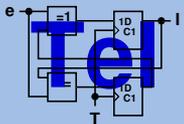


# Erstellen von Präsentationen mit der $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Klasse `iteslides`

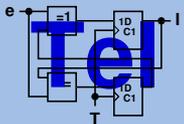
Martin Zabel

`zabel@ite.inf.tu-dresden.de`

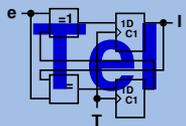
Technische Universität Dresden  
Institut für Technische Informatik



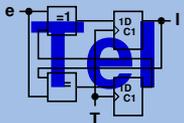
Abschnitt	Folie
1 Motivation	3
2 Dokumentklasse <code>iteslides</code>	5
3 <code>prosper</code> -Konvertierung	21
4 Hinweise zur Anwendung	25
5 Zusammenfassung und Ausblick	35



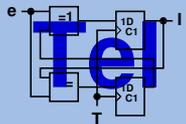
# 1 Motivation



- ◆ Ersatz für bisher verwendete  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Klasse `prosper`
  - Fehlerhafte `prosper.cls` fügt ungewollt Leerzeichen ein.  
⇒ Textbereich nicht auf Folie zentriert.
  - Alle Folien sind zusätzlich in einer `minipage` eingebettet.  
⇒ unbrauchbare Standardeinstellungen für die Textformatierung
  - Viele weitere Fehler
- ◆ `iteslides` setzt direkt auf `seminar` auf
- ◆ Unterstützung für Gliederung mit automatischer Nummerierung, Kopfzeile und Inhaltsverzeichnis
- ◆ Verwendung sowohl für Vorträge als auch Vorlesungsskript

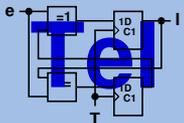


## 2 Dokumentklasse iteslides



- ◆ **Einbindung:** `\documentclass [<Optionen>] {iteslides}`
- ◆ **Klassensoptionen:**
  - `notes`: **Folien und Notizen**
  - `slidesonly`: **nur Folien**
  - `colored/uncolored`: **farbig/schwarzweiß**
  - `fillbg/nofillbg`: **Hintergrund füllen/nicht füllen**

Weitere folgen später ...

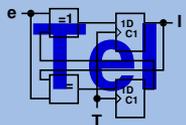


## ◆ Erforderliche Angaben im Vorspann

- `\title{<Titel des Vortrags>}`
- `\author{<Autor/Vortragender>}`
- `\institution{<Einrichtung/Institution>}`

## ◆ Optionale Angaben im Vorspann

- `\subtitle{<Untertitel>}`
- `\email{<E-Mail-Adresse>}`
- `\date{<Datum>}` (Standard: aktueller Tag)

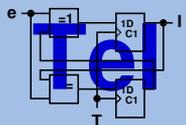


- ◆ Neue Umgebungen `slide` und `slide*`
- ◆ Erforderlicher Parameter: Titel der Folie
- ◆ Erzeugen einer Folie im Querformat (Normalform) ...

```
\begin{slide}{<Titel der Folie>}  
<Folieninhalt>  
\end{slide}
```

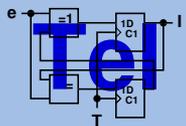
- ◆ ... und im Hochformat (Sternform)

```
\begin{slide*}{<Titel der Folie>}  
<Folieninhalt>  
\end{slide*}
```



- ◆ Neue Umgebung `note`
- ◆ Erzeugen einer Notizseite (immer im Hochformat)

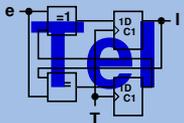
```
\begin{note}  
<Notizen>  
\end{note}
```



Dies ist ein Beispiel für eine Notiz.

## Kompilierung einer Datei `test.tex`

1. Kopieren der Datei `iteslides.cls` und `tei.eps` in das Verzeichnis von `test.tex` oder Anpassen des Suchpfades von  $\text{\LaTeX}$ .
2.  $\text{\LaTeX}$ -Durchlauf (mehrmals, um alle Referenzen aufzulösen)  
`latex test && latex test.tex && latex test.tex`
3. Erstellen der Ausgabedatei und Anzeige:
  - ◆ PS-Datei: `dvips test && gv test.ps`
  - ◆ PDF-Datei: `dvipdf test && acroread test.pdf`
  - ◆ Verwendung von X $\text{\DVI}$  nicht möglich, da Feature für gedrehten Text nicht implementiert



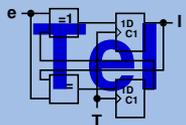
- ◆ Erzeugung der Titelseite, wie gehabt, mit `\maketitle`
- ◆ Dieser Befehl muss jedoch
  - entweder innerhalb einer `slide/slide*`-Umgebung

```
\begin{slide}{}  
\maketitle  
\end{slide}
```

- oder innerhalb einer `note`-Umgebung

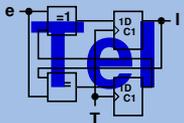
```
\begin{note}  
\maketitle  
\end{note}
```

stehen.



