

LaTeX Presentation Packages

Andre Gass, Jörn Spannhacke

28. November 2003

Das Seminar Package

Allgemein

Befehle

Das Prosper Package

Allgemein

Beispiel

Befehle

PDFscreen

Befehle

Beispiel

Nachteile

Beamer

Befehle

Nachteile

Themes

Das Seminar Paket

- ▶ Von Timothy Van Zandt.
- ▶ Einfaches Layout.
- ▶ Kein nach und nach Einblenden von Informationen.
- ▶ Grundlage für das verbreitete Prosper Paket.

Quellcode:

```
\documentstyle{seminar}  
\begin{document}  
\begin{slide}  
Eine Folie im Landscape Format.  
\end{slide}  
\end{document}
```

Commandline:

```
latex %1  
dvips -t landscape %1  
epstopdf %1.ps
```

Eine Folie im Landscape Format.

Quellcode:

```
\documentstyle[portrait]{seminar}  
\begin{document}  
\begin{slide*}  
Eine Folie im Portrait Format.  
\end{slide*}  
\end{document}
```

Commandline:

```
latex %1  
dvips %1  
epstopdf %1.ps
```



Eine Folie im Portrait Format.

Innerhalb einer von `\begin{slide}` und `\end{slide}` kann mit `\newslide` erzwungen werden, dass eine neue Folie beginnt, ansonsten macht L^AT_EX automatisch neue Folien.

Folieninhalt vergrössern oder verkleinern mit `\slidesmag`.

Quellcode:

```
\documentstyle[portrait]{seminar}  
\slidesmag{9}  
\begin{document}  
\begin{slide*}  
Text  
\end{slide*}  
\end{document}
```

`slidesmag{n}` wird als 1.2^n berechnet. n ist eine ganze Zahl zwischen -5 und 9. In diesem Beispiel ist $n = 9$.

```
\slidewidth{breite}
```

gibt die Breite der Folien an.

```
\slideheight{höhe}
```

gibt die Höhe der Folien an.

```
\begin{slide}[breite,höhe]
```

gibt die Grösse jeder Folie einzeln an.

... und viele weitere Möglichkeiten die Grössen zu ändern.

Mit `\centerslidestrue` oder `\centerslidesfalse` kann die vertikale Zentrierung geändert werden.
Default ist `\centerslidestrue`.

Quellcode:

```
\documentstyle[portrait]{seminar}  
\centerslidesfalse  
\begin{document}  
\begin{slide*}  
Dieser Text ist nicht zentriert  
\end{slide*}  
\end{document}
```

Dieser Text ist nicht zentriert

```
\slideframe[options]{style}
```

ermöglicht es, die Ränder der Folien anders zu gestalten.

Mögliche Stile sind:

- ▶ none
- ▶ plain

Weite Stile können z.B. mit

```
\documentstyle [fancybox]{seminar}
```


eingebunden werden:

- ▶ shadow
- ▶ double
- ▶ oval
- ▶ Oval

Beispiel für double.

Quellcode:

```
\documentstyle
  [portrait,fancybox]{seminar}
\slideframewidth 0.5cm
\slideframe{double}
\begin{document}
\begin{slide*}
Folie mit dickem, doppeltem Rand.
\end{slide*}
\end{document}
```



Folie mit dickem, doppeltem Rand.

Das Prosper Paket

- ▶ von Frédéric Goualard und Peter Møller Neergaard
- ▶ Erstellt PowerPoint ähnliche Präsentationen
- ▶ Baut auf dem Seminar Paket auf
- ▶ Viele vorgegebene Themes
- ▶ Deutlich erweiterbare Möglichkeiten gegenüber dem Seminar Paket

Quellcode:

```
\documentclass[pdf]{prosper}  
\title{prosper}  
\subtitle{Vorlage}  
\author{J{\\"o}rn Spannhacke}  
\begin{document}  
\maketitle  
\end{document}
```

prosper

Vorlage

Jörn Spannhacke

Commandline:

```
latex %1  
dvips %1  
epstopdf %1.ps
```

Überschriften werden mit
`\begin{slide}{Titel}`
angegeben.

