortoks
600 \newtoks\@melodytoks
601 \newtoks\@revyrevynametoks
602 \newtoks\@revyrevyyeartoks
603 \let\end@document=\enddocument
604 \def\enddocument{\label{lastpage}\write\@auxout
605   {\string\@ifundefined{revyinfo}{\string\def\string
606   \revyinfo\string##1\string##2\string##3\string
607   ##4\string##5\string##6\string##7\string##8{}}{}}\@titletoks=
608   \expandafter{\@title}\@titletoks=
609   \expandafter{\@title}\@authortoks=
610   \expandafter{\@author}\@melodytoks=
611   \expandafter{\@melody}\@revyrevynametoks=
612   \expandafter{\@revyname}\@revyrevyyeartoks=
613   \expandafter{\@revyyear}\write\@auxout
614   {\string\revyinfo{\the\@titletoks}{\@version}
615     {\today}%
616     {\the\@authortoks}%
617     {\the\@melodytoks}%
618     {}}%
619     {\the\@revyrevynametoks}%
620     {\the\@revyrevyyeartoks}%
621   }%
622 \end@document
623 }
```

Litteratur

- [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] Peter Müller, Kristoffer Stensbo-Smidt, Troels Henriksen, Sebastian Paaske Tørholm, Sarah Brofeldt, Anders Komár Ravn, Kristoffer Levin Hansen, Asbjørn Preuss, and Mikkel Noer. *FysikRevyTeX*. <https://github.com/FysikRevy/FysikRevyTeX>, 2025.
- [4] Overleaf. Understanding packages and class files. https://www.overleaf.com/learn/latex/Understanding_packages_and_class_files.
- [5] Till Tantau. *The TikZ and PGF Packages*. <https://mirrors.dotsrc.org/ctan/graphics/pgf/base/doc/pgfmanual.pdf>, 2023.

Implementering

Index

Numbers written in *italics* 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		
<code>\@author</code> 328, 333, 352, 610	<code>\@ifundefined</code> 308, 605	<code>\@oddhead</code> 393, 426
<code>\@authorfalse</code> 323	<code>\@instructorname</code> 479, 492, 497	<code>\@propname</code> 461, 490, 495
<code>\@authortoks</code> 599, 609, 616	<code>\@instructorwith</code> 480, 483	<code>\@responsible</code> 331, 336, 357
<code>\@authortrue</code> 328	<code>\@instructorwithout</code> 481, 483	<code>\@responsiblefalse</code> 326
<code>\@auxout</code> 604, 613	<code>\@leftsingpad</code> 548, 550,	<code>\@responsiblenum</code> 377, 380
<code>\@carriagereturn</code> 571, 586	558, 560, 561	<code>\@responsibletext</code> 341, 345, 357
<code>\@costumename</code> 467, 489, 494	<code>\@melody</code> 332, 337, 353, 611	<code>\@responsiblettrue</code> 331
<code>\@curr@file</code> 112	<code>\@melodyfalse</code> 327	<code>\@revyname</code> 311, 321, 350,
<code>\@eta</code> 330, 335, 358	<code>\@melodyname</code> 340, 344, 353	395, 402, 612
<code>\@etafalse</code> 325	<code>\@melodytoks</code> 600, 610, 617	<code>\@revyrevynametoks</code> 601, 611, 619
<code>\@etanum</code> 379, 382	<code>\@melodytrue</code> 332	<code>\@revyrevyyeartoks</code> 602, 612, 620
<code>\@etattrue</code> 330	<code>\@micname</code> 473, 491, 496	<code>\@revyyear</code> 312, 320, 350,
<code>\@evenfoot</code> ... 407, 430	<code>\@newparfalse</code> 577, 579	395, 402, 613
<code>\@evenhead</code> ... 400, 429	<code>\@newpartrue</code> 573	<code>\@rightsingpad</code> 549, 551,
<code>\@firstofone</code> 28	<code>\@oddfoot</code> 407, 430	
<code>\@gobble</code> 30		
<code>\@gobblecr</code> 587		
