The jslectureplanner bundle

Jürgen Spitzmüller*

Version 1.2, 2016/05/11

Abstract

The jslectureplanner bundle provides a collection of packages I have written in order to facilitate, and somewhat systematize, the planning of my university courses and the generation of course material (and thus, ultimately, to relieve me from some of the more boring and time-consuming part of teaching work). Currently, the bundle consists of the main, title-giving package jslectureplanner that provides an interface to central, partly dynamically generated course metadata as well as routines to generate programs, bibliographies etc., and one smaller companion package: jsmembertable, a package that can construct course member and presence tables. This manual documents these packages.

Contents

*Please report issues via `https://github.com/jspitz/jslectureplanner`.

Part I
The jslectureplanner package

This part of the manual describes the basic user interface of the jslectureplanner package. Please also refer to the example files included in the bundle in order to see how jslectureplanner is supposed to be used "in practice".

1 Aim of this package

The basic idea of the jslectureplanner package is that you record all general data of a course (date, semester, type, course title, session titles, general information etc.) in a central tex file and then input this file to all documents related to the course (such as the course program, bibliography, scripts, handouts, beamer presentations, exercises etc.), in order to re-use the recorded data. This is particularly helpful if you have standard courses which are held every other semester: instead of changing the data in every single file, you just need to change it once, in the metadata file.

To further facilitate the planning, the jslectureplanner package is able to calculate the session dates of a whole semester, if the sessions of the course follow at regular intervals. You only need to set the date of the first session, the package does the rest. Moreover, the package can be used to generate a sectioned bibliography for the course via biblatex.

Since the package has been written for my own needs, it is somewhat tied to my workflow. Particularly, it is tied to my context: teaching in the Humanities at a German-speaking university. Hence, the categorization of the package draws on this context. In particular, all strings and formats are German. However, it is possible to add new categories as well as to customize and translate the strings. This is described later in this manual.

2 Requirements of jslectureplanner

The following packages are required and loaded by jslectureplanner:

- advdate
- calc
- datetime
- etoolbox
- ifthen
- xkeyval

3 General idea

The general idea goes as follows:

- Set up a *metadata.tex* file (or whatever name you prefer) and record all meta information of the course in it (as described in sec. 4 and 5).
- In your documents, load the package (`\usepackage{jslectureplanner}`) and/or[1] input the metadata file (`\input{metadata.tex}`).
- Finally, use the macros described in sec. 6 and 7 to retrieve the respective data and re-use the recorded information in your documents.

The procedure is detailed in the following sections.

4 Setting up course metadata

You can set up the course metadata either via package options (if you load the package in the metadata file; see sec. 4.1) or via specific macros (see sec. 4.2).

4.1 Via package options

`\usepackage[<comma-separated options>]{jslectureplanner}`

4.1.1 Course classification

The default course classification draws on the conventions at German-speaking universities (or those where I have worked, for that matter). Each type is connected to a verbose string (such as *Seminar*), a short form (such as *SE*) and a form that is used to denote individual sessions (as in "the second *session* of this course").

`type=<type>` defines the course type. Predefined values are:

- `vl`: A lecture (verbose form: *Vorlesung*, short form: *VL*, session form: *Vorlesung*)
- `ps`: Undergraduate seminar (verbose form: *Proseminar*, short form: *PS*, session form: *Sitzung*)
- `se`: Seminar (verbose form: *Seminar*, short form: *SE*, session form: *Sitzung*)
- `ue`: Practical course (verbose form: *Übung*, short form: *UE*, session form: *Sitzung*)
- `ko`: Colloquium (verbose form: *Kolloquium*, short form: *KO*, session form: *Sitzung*)
- `pv`: Research seminar (verbose form: *Privatissimum*, short form: *PV*, session form: *Sitzung*)

Section 8 explains how to extend this list and how to customize the strings. Please also cf. sec. 8.5 on the concept of "styles".

[1] If you prefer to record the data via package options (see sec. 4.1) instead of macros (see sec. 4.2), you need to load the package in the metadata file itself.