Quellcode:

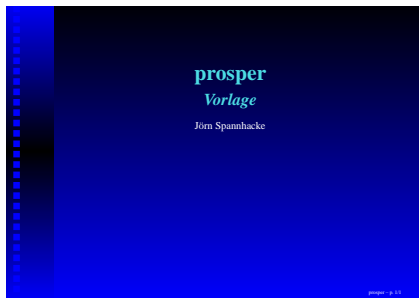
```
\documentclass [pdf] {prosper}  
\begin{document}  
\begin{slide}{Titel}  
Unser Text.  
\end{slide}  
\end{document}
```

Unser Text.

Titel

Quellcode:

```
\documentclass[pdf,azure,  
  slideColor,colorBG,  
  total]{prosper}  
\title{prosper}  
\subtitle{Vorlage}  
\author{J{\\"o}rn Spannhacke}  
\begin{document}  
\maketitle  
\end{document}
```



Quellcode:

```
\documentclass[pdf,gyom,  
  slideColor,colorBG,  
  total]{prosper}  
\title{prosper}  
\subtitle{Vorlage}  
\author{J{\\"o}rn Spannhacke}  
\begin{document}  
\maketitle  
\end{document}
```



Optionen für `\documentclass[...]{prosper}`:

<code>final</code>	Finale Version
<code>draft</code>	Entwicklungsversion
<code>total</code>	Seitennummer und Gesamtseitenzahl
<code>nototal</code>	Seitennummer
<code>slideBW</code>	Eine eingeschränkte Menge an Farben, für B/W
<code>slideColor</code>	Der volle Umfang an Farben wird benutzt
<code>colorBG</code>	Benutzt die Farben des Styles
<code>nocolorBG</code>	Die Hintergrundfarbe ist weiß
<code>noaccumulate</code>	Overlays werden beibehalten
<code>accumulate</code>	Overlays werden unterdrückt

Die `itemstep` Umgebung erzeugt eine Aufzählung, deren Punkte nach und nach eingeblendet werden.

```
\begin{itemstep}
  \item Punkt 1
  \item Punkt 2
  \item ...
\end{itemstep}
```

Die `overlays` Umgebung eignet sich dafür, Informationen nach und nach einzublenden oder auch wieder auszublenden.

```
\overlays{n}{  
  \begin{slide}{Folientitel}  
    ...  
  \end{slide}  
}
```

n gibt die Anzahl der Überlagerungen an.

Weitere Befehle für Overlays:

<code>\fromSlide{p}{Inhalt}</code>	Inhalt erscheint von Folie p bis n
<code>\onlySlide{p}{Inhalt}</code>	Inhalt erscheint nur auf Folie p
<code>\untilSlide{p}{Inhalt}</code>	Inhalt erscheint von Folie 1 bis p
<code>\FromSlide{p}</code>	Nachfolgendes erscheint von Folie p bis n
<code>\OnlySlide{p}</code>	Nachfolgendes erscheint nur auf Folie p
<code>\UntilSlide{p}</code>	Nachfolgendes erscheint von Folie 1 bis p

Mit `\href{URL}{Text}` kann ein Link ins Internet erstellt werden.

Quellcode:

```
\documentclass[pdf]{prosper}
\begin{document}
\begin{slide}{Suchmaschinen}
\large{ Link zu
\href{http://www.google.de}
      {Google}.}
\end{slide}
\end{document}
```

Suchmaschinen

Link zu Google.

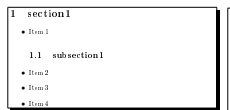
PDFscreen:

- ▶ Autor: C.V. Radhakrishnan
- ▶ Quelle: <http://river-valley.com/>

- ▶ zum Aufbereiten von bereits vorhandenen Dokumenten für Präsentationen
- ▶ daher nur ein Style-File und keine Dokumentenklasse
- ▶ Als Klasse wird z.B. `article` oder `book` verwendet
- ▶ kann zwei Versionen in einem `.tex` File verwalten

- ▶ `\usepackage[options]{pdfscreen}` zum Benutzen von PDFscreen. Optionen:
 - ▶ `article` teilt PDFscreen mit, dass als Klasse `Article` benutzt wird
 - ▶ `print` oder `screen` um entweder die Präsentationsversion oder die Druckversion zu erzeugen
 - ▶ `panelleft` oder `panelright` um ein Navigationpanel zu erzeugen
- ▶ `\begin{screen} ... \end{screen}` für Text, der nur in der `screen`-Version sichtbar ist
- ▶ `\begin{print} ... \end{print}` analog zu `\begin{screen}...`
- ▶ `\screensize{12cm}{16cm}` um die Seitengröße festzulegen

- ▶ `\marginwidth{1cm}{1cm}{1cm}{1cm}`
um die Größe des Randes festzulegen
- ▶ `\begin{slide}... \end{slide}` um
Folien zu machen. Nachteil: die Folien
werden in der Druckversion im
gleichen Format eingebunden wie in
der Präsentation



```
\documentclass[a4paper]{article}
\usepackage[screen,article]{pdfscreen}
\begin{screen}
\screensize{12cm}{16cm}
\marginsize{1cm}{1cm}{1cm}{1cm}
\end{screen}
\begin{document}
\maketitle \newpage
\tableofcontents \newpage
...
\end{document}
```

Inhaltsverzeichnis

1	section1	2
1.1	subsection1	2
1.2	subsection2	3
2	section2	4
2.1	subsection1	5
2.2	subsection2	6

Home Page

Print

Title Page

Contents



Page 1 of 6

Go Back

Full Screen

Close

Quit

- ▶ funktioniert nur mit PDF_Latex
- ▶ bei Komplizierteren Dokumenten ist es nicht mehr einfach die `screen` und die `print` Version in einer Datei zu verwalten
- ▶ Overlays lassen sich nur schlecht umsetzen

Beamer:

- ▶ Autor: Till Tantau
- ▶ Quelle: <http://latex-beamer.sourceforge.net>

- ▶ Beamer ist recht einfach zu erlernen
- ▶ kann ebenso wie PDFscreen neben der Präsentationsversion noch eine Druckversion verwalten
- ▶ bietet einfach mittels `\usepackage{}` zu benutzende Themes

- ▶ `\frame{...text...}` für ein Frame

- ▶ `\frame{...text...}` für ein Frame
- ▶ `<n>` Für Overlays, wobei `n` eine Aufzählung der Slides ist, in denen der entsprechende Punkt erscheinen soll. `\item<1-3,5>` würde den Punkt auf den Slides 1 bis 3 und 5 erscheinen lassen.

- ▶ `\frame{...text...}` für ein Frame
- ▶ `<n>` Für Overlays, wobei `n` eine Aufzählung der Slides ist, in denen der entsprechende Punkt erscheinen soll. `\item<1-3,5>` würde den Punkt auf den Slides 1 bis 3 und 5 erscheinen lassen.
- ▶ Der Code für diese Seite:
`\item<1-> ... \item<2-> ... \item<3-> ...`

- ▶ Zum einbinden von Grafiken wird pgf verwendet, ein Paket das ebenfalls von Till Tantau entwickelt wird
- ▶ PGF ist sehr mächtig
- ▶ `\pgfimage{dateiname}` zum einbinden von Bildern, die Grösse kann man als Option mitgeben
- ▶ `\plainframe` erzeugt ein komplett weisses Frame, z.B. um Screenshots zu zeigen

- ▶ `\note` um Notizen in die Präsentation einzubinden
- ▶ `\presentation`, `\common` und `\article` um Teile entweder nur in der Präsentation, in beiden oder nur in der Druckversion erscheinen zu lassen

- ▶ Um mehrere Spalten nebeneinander zu bekommen, kann man `\begin{columns}\begin{column}{3cm}...\end{column}\end{columns}` benutzen
- ▶ in Kombination mit `\pgfimage{}` kann man so Beispiele direkt neben den dazugehörigen Text setzen, wie das in Section 1 und 2 zu sehen war

- ▶ Wie bei PDFscreen ist es schwierig, zwei Versionen in einer Datei zu haben
- ▶ `verbatim`s lassen sich nur etwas kompliziert in Frames einbauen. Man muss dazu entweder `\frame[all:1]{...text...}` angeben, wodurch das Frame nur ein slide hat, oder man muss die `verbatim`s vor dem Frame umständlich mit `\defverb\name!...!` oder `\defverbatim\name{\begin{verbatim}...\end{verbatim}}` definieren und hinterher mit `\name` einbinden

L^AT_EX Presentation Packages

Andre Gass, Jörn Spannhacke

28. November 2003

L^AT_EX Presentation Packages

Andre Gass, Jörn Spannhacke

28. November 2003