<code>\@ifnextchar</code> .. 450, 483, 516, 547, 572, 575, 576		

559, 560, 561

`\@rolename`
 455, 488, 493

`\@says` 516, 519

`\@saysas` 516, 517

`\@sings` 547, 550

`\@singsas` 547, 548

`\@status` . 329, 334, 356

`\@statusfalse` 324

`\@statusnum` .. 378, 381

`\@statustext`
 342, 346, 356

`\@statustrue` 329

`\@thumbindexfalse` .. 2

`\@thumbindextrue` ... 3

`\@title` ... 315, 319,
 322, 351, 410,
 420, 608, 609

`\@titletoks` ... 598,
 607, 608, 614

`\@undefined` 52

`\@version` 307,
 308, 362, 367,
 396, 405, 614

`\@writtenbyname` ...
 339, 343, 352

A

`\act` 38,
 43, 60, 61, 114,
 121, 133, 140,
 144, 151, 594

`\addtolength`
 18, 390, 391

`\article` 3

`\auteurs` 338

`\author` 323, 338

B

`\baselineskip`
 536, 537, 540

`\bf` ... 351, 432, 446,
 447, 480, 517,
 519, 548, 550

`\bfseries` 196, 201

`\bigheaderwidth` ...
 355, 386

`\bigskip` 577

`\bigskipamount` ... 535

C

`\category` 34

`\centering` 350

`\costumename` 488

costumes (env.) ... 460

`\csname` 449

`\CurrentOption`
 6, 8, 10

D

`\DeclareOption` ... 3, 5

`\divide` 165

`\does` 593

`\done` 83,
 130, 149, 164,
 222, 231, 250

`\doWhenMatch` .. 237,
 240, 245, 247

E

`\end@document`
 603, 622

`\endcsname` 449

`\enddocument` . 603, 604

`\endlinechar` 93

environments:

 costumes 460

 instructors ... 460

 mics 460

 props 460

 revy@bodycommon
 29, 591

 revy@list 431

 roles 454

 sketch 498

 song 525

 varwidth 20

`\eta` 34, 323

`\evensidemargin` .. 418

`\expandafter` . 8, 28,
 30, 68, 77, 105,
 129, 130, 449,
 608, 609, 610,
 611, 612, 613

`\expandFiHere` ... 71,
 74, 87, 89, 159,
 224, 232, 251

F

`\fill` 195, 210

`\filtername` 236, 265,
 278, 296, 302

`\findStubCommands` .
 62, 105

`\flushingsleft` ...
 561, 562

`\flushingsright` . 560

G

`\gdef` 392, 425,
 571, 584, 589

`\getpagerefnumber` .
 364, 371

`\global` 310

H

`\hbox` 394, 395,
 396, 401, 402,
 403, 427, 429

`\headheight` 349

`\headsep` 349

`\hfil` .. 394, 397, 402,
 404, 427, 445

`\hfill` 446, 447,
 480, 517, 519

`\hfuzz` 563

`\hoffset` 413, 419

`\hrule` 399, 406

`\hskip` 548,
 549, 550, 551

`\hspace` 360, 368, 370,
 412, 413, 418,
 419, 593, 596

I

`\if@author` ... 323, 352

`\if@eta` 325, 358

`\if@leftfields`
 354, 376

`\if@melody` ... 327, 353

`\if@newpar`
 568, 577, 579

`\if@responsible` ...
 326, 357, 377

`\if@status` 324,
 356, 378, 379

`\if@thumbindex` .. 2,
 5, 19, 409, 417

`\IfBeginWith` ... 6, 108

`\IfEndWith` 109

`\ifeof` 97, 101, 132

`\ifnum` 165, 347,
 380, 381, 382

`\IfStrEq` 107, 239

`\ifstub` 62, 106

`\iftrue` .. 377, 378, 379

`\ifwritetitles` ...
 193, 200

<code>\ifx</code>	52, 53, 64, 67, 73, 83, 86, 164, 222, 231, 250, 315, 377, 378, 379			
<code>\instructor</code>	482			
<code>\instructorname</code>	488			
<code>instructors (env.)</code>	460			
<code>\it</code>	517, 520, 548, 556			
<code>\item</code>	446, 447, 480, 481, 510, 517, 519, 540, 548, 550			
<code>\itemsep</code>	442, 509, 537			
L				
<code>\label</code>	604			
<code>\labelsep</code>	437, 504, 532			
<code>\labelwidth</code>	434, 501, 529			
<code>\LARGE</code>	351			
<code>\Large</code>	215, 350, 432			
<code>\large</code>	352, 397, 404			