4.1.2 Course title

title=<title> Main title of the course

titlesep=<separator> Separator between title and subtitle

subtitle=<subtitle> Subtitle of the course

shorttitle=<shorttitle> Short title of the course (for headings etc.)

4.1.3 Date and place

semester=<term>, where **<term>** is one of

- **fs**: Spring term (*Frühjahrssemester*, short form: *FS*)
- **hs**: Fall term (*Herbstsemester*, short form: *HS*)
- **ss**: Summer term (*Sommersemester*, short form: *SoSe*)
- **ws**: Winter term (*Wintersemester*, short form: *WS*)

Sec. 8 explains how to customize and extend this list. Also cf. sec. 8.5.

year=<year> Year of the term

uni=<university> Your university

institute=<institute> Your institute/department

room=<room> The room where the course generally takes place. It is possible to specify diverging rooms for individual sessions (see sec. 5).

startdate=<startdate> The date of the first session (the argument *must* have the form DD/MM/YYYY). If no startdate is given, `\today` is used.

interval=<interval> The interval (in days) between two sessions. The preset interval is 7 (= weekly courses).

4.1.4 Additional information

instructor=<name> (General) course instructor's name. It is possible to specify different or specific instructors for individual sessions (see sec. 5).

platform=<data> Information of the e-learning platform (such as the URL)

officehours=<data> Date of your office hours

officenumber=<data> Your office (room) number

4.2 Via macros

The following macros provide an alternative way to set up the course metadata.

4.2.1 Course classification

- **`\LecType{<course type>}`**: Set course type; **`<course type>`** is one of[2]
  - **`vl`**: Lecture (*Vorlesung*)
  - **`ps`**: Undergraduate seminar (*Proseminar*)
  - **`se`**: Seminar (*Seminar*)
  - **`ue`**: Practical course (*Übung*)
  - **`ko`**: Colloquium (*Kolloquium*)
  - **`pv`**: Research seminar (*Privatissimum*)

4.2.2 Course title

- **`\LecTitle[<package options>]{<title>}`**: Record course title and options; **`<package options>`** might be any set of package options described in sec. 4.1, so this macro can actually be used to record all metadata.
- **`\LecTitleSep{<title separator>}`**: Specify a separator between course title and subtitle (e. g., ". " or " – ")
- **`\LecSubTitle{<subtitle>}`**: Record course subtitle

4.2.3 Date and place

- **`\LecYear{<term>}{<year>}`**, Record term and year of the course; **`<term>`** is one of[3]
  - **`fs`**: Spring term (*Frühjahrssemester*)
  - **`hs`**: Fall term (*Herbstsemester*)
  - **`ss`**: Summer term (*Sommersemester*)
  - **`ws`**: Winter term (*Wintersemester*)
- **`\LecUni{<university>}`**: Record the name of your university
- **`\LecInstitute{<institute>}`**: Record your institute's/department's name
- **`\LecRoom{<room>}`**: Record the room where the course (generally) takes place
- **`\LecStartDate{<startdate>}`**: Set the date of the first session (the argument *must* have the form DD/MM/YYYY)

[2] See above sec. 4.1.1 for a detailed description.
[3] See above sec. 4.1.3 for a detailed description.

- **\LecInterval{<interval>}**: Adjust the interval (in days) between two sessions. The preset interval is 7 (= weekly courses).
- **\SetAutoOffset{<n>}{<days>}**: Automatically shift all subsequent sessions by <days> extra days after every <n>th session. This is useful if you have each week two regular sessions and then a pause until the next week (for instance, a Tuesday + Thursday course could be defined via **\LecInterval{2}** and **\SetAutoOffset{2}{3}**). <n> must be a positive integer, <days> can also be a negative integer value.

4.2.4 Additional information

- **\LecInstructor{<name>}**: Record the (general) instructor's name
- **\SetOfficeHours{<office hours>}**: Record date of your office hours
- **\SetOfficeNumber{<office number>}**: Record your office (room) number
- **\SetPlatform{<platform information>}**: Record information (such as URL) of the e-learning platform

5 Setting up a course schedule

In the metadata file, you can set up a schedule for the course by entering session information in chronological order. This is done via the following macro:

- **\NewSession[<options>]{Session title}**

  Valid **<options>** include:

  - **titlesep=<separator>**: Separator between session title and subtitle
  - **subtitle=<subtitle>**: Subtitle of the session
  - **shorttitle=<shorttitle>**: Short title (for headings etc.)
  - **instructor=<name>**: Specific instructor for this session (if it differs from the usual course instructor)
  - **presstudents=<names>**: Students presenting in this session (used for student presentation lists, see below sec. 6.4)
  - **room=<room>**: Specific room for this session (if it differs from the usual course room)
  - **bibsec=<keyword>**: Session keyword for the bibliography (see below sec. 6.3 for its use)
  - **cancel=<true|false>**: Marks this session as "canceled". If you specify **cancel=true**, the session will be printed in the program in bold type and appended by a verbal comment. For example,

    \NewSession[cancel=true]{Dies Academicus}

    will expand to

15. 05.: Dies Academicus – keine Sitzung

(i. e., "Dies Academicus – no session").

Furthermore, a canceled session is not counted in the session counter that is used to retrieve session-specific information (see below sec. 7).

- **\SetBreak[]{Break}**: Add a semester break (e. g., holidays) that affects one session. The macro increments the internal counters respectively and expands in the program to

  (Break)

  By means of the optional argument, the real time span of the break might be given if you want to print it on the program. If you pass an optional argument ****, the macro will expand to

  (: Break)

  So for instance, **\SetBreak[28.05.--3.06.]{Whitsun holidays}** expands to

  (28.05.–3.06.: Whitsun holidays)

- **\SetBreaks[<options>]{Break}**: Add a semester break that affects more than one session (e. g., two-week holidays). The output in the program is identical to the **\SetBreak** macro, but the counter incrementation can be adjusted to the needed time span.

  Possible **<options>** are:

  - **units=<n>**: Number of interval units the break takes. One *unit* is the number of days as specified by the **interval** option (one week by default[4]). This value is needed in order to increment the internal counter that is used to calculate the session dates. If **units** is not specified, the counter is incremented by one.[5] <n> must be a positive integer.
  - **span=**: Real time span (verbally output on the program)

- **\SetLecOffset{<n>}**: Shift the date of all subsequent sessions by <n> days. <n> can also be negative.

- **\SetBeamerFrameBreak**: Add a frame break in the beamer program at this position (more precisely, start a new frame). This also increments the frame counter which is used for subsequent program frame titles. Note that you can insert maximally 3 breaks.

- **\SetBeamerHook{<code>}**: Add arbitrary LaTeX code to the beamer program

[4]The former option **weeks** is deprecated as of v. 0.9, since the interval can be adjusted now.

[5]That is: **\SetBreak{Break}** = **\SetBreaks{Break}** = **\SetBreaks[units=1]{Break}**.

- **\begin{SessionBlock}{<Block title>}**
  …
  \end{SessionBlock}:
  Thematic block consisting of several sessions. Respective sessions are nested inside this block.

6 Retrieving general course information

Once the metadata file is set up as documented above and input to your document(s), you can retrieve the recorded as well as some concatenated information by means the following macros.

6.1 Course data

The following macros output general course-specific information:

- **\lectype**: Outputs the course type in short form (e. g., "SE")
- **\lectypeverb**: Outputs the course type in verbose form (e. g., "Seminar")
- **\lectypesession**: Outputs the appropriate "session" string (e. g., "Vorlesung" or "Sitzung")
- **\lectitle**: Outputs the main title of the course
- **\lectitlesep**: Outputs the separator between title and subtitle
- **\lecsubtitle**: Outputs the subtitle of the course
- **\lecfulltitle**: Outputs the course's full title (title, separator, subtitle)
- **\lecshorttitle**: Outputs the short version of the course title
- **\lecsemshort**: Outputs the short version of the term type (e. g., "WS")
- **\lecsemverb**: Outputs the verbose version of the term type (such as "Wintersemester")
- **\lecyear**: Outputs the year
- **\lecsemester**: Outputs the short version of the semester (e. g., "WS 2014")
- **\lecsemesterverb**: Outputs the verbose version of the semester (such as "Wintersemester 2014")
- **\lecuniversity**: Outputs the university's name
- **\lecinstitute**: Outputs the institute's/department's name
- **\lecinstructor**: Outputs the (general) instructor's name

