

Introduction to ParaView

Raphael Münster

December 5, 2017

<https://depot.tu-dortmund.de/xlu8z>

About ParaView

ParaView Features

- Open-source and multi-platform visualization software
- Visualization backend provided by VTK library
- Huge number of visualization filters
- Extension is possible by programmable filters (Python) or
- User defined plugins
- Import and export of data to various formats used in CFD packages
- Easy access to numerical data for internal or external plotting etc.
- Tasks can be automated by using the PvPython interface
- Client/Server paradigm to view Big Data on remote clusters

ParaView Software Ecosystem

CMake

- Script language to control the building process
- Generates a wide range of specific build files

VTK

- Visualization backend
- Uses OpenGL

Qt (cute)

- Provides widgets and other GUI controls
- Support for modular plugins

ParaView

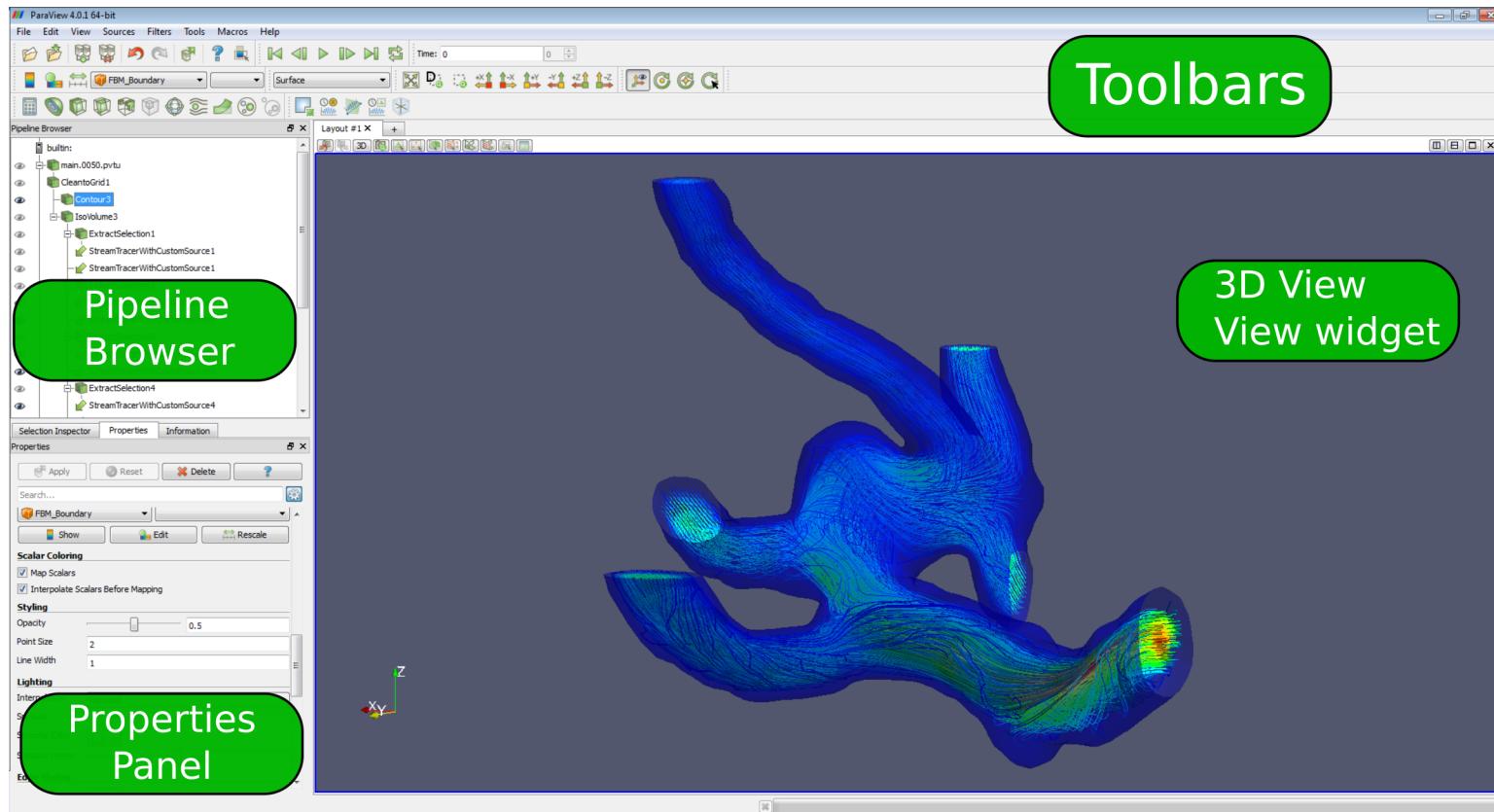
- VTK visualization by GUI
- Scripting via Python
- Extension by plugins

Use Cases of Visualization Software

Typical ParaView Use Cases

- Provide an intuitive illustration of raw simulation data
- Highlight key features of specific flow features
- Provide an intuitive understanding for non-expert viewers
- Help in the testing/debugging process of CFD software
- Assist during result validation by providing analysis tools
- Convert data in different formats in order to communicate with scientific partners

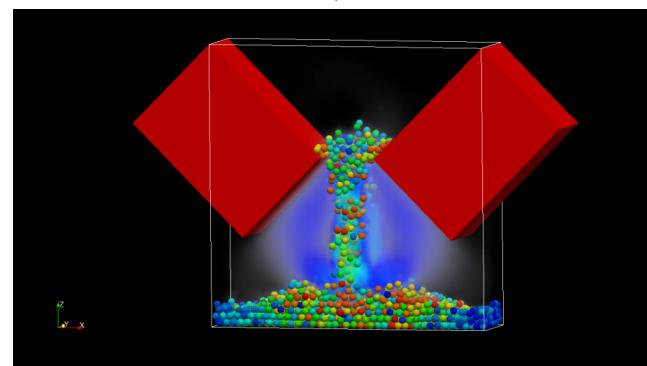
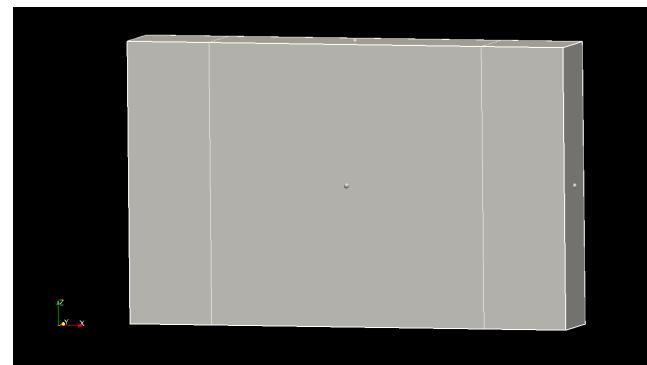
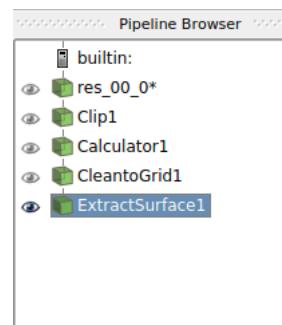
ParaView Interface



Filter Pipeline Concept

Filter Pipeline

- A **filter** is an operation on an input data set
- Filters can be chained (pipelined)
- More complex visualizations require multiple filters



Camera and Axes Toolbar



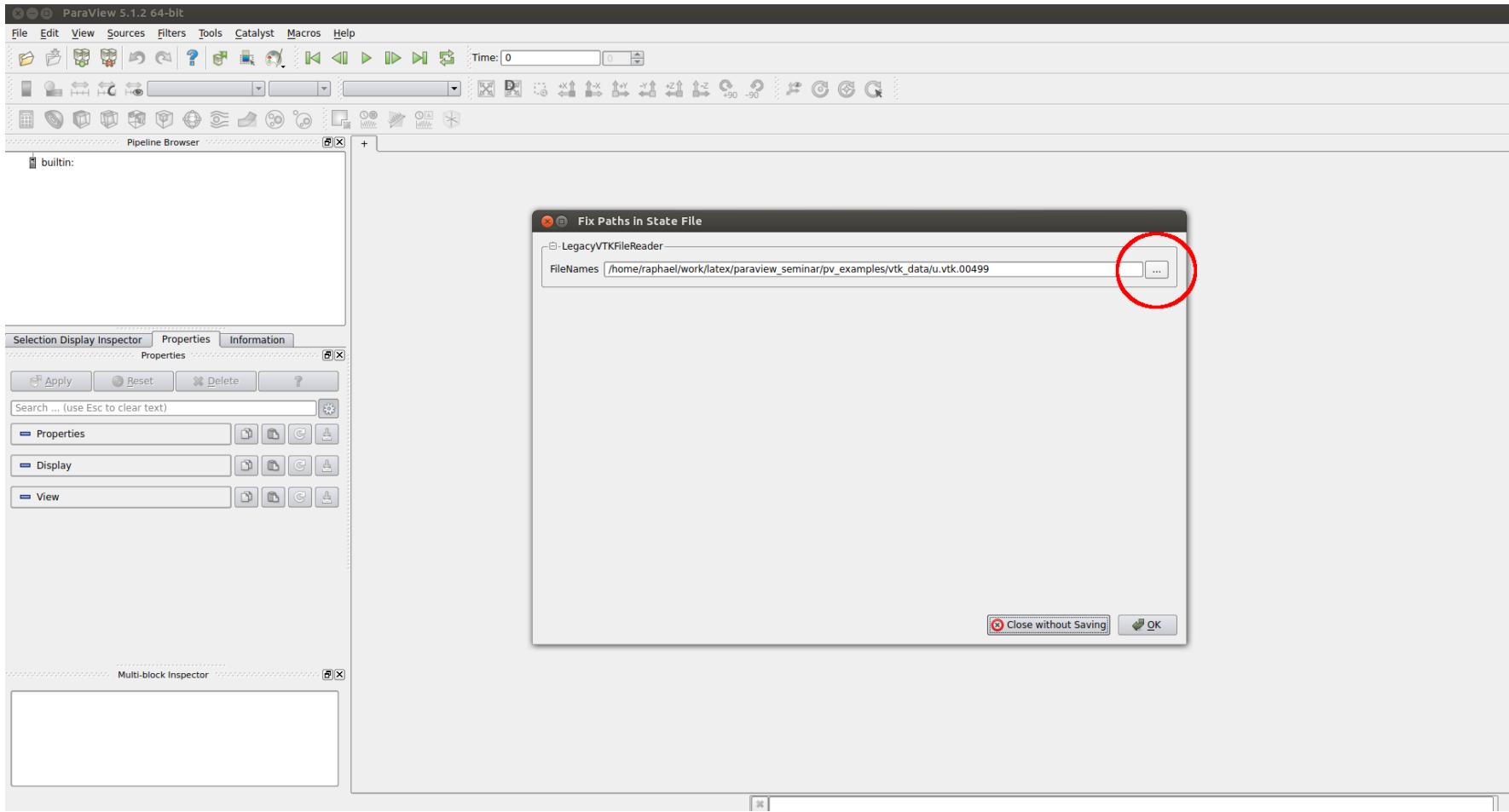
- Reset the camera to its default parameters
- Magnify a rectangular selection of the view area
- Choose a certain coordinate axes plane to look at
- Toggle rendering of orientation axes
- Toggle rendering of center of rotation
- Select a new center of rotation

Basic Filters

- **Slice**: Extract a plane, box-, sphere- or cylinder surface out of a data set
- **Clip**: Extract a plane, box-, sphere- or cylinder volume out of a data set
- **Warp by Scalar**: Visualizes a scalar value by a height extrusion on a 2D data set
- **Contours**: Increases visibility of different solution contour levels
- **Surface LIC**: Streamlines on pixel basis, highlights small scale flow features
- **Glyphs**: Streamlines on pixel basis, highlights small scale flow features
- **Stream Traces**: Visualizes the pathline of a particle through a *stationary* vector field

State Files

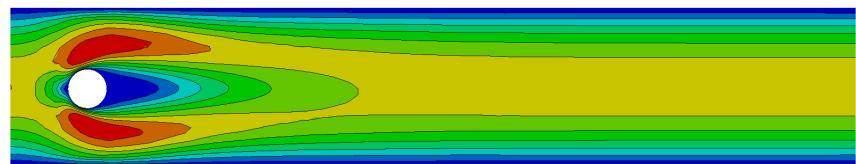
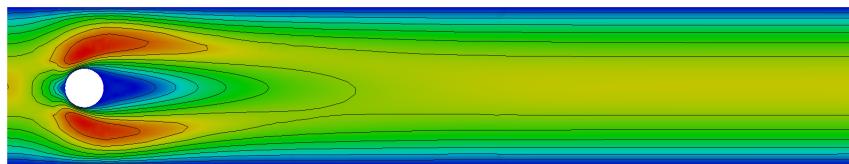
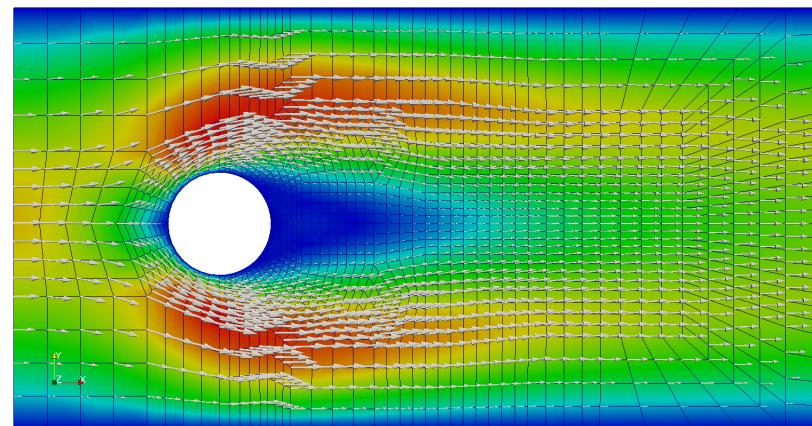
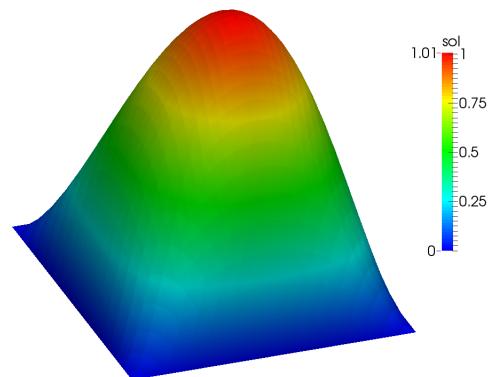
- **State file:** XML format based file with the ending .pvsm that is used to store the currently applied filters and a reference to the currently loaded data sets to a file
- Used to quickly restore a state or to recover from a crash
- The reference to the data set can be changed upon loading the state file in order to apply the filters to a different data set or if the location of the data on the hard drive has changed
- This way state files can be used to exchange a ParaView visualization with collaborators, they only need to set the location of their data set upon loading the state file
- State files can be exchanged between different version of ParaView
- Accessible from Menu: **File->Load State...**



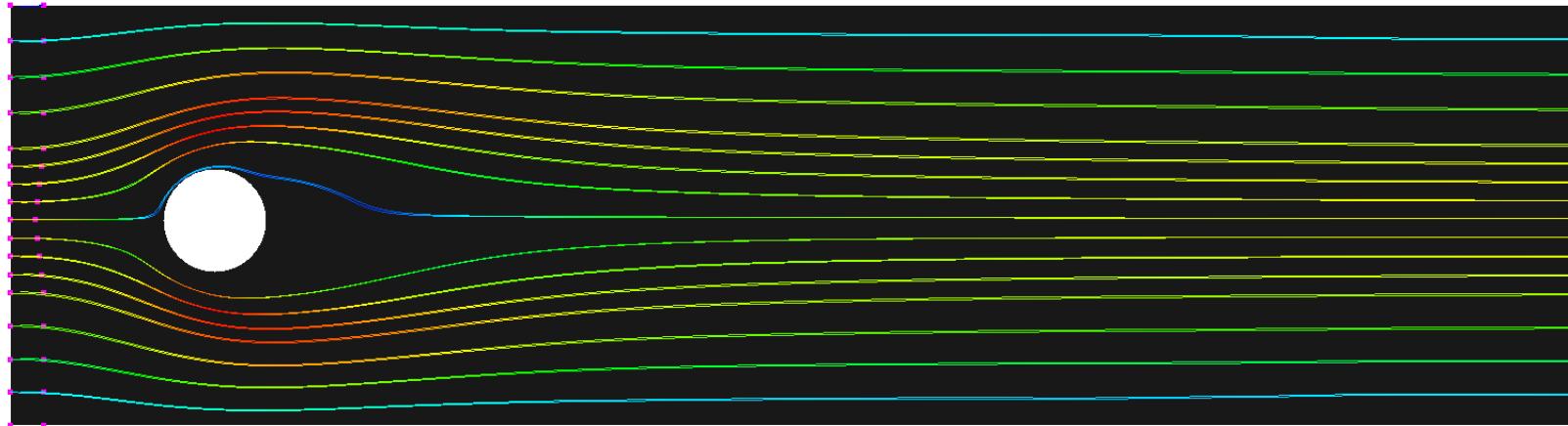
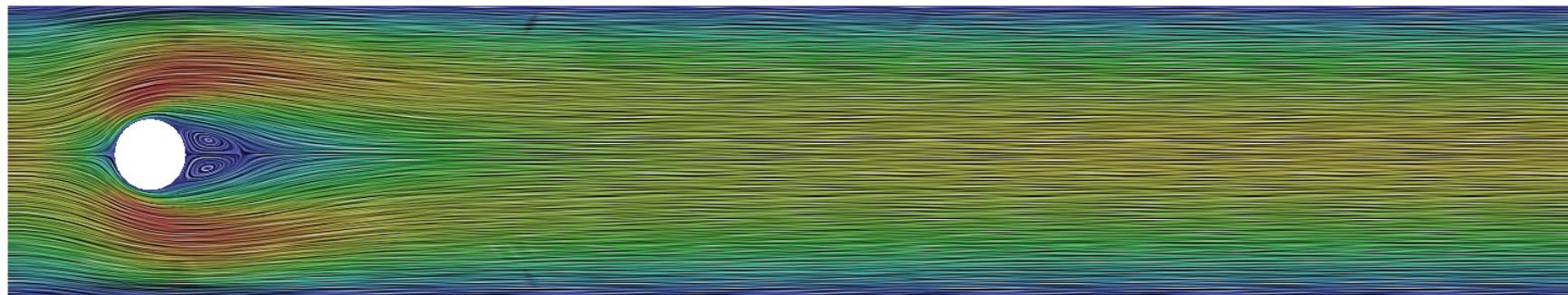
Examples Files

- ParaView examples are provided as state files
- Download archive of ParaView examples from:
- Extract archive to **/mypath/to/examples/**
- Basic filter examples are located in **/mypath/to/examples/pv_examples**
- Basic filter examples are located in
/mypath/to/examples/pv_examples/BasicFilters
- To load a basic filter example, load the state file and set data path to:
/mypath/to/examples/pv_examples/vtk_data/u.vtk

Basic Filter Gallery I



Basic Filter Gallery II



Client-Server Mode

- In default mode ParaView is both the client and the server
- When client/server are different rendering and data processing can be handled by different computers
- Simple X forwarding works adequately only if the network speed is fast
- Client-Server is preferable to access data on remote (non-local) clusters
- Client-Server steps: **port forwarding, starting the remote server, connecting the client to the server**

Port Forwarding

- Establish an ssh tunnel to forward the local port to the remote server:

```
> ssh lidong1.itmc.tu-dortmund.de \
-L 11111:lidong1.itmc.tu-dortmund.de:11111
```

Start the remote Server

- Start a ParaView data server on the remote machine

```
> pvserver --server-port=11111 --use-offscreen-rendering
```

Connecting to the remote Server

- Start a ParaView client locally
- Press the <Connect> button on the toolbar
- Manually configure the Server dialog:

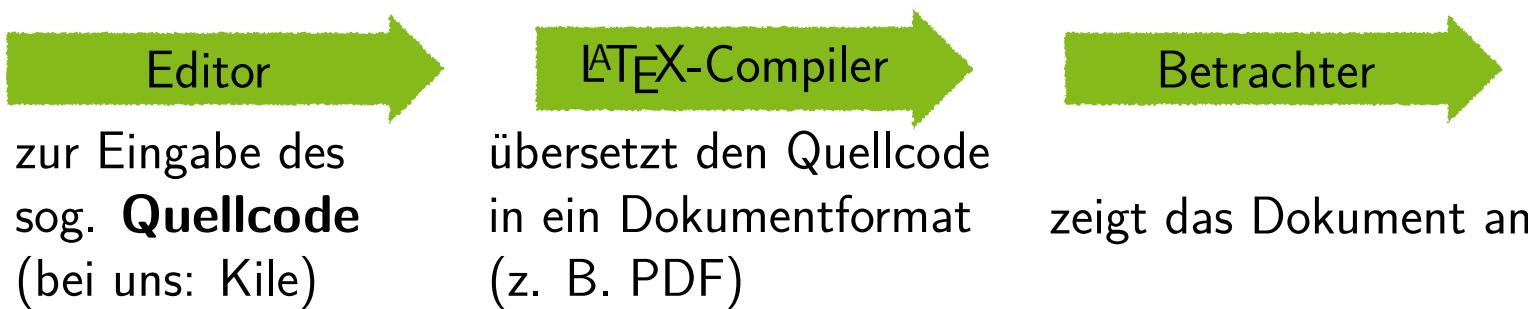
- Name: myname
- Server Type: Client/Server
- Host: localhost
- Port: 11111

Pitfalls:

- You **have to** make use the same ParaView version of the client and the server
- Check that the port is not occupied, otherwise use a different port:

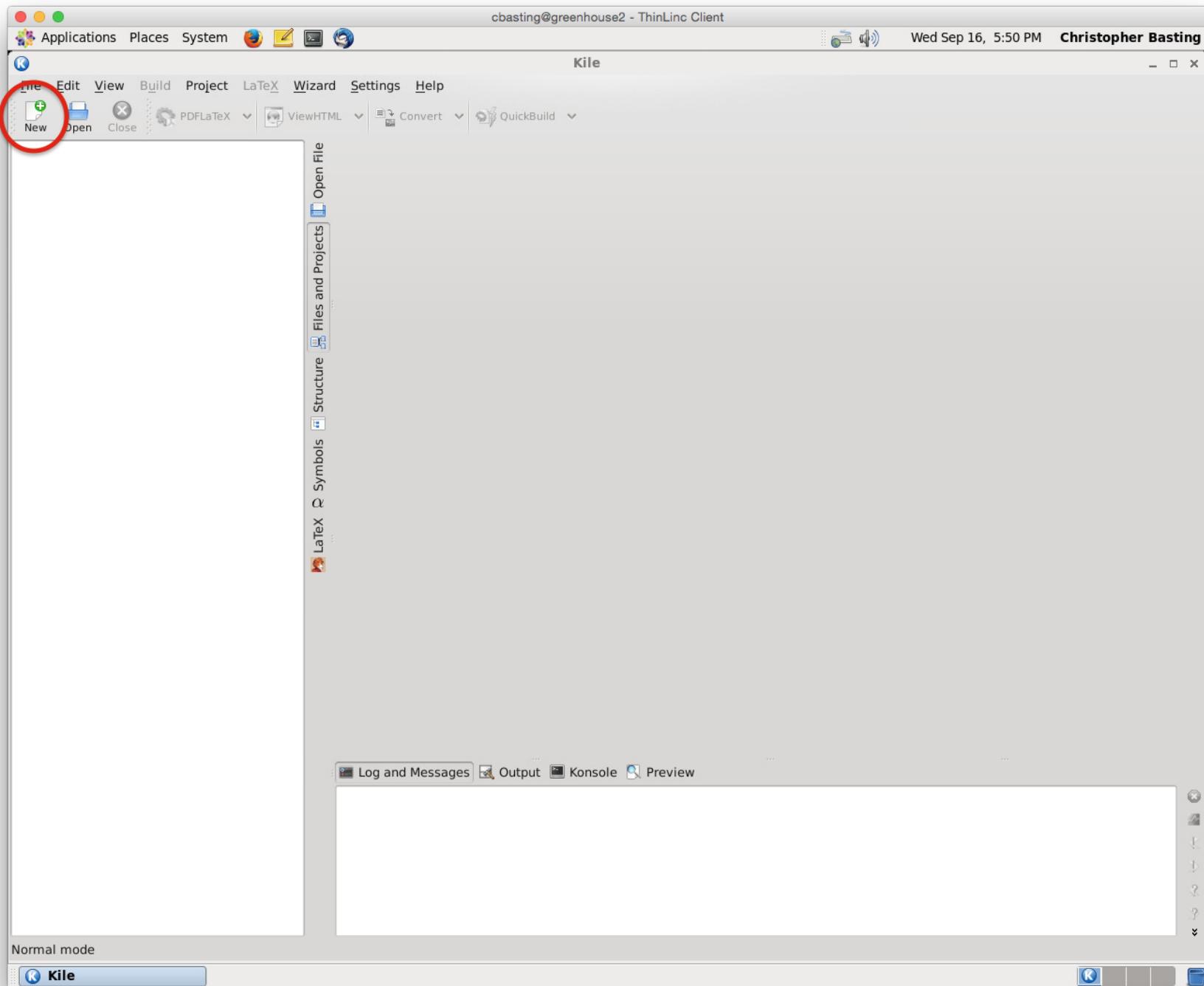
```
> lsof -i:11111
```

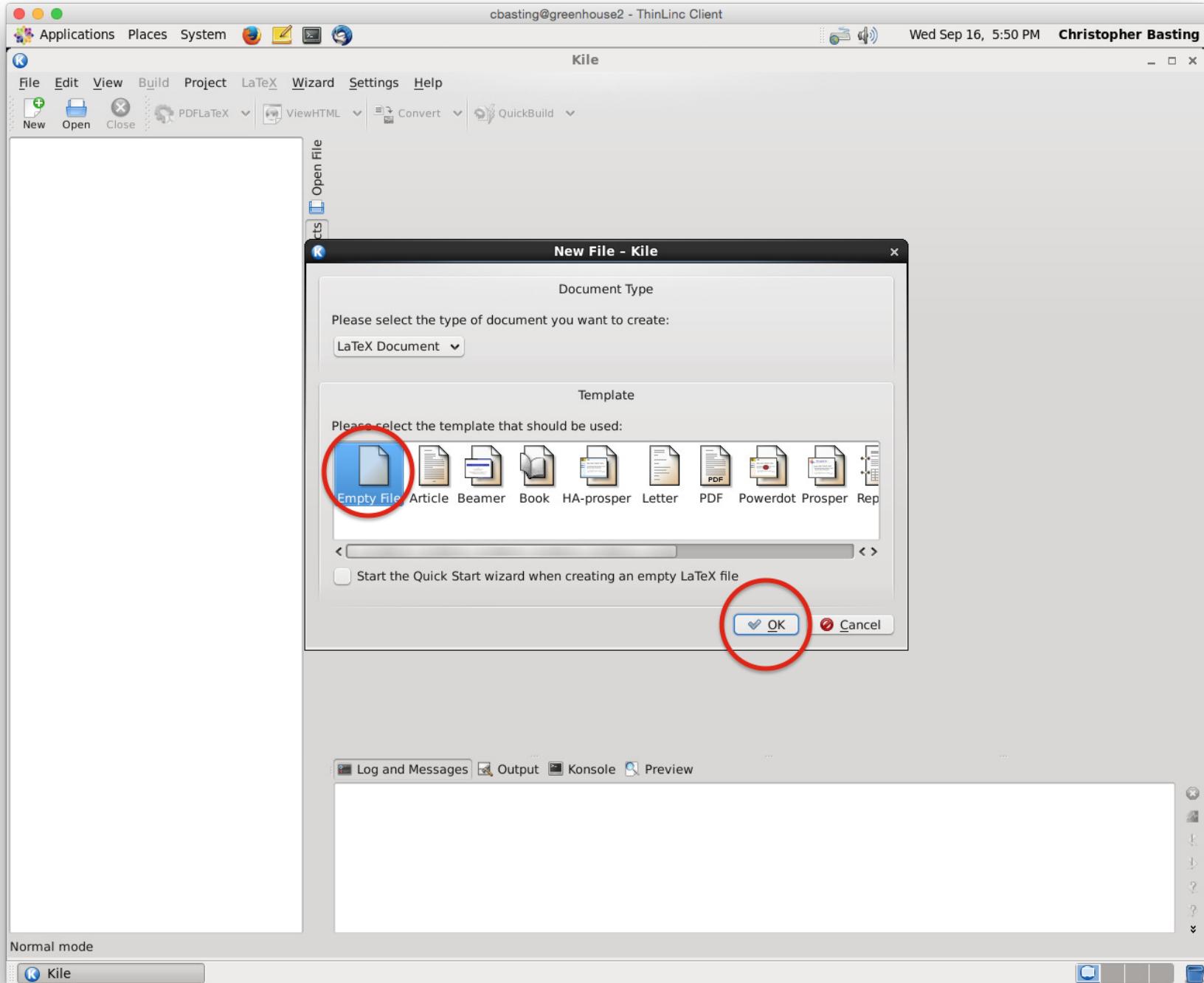
LATEX: Ablauf der Dokumenterstellung

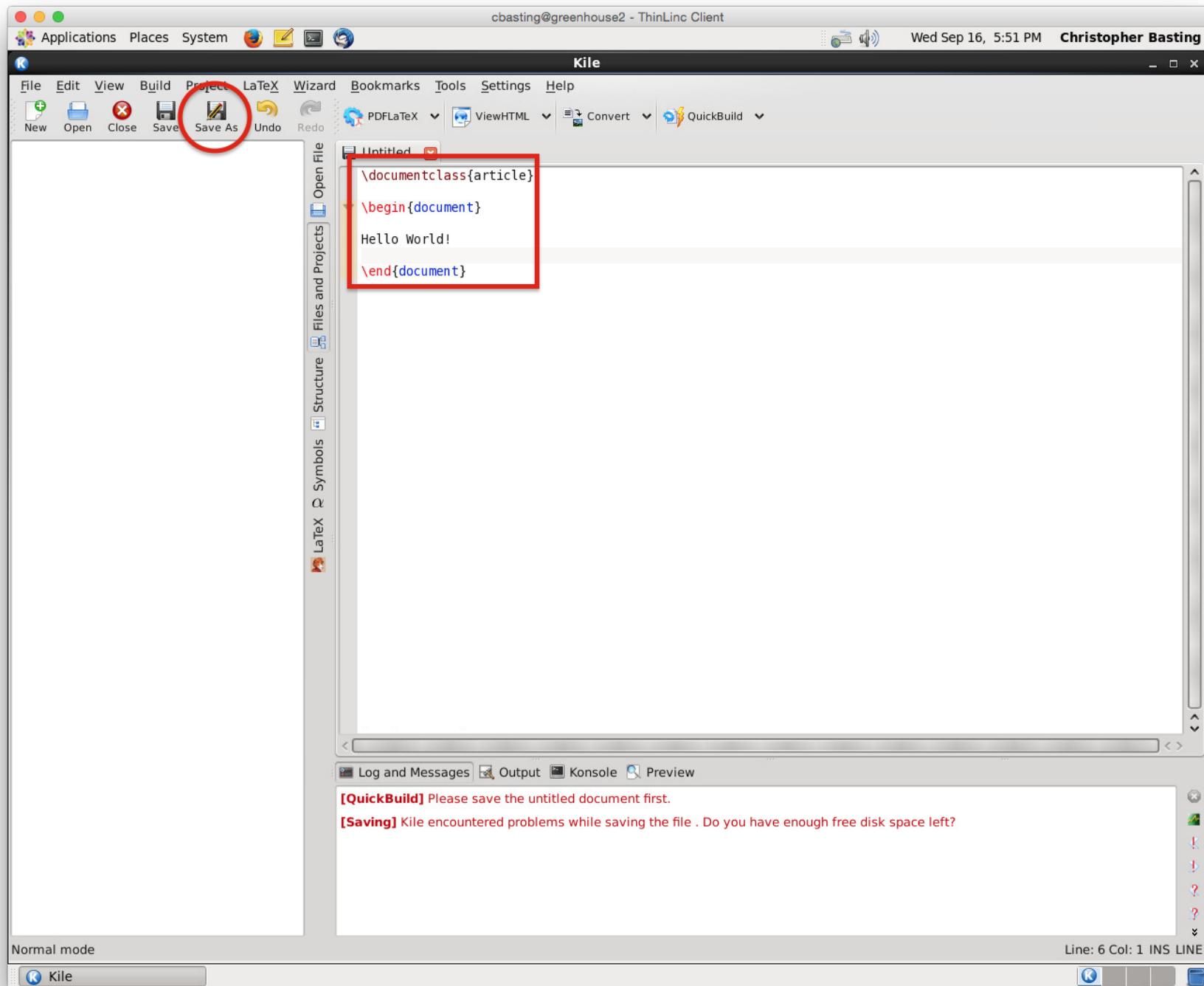


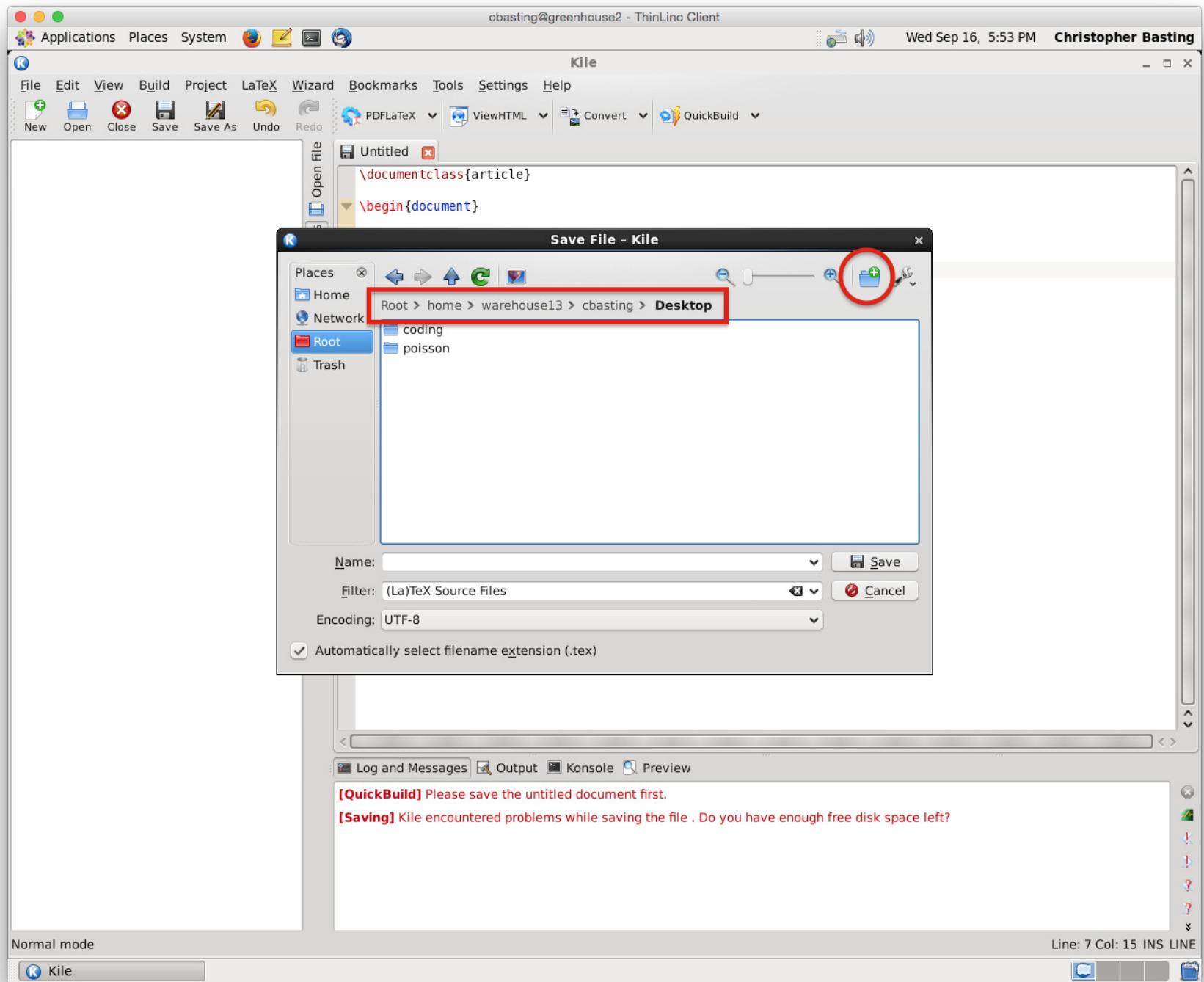
Kile - **LATEX**Entwicklungsumgebung

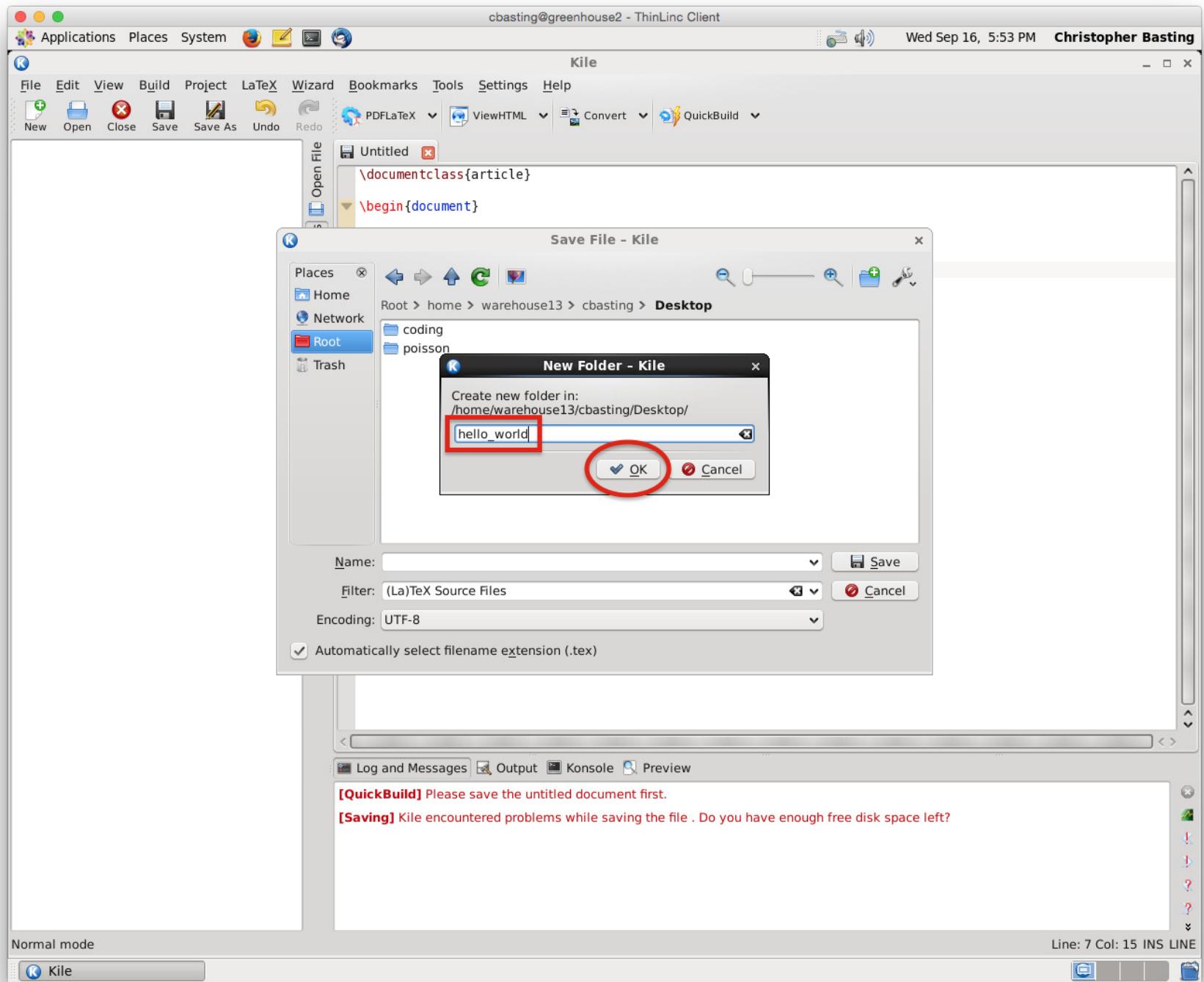
- erleichtert das Bearbeiten von mehreren Dokumenten
- Auto vervollständigung
- Hilfe zum Kompilieren der Dokumente
- Syntaxhervorhebung
- Rechtschreibprüfung
- Code-Faltung

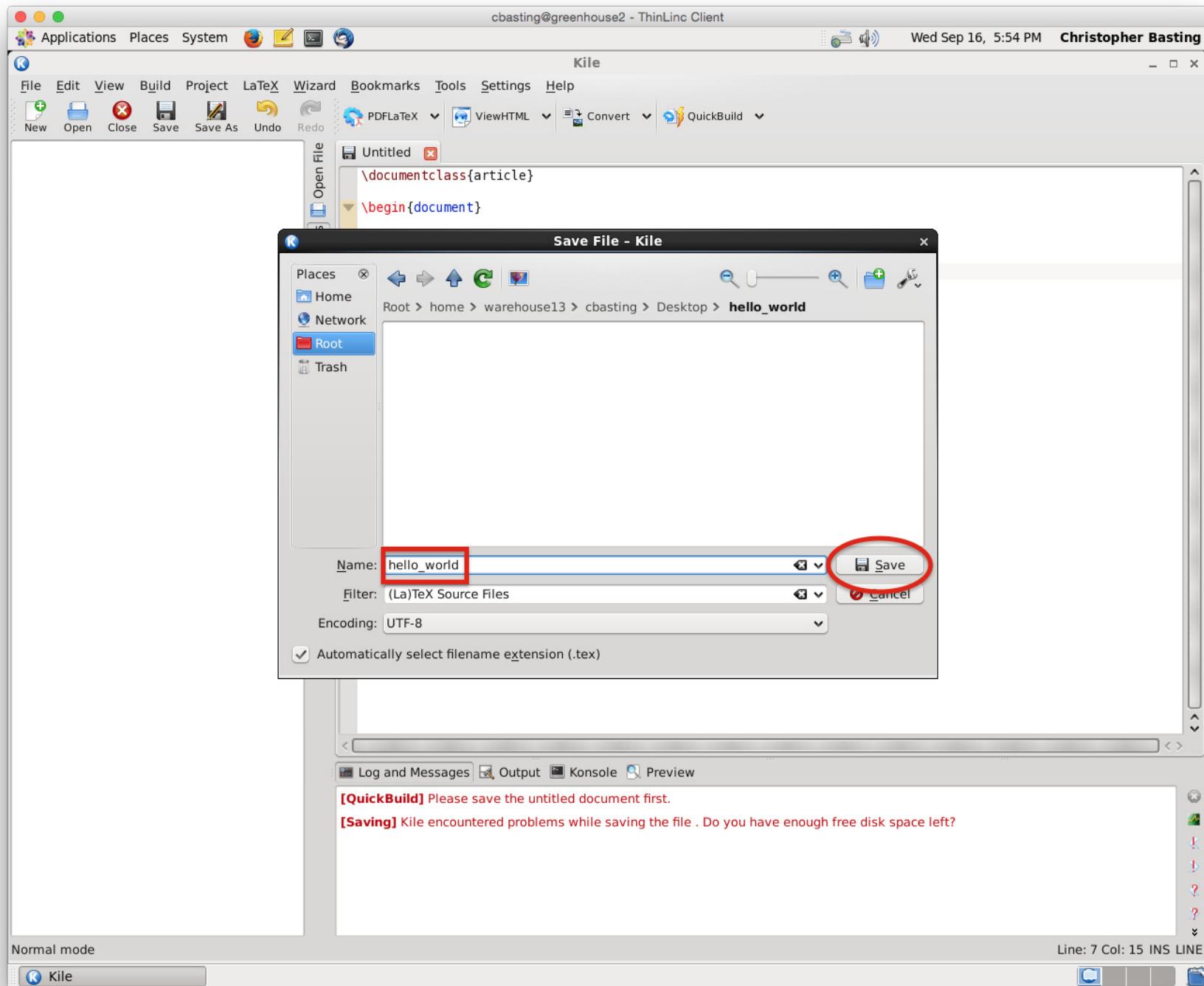


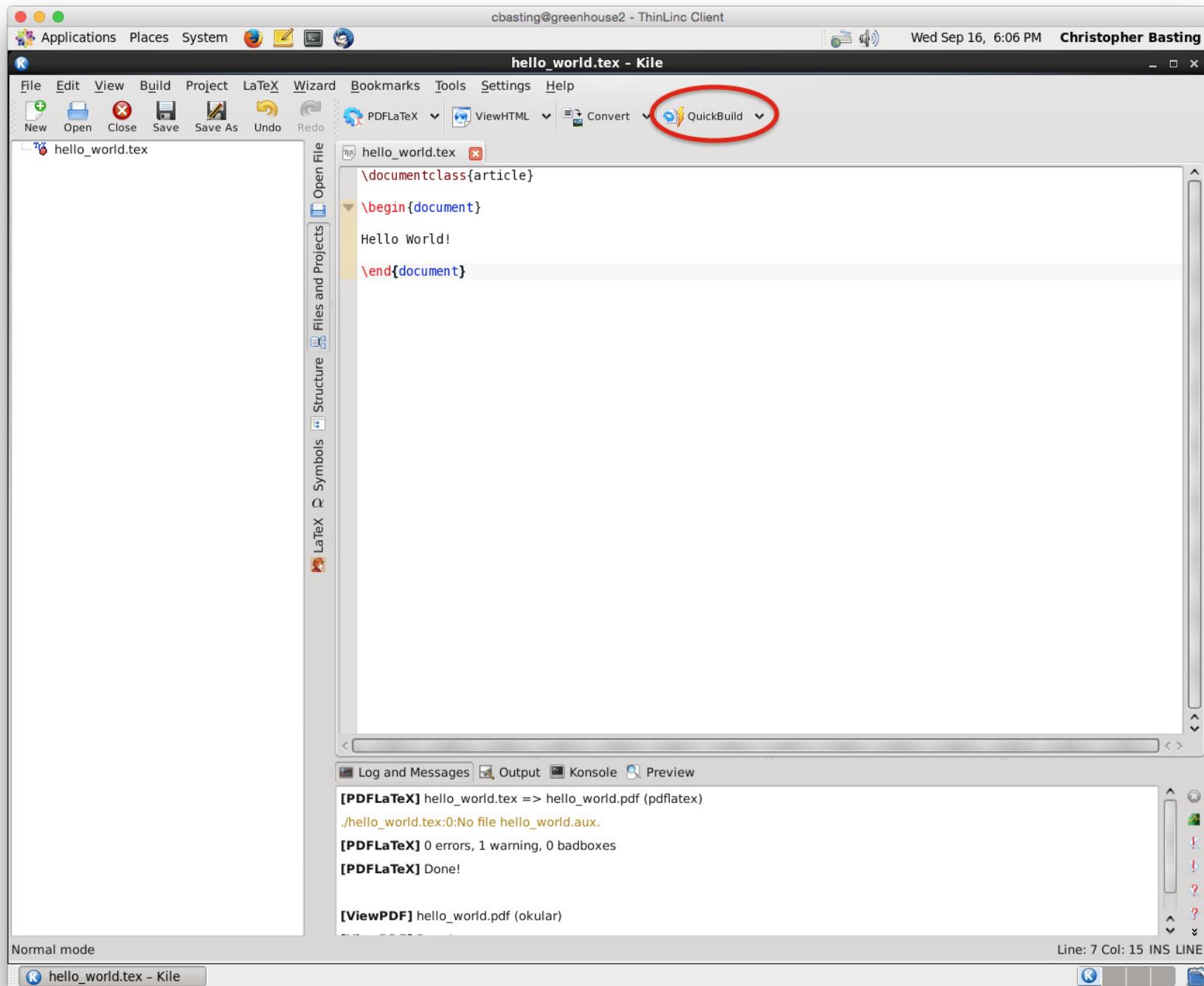