<code>\leftmargin</code>	435, 502, 530			
<code>\leftskip</code>	512, 514, 518, 519, 520, 542, 544, 549, 551, 555			
<code>\let</code>	50, 57, 102, 223, 310, 313, 338, 552, 554, 586, 592, 597, 603			
<code>\line</code>	22, 105, 107, 108, 109, 111, 119			
<code>\listparindent</code>	438, 505, 533			
<code>\LoadClass</code>	14			
<code>\loop</code>	97			
M				
<code>\maketitle</code>	307, 348			
<code>\markStubCommands</code>	47			
<code>\medskip</code>	445, 520, 554, 555			
<code>\melody</code>	34, 323			
<code>\melodyname</code>	339			
<code>\micname</code>	488			
<code>mics (env.)</code>	460			
<code>\movetoleftside</code>	170, 284, 289, 295, 301			
N				
<code>\newcount</code>	96			
<code>\newdimen</code>	160, 511, 513, 541, 543			
<code>\newif</code>	2, 62, 193, 323, 324, 325, 326, 327, 568			
<code>\newlength</code>	386, 388			
<code>\newread</code>	78, 94			
<code>\newskip</code>	558, 559			
<code>\newtoks</code>	32, 95, 598, 599, 600, 601, 602			
<code>\noexpand</code>	39, 60, 114, 116, 118, 120, 133, 135, 137, 139, 144, 146, 148, 149, 150			
<code>\noindent</code>	432, 520, 556			
<code>\nothing</code>	47			
<code>\number</code>	117, 136, 147			
O				
<code>\obeycr</code>	546, 563, 568			
<code>\octothorpe</code>	20, 108			
<code>\oddsidemargin</code>	412			
<code>\old@leftskip</code>	511, 512, 518, 519, 541, 542, 549, 551			
<code>\old@sings</code>	552, 554			
<code>\openin</code>	79, 112			
<code>\overtab</code>	135, 168, 218, 238			
<code>\overtableft</code>	170, 173, 182, 185, 197			
<code>\overtabright</code>	171, 172, 183, 184, 198			
P				
<code>\pageref</code>	398, 403			
<code>\pagessum</code>	347, 364, 371			
<code>\pagestyle</code>	424			
<code>\paperheight</code>	17, 161, 259, 411, 421			
<code>\paperwidth</code>	412			
<code>\par</code>	445, 510, 520, 540, 555, 577, 579			
<code>\parsep</code>	441, 508, 536			
<code>\partopsep</code>	440, 507, 535			
<code>\PassOptionsToClass</code>	10			
<code>\path</code>	206, 208, 211, 259			
<code>\patheat</code>	72, 74, 77			
<code>\planfile</code>	4, 7, 8, 9, 77, 79			
<code>\planfileAllowedMacros</code>	32, 39, 40, 60			
<code>\planfileAllowMacros</code>	36			
<code>\planpath</code>	77, 112			
<code>\planread</code>	22, 78, 79, 97, 98			
<code>\printovertab</code>	194, 219, 241			
<code>\printundertab</code>	205, 229, 246			
<code>\ProcessOptions</code>	13			
<code>\propname</code>	488			
<code>props (env.)</code>	460			
<code>\ps@empty</code>	22, 425			
<code>\ps@revyheadings</code>	392			
Q				
<code>\quad</code>	396, 405			
R				
<code>\raggedleft</code>	355			
<code>\raggedright</code>	361, 528			
<code>\raisebox</code>	596			
<code>\read</code>	22, 127			
<code>\readline</code>	102, 128, 130, 138			
<code>\rectoheaderthumbtab</code>	394, 408, 427			
<code>\rectothumbindex</code>	273			
<code>\rectothumbindexwithtitles</code>	268			
<code>\rectothumbtabfor</code>	263, 410			
<code>\rectothumbtabwithtitlefor</code>	276			
<code>\relax</code>	13, 50, 53, 60, 67, 73, 77, 105, 314, 315, 442, 509, 538, 556, 560, 561, 597			
<code>\repeat</code>	143			
<code>\RequirePackage</code>	1, 156, 374, 375			
<code>\responsible</code>	323			