- **\lecroom**: Outputs the (general) course room
- **\lecplatform**: Outputs information on the e-learning platform
- **\officehours**: Outputs the office hours
- **\officenumber**: Outputs the office (room) number

6.2 Generating course programs

- **\makeprogram**: Generates a course program in the form:

```
\begin{labeling}{\lecprogramlistindent}
\item[<short date>] <session full title>
...
(break)
...
\item[<short date>] <session full title>
\end{labeling}
```

  The **labeling** environment is defined by the KOMA classes. The package provides a fallback, however, if a different class is used.

- **\makebeamerprogram[<options>]**: Generates a beamer-suited lecture program from the metadata in the form

```
\begin[label=beamerprogram<nr>,<bfoptions>]{frame}
\frametitle{<Heading>}
\begin{description}[\lecprogramlistindent]
\item[<short date>] <session full title>
...
(break)
...
\item[<short date>] <session full title>
\end{description}
\end{frame}
```

  <options> can be one of

  - **title=<title>**: Program title (<Heading>)
  - **options=<bfoptions>**: Beamer frame options. Note that key-value beamer frame options need to be embraced, as in **\makebeamerprogram[options={shrink=10}]**

  Note that multiple subsequent frames are generated if the metadata file contains **\SetBeamerFrameBreak** macros. Also note that you can refer to the frames via the automatically generated labels *beamerprogram* (for the first frame), *beamerprogram2* for the second, etc. Finally note that the number of program frames is currently limited to 4. If you need more, you should probably rethink your program structure.

6.3 Generating bibliographies

The package provides macros for easy generation of sectioned bibliographies via biblatex. The macro

- **\makesessionbib**

generates code in the form

```
\section{<session full title>}
\nocite{*}
\printbibliography[keyword=<session keyword>,heading=none]
```

for each session of the lecture which has been linked to a biblatex keyword via the **bibsec** option (see above sec. 5).

So if you specify your session with a keyword via the **bibsec** option, and tag your BibTEX database entries with that keyword, you will get a list of session-specific literature.

6.4 Generating student presentation lists

The package provides macros to produce a list with the name of the students presenting stuff, or generally being particularly involved, in specific sessions.

The names of the presenting students are assigned in the metadata to specific sessions by means of the **presstudents** option of the **\NewSession** command (see above sec. 4.2.4). The macro

\makepreslist

outputs a list that consists of the session dates and titles as well as the names of the presenting students, i. e., it expands to code in the form:

```
\begin{labeling}{\lecprogramlistindent}
\item[<short date>] <session full title>\par
                    <presenting students>
...
\end{labeling}
```

If no presenting students have been assigned to a given session, an em-dash (—) is output instead.

A starred version of the macro outputs a "blank" presentation list that just consists of the session dates and titles with enough space to fill in student names manually. This version might be useful to schedule the presentations of your students in the first session:

\makepreslist*

For beamer presentations, finally, a specific macro

\makebeamerpreslist

is provided (with no starred version). The main difference is the more compact design that makes the list fit better on a slide.

If you want to start your presentation list only from a given session number (e. g., since you do not have any presentations in the first session of the course), you can use

```
\setfirstpressession{<session number>}
```

to set the first session number to be printed on the list. Likewise,

```
\setlastpressession{<session number>}
```

can be used to determine the last session number to be printed on the list.

7 Retrieving session-specific information

In session-specific documents, you also need information specific to the current session. This is done as follows.

7.1 Setting the current session

Use **\ThisSession{<nr>}** to tell the package which session is current. **<nr>** is an integer value, e. g. **\ThisSession{3}** for the 3rd session of the lecture.

7.2 Generating session-specific information for the current session

If you have specified the current session via the macro **\ThisSession**, the following macros output general session-specific information:

- **\sesdate**: Date of the current session
- **\sesshortdate**: Date of the current session, short form (no year)
- **\sestitle**: Main title of the current session
- **\sestitlesep**: Title-subtitle separator of the current session
- **\sessubtitle**: Subtitle of the current session
- **\sesfulltitle**: Full title of the current session (title, separator, subtitle)
- **\sesinstructor**: Current session's instructor's name
- **\sespresstudents**: Students presenting in this session
- **\sesnr**: Number of the current session
- **\sesroom**: Room of the current session

7.3 Generating session-specific information for adjacent sessions

Depending on the value of **\ThisSession**, you can retrieve the following information for sessions that follow or precede the current session by a specific interval unit via the following macros:

- **\AdjSessionTitle[<offset>]**: Outputs the main title of the adjacent session
- **\AdjSessionFullTitle[<offset>]**: Outputs the full title (main title, separator, subtitle) of the adjacent session
- **\AdjSessionShortTitle[<offset>]**: Outputs the short title of the adjacent session
- **\AdjSessionDate[<offset>]**: Outputs the date of the adjacent session
- **\AdjSessionShortDate[<offset>]**: Outputs the short date (no year) of the adjacent session
- **\AdjSessionInstructor[<offset>]**: Outputs the name of the instructor of the adjacent session
- **\AdjSessionPresStudents[<offset>]**: Outputs the name of the students presenting in the adjacent session
- **\AdjSessionRoom[<offset>]**: Outputs the room of the adjacent session

Via the numerical **<offset>** option, you can specify which session, relative to the current one, you want to output. The default is **1**, i. e., **\AdjSessionTitle** without argument outputs the title of the *next* session. Negative values are valid: **\AdjSessionTitle[-1]**, thus, outputs the title of the *previous* session.

7.4 Generating session-specific information for an arbitrary session

Independent of the value of **\ThisSession**, you can retrieve the following information for arbitrary sessions (session number **<nr>**) via the following macros:

- **\SessionTitle{<nr>}**: Outputs the main title of session number **<nr>**
- **\SessionFullTitle{<nr>}**: Outputs the full title (main title, separator, subtitle) of session number **<nr>**
- **\SessionShortTitle{<nr>}**: Outputs the short title of session number **<nr>**
- **\SessionDate{<nr>}**: Outputs the date of session number **<nr>**
- **\SessionShortDate{<nr>}**: Outputs the short date (no year) of session number **<nr>**
- **\SessionInstructor{<nr>}**: Outputs the name of the instructor of session number **<nr>**

- **\SessionPresStudents{<nr>}**: Outputs the name of the students presenting in session number **<nr>**
- **\SessionRoom{<nr>}**: Outputs the room of session number **<nr>**
- **\MakeProgramline{<nr>}**: Outputs a list item line for session number **<nr>** in the form

```
\item[{\bfseries <short date>}] <session full title>
```

8 Customizing the output

8.1 Defining and customizing course types

The following macro allows to define new course types and redefine existing ones:

- **\DefLecType{<key>}{<short form>}{<title>}{<session title>}**

A new type, say summer school, thus, could be defined as follows:

- **\DefLecType{ss}{SS}{Summer School}{Session}**

This could then be set via **\LecType{ss}** or the package option **type=ss**, respectively.

Likewise, you can use the macro to redefine existing styles, e. g. change the "session" string of type **vl** via

- **\DefLecType{vl}{VL}{Vorlesung}{Einheit}**

If you need to change or add types, consider the use of styles (see sec. 8.5).

8.2 Defining and customizing term types

The available term (semester) types can be changed and extended via the macro:

- **\DefSemType{<key>}{<short form>}{<verbose form>}**

Thus, you could add a new term type "spring term" via

- **\DefSemType{st}{ST}{Spring Term}**

8.3 Customizing the date format

If you need to change the date format, redefine the macros **\lecdateformat** (for the long format) and **\lecdateshortformat** (for the short format), using the syntax provided by the datetime package. By default, the two formats are defined as follows:

```
\newcommand*\lecdateformat{%
    \THEDAY.\,\twodigit{\THEMONTH}.\,\THEYEAR}
\newcommand*\lecdateshortformat{%
    \THEDAY.\,\twodigit{\THEMONTH}.}
```

Use `\renewcommand*` to redefine them.

8.4 Misc. customizing

The canceling output is “Session title – keine <Session>” (“keine” is German for “no”). To modify or translate this, redefine the following macro

```
\newcommand*\leccancel{\ -- keine \lectypesession}
```

The default program string used in beamer frames can be changed by redefining

```
\newcommand*\lecprogram{Programm}
```

The indentation used in program lists can be changed by redefining

```
\newcommand*\lecprogramlistindent{\textbf{88.\,88.}}
```

Again, use `\renewcommand*` for the redefinitions.

8.5 Using styles

The most elegant way to do the customizing is to use *lecture planner style* (*.lps) files. These files can be loaded via the **`style=<style>`** package option, where <style> is the file name without *.lps extension.

Currently, the package ships two styles:

1. **`german-default`** simply includes the default strings. This style is meant to be used as a model for new styles.
2. **`cologne`** provides some specifics current at the University of Cologne (*Universität zu Köln*):
    - Additional course types: **`ak`** (AK, *Arbeitskurs*), **`as`** (AS, *Aufbauseminar*), **`es`** (ES, *Einführungsseminar*), **`evl`** (EVL, *Einführungsvorlesung*), **`hs`** (HS, *Hauptseminar*).

If you provide me with other style files, in particular translations to other languages than German (but also other university conventions), I consider them for inclusion to the package.

Part II
Companion packages

9 The jsmembertable package

This section of the manual describes the user interface of the jsmembertable package. Please also refer to the example files included in the bundle.

9.1 Aim of this package

This package provides a way to easily generate tables to track the members of a university course (a table where you, or your students, can fill in names, student IDs and e-mail adresses) and a table to record their presence, should this be a requisite at your institute.

The package somewhat integrates with jslectureplanner and can use the metadata and styles of that package where necessary (see part I of this manual for details on metadata and styles).

Since the package was written for my own needs, the table design might not fit yours. However, with some basic LaTeX knowledge, it should not be too hard to adapt it. Like for jslectureplanner, the default strings are German. However, it is possible to customize and translate them. This is described later in this manual.

9.2 Requirements of jsmembertable

The following packages are required and loaded by jsmembertable:

- datatool (only if the option **`datafile`** is used)
- ifthen
- calc
- longtable
- hhline
- xkeyval

jsmembertable can be used independently of jslectureplanner to some degree. The latter package is only required for specific features (see below) and must be loaded explicitly in that case.

9.3 Loading the package

The package is loaded as usual:

```
\usepackage[<options>]{jsmembertable}
```

Currently, one package option is available:

- **`datafile=<csv-file>`**: Use external student data file (see sec. 9.4).

9.4 Loading external data

Both course member and presence tables can be filled with external student data. To this end, the package option **datafile=<csv-file>** is provided. If it is used, the package tries to access a comma-separated data file. Note that the external file must have the extension *.csv*, and the extension must be omitted in the option, so

```
\usepackage[datafile=mystudents]{jsmembertable}
```

loads the file `mystudents.csv`.

The external csv file must follow the following structure:

```
Prename1, Surname1, StudentID1, Email1
Prename2, Surname2, StudentID2, Email2
...
```

9.5 Generating a course member table

In order to generate a course member table, simply insert

```
\makemembertable{<number of blank rows>}
```

If an external data file has been loaded, **<number of blank rows>** can either be empty or a positive integer value. If no external data file has been loaded, **<number of blank rows>** *must* be a positive integer value.

If no external data file has been loaded, the command generates a table with **<number of blank rows>** blank rows, to which your students can fill in their name, student ID and e-mail address.

If an external data file has been loaded, an appropriate number of rows is generated and the respective student data is filled in. Additionally, **<number of blank rows>** blank rows are *appended* after the filled-in rows (for the late birds that have not registered themselves yet). If the command's argument is empty or **0**, no extra rows are appended.

The number of columns of the member table is always 3, tracking *student id*, *name* and *e-mail address*. If you want to have more or less columns, you need to redefine the **\makemembertable** command. The header strings, though, can easily be changed by redefining the following three macros (using `\renewcommand*`), for instance like this:

```
% Student Name
\renewcommand*\jsmnameheader{\textbf{Student name}}
% Student ID
\renewcommand*\jsmidheader{\textbf{Student ID}}
% Email
\renewcommand*\jsemailheader{\textbf{Email address}}
```

If you use the jslectureplanner package, you can use the jslectureplanner style files in order to store those redefinitions globally. See the `german-default.lps` style file for a model.[6] Of course, you need to load jslectureplanner with the respective **style** option then (*before* jsmembertable). See above section 8.5 for details on styles.

[6] Note that within styles, you must use `\newcommand*` instead of `\renewcommand*`, since the styles are loaded before the jsmembertable package.

9.6 Generating a presence table

Presence tables consist of rows with your students' names and columns for each session of your course where each student can sign and mark her/his presence in the respective session. As with the member tables, the package can fill in student data (the student name in this case) from an external data file if the **datafile** package option is used.

The package provides two kinds of presence tables: A presence table with real session dates and a simpler one just with session numbers ("Session 1" etc.). The first variant requires the jslectureplanner package, since it uses the course metadata in order to generate a matching number of columns with the respective session date in the column headers. The second one also works without jslectureplanner since the session numbers can be simply derived from an internal counter and you can manually set the total number of sessions.

Note that if you use jslectureplanner, you must load the jslectureplanner package and also input the metadata file *before* jsmembertable.

The first presence table variant (with real session dates) is produced with the following macro:

```
\makeprestable{<number of blank rows>}
```

As with the member table, an appropriate number of rows, pre-filled with student data, is generated if an external data file has been loaded. Then, **<number of blank rows>** blank rows are *appended* after the filled-in rows (if **<number of blank rows>** is not empty or **0**). If no external data file has been loaded, simply a table with **<number of blank rows>** blank rows is constructed.

The number of columns/sessions is automatically calculated from the metadata. If you want a different number, you can use the optional argument of the command to override this calculation:

```
\makeprestable[<number of sessions>]{<number of blank rows>}
```

The second kind of presence table (with session numbers instead of session dates) is provided via the starred version of the command:

```
\makeprestable*[<number of sessions>]{<number of blank rows>}
```

Like with the unstarred version, if the jslectureplanner metadata is used/available, the number of columns/sessions is calculated from the metadata. Again, the optional argument can be used to override this calculation. If the jslectureplanner metadata is not used/available, and no session number is given via the optional argument, the table will use a preset number of sessions, namely 15. The starred version uses, like the unstarred variant, the external student data if the **datafile** option has been used.

For courses of 2–16 sessions, the package tries to produce reasonably sized tables that fit the page width (more than 16 sessions are currently not supported). From 7 sessions upwards, two separate tables, each for half a term, are produced. In general, it is assumed that you use landscape page format for presence tables. The tables will be too wide for portrait page format. Also, the tables have been optimized for A4 page size.

The header strings of the presence tables can be adjusted by redefining the following macros (using `\renewcommand*`), for instance like this:

```
% Student Name
\renewcommand*\jsmnameheader{\textbf{Student name}}
% "Session" string for the starred macro version
\renewcommand*\jsmsession{Session}}
% Session header for the starred macro version
% (\thesession outputs the counter value)
\renewcommand*\jsmsessionheader{\textbf{\jsmsession\ \thesession}}
% The advice for students to sign
\renewcommand*\jssigheader{Students' signature}
```

Again, if you use jslectureplanner, you can use the jslectureplanner style files in order to store those redefinitions globally. See the `german-default.lps` style file for a model.[7] Of course, you need to load jslectureplanner with the respective **`style`** option then (*before* jsmembertable). See above section 8.5 for details on styles.

Note, furthermore, that the "Session" string (value of **`\jsmsession`**) is by default linked to jslectureplanner's **`\lectypesession`** value if jslectureplanner is loaded before jsmembertable. So if **`\lectypesession`** fits, you do not have to redefine **`\jsmsession`**.

[7] Note that within styles, you must use `\newcommand*` instead of `\renewcommand*`, since the styles are loaded before the jsmembertable package.

Part III
Appendix

A Acknowledgements

I could not have written the packages of the jslectureplanner bundle without the expertise of the incredible LaTeX community. The packages heavily rely on functions and features provided by other packages (which are listed in sec. 2 and 9.2). I am deeply indepted to the authors of those packages. Furthermore, *stackexchange* proved to be a great source of knowledge which provided me with solutions for specific tasks: jsmembertable heavily draws on code for dynamic table generation that has been provided by Herbert Voss (`http://tex.stackexchange.com/a/7594`); jslectureplanner and jsmembertable use code to test for `\romannumeral` arguments which has been taken from a *stackexchange* post by Enrico Gregorio (`http://tex.stackexchange.com/a/50131`). Thank you, Herbert and Enrico! Finally, Dominik Waßenhoven tested the package and provided me with multiple suggestions, which effectively manifested in major new features such as the jslectureplanner style files. Thanks, Dominik.

B Release History

- 2016/05/11 (v. 1.2):
  - Fix line breaking with empty session titles before course breaks.
  - New commands for getting adjacent session data:
    - **`\AdjSessionFullTitle`**
    - **`\AdjSessionDate`**
    - **`\AdjSessionShortTitle`**
    - **`\AdjSessionTitle`**
    - **`\AdjSessionShortDate`**
    - **`\AdjSessionRoom`**
    - **`\AdjSessionInstructor`**
    - **`\AdjSessionPresStudents`**
- 2015/07/15 (v. 1.1):
  - Include and document the jsmembertable package.
  - Add support for student presentation lists. This introduces the new macros **`\makepreslist`**, **`\makepreslist*`**, **`\makebeamerpreslist`**, **`\setfirstpressession`** and **`\setlastpressession`**.
  - Major revision of the manual.

- 2015/07/07 (v. 1.0.2):
  - Permit **\LecType** and **\LecYear** in the document body again (fix a regression introduced by v. 1.0).
- 2015/02/27 (v. 1.0.1):
  - Properly expand **\leccancel** (fix a regression introduced by v. 1.0).
- 2015/02/27 (v. 1.0):
  - Lecture types and term types are not hardcoded anymore and can be freely modified and extended via **\DefLecType** and **\DefSemType**.
  - Add option to add the instructor's name to both the course (via macro **\LecInstructor** or **instructor** package option) and to specific sessions (via **instructor** option of **\NewSession**).
  - Add option to add names of presenting students to a session (via option **presstudents**).
  - Add option to record the office number (**\SetOfficeNumber** or package option **officenumber**).
  - Fix expansion issue in **options** option of **\makebeamerprogram**.
  - Add option to automatically shift all subsequent sessions by a certain value after every *n*th session (**\SetAutoOffset**).
  - Add **cologne** style file (suggested by Dominik Waßenhoven).
- 2015/02/08 (v. 0.9):
  - Add a way to adjust the interval between two sessions (**interval** package option and **\LecInterval** macro).
  - Add a way to shift session dates from the regular interval (by means of macro **\SetLecOffset**).
  - Add option and macro to set a room for the course as well as a session-specific room.
  - Rename option **weeks** of the **\SetBreaks** macro to **units** (**weeks** still works, but is marked deprecated).
- 2015/01/17 (v. 0.8):
  - Fix corruption of `\today`.
  - Add framework for different styles.
  - Add option and macro to set a room for the course as well as a session-specific room.
  - Remove redundant lecture type **vo** from the documentation.
  - Change **\leccancel** default definition in favour of better translatability.

- 2015/01/04 (v. 0.7):
  - Add macro for resetting `\today`.
  - Minor corrections of the manual.
- 2014/12/12 (v. 0.6 b): Minor corrections of the manual and example files.
- 2014/12/12 (v. 0.6): Initial release to CTAN.