

# L<sup>A</sup>T<sub>E</sub>X-Beamer-Kurs

## Beamer

Martin Glatz  
ig-mathe



**1. Paket**

**2. Basics**

**3. Effekte – Overlays**

**4. Beamer-Spezialitäten**

**5. Design**

**6. eigenes Design**



# Leseanleitung

Programmcode:

```
abgesetzter Programmcode  
(Leerzeichen notfalls entfernen)  
%Kommentare im Code
```

Normaler Text sowie Programmcode im Text normaler Text

## Achtung! Gefahr?

Besonders Wichtiges oder Gefährliches ist so gekennzeichnet! Also eventuelle Unstimmigkeiten, Kompatibilitätsprobleme, ...

Solche Hinterlegungen dienen nur der optischen/inhaltlichen Strukturierung und haben ansonsten keine besondere Bedeutung.

## Starke Gliederung

Falls notwendig und sinnvoll, gibt es auch Gliederungen mit Überschriften ...



# beamer-Paket

```
\documentclass[xcolor=x11names,10pt]{beamer}
```

- ⊙ Das Beamer-Paket erzeugt pdf-Seiten als Folien (frames bzw. slides).
- ⊙ Präsentation = Durchklicken durch Folien.
- ⊙ Jede Animation ist eine eigene Seite (slide)
- ⊙ praktisch sämtliche L<sup>A</sup>T<sub>E</sub>X-Kommandos nutzbar: Mathematik, Grafiken, Tabellen ...
- ⊙ übliche Zusatzpakete können eingebunden werden
- ⊙ Optionen `handout` ignoriert Aufdeck-Effekte (jede frame wird nur einmal gedruckt)
- ⊙ Standardfoliengröße: 128mm breit, 96mm hoch (passend für übliche Schriftgrößen: 10, 11 pt)
- ⊙ alternativ als Klassen-Option:  
`aspectratio=169` für 16:9  
`aspectratio=1610` für 16:10



# Grundbefehle

Erstellen einer Folie:

```
%\begin{frame}[<+>] [Folienoptionen]  
%{Folientitel}{Folienuntertitel}  
%... Folieninhalt ...  
\end{frame}
```

Optionen:

- ⊙ `plain` leere Folie
- ⊙ `fragile` für `\verb+...+ Befehl` bzw. `lstlisting` Umgebung
- ⊙ `c, t, b` vertikale Ausrichtung des Folieninhalts  
(zentriert, oben, unten)
- ⊙ `label=....` gibt der Folie einen Namen  
(evtl. späterer Aufruf mit `\againframe{...}`)
- ⊙ `squeeze` für kleinere vertikale Abstände
- ⊙ `[<+>]` alle Aufzählungspunkte und Blöcke etc  
werden item-weise aufgedeckt
- ⊙ `\titlepage` als Folieninhalt erzeugt Titelseite



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Blöcke und Auszeichnungen

beamer-Zusatzbefehle

Text mit `\alert{...}` **ausgezeichnet**

Text mit `\structure{...}` **ausgezeichnet**

Inhalt

Paket

Basics

frames

Blöcke

Spalten

Effekte –

Overlays

Beamer-

Spezialitäten

Design

eigenes Design

## alertblock-Umgebung-Titel

```
\begin{alertblock}{Titel}
...
\end{alertblock}
```

## block-Umgebung-Titel

```
\begin{block}{Titel}
...
\end{block}
```

## exampleblock-Umgebung-Titel

```
\begin{exampleblock}{Titel}
...
\end{exampleblock}
```



# Spalten

```
\begin{columns}[totalwidth=gesamtbreite]  
\column[Option]{breite1}  
... Text in Spalte 1 ...  
  
\column[Option]{breite2}  
... Text in Spalte 2 ...  
\end{columns}
```

- ⊙ alternativ:

```
\begin{column}{breite}[Option] ... \end{column}
```

innerhalb der columns-Umgebung

- ⊙ Optionen:

- b
- c
- t
- T

ausrichten an Grundlinie  
zentriert ausrichten,  
an erster Zeile ausrichten  
oben ausrichten

- ⊙ Für Breite auf voriger Folie (inkl. block):  
gesamtbreite = 1.025\textwidth



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-

Spezialitäten

Design

eigenes Design

# Effekte - Basics

- schrittweises Aufdecken von Listen etc möglich (Overlay)
- schrittweises Aufdecken/Zusammenbauen von Tabellen
- Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc
- viele Befehle mit verschiedenen Optionen
- nur eine frame nötig, die nötigen slides werden automatisch erstellt.
- Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$

- `\setbeamercovered{true}` → Wirkung: Text wird eingekreuzt
- `\setbeamercovered{false}` → Wirkung: Text wird nicht eingekreuzt
- `\setbeamercovered{transparent}` → Wirkung: Text wird eingekreuzt, aber nicht sichtbar
- `\setbeamercovered{opacity=0.5}` → Wirkung: Text wird eingekreuzt, aber nicht sichtbar
- `\setbeamercovered{opacity=0.5,opacity=0.5}` → Wirkung: Text wird eingekreuzt, aber nicht sichtbar
- `\setbeamercovered{opacity=0.5,opacity=0.5}` → Wirkung: Text wird eingekreuzt, aber nicht sichtbar

```
\setbeamercovered{transparent}
```

»nichteingebledeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle  
Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Effekte - Basics

## ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)

⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen

⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc

⊙ viele Befehle mit verschiedenen Optionen

⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.

⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$

- $n$ : Wirkung bei slide  $n$
- $n, m$ : Wirkung bei slide  $n$  und  $m$
- $n-m$ : Wirkung von slide  $n$  bis  $m$
- $n-$ : Wirkung ab slide  $n$
- $-n$ : Wirkung bis slide  $n$
- $\langle 1, 3-4, 7 \rangle$  Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-

Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ **schrittweises Aufdecken/Zusammenbauen von Tabellen**
- ⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc
- ⊙ viele Befehle mit verschiedenen Optionen
- ⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.
- ⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$

- $n$ : Wirkung bei slide  $n$
- $n, m$ : Wirkung bei slide  $n$  und  $m$
- $n-m$ : Wirkung von slide  $n$  bis  $m$
- $n-$ : Wirkung ab slide  $n$
- $-n$ : Wirkung bis slide  $n$
- $\langle 1, 3-4, 7 \rangle$ : Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen
- ⊙ **Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc**
- ⊙ viele Befehle mit verschiedenen Optionen
- ⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.
- ⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$ 
  - $n$ : Wirkung bei slide  $n$
  - $n, m$ : Wirkung bei slide  $n$  und  $m$
  - $n-m$ : Wirkung von slide  $n$  bis  $m$
  - $n-$ : Wirkung ab slide  $n$
  - $-n$ : Wirkung bis slide  $n$
  - $\langle 1, 3-4, 7 \rangle$ : Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen
- ⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc

- ⊙ **viele Befehle mit verschiedenen Optionen**

- ⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.

- ⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$

- $n$ : Wirkung bei slide  $n$
- $n, m$ : Wirkung bei slide  $n$  und  $m$
- $n-m$ : Wirkung von slide  $n$  bis  $m$
- $n-$ : Wirkung ab slide  $n$
- $-n$ : Wirkung bis slide  $n$
- $\langle 1, 3-4, 7 \rangle$ : Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-

Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen
- ⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc
- ⊙ viele Befehle mit verschiedenen Optionen
- ⊙ **nur eine frame nötig, die nötigen slides werden automatisch erstellt.**
- ⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle \dots \rangle$ 
  - $n$ : Wirkung bei slide  $n$
  - $n,m$ : Wirkung bei slide  $n$  und  $m$
  - $n-m$ : Wirkung von slide  $n$  bis  $m$
  - $n-$ : Wirkung ab slide  $n$
  - $-n$ : Wirkung bis slide  $n$
  - $\langle 1,3-4,7 \rangle$ : Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen
- ⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc
- ⊙ viele Befehle mit verschiedenen Optionen
- ⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.
- ⊙ **Aufruf der jeweiligen slides per Zähler möglich**  $\langle . . . \rangle$ 
  - $n$ : Wirkung bei slide  $n$
  - $n, m$ : Wirkung bei slide  $n$  und  $m$
  - $n-m$ : Wirkung von slide  $n$  bis  $m$
  - $n-$ : Wirkung ab slide  $n$
  - $-n$ : Wirkung bis slide  $n$
  - $\langle 1, 3-4, 7 \rangle$  Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Effekte - Basics

- ⊙ schrittweises Aufdecken von Listen etc möglich (Overlay)
- ⊙ schrittweises Aufdecken/Zusammenbauen von Tabellen
- ⊙ Hinzufügen von Objekten, Farbänderungen (von Text), von Grafiken etc
- ⊙ viele Befehle mit verschiedenen Optionen
- ⊙ nur eine frame nötig, die nötigen slides werden automatisch erstellt.
- ⊙ Aufruf der jeweiligen slides per Zähler möglich  $\langle . . . \rangle$ 
  - $n$ : Wirkung bei slide  $n$
  - $n, m$ : Wirkung bei slide  $n$  und  $m$
  - $n-m$ : Wirkung von slide  $n$  bis  $m$
  - $n-$ : Wirkung ab slide  $n$
  - $-n$ : Wirkung bis slide  $n$
  - $\langle 1, 3-4, 7 \rangle$  Mehrfachwirkung mit Beistrich trennen

```
\setbeamercovered{transparent}
```

»nichteingblendeter« Text wird hellgrau



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Items aufdecken

alle items nacheinander aufdecken

```
\begin{itemize}[<+>]  
\item ...  
\item ...  
\end{itemize}
```

nur gewünschte aufdecken

```
\begin{itemize}  
\item ...  
\item<3-> ...  
\item<4-> ...  
\item<2-> ...  
\end{itemize}
```

- ⊙ `\item<n- | alert@m>` deckt dieses Item ab  $n$  auf und markiert es bei  $m$  mit dem `\alert`-Befehl  
alternativ Formatierungen: `structure`, `uncover`, ...
- ⊙ Analoges für `description` und `enumerate`