```

\melody{Queen: „Bohemian Rhapsody`` (\url{https://youtu.be/fJ9rUzIMcZQ})}

\begin{document}
\maketitle

\begin{roles}
\role{PH}[Jophiel] Per Hedegaard
\role{I} Vred instruktør
\role{A9}[A--ni] Anine
\end{roles}
%% instructors- og props-miljøet kan fjernes, hvis der ikke er brug
%% for det
\begin{instructors}
\instructor[Instruktør] Raben
\end{instructors}
\begin{props}
\prop{Hedegaard--hår}[Person, der skaffer]
\prop{Didgeridoo}[Anine]
\end{props}

\begin{sketch}
\scene{Lys op}
\says{PH} Hej venner. I den her sketch, der spiller jeg Per
Hedegaard. Det kan man se ud fra farven på mit hår. Men bare rolig,
jeg er ikke Per Hedegaard i virkeligheden, der er ikke nogen
regneøvelser, I har glemt at lave.

Se, hvis jeg tager håret af \act{fjerner parykken}, så er det helt
tydeligt, at jeg ikke er den rigtige Hedegaard.

\says{I}[Stormer ind på scenen] \emph{NEEEEEJ}, for helvede. Du må
ikke pille ved den fjerde væg. Det er simpelthen så uprofessionelt,
det der!

\says{PH} Se, venner, det var meningen, at der skulle have været en
anden replik der, men fordi der ikke er sat nogen på den rolle i
rollelisten, så kunne vi ikke høre nogen som helst kritik af min
performance.

\does{I} reagerer med åbenmundet vantro.

\says{PH} Det må også være svært at følge skuepil--instruktioner,
når man ikke eksisterer, kan jeg forestille mig.

\says{A9} Hej Jophiel.
\does{PH} tager sin paryk på igen.
\says{A9} Hej, Per Hedegaard.
\says{A9} Var det ikke meningen, at det her skulle være en sang?
\says{PH} Jo, for dølen. Har du en didgeridoo med?
\says{A9} Didgeridoo?

```

```

\says{PH} Ja. Der står i rekvisitlisten, at du skaffer en.
\says{A9} Jamen så må det jo passe.
\says{PH} Ja, ellers skriver vi "`Person, der skaffer," for at
minde os om at skrive et navn ind.
\says{A9} Nå, men så må jeg jo hellere begynde at lede. \act{Vender
sig mod Bandet} Bandet, har I...?
\scene{Hvert bandmedlem holder sin egen didgeridoo op.}
\says{A9}[Til PH] Den ser ud til at være i vinkel.
\says{PH} Sejt.
\says{PH}[Råber til Bandet] Og \emph{tre!} \emph{Fire!}
\end{sketch}
\begin{song}%
\scene{Bandet spiller ``Bohemian Rhapsody" af Queen, udelukkende på
didgeridoo.}

\sings{A9} Åh, Per Hede
          Er den lede
          Og han dumped' mig i kvant!

\sings{PH} Per Hedegaard har er reksamenssæt til dig
\sings{A9} Til mig?
\sings{PH} Til dig!
\scene{Episk didgeridoo--solo, mens A9 løser reksamenssæt.}

\scene ...

\does{A9} rækker hånden i vejret, og bandet stopper.
\end{song}%
\begin{sketch}
\says{PH}[Er et stykke tid om at lægge mærke til A9, men kommer til
sidst over] Ja, spørgsmål?
\says{A9} Det her eksamenssæt er jo trivielt...?
\scene{Lys ned}
\end{sketch}

\end{document}
\endinput
%%
%% End of file `Eksempel.tex'.

```

B Eksempel på registerindeks

```

\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}

```