

# Quellcode

Robert Hilbrich

hilbrich@informatik.hu-berlin.de

# Warum mit LaTeX?

- ❑ Lesbarkeit
- ❑ Übersichtlichkeit
- ❑ geringer Formatierungsaufwand
- ❑ geringer Wartungsaufwand
- ❑ Konsistenz (z.B. Diplomarbeit)
  
- ❑ Es sieht einfach gut aus. 😊

# Bordmittel

- ❑ `\verb{ <Quellcode> }`
- ❑ `\begin{verbatim}`  
`<Quellcode>`  
`\end{verbatim}`
- ❑ "fixed" Font
- ❑ Keine Interpretation der Zeichen
- ❑ Reicht das?

# de-tex-faq.pdf

## 7.3.4 Wie kann ich Programm-Code in mein LATEX – Dokument einbinden?

Neben der Möglichkeit, kurze Abschnitte 'verbatim' einzubinden, gibt es verschiedene Tools, die automatisch LATEX-Code mit den entsprechenden Syntaxhervorhebungen erzeugen:

- ❑ **listings** - Reine LATEX-Lösung; unterstützt viele Programmiersprachen; in neueren Versionen kann, 'listings' zusätzlich einige der Möglichkeiten des Pakets 'fancyvrb' nutzen.
- ❑ **lgrind** - C-Programm und LATEX-Style, für LaTeX; unterstützt sehr viele Programmiersprachen.

# Schwerpunkte des Vortrages

## □ Inhalt:

- "listings" als LaTeX Lösung
- "Igrind" als kombinierte Präprozessor Lösung

## □ Form:

- Keine Referenz
- Fallbeispiele, Szenarien

```
\usepackage{ listings }
```

<ftp://ftp.dante.de/tex-archive/macros/LaTeX/contrib/supported/listings>

# Installation

- Download (<ftp.dante.de>)
  - README lesen
  - LaTeX listings.ins
  - texhash
- 
- sehr gute Dokumentation:  
listings.dvi

# Aufgabenstellung

## □ Aufgabe 1:

- Schreiben Sie eine Funktion, die das **Maximum** zweier Zahlen berechnet.
- Erläutern Sie verwendete Variablen.

## □ Aufgabe 2:

- Schreiben Sie eine Funktion, die das **Minimum** zweier Zahlen berechnet.
- Erläutern Sie verwendete Variablen.

# Lösung (/home/user/loesung.c)

```
/* Aufgabe 1 */
int maximum (int a, int b)
{
    printf("Berechne das Maximum ... denke ... denke ... denke ... denke ... \n");

    if (a >= b) return a;
    else return b;
}

/* Aufgabe 2 */
int minimum (int a, int b)
{
    printf("Berechne das Minimum ... denke ... denke ... denke ... denke ... \n");

    if (a < b) return a;
    else return b;
}
```

# Umsetzung in LaTeX

## Stufe 1

### □ Zielstellung:

- Setzen des Quelltextes außerhalb des Fließtextes
- Automatischer Umbruch überlanger Zeilen
- Benutzung von Syntaxhighlighting
- Betonung der Variablen **a** und **b**

# Umsetzung in LaTeX

## Stufe 1

```
\documentclass{ article}
\usepackage{ listings}
\begin{ document}
\lstset{ language=C}

\section{ 1. Aufgabe}
\begin{ lstlisting}[emph={ a, b}, emphstyle=\underbar, breaklines=true]

/*   Aufgabe 1   */
int maximum (int a, int b)
{
    printf("Berechne das Maximum ... denke ... denke ... denke ... denke ... \n");

    if (a >= b) return a;
    else return b;
}
\end{ lstlisting}

[... ]
\end{ document}
```

# Umsetzung in LaTeX

## Stufe 1

```
\documentclass{article}
\usepackage{listings}
\begin{document}
\lstset{language=C}

\section{1. Aufgabe}
\begin{lstlisting}[emph={a, b}, emphstyle=\underbar, breaklines=true]

/*  Aufgabe 1  */
int maximum (int a, int b)
{
    printf("Berechne das Maximum ... denke ... denke ... denke ... denke ... \n");

    if (a >= b) return a;
    else return b;
}
\end{lstlisting}

[...]
```

# Ausgabe von LaTeX

## Stufe 1

Ausgabe

# Fazit

## Stufe 1

- `\lstset{options}` setzt globale Optionen
  - **language** - Programmiersprache
  - **emph** – Liste von hervorzuhebenden Schlüsselwörtern
  - **emphstyle** – Art der Hervorhebung von Schlüsselwörtern
  - **breaklines** – Bricht überlange Zeilen um
  
- Umgebung **lstlisting** setzt Quelltext außerhalb vom Text

# Erweiterung

## Stufe 2

- **Erweiterte Zielstellung**
  - externe Datei einbinden
  - Teilausschnitte der Datei einbinden
  - Zeilennummern ausgeben

# Umsetzung in LaTeX

## Stufe 2

```
\documentclass{article}
\usepackage{listings}
\begin{document}
\lstset{language=C}
```

```
\section{1. Aufgabe}
```

```
\lstinputlisting[emph={a, b}, emphstyle=\underbar, breaklines=true,
                 firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2]
                 {/home/user/loesung.c}
```

```
\section{2. Aufgabe}
```

```
\lstinputlisting[emph={a, b}, emphstyle=\underbar, breaklines=true,
                 firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2]
                 {/home/user/loesung.c}
```

```
\end{document}
```

# Umsetzung in LaTeX

## Stufe 2

```
\documentclass{ article}  
\usepackage{ listings}  
\begin{ document}  
\lstset{ language=C}
```

```
\section{ 1. Aufgabe}
```

```
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,  
                firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2]  
                {/home/user/loesung.c}
```

```
\section{ 2. Aufgabe}
```

```
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,  
                firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2]  
                {/home/user/loesung.c}
```

```
\end{ document}
```

# Ausgabe von LaTeX

## Stufe 2

[Ausgabe](#)

# Fazit

## Stufe 2

- **\lstinputlisting[option]{file}**  
bindet externe Datei ein
  - **firstline, lastline** – Ausschnitt einbinden
  - **numbers=left** – Zeilennummern links ausgeben
  - **numberstyle** – Stil von Zeilennummern
  - **stepnumber** – Schrittweite von Zeilennummern
  
- Dokumentation zum Quelltext kann so per **Makefile** aktualisiert werden

# Erweiterung

## Stufe 3

- **Erweiterte Zielstellung**
  - Überschrift vergeben
  - Rahmen zeichnen
  - Label für Referenz im Text setzen

# Umsetzung in LaTeX

## Stufe 3

```
\documentclass{ article}
\usepackage{ listings}
\begin{document}
\lstset{ language=C}

\section{1. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2,
                caption={ Aufgabe 1}, label=aufgabe1, frame=trBL]
                {/home/user/loesung.c}

\section{2. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2,
                caption={ Aufgabe 2}, label=aufgabe2, frame=trBL]
                {/home/user/loesung.c}

Wie in Quellcode \ref{ aufgabe2} gesehen, ...
\end{document}
```

# Umsetzung in LaTeX

## Stufe 3

```
\documentclass{ article}
\usepackage{ listings}
\begin{document}
\lstset{ language=C}

\section{1. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2,
                caption={ Aufgabe 1}, label=aufgabe1, frame=trBL]
                {/home/user/loesung.c}

\section{2. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2,
                caption={ Aufgabe 2}, label=aufgabe2, frame=trBL]
                {/home/user/loesung.c}

Wie in Quellcode \ref{aufgabe2} gesehen, ...
\end{document}
```

# Ausgabe von LaTeX

## Stufe 3

Ausgabe

# Fazit

## Stufe 3

- **caption** – setzt eine Überschrift
- **frame=[t|T][r|R][b|B][l|L]** –  
setzt einen Rahmen  
(Großbuchstabe = Doppellinie)
- **label** – setzt ein Label → `\ref{label}`

# Erweiterung

## Stufe 4 (letzte Stufe)

- Erweiterte Zielstellung:
  - Inhaltsverzeichnis über alle enthaltenen Quellcodes im Dokument

"list of listings"

# Umsetzung in LaTeX

## Stufe 4

```
\documentclass{ article}
\usepackage{ listings}
\begin{document}
\lstset{ language=C}  \
Inhaltsverzeichnis:  \
\lstlistoflistings   \
\section{1. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                 firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2,
                 caption={Aufgabe 1}, label=aufgabe1, frame=trBL]
                 {/home/user/loesung.c}

\section{2. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines=true,
                 firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2,
                 caption={Aufgabe 2}, label=aufgabe2, frame=trBL]
                 {/home/user/loesung.c}

Wie in Quellcode \ref{aufgabe2} gesehen, ...
\end{document}
```

# Umsetzung in LaTeX

## Stufe 4

```
\documentclass{ article}
\usepackage{ listings}
\begin{ document}
\lstset{ language=C}  \
Inhaltsverzeichnis:  \
\lstlistoflistings   \
\section{ 1. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines= true,
                 firstline=1, lastline=7, numbers=left, numberstyle=\tiny, stepnumber=2,
                 caption={ Aufgabe 1 }, label=aufgabe1, frame=trBL]
                 {/home/user/loesung.c}

\section{ 2. Aufgabe}
\lstinputlisting[emph={ a, b}, emphstyle=\underbar, breaklines= true,
                 firstline=8, lastline=15, numbers=left, numberstyle=\tiny, stepnumber=2,
                 caption={ Aufgabe 2 }, label=aufgabe2, frame=trBL]
                 {/home/user/loesung.c}

Wie in Quellcode \ref{aufgabe2} gesehen, ...
\end{ document}
```

# Ausgabe von LaTeX

## Stufe 4

Ausgabe

# Fazit

## Stufe 4

- ❑ **\lstlistoflistings** –  
erzeugt ein Quelltextverzeichnis
- ❑ Einträge entsprechen den jeweiligen  
"caption" Werten
- ❑ 2x kompilieren

# Zusammenfassung: Quellcode – Wo im Dokument?

- "code snippets"
  - **innerhalb** eines Fließtextes
  - `\stinline[Optionen]{...}` (vgl. `$...$`)
  
- "displayed code"
  - **außerhalb** eines Fließtextes
  - Istlisting Umgebung (vgl. `$$...$$`)
  - `\begin{Istlisting} [Optionen]`  
    `...`  
    `\end{Istlisting}`
  
- "standalone files"
  - **externe** Datei außerhalb des Textes
  - `\stinputlisting[Optionen]{Pfad}`

# Zusammenfassung: Features von listings

- Syntaxhighlighting
- Zeilennummerierung
- Hervorhebung von Literalen
- externe Dateien einlesen
- Rahmen setzen
- Quellcodeverzeichnis
- [...]

--> RTFM!

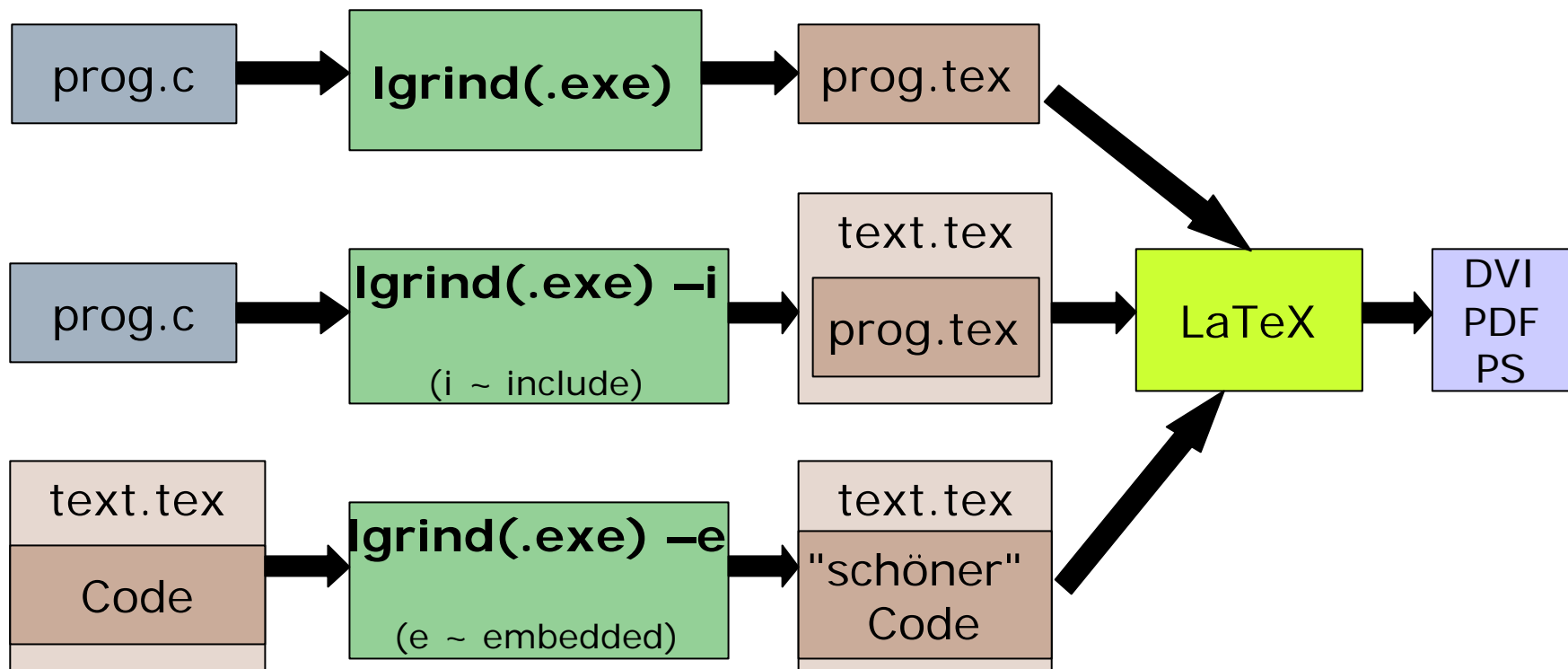
```
\usepackage{ lgrind }
```

<ftp://ftp.dante.de/tex-archive/support/lgrind>

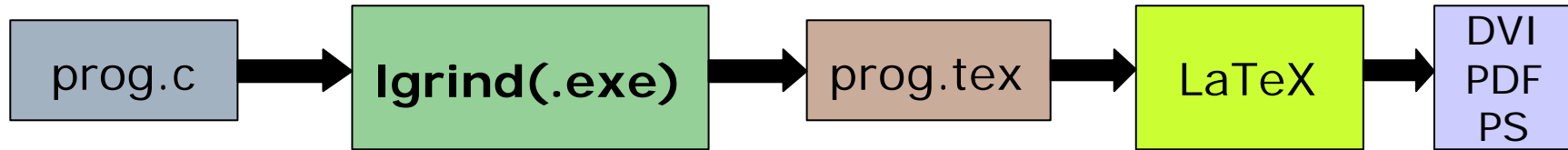
# Installation

- Download (<ftp.dante.de>)
- README lesen
- make && make install
  - erstellt das Binary "lgrind" und
  - das LaTeX Package "lgrind"
- texhash
  
- Dokumentation: lgrind.dvi

# Funktionsmodi von Igrind



# Beispiel 1: "stand-alone" Modus



- **Aufruf:**  
lgrind -lC loesung.c > loesung.tex  
latex loesung.tex
- **Optionen:**
  - -l (~ language) setzt die Programmiersprache manuell
  - -o (~ output) Ausgabe in Datei umlenken
- **Ergebnis:**  
loesung.dvi

# Beispiel 1:

## Ausgabe von Igrind

```
\documentclass[a4paper,twoside]{article}
\usepackage[procnames,noindent]{Igrind}
\usepackage{fancyhdr,a4wide}
\usepackage{german}
\usepackage{makeidx}
\pagestyle{fancy}

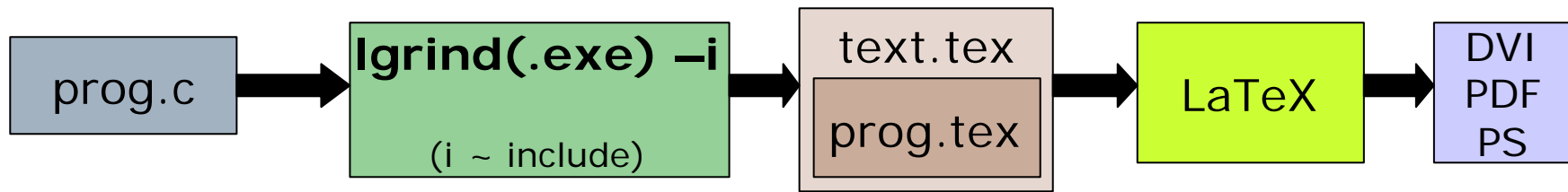
\makeindex

\begin{document}
\renewcommand{\footrulewidth}{0.4pt}
\fancyhead[C]{\Igrindhead}
\fancyhead[LO,RE]{\Igrindfilesize~Bytes\\Igrindmodtime}
\fancyhead[RO,LE]{\bfseries \Igrindfilename\\Igrindmodday.\Igrindmodmonth.\Igrindmodyear}
\fancyfoot[C]{\bfseries\thepage}
\setlength{\headheight}{24pt}
\begin{Igrind}
\BGfont
\File{loesung.c}{2003}{6}{21}{16:20}{349}
\L{\LB{\C}/*}\Tab{4}{Aufgabe_1}\Tab{15}{*/\CE{}}
\index{maximum}\Proc{maximum}\L{\LB{\K{int}_\V{maximum}}_{\K{int}_\V{a},_\K{int}_\V{b}}}}
\L{\LB{\}}
\L{\LB{}}\Tab{8}{\V{printf}(\S{}}\3Berechne_das_Maximum_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_\2n\3\SE{}});}\Tab{88}{}}
\L{\LB{}}\Tab{8}{\K{if}_(\V{a}_\>=_\V{b})_\K{return}_\V{a};}\Tab{40}{}}
\L{\LB{}}\Tab{8}{\K{else}_\K{return}_\V{b};}\Tab{32}{}}
\L{\LB{\}}
\L{\LB{}}
\L{\LB{\C}/*}\Tab{4}{Aufgabe_2}\Tab{15}{*/\CE{}}
\index{minimum}\Proc{minimum}\L{\LB{\K{int}_\V{minimum}}_{\K{int}_\V{a},_\K{int}_\V{b}}}}
\L{\LB{\}}
\L{\LB{}}\Tab{8}{\V{printf}(\S{}}\3Berechne_das_Minimum_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_denke_.\.,\.,_\2n\3\SE{}});}\Tab{88}{}}
\L{\LB{}}\Tab{8}{\K{if}_(\V{a}_\<_\V{b})_\K{return}_\V{a};}\Tab{48}{}}
\L{\LB{}}\Tab{8}{\K{else}_\K{return}_\V{b};}\Tab{48}{}}
\L{\LB{\}}
\end{Igrind}
\printindex
\end{document}
```

# Beispiel 1: Ausgabe von LaTeX

Ausgabe

# Beispiel 2: "include" Modus



**Analog** zu Beispiel 1

Unterschiede:

■ Aufruf:

`lgrind -i` aufrufen

■ `text.tex`:

`\usepackage[Optionen]{lgrind}`

`\lgrindfile{file}` fügt `file` in Dokument ein

# Beispiel 2:

## LaTeX Dokument + Igrind Import

```
\documentclass{article}
```

```
\usepackage{Igrind}
```

```
\begin{document}
```

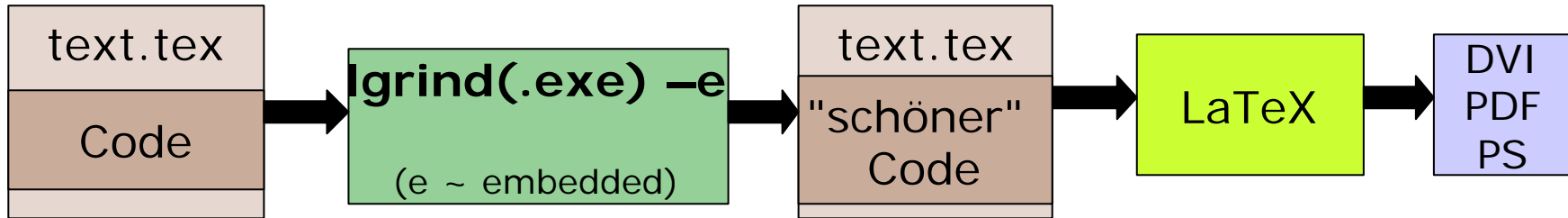
```
\Igrindfile{loesung-include.tex}
```

```
\end{document}
```

# Beispiel 2: LaTeX Ausgabe

Ausgabe

# Beispiel 3: "embedded" Modus



- ❑ "Markierter" Quellcode wird "überarbeitet"
- ❑ **Markierung** erfolgt durch:
  - Innerhalb von Fließtext: @ ... Quellcode ... @
  - Außerhalb von Fließtext: %[ ... Quellcode ... %]
- ❑ **Regeln** bzgl. %[ bzw. %]:
  - ersten zwei Zeichen pro Zeile
  - einzigen zwei Zeichen pro Zeile
- ❑ `\usepackage[Optionen]{lgrind}` nicht vergessen

# Beispiel 3:

## Dokument mit embedded Quellcode

```
\documentclass{article}
\usepackage{lgrind}
\begin{document}
%[
/* Aufgabe 1 */
int maximum (int a, int b)
{
    printf("Berechne das Maximum ... denke ... denke ... denke ... \n");
    if (a >= b) return a;
    else return b;
}

/* Aufgabe 2 (gekürzt) */

}
%]
\end{document}
```

Beispiel 3:

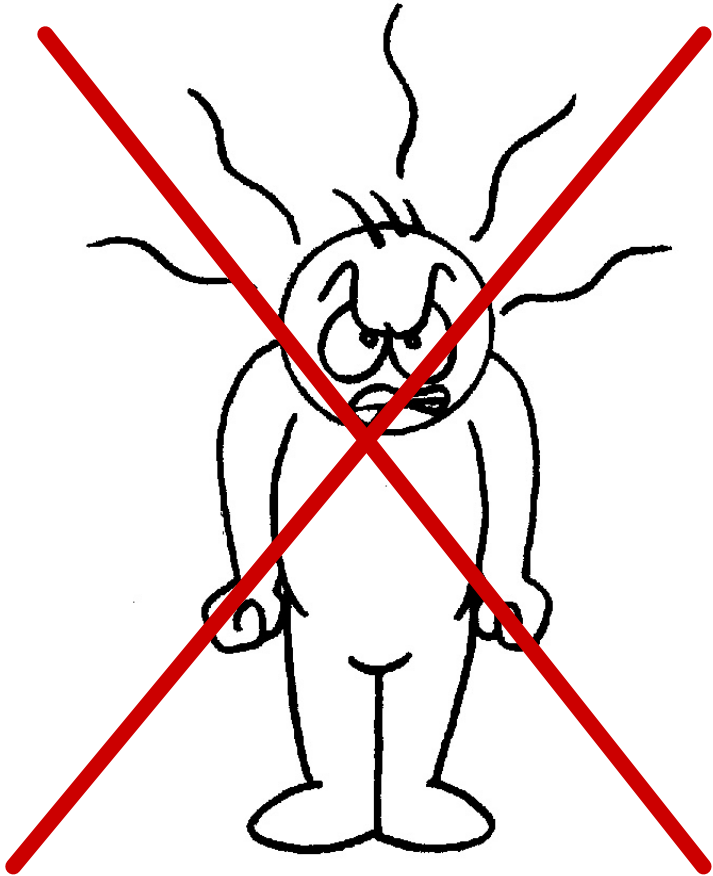
`lgrind -e && latex` → Ausgabe

Ausgabe

# Zusammenfassung: Igrind

- Igrind arbeitet als LaTeX Präprozessor
- setzen von Optionen
  - als Parameter für den Aufruf des Programms
  - als Parameter für das LaTeX Package
- 3 Modi (stand-alone, include, embedded)

# Fazit



- Quellcode mit LaTeX zu setzen ist:
  - einfach
  - bequem
  - konsistent
  - automatisierbar
  
- Es lohnt sich, "**listings**" und "**lgrind**" zu benutzen.