## Items aufdecken II

⊙ `\begin{itemize}[<+>]`

globales schrittweises Aufdecken

⊙ `\begin{itemize}[<+ | alert@+]`

Kennzeichnung mit alert beim schrittweisen Aufdecken

⊙ `\begin{itemize}[<+ | structure@+]`

Kennzeichnung mit alert beim schrittweisen Aufdecken

Problem: bei diesen Optionen werden auch Unter-items derart aufgedeckt. Lösung:

```
\begin{itemize}[<+ | alert@+]  
\item ...  
\item ...  
\item<3-> \alert<3>{ ... }  
    \begin{itemize}  
    ...  
    \end{itemize}  
\item ...  
\end{itemize}
```



# Textauszeichnung auf slides

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen  
Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

- ⊙ `\textbf<...>{fetter Text}` **fett** machen
- ⊙ `\textit<...>{...}` *kursiv* machen
- ⊙ `\textsl<...>{...}` *schräg* machen
- ⊙ `\textrm<...>{...}` roman machen
- ⊙ `\textsf<...>{...}` sanserif machen
- ⊙ `... { \color<...>{green} grüner Text } ...` einfärben
- ⊙ `\alert<...>{...}` Auszeichnung mit **alert**
- ⊙ `\structure<...>{...}` Auszeichnung mit **structure**



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen  
Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Erscheinen und Verschwinden lassen

○ `\only<...>{Inhalt}`

Inhalt erscheint nur bei den angegebenen slides  
(entstehende Textlücke wird aufgefüllt)

○ `\visible<...>{...}`

Inhalt erscheint nur bei den angegebenen slides  
(Textlücke bleibt bestehen, echt unsichtbar)

○ `\uncover<...>{...}`

Inhalt wird bei den angegebenen slides aufgedeckt

○ `\invisible<...>{..}`

Inhalt wird bei den angegebenen slides unsichtbar

○ `\alt<...>{Inhalt a}{Inhalt b}`

Inhalt a erscheint bei den angegebenen slides, Inhalt b  
ansonsten.

○ `\temporal<n>{vorher}{Inhalt bei slide n}{danach}`

Es gibt diese Befehle auch als Umgebungen (Dok, 9.4)



# Überdrucken

## Möglichkeit 1

```
\begin{overprint}  
\onslide<1> Inhalt 1  
\onslide<2-> Inhalt 2  
\end{overprint}
```

- ⊙ (Platz richtet sich nach dem größeren Bedarf)
- ⊙ der `\onslide<...>`-Befehl arbeitet wie der `\item`-Befehl

## Möglichkeit 2

```
\begin{overlayarea}{breite}{hoehe}  
...  
\end{overlayarea}
```

- ⊙ stellt festen Bereich zur Verfügung
- ⊙ innen `\only<...>{...}` verwenden



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Tipps

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

Design

eigenes Design

- weniger ist mehr!
- Itemweises Aufdecken sollte wohlüberlegt sein, da es sonst zu ständigem Geklicke ausartet
- Aufdeck- und Ersetz-Befehle können sinnvoll sein
- Auszeichnungen mit Farben/structure/alert durchaus sinnvoll, um ohne Laser-Pointer die Aufmerksamkeit auf einzelne Ausdrücke zu bringen!



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

Design

eigenes Design

# Tipps

- ⦿ weniger ist mehr!
- ⦿ Itemweises Aufdecken sollte wohlüberlegt sein, da es sonst zu ständigem Geklicke ausartet
- ⦿ Aufdeck- und Ersetz-Befehle können sinnvoll sein
- ⦿ Auszeichnungen mit **Farben/structure/alert** durchaus sinnvoll, um ohne Laser-Pointer die Aufmerksamkeit auf einzelne Ausdrücke zu bringen!



# Beispiele

$$f(x) \approx f(0) + f'(0) \cdot x + f''(0) \cdot x^2$$

```
f(x) \alt<2>{=}{\approx}  
f(0) + f'(0) \cdot x + f''(0) \cdot x^2  
\visible<2->{+ R(0,x)}
```

$$f(x) \approx f(0) + f'(0) \cdot x + f''(0) \cdot x^2 + R(0, x)$$

```
f(x) \alt<2>{=}{\approx}  
f(0) + f'(0) \cdot x + f''(0) \cdot x^2  
\uncover<2->{+ R(0,x)}
```



# Beispiele

$$f(x) = f(0) + f'(0) \cdot x + f''(0) \cdot x^2 + R(0, x)$$

```
f(x) \alt<2>{=}{\approx}  
f(0) + f'(0) \cdot x + f''(0) \cdot x^2  
\visible<2->{+ R(0,x)}
```

$$f(x) = f(0) + f'(0) \cdot x + f''(0) \cdot x^2 + R(0, x)$$

```
f(x) \alt<2>{=}{\approx}  
f(0) + f'(0) \cdot x + f''(0) \cdot x^2  
\uncover<2->{+ R(0,x)}
```



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

**Tabellen**

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Tabellen aufdecken

Zeilenweises Aufdecken:

`\pause` vor dem jeweiligen Zeilenumbruch in der Tabelle angeben:  
`... & ... &... \pause \\`

Spaltenweises aufdecken:

`\onslide<n-> ...`  
zeigt sämtliches Nachfolgendes ab Slide  $n$  bis zum nächsten Frame.

```
\begin{tabular}{  
l<{\onslide<2->} c<{\onslide<3->} c  
}
```

- ⦿ erste Spalte ist immer da
- ⦿ zweite Spalte ab slide 2
- ⦿ dritte Spalte ab slide 3



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Grafiken aufdecken

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

**Grafiken**

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

```
\includegraphics<...>[opt. Parameter]{grafik1}
```



# semiverbatim-Umgebung

- ermöglicht schrittweises Aufdecken von Code
- `\`, `{`, `}` behalten die übliche L<sup>A</sup>T<sub>E</sub>X-Bedeutung
- `\\`, `\{`, `\}` verwenden, um die Zeichen `\`, `{`, `}` zu erhalten

```
# include <iostream>
using namespace std;
int main(void)
{
    cout << 'Hello World' << endl;
}
```

```
\begin{semiverbatim}
\uncover<1->{# include <iostream>}
\uncover<2->{\alert<2>{using namespace std;}}
\uncover<3->{int main(void)}
\uncover<3->{\}
\uncover<4->{ cout << 'Hello World' << endl;}
\uncover<3->{\}
\end{semiverbatim}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Und das Ergebnis ...

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

Design

eigenes Design

```
# include <iostream>
using namespace std;
int main(void)
{
    cout << 'Hello World' << endl;
}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Und das Ergebnis ...

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

Design

eigenes Design

```
# include <iostream>
using namespace std;
int main(void)
{
    cout << 'Hello World' << endl;
}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

eigenes Design

# Und das Ergebnis ...

```
# include <iostream>
using namespace std;
int main(void)
{
    cout << 'Hello World' << endl;
}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Und das Ergebnis ...

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-

Spezialitäten

Design

eigenes Design

```
# include <iostream>
using namespace std;
int main(void)
{
    cout << 'Hello World' << endl;
}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Visuelle Effekte - Ausblick

Inhalt

Paket

Basics

Effekte –  
Overlays

Grundlagen

Listen

Overlay-Befehle

Tipps/Beispiele

Tabellen

Grafiken

Weiteres

Beamer-  
Spezialitäten

Design

Design

eigenes Design

- man kann eigene Befehle Overlay-fähig machen (Dok 9.6)
- man kann »Folienübergänge« machen (schnell wieder vergessen!)
- der `\framezoom`-Befehl erlaubt das Vergrößern eines vordefinierten Bereiches (Dok 11.3)
- ...



# Multimedia

```
\usepackage{multimedia}
```

- ⊙ Einbindung von Multimedia-Dateien  
(Sounds, Filme als mpg oder avi) (Dok 14)
- ⊙ keine Einbettung!

```
\movie[...] {xxx} {video.mpg}
```

- ⊙ xxx ist dabei der Platzhalter, der als Startbild erscheint  
(für ein Bild: Grafik einfügen; Text hineinschreiben, ...)
- ⊙ Optionen:
  - poster (als Startbild wird das erste Filmbild genommen)
  - showcontrols (Steuerleiste des Players,...)
  - height=0.3\textwidth, width=0.4\textwidth  
für Größenvorgaben des Filmfensters
  - externalviewer ruft externen Player auf
  - ...



# Inhaltsverzeichnis-Einstellungen

## `\tableofcontents [...]`

- `currentsection`                      alles bis auf die momentane section (samt subsections) wird hellgrau gemacht
- `firstsection=Nr`                      Deklariert, welche section als section 1 gezählt wird
- `hideallsubsections`                      subsections werden versteckt
- `hideothersubsections`                      nur subsections der aktuellen section werden angezeigt
- `part=Nr`                                  zeigt nur den gewünschten part-Inhalt an
- `pausesections`                              nach jeder section wird pausiert
- `sections={...}`                              nur gewünschte sections werden angezeigt  
    `sections={2-4}`                              zeigt section 2 bis 4 (weitere Einstellungen möglich!)
- ...



# Links & Buttons

- ⊙ Ziele (Hypertarget) und Buttons können erzeugt werden

- ⊙ `\hypertarget{name}{...Inhalt...}`

deklariert den `...Inhalt...` als Ziel.  
`name` ist der interne Name dieses Ziels

- ⊙ `\hyperlink{name}{...Link...}`

Klickt man auf `...Link...` so, springt man zum Ziel

- ⊙ `\hyperlink{name}{\beamerbutton{Zum Ziel}}`

erzeugt einen Knopf mit dem Text »Zum Ziel«.

- ⊙ vordefinierte Buttons: `\beamergotobutton`,  
`\beamerskipbutton` und `\beamerreturnbutton`

- ⊙ falls frames ein label haben:

`\hyperlink{folienbennung}{...Inhalt...}`

- ⊙ falls frames ein label und mehrere slides haben:

`\hyperlink{folienbennung<2>}{...Inhalt...}`

führt zu slide 2



# Def-Satz-Beweis

- ⊙ beamer bietet vordefinierte amsthm-Umgebungen:
  - Definition, Satz, Beweis
  - Problem, Lösung
  - Folgerung, Lemma, Fakt
  - Beispiel und Beispiele
- ⊙ (Es gibt auch englische Umgebungen)
- ⊙ Das Design richtet sich nach den block/exampleblock-Umgebungen

```
\begin{Satz}<2->[Hauptsatz]  
Ein Hauptsatz kennzeichnet sich ...  
\end{Satz}
```

## Satz 1 (Hauptsatz)

*Ein Hauptsatz kennzeichnet sich dadurch, dass das Prädikat an zweiter Stelle steht*



# Def-Satz-Beweis

- ⊙ beamer bietet vordefinierte amsthm-Umgebungen:
  - Definition, Satz, Beweis
  - Problem, Lösung
  - Folgerung, Lemma, Fakt
  - Beispiel und Beispiele
- ⊙ (Es gibt auch englische Umgebungen)
- ⊙ Das Design richtet sich nach den block/exampleblock-Umgebungen

```
\begin{Satz}<2->[Hauptsatz]
```

Ein Hauptsatz kennzeichnet sich ...

```
\end{Satz}
```

## Satz 1 (Hauptsatz)

*Ein Hauptsatz kennzeichnet sich dadurch, dass das Prädikat an zweiter Stelle steht*



# Aussehen

## Definition 2 (opt. Titel)

und hier der Text samt einer Formel:

$$f'(x) = x^2 - \ln(x)$$

## Beispiel 3 (opt. Titel)

und hier der Text samt einer Formel:

$$f'(x) = x^2 - \ln(x)$$

## Titel 4 (Meine Umgebung)

*und hier der Text samt einer Formel:*

$$f'(x) = x^2 - \ln(x)$$



# Def Satz Beweis umdefinieren

- `\setbeamertemplate{qed symbol}{neues Symbol}`  
ändert das qed-Symbol
- `\qedhere` für manuelles qed-Symbol
- `\setbeamertemplate{theorems}[numbered]`  
macht gemeinsame Nummerierung

- `\setbeamertemplate{theorems}[normal font]`

Sätze werden nicht mehr kursiv geschrieben

- `\newtheorem{name}[theorem]{Titel}`

definiert eine eigene Umgebung. Zählung erfolgt mit `theorem` wie die üblichen `theorem`-Umgebungen



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Presentation-Themes

Vollständige Designvorlagen (Farben, Schrifteinstellungen, Navigationsleisten, ...)

```
\usetheme{name}
```

Eine Auswahl:

- Ohne Navigation:  
default (Standardtheme), Rochester, boxes (dazu später mehr), sowie: Bergen, Bordilla, Madrid, AnnArbor, CambridgeUS, Pittsburgh
- Baumartige Navigation in der Kopfzeile  
Antibes, JuanLesPins, Montpellier
- Seitliche Inhaltsangabe:  
Berkeley, PaloAlto, Goettingen, Marburg, Hannover
- Mini Navigation (frames als Punkte in der Kopfzeile)  
Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt, Singapore, Szeged
- Gliederung:  
Copenhagen, Luebeck, Malmoe, Warsaw

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik

Schriften

eigenes Design



IATeX-Kurs  
Beamer

# Foliendesign I – outer Theme

Äußeres Erscheinungsbild: Kopf- bzw. Fußzeilen, sidebars  
(Seitenleiste), etwaige Navigationselemente,...

```
\useoutertheme [optionen] {name}
```

- ⊙ `default` Standardlayout ohne Kopf- und Fußzeilen
- ⊙ `infolines` Minimale Kopf- und Fußzeile (Titel, ...)
- ⊙ `miniframes` Folien pro section in Kopfzeile als Punkte sichtbar, Optionen z. B.
  - `subsection=false` unterdrückt Subsections,
  - `footline=authortitle` zeigt Titel, AutorIn in der Fußzeile
- ⊙ `split` geteilte Kopf- und Fußzeile
- ⊙ `shadow` wie `split` mit Farbverlauf
- ⊙ `smoothtree` Gliederung in der Kopfzeile mit weichen Übergängen, Option `hook`

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik  
Schriften

eigenes Design



# Foliendesign I – outer Theme sidebar

```
\useoutertheme[optionen]{sidebar}
```

Ein (anklickbares Inhaltsverzeichnis) ist als seitlicher Balken zu sehen (vgl. diese Präsentation)

Optionen:

- `height=...` Höhe des Titels  
(Opt: Logo verschwindet und Titel wird in Seite integriert)
- `hideothersubsections` nur aktuelle Subsection wird gezeigt
- `left` Balken ist auf linker Seite der Folie
- `right` Balken ist auf rechter Seite der Folie
- `width=...` Breite des Balkens



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

# Foliendesign II – inner Themes

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt  
outer

inner

Farben

Mathematik  
Schriften

eigenes Design

```
\useinnertheme{name}
```

Inner Themes für z. B. itemize-Umgebungen, block-Umgebungen, Theorem und proof-Umgebungen, ...

- default Dreiecke als Aufzählungspunkte
- rectangles Quadrate für Aufzählungen und Nummerierungen
- circles Kreise als Aufzählungszeichen
- rounded Kreise und abgerundete Ecken



# Farb-themes

Voreingestellte Farbthemen:

```
\usecolortheme[option]{name}
```

Gesamt-themes:

- ⊙ crane gelb-orange, dunkelblaue structure-Schrift
- ⊙ dove unaufdringlich, grau. Schwarze Schrift
- ⊙ seagull ähnlich wie dove, etwas auffälligere Blöcke
- ⊙ beaver unaufdringlich gräulich, mit roter bzw. blauer Schrift

Outer-themes (für Kopf- und Fußzeile,...) (zusätzlich/alternativ)

- ⊙ whale dunkelblau
- ⊙ seahorse blassblau

Inner-themes (für blocks,...) (zusätzlich/alternativ)

- ⊙ lily löscht sämtliche Farben für Blöcke ...
- ⊙ orchid weiß auf dunkel (Blöcke bekommen structure-Farbe)
- ⊙ rose blasse, unaufdringliche Blockfarben (Pastelltöne)



# Template-System für Farben

- ⊙ Farbänderungen für einzelne Bestandteile greifen meist auf templates zurück.
- ⊙ Meistens Vordergrund (Schrift, fg) und Hintergrund (bg)
- ⊙ Zum Teil komplizierte Abhängigkeiten gegeben (parents,...)

Beispiel:

```
\setbeamercolor{frametitle}{fg=yellow, bg=red}
```

erzeugt auf den Folien gelbe Schrift und blauen Hintergrund beim Folientitel. Vor/Außerhalb der gewünschten Folie bzw. in der Präambel einfügen.

```
\setbeamercolor{frametitle}{fg=yellow}
```

verändert nur die Schriftfarbe des Folientitels



# Farb-Templates (Auswahl)

- ⊙ `structure` (für `structure`-Befehl (Achtung: Abhängigkeiten!))
- ⊙ `alerted text` für den `alert`-Befehl

## Blöcke-Umgebungen-Farben:

- ⊙ `block title` (für Schrift-, Hintergrund-Farbe des Block-Titels)
- ⊙ `block body` (für Schrift-, Hintergrund-Farbe des Block-Körpers)
- ⊙ `block title example`
- ⊙ `block body example`
- ⊙ `block title alerted`
- ⊙ `block body alerted`

## Folienhintergrund:

- ⊙ `normal text` für Hintergrundfarbe der Folie
- ⊙ Farbverlauf von `farbe1` zu `farbe2`

```
\setbeamertemplate{background canvas}  
[vertical shading] [top=farbe1,bottom=farbe2]
```

- ⊙ pdf-Seite als Hintergrund (Paket `pdfpages` nötig):

```
\setbeamercolor{background canvas}{bg=}  
\includepdf{pdfseite.pdf}
```



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt  
outer  
inner

Farben

Mathematik  
Schriften

eigenes Design

# Beamercolorbox

Für Boxen ohne Titelzeile:

```
\begin{beamercolorbox}[optionen]{beamerfarbe}  
...  
\end{beamercolorbox}
```

Als beamerfarbe kann entweder ein bereits existierendes Template genommen werden, oder ein neues (z. B. in der Präambel) definiert werden, z. B.

```
\setbeamercolor{bunt}{fg=red,bg=yellow}
```

Der Inhalt einer Box ohne Optionen und mit  
manuellem Zeilenumbruch



## Optionen für beamercolorbox (Auswahl)

- ⦿ `wd` Boxbreite (z. B. `0.5\textwidth`)
- ⦿ `left, right, center` Text: links-, rechtsbündig oder zentriert
- ⦿ `sep` Randbreite um den Text

```
\begin{beamercolorbox}
[wd=0.5\textwidth, sep=0.5\baselineskip, center]
{subsection in sidebar shaded}
Box mit Optionen ...
\end{beamercolorbox}
```

Box mit Optionen ...

```
\newenvironment{block1}
{\begin{beamercolorbox}
[colsep*=.75ex]{farbe1}}
{\end{beamercolorbox}}
```

erzeugt eine Umgebung `block1` wie die `block`-Umgebung (ohne Titel) mit der Farbe `farbe1`



# beamerboxesrounded

```
\begin{beamerboxesrounded}[optionen]{Kopf}  
...  
\end{beamerboxesrounded}
```

erzeugt eine abgerundete Box. Kopf wird als Titel genommen, kann aber auch leerbleiben:

Mit Text im Kopf

restlicher Text ...

Optionen:

- upper (Farbe für Kopfteil)
- lower (Farbe für Text-Teil)
- shadow=true (erzeugt Schatten)
- width=... (legt Breite fest)

Wird Breite nicht angegeben, so ist die Box `\textwidth` breit, wobei die Box je 4pt auf der Seite übersteht.)

Ohne Text im Kopf.



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt  
outer  
inner

Farben

Mathematik  
Schriften

eigenes Design

# Farben für sidebar-theme

Grundthema:

```
\usecolortheme{sidebartab}
```

Farben für seitlichen Balken:

```
sidebar left
```

Farben für aktuelle Einträge

```
section in sidebar  
subsection in sidebar
```

Farben für nicht aktuelle Einträge

```
section in sidebar shaded  
subsection in sidebar shaded
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik

Schriften

eigenes Design

# Die Definition einer theorem-Umgebung

Am Beginn wird dieses Template aufgerufen:

```
\setbeamertemplate{theorem begin}{%  
\begin{\inserttheoremblockenv}  
{% das kommt alles in den block-Titel  
  \inserttheoremheadfont  
  \inserttheoremname  
  \inserttheoremnumber  
  \ifx\inserttheoremaddition\@empty\else\  
  (\inserttheoremaddition)\fi%  
  \inserttheorempunctuation  
}% Ende des block-Titels  
}
```

Am wird dieses Template aufgerufen:

```
\setbeamertemplate{theorem end}{  
\end{\inserttheoremblockenv}  
}
```



L<sup>A</sup>T<sub>E</sub>X-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik

Schriften

eigenes Design

# theorem-Umgebungen ohne Blöcke

Anfang:

```
\setbeamertemplate{theorem begin}{%  
\normalfont % normale Schrift des Textes  
{%  
\inserttheoremheadfont  
\inserttheoremname  
\inserttheoremnumber  
\ifx\inserttheoremaddition\@empty\else\  
(\inserttheoremaddition)\fi%  
\inserttheorempunctuation  
\newline % Zeilenumbruch, falls gewollt  
}%  
}
```

Ende:

```
\setbeamertemplate{theorem end}{{}
```

verschiedenfärbige Hintergründe nun mit z. B. Colorboxen möglich



# fonts – Schriftgestaltung I

Schriftpakete wie üblich einbinden. Zusätzliche font-themes:

```
\usefonttheme [optionen] {bezeichnung}
```

bezeichnungen:

- ⊙ default ohne Auszeichnungen, alles sansserif.
- ⊙ professionalfonts Einbindung von professionellen Schriften
- ⊙ serif macht serifige Schrift.

Optionen zur Einstellung serif

- ⊙ stillsansserifmath (math. Text bleibt sansserif),
- ⊙ stillsansserifsmall (kleiner Text bleibt sansserif),
- ⊙ stillsansseriflarge (besonders großer Text bleibt sanserif),
- ⊙ stillsansseriftext (normale Textgröße bleibt sannserif)
- ⊙ onlymath (nur math. Text wird serifig).



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik

Schriften

eigenes Design

# fonts – Schriftgestaltung II

Schriftpakete wie üblich einbinden. Zusätzliche font-themes (auch in Kombination zu den vorigen)

```
\usefonttheme [optionen] {bezeichnung}
```

weitere bezeichnungen:

- `structurebold` Titel etc werden fett.
- `structureitalicserif` Titel etc werden kursiv-serif.
- `structuresmallcapsserif` Titel werden Kapitälchen.

Verfügbare Optionen:

- `onlysmall` (Auswirkung nur auf kleinste Textgrößen),
- `onlylarge` (Auswirkung nur auf große Schrift).



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt

outer

inner

Farben

Mathematik

Schriften

eigenes Design

# Schriften: weitere Einstellungen

Auch mehrere font-themes können nacheinander/gleichzeitig genutzt werden:

Mit eigens geladener Mathematik-Schrift

```
\usepackage{eulervm}  
\usefonttheme[onlysmall]{structurebold}  
\usefonttheme[onlymath]{serif}
```

Mit Palatino-Schrift:

```
\usefonttheme{professionalfonts}  
\usepackage[sc]{mathpazo}  
\linespread{1.05}  
\usefonttheme{serif}  
\usefonttheme{structurebold}
```



# font-template-System für Schriften

Schrifteinstellungen (z. B. Folientitel, ...) laufen über Templates

```
\setbeamerfont{templatename}{einstellungen}  
\setbeamerfont*{templatename}{einstellungen}
```

Der \*Befehl löscht die alten Schrifteinstellungen, bevor die neuen einstellungen hinzugefügt werden, z. B.

```
\setbeamerfont{frametitle}{  
size=\large, series=\bfseries}
```

Mögliche einstellungen sind u.a.:

- ⦿ size (für Größen, z. B. tiny ...)
- ⦿ series (für Stärke: \bfseries oder \mdseries)
- ⦿ shape (für die Form: kursiv: \itshape, schräg: \slshape, Kapitälchen: \scshape, aufrecht: \upshape.)
- ⦿ family (für Schriftfamilie: serif: rmfamily, sansserif: \sffamily, Typewriter: \ttfamily)



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

gesamt  
outer

inner

Farben

Mathematik

Schriften

eigenes Design

# Nützliche font-templates

font-template benutzen:

`\usebeamerfont{name}` vor dem gewünschten Text (falls nötig)

Keine vollständige Liste, sondern als Ausgangsbasis zum suchen in der beamer-Dokumentation gedacht.

- ◉ `frametitle` Überschrift der Folie
- ◉ `structure` Für structure-Auszeichnungen
- ◉ `alerted text` Für Text mit `\alert{...}`
- ◉ `block title` Schrifteinstellung für den Titel eines Blocks.  
(vgl. Farb-templates für Block-Umgebungen)
- ◉ `block body` Schrifteinstellung für den Körper eines Blocks.
- ◉ `title page` Für Titelfolie
- ◉ `section in toc` Sectioneintrag im Inhaltsverzeichnis
- ◉ `item` Textgestaltung der Aufzählungssymbole/Einträge  
(z.B description-Umgebung)



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

eigenes Design

boxes

Kopf & Fuß

Listen

Ränder

# beamertheme: boxes

einfaches Hinzufügen von Einträgen in die Kopf- oder Fußzeile

```
\usetheme[headheight=..., footheight=...]{boxes}
```

```
\addheadbox{beamercolor}{Kopfzeilen-Boxinhalt}
```

```
\addfootbox{beamercolor}{Fuzeilen-Boxinhalt}
```

- ⊙ `headheight=...` Höhe der Kopfzeilenbox
- ⊙ `footheight=...` Höhe der Fußzeilenbox

- ⊙ Jede neu geaddete Box wird hinzugefügt,
- ⊙ die Box-Breiten grundsätzlich symmetrisch aufgeteilt.
- ⊙ Wird `beamercolor` weggelassen, so erhält der Inhalt die Standardfarben.



# Beispiel für ein Fußzeile mit boxes

```
\usepackage[footheight=4mm]{boxes}
```

Autor(in), Abschnitt, Unterabschnitt werden in der Fußzeile gezeigt.

```
\addfootbox{frametitle}{\tiny \quad \insertauthor}
```

```
\addfootbox{}{\tiny \quad \insertsection{}  
\quad \insertsubsection{}}
```

Zusätzlich (Navigationselemente, frame-Zahlen):

```
%Entfernen der Standard-Navigationssymbole
```

```
\setbeamertemplate{navigation symbols}{}  
  
\addfootbox{}{\hfill \quad  
\insertframenavigationssymbol \quad  
\insertsectionnavigationssymbol  
\quad \insertframenummer{} / \inserttotalframenummer  
\quad}
```



# Kopf- Fußzeilen selbst basteln

siehe beamer-Dokumentation Abschnitt 8.2.1. (ab Seite 64)

```
\setbeamertemplate{headline}{Kopfzeileninhalt}  
\setbeamertemplate{footline}{Fusszeileninhalt}
```

Beispiel:

```
\setbeamertemplate{headline}  
{\leavevmode  
  \begin{beamercolorbox}  
    [width=0.7\paperwidth, ht=3mm]{farbe1}  
  \end{beamercolorbox}%  
  \begin{beamercolorbox}  
    [width=0.3\paperwidth, ht=3mm]{farbe2}  
  \end{beamercolorbox}  
}
```

- ⦿ erzeugt als Kopfzeile eine zweifarbige Zierleiste:  
linker Teil ist mit `farbe1` gefärbt, rechter Teil mit `farbe2`.
- ⦿ (Das Kommando `\leavevmode` verlässt den vertikalen Modus, weswegen Objekte grundsätzlich innerhalb einer Zeile erzeugt werden.)



# itemize und enumerate-Einstellungen I

Grundsätzliche Templates (Voreinstellung):

```
\setbeamertemplate{enumerate items}[aussehen]
```

aussehen:

- ⦿ square (item-Symbole sind Quadrate)
- ⦿ triangle (item-Symbole sind Dreiecke)
- ⦿ circle (item-Symbole sind rund)
- ⦿ ball (item-Symbole sind rund mit 3d-Effekt)

Zusätzliche templates (Voreinstellung, font, color) für itemize item bzw. itemize subitem bzw. itemize subsubitem

Beispiele:

```
\setbeamertemplate{itemize subitem}[square]  
\setbeamercolor{itemize subitem}[fg=red]
```

für rote Quadrate als Aufzählungszeichen in zweiter Ebene.  
(Analoges für die enumerate-Umgebung: z. B. enumerate item statt itemize)



# itemize und enumerate-Einstellungen II

## Schriften ändern

Ändern der Schriften/Farben auf den einzelnen Ebenen:  
(font bzw. color - templates)

- ⊙ `itemize/enumerate body` ( $\geq$  1. Ebene)
- ⊙ `itemize/enumerate subbody` ( $\geq$  2. Ebene)
- ⊙ `itemize/enumerate subsubbody` (3. Ebene)

```
\setbeamerfont{itemize/enumerate body}  
{family = \rmfamily}
```

```
\setbeamercolor{item projected}{fg=...,bg=...}
```

verändert Schriftfarbe und Hintergrundfarbe bei den  
Aufzählungen »Zahl auf Quadrat«



# itemize und enumerate-Einstellungen III

## Symbole ändern

mühsam:

händisch das jeweilige itemize-Symbol machen:

`\item[Symbol]` statt `\item` verwenden.

### Sinnvoller:

eigenen Befehl machen, der sämtliche Funktionsweisen behält.

```
\newcommand{\itemn}{\item[eigenes Symbol]}
```

Aufruf mit `\itemn`

Auch Grafiken einfügbar, evtl. mit dem `\mbox{...}`-Befehl



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

eigenes Design

boxes

Kopf & Fuß

Listen

Ränder

# itemize und enumerate-Einstellungen III

## Symbole ändern

Enumerate-Aufzählungszeichen:

```
\begin{enumerate}[Option]  
...
```

Beispiele für Option:

- ⊙ (i)                   macht römische Kleinbuchstaben in Klammer
- ⊙ A)                    macht Großbuchstaben,
- ⊙ 1.)                   macht Zahlen mit Punkt und Klammer etc.



IATeX-Kurs  
Beamer

Inhalt

Paket

Basics

Effekte –  
Overlays

Beamer-  
Spezialitäten

Design

eigenes Design

boxes

Kopf & Fuß

Listen

Ränder

# Layout und Ränder-Einstellungen

beliebige Größen ändern mit

```
\setbeamersize{name=wert}
```

Beispiele:

- ⦿ `text margin left`                    Abstand von linker sidebar zum Text
- ⦿ `text margin right`                    Abstand rechts zum Text
- ⦿ ...

```
\setbeamersize{text margin left = 2cm}
```