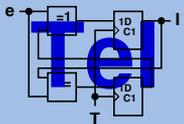
Die Folien werden standardmäßig um den Faktor  $(1.2)^4 \approx 2.1$  skaliert. Dies betrifft alle Elemente wie Text, Grafiken und Maßangaben.

- ◆ Grundschriftart der Folien ist Helvetica in 10 pt  
⇒ Schriftgröße von 21 pt auf Papier/Monitor
- ◆ 1 cm im Quelltext ⇒ 2.1 cm in der Ausgabe, im Gegensatz dazu keine Anpassung von `1ex` und `1em` nötig
- ◆ Alternative: `1\semcm` ⇒ 1 cm in der Ausgabe
- ◆ Grafiken, die in realer Größe erstellt wurden, müssen vorher skaliert werden:
  - EPS: `\includegraphics[scale=.48]{...}`
  - PSTEX: beim Export in XFIG *Magnification* von 48% einstellen

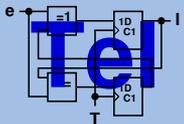


# Ausrichtung Folieninhalt

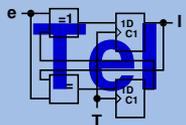
- ◆ Folieninhalt oben bündig (Standard): `\centerslidesfalse`
- ◆ Folieninhalt vertikal zentriert: `\centerslidestrue`
- ◆ Gewünschter Befehl
  - muss vor `\begin{slide}` stehen
  - und gilt von dort an für alle folgenden Folien, bis sein Gegenstück aufgerufen wird



Diese Folie ist vertikal zentriert.



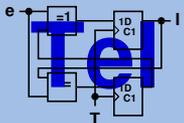
- ◆ Höhe des Textbereiches: 150 mm (reale Größe)
- ◆ Breite des Textbereiches: 222 mm (reale Größe)
- ◆ Ändern der Breite des Textbereiches mit `\slidewidth<Breite>`
  - entweder im  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Vorspann
  - oder vor `\begin{slide}` (gilt für alle folgenden Folien)
  - Beispiel: `\slidewidth257mm`
- ◆ Anzeige des zur Verfügung stehenden Textbereiches mit der Klassenoption `drawslidearea`



Zwecks bildschirmfüllender Darstellung der PDF-Datei auf einem Beamer mit dem Seitenverhältniss 4:3 wird die A4-Seite links und rechts beschnitten.

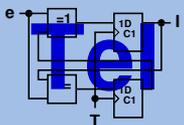
- ◆ Anzeige der Schnittlinien mit der Klassenoption `drawpdfcrop`
- ◆ Sinnvoll nur dann, wenn während der Erstellung des Foliensatzes die PS-Datei zur Kontrolle verwendet wird. Denn hier wird immer die gesamte A4-Seite angezeigt.

Weiteres Feature der PDF-Datei: automatische Generierung von Lesezeichen (Bookmarks) für jede Folie, inklusive Gliederung.



Zwei (disjunkte) Arten der Gliederung sind möglich:

- ◆ Standard: Gliederung mit `\section`, `\subsection`, ...
  - Aufruf der Befehle innerhalb der `slide`-Umgebung (Folieninhalt)
  - Kopfzeile der Folie enthält automatisch Abschnitt (oben links) und Unterabschnitt (oben rechts)
  - Automatisches Inhaltsverzeichnis
  - *Formatierung der Überschriften im Textbereich noch offen!*
- ◆ Klassenoption `slidesection`:
  - Neuer Befehl: `\slidesection{<Überschrift>}`
  - Aufruf vor `\begin{slide}`
  - Kopfzeile der Folie enthält automatisch den Abschnittsnamen (oben rechts)



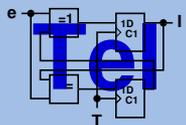
## ◆ Festlegung eines Overlay-Satzes mit

```
\overlays{<Anzahl>}{  
  \begin{slide}{<Folientitel>}  
    ...  
  \end{slide}  
}
```

## ◆ Bedingte Darstellung eines Textstückes mit

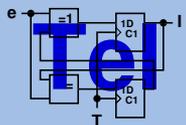
```
\untilSlide{<Overlay-Nummer>}{<Text>}  
\onlySlide{<Overlay-Nummer>}{<Text>}  
\fromSlide{<Overlay-Nummer>}{<Text>}
```

- Zählung der Overlays beginnt mit 1.
- Normal- und Sternform verfügbar.



- ◆ Normalform: Wird das Textstück auf dem aktuellem Overlay nicht angezeigt, dann wird stattdessen ein entsprechender Freiraum eingefügt.
- ◆ Sternform: Das Textstück wird im Gegensatz zu oben vollständig ignoriert.

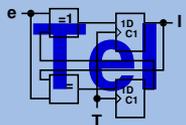
In diesem                    ist ein Freiraum.  
Und der gleiche , nochmal ohne Freiraum.



- ◆ Normalform: Wird das Textstück auf dem aktuellem Overlay nicht angezeigt, dann wird stattdessen ein entsprechender Freiraum eingefügt.
- ◆ Sternform: Das Textstück wird im Gegensatz zu oben vollständig ignoriert.

In diesem Lückentext ist ein Freiraum.

Und der gleiche Lückentext, nochmal ohne Freiraum.



## ◆ Eigene Gestaltung

- der Titelseite (Folie):

```
\renewcommand{\slidemaketitle}{...}
```

- der Kopfzeile:

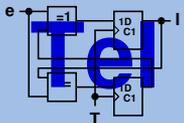
```
\renewcommand{\slideheader}{...}
```

- der Fußzeile:

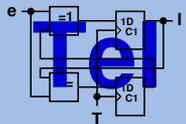
```
\renewcommand{\slidefooter}{...}
```

- bereitgestellte Makros:

```
\thetitle, \theauthor, \theinstitution,  
\thesubtitle, \theemail, \thedate,  
\theslidetitle, \theslidesection
```



## 3 prosper-Konvertierung



## ◆ Dokumentklasse

## ● Präsentation:

```
\documentclass[slidesection]{iteslides}
```

● Druck: `\documentclass[slidesection, uncolored, nofillbg]{iteslides}`

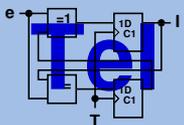
## ◆ \Logo nicht mehr unterstützt, Gestaltung der Fußzeile mittels

```
\renewcommand{\slidefooter}{...}
```

◆ Die Titelfolie muss in einer `slide`-Umgebung stehen.

```
\begin{slide}{}  
  \maketitle
```

```
\end{slide}
```

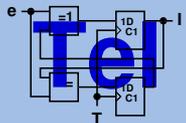


- ◆ Zwecks besserer Lesbarkeit, größere Abstände von Absätzen und Textzeilen
- ◆ Wiederherstellung der prosper-Eigenschaften mittels

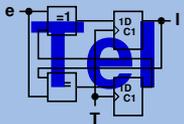
```
\renewcommand{\slideparskip}{0cm}
```

```
\renewcommand{\slidestretch}{1}
```

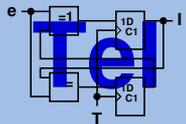
im  $\text{\LaTeX}$ -Vorspann



- ◆ Farbauswahl des `pstricks`-Pakets funktioniert nicht zusammen mit dem `color`-Paket und `\item`.
- ◆ Workaround: `\item \rule{0cm}{0cm} \darkred Text`
- ◆ Besser: Kommandos des `color`-Pakets
  - Farbe definieren  
`\definecolor{darkred}{rgb}{.7, .2, .2}`
  - Farbe verwenden  
`{\color{darkred} Text}`  
`\textcolor{darkred}{Text}`
  - Alternativ: direkte Angabe  
`{\color[rgb]{.7, .2, .2} Text}`  
`\textcolor[rgb]{.7, .2, .2}{Text}`



## 4 Hinweise zur Anwendung



## Something's wrong--perhaps a missing \item

### Fehlermeldung:

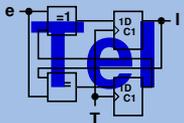
„Something's wrong--perhaps a missing \item“ im Zusammenhang mit `\fromslide`, `\untilslide` und `\onlyslide`

**Ursache:** Die Argumente zu den o. g. Overlay-Befehlen enthalten  $\text{\LaTeX}$ -Kommandos die einen Absatz erzeugen.

**Lösung:** Es muss explizit eine vertikale Box mittels `\parbox` oder `\begin{minipage}\end{minipage}` angelegt werden, die dann  $\text{\LaTeX}$ -Kommandos für Absätze enthalten darf. Als Breite für die Box eignet sich `\linewidth`.

**Beispiel:** `\fromslide{2}{\parbox{\linewidth}{%  
\begin{flushright} rechtsbündig \end{flushright}%  
}}`

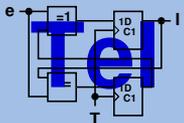
**Tip:** Möchte man nur etwas links-, rechtsbündig oder zentriert in einer Zeile ausrichten, kann man auch die Befehle `\leftline{...}`, `\rightline{...}`, `\centerline{...}` ohne Zusatzbox verwenden.



Familie	Sans Serif	<code>\sffamily</code>	<code>\textsf{&lt;Text&gt;}</code>
	Roman	<code>\rmfamily</code>	<code>\textrm{&lt;Text&gt;}</code>
	Typewriter	<code>\ttfamily</code>	<code>\texttt{&lt;Text&gt;}</code>
Serie	Medium	<code>\mdseries</code>	<code>\textmd{&lt;Text&gt;}</code>
	<b>Fett</b>	<code>\bfseries</code>	<code>\textbf{&lt;Text&gt;}</code>
Form	Aufrecht	<code>\upshape</code>	<code>\textup{&lt;Text&gt;}</code>
	<i>Geneigt</i>	<code>\slshape</code>	<code>\textsl{&lt;Text&gt;}</code>
	<i>Kursiv</i>	<code>\itshape</code>	<code>\textit{&lt;Text&gt;}</code>
	SMALL CAPS	<code>\scshape</code>	<code>\textsc{&lt;Text&gt;}</code>

- ◆ Obige Befehle verändern immer nur ein Attribut
- ◆ Hingegen stellen die Befehle `\bf`, `\it`, ... immer zuerst die Grundschriftart ein bevor sie das gewünschte Attribut verändern.

`\rm\bf Text` ⇒ **Text**



# Ausrichtung von Text (1)

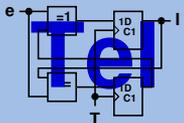
Ausrichtung	Umgebung für Absatz	Zeile
linksbündig	<code>flushleft</code>	<code>\leftline{&lt;Text&gt;}</code>
rechtsbündig	<code>flushright</code>	<code>\rightline{&lt;Text&gt;}</code>
zentriert	<code>center</code>	<code>\centerline{&lt;Text&gt;}</code>

- ◆ Umgebungen in 2. Spalte erzeugen immer einen Absatz und fügen daher einen Absatzabstand (`\parskip`) vor und nach der Umgebung ein.

Zentrierter Text, der genau unter dem Anstrich steht.

- ◆ Befehle in 3. Spalte richten jeweils eine Textzeile (Parameter) aus, die ggf. in den Rand hineinragt.

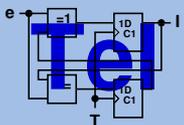
Dies ist eine zentrierte Textzeile, die auf beiden Seiten in den Rand hineinragt.  
Es wird kein neuer Absatz erzeugt.



## Manuelle Korrektur des Layouts

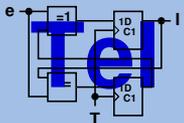
- ◆ **Horizontaler Abstand:** `\hspace{1cm}`, `\hspace*{1cm}`
- ◆ **Vertikaler Abstand:** `\vspace{1cm}`,  
`\vspace*{1cm}`, `<Text/Tabellenzeile>\[1cm]`
- ◆ **Normalform:** Befehl wird am Seiten-/Folienanfang bzw. Zeilenanfang ignoriert.
- ◆ **Vertikaler Abstand**
  - immer zzgl. normalem Abstand der Grundlinien
  - wird erst am Zeilenende eingefügt

Der Befehl `\vspace{1\semcm}` steht hier, die Textzeile wird aber zunächst beendet, bevor der Abstand eingefügt wird.



# Informationen zum Textbereich

- ◆ Höhe des Textbereiches: `\textheight`
  - ◆ Breite des gesamten Textbereiches über alle Spalten  
`\textwidth`
- 
- ◆ Breite der aktuellen Textspalte: `\columnwidth`
  - ◆ Breite der aktuellen Textzeile (Absatz): `\linewidth`
- 



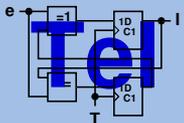
Mit den folgenden Befehlen kann innerhalb einer Textzeile ein eigenständiger Absatz in einer Absatzbox gesetzt werden:

- ◆ `\parbox [<Pos>] {<Breite>} {<Text>}`, wobei der Parameter `Text` selbst keine Befehle für Absätze enthalten darf
- ◆ Befehle für Absätze sind nur in `minipage` erlaubt, die jedoch alle Formatierungseinstellungen auf Standardwerte zurückstellt

```
\begin{minipage} [<Pos>] {<Breite>}  
<Text mit Absätzen>  
\end{minipage}
```

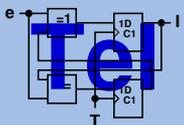
- ◆ Tabellen sind ebenfalls Absatzboxen

```
\begin{tabular} [<Pos>] {<Format>}  
<Tabellenzeilen>  
\end{tabular}
```



Der Positionierungsparameter  $\langle pos \rangle$  kann entfallen oder die Werte  $t$ , oder  $b$  annehmen:

- ◆ (laufende Zeile) Ohne Angabe wird der Absatztext zentriert zur laufenden Zeile ausgerichtet.
- ◆ (laufende Zeile) Mit dem Wert  $t$  wird die oberste Zeile des Absatztextes auf die laufende Zeile ausgerichtet.  
Mit dem Wert  $b$  wird die unterste Zeile des Absatztextes auf die laufende Zeile ausgerichtet.
- ◆ (laufende Zeile) Mit dem Wert  $c$  wird der Absatztext zentriert zur laufenden Zeile ausgerichtet.



# `\centerslidesfalse` (1)

Bei `\centerslidesfalse` ist zu beachten:

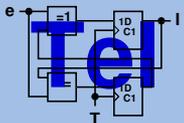
1. Der Abstand der Grundlinie der ersten Textzeile von der Kopfzeile (`\topskip`) ist auf `1.2\baselineskip` der Standardschriftgröße (`\normalsize`) eingestellt.
2. Beginnt Seite mit `minipage`, `tabular` oder `\parbox`, müssen diese oben ausgerichtet werden, damit dies auch für die erste Zeile der Absatzbox gilt

```
\begin{minipage}[t]{...}
```

```
\begin{tabular}[t]{...}
```

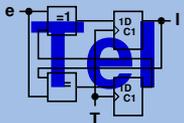
```
\begin{parbox}[t]{...}
```

- ◆ Wird dies nicht beachtet, dann klebt der Inhalt der Absatzbox am Balken der Kopfzeile.

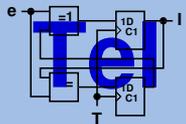


## Grafik

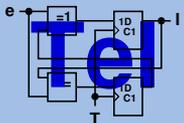
- ◆ Grundlinie von Grafiken (EPS, PSTEX) befindet sich am unterem Rand  $\hat{=}$  Positionierung mit `b`
- Konflikt mit `\centerslidesfalse`, wenn Grafik am Folienanfang
- ◆ Lösung: leere Zeile mit Abstandsangabe
  - Beispiel: `~\ [-1ex]`
  - Abstand ist zusätzlich zu `\topskip!`
  - Ebenfalls anwendbar für Absatzboxen



## 5 Zusammenfassung und Ausblick



- ◆ Korrektes Layout der Seite mit sinnvollen Einstellungen für die Formatierung des Textes
- ◆ Unterstützung der Gliederung mit `\section, ...`  
→ Einstellungen für Formatierung noch offen
- ◆ Unterstützung von Overlays
- ◆ Eigene Gestaltung von Kopf-/Fußzeile etc. bei Bedarf möglich
- ◆ Zusätzliche Features für PDF-Datei wie Beschneidung auf 4:3 und Lesezeichen (mit Gliederung)



- ◆ Wünsche, Anregungen?
- ◆ Einbau neuer Features unter Berücksichtigung der Abwärtskompatibilität
- ◆ Ist Skalierung Nachteil oder Vorteil?

Mögliche weitere Themen:

- ◆ Schriftarten und  $\text{\LaTeX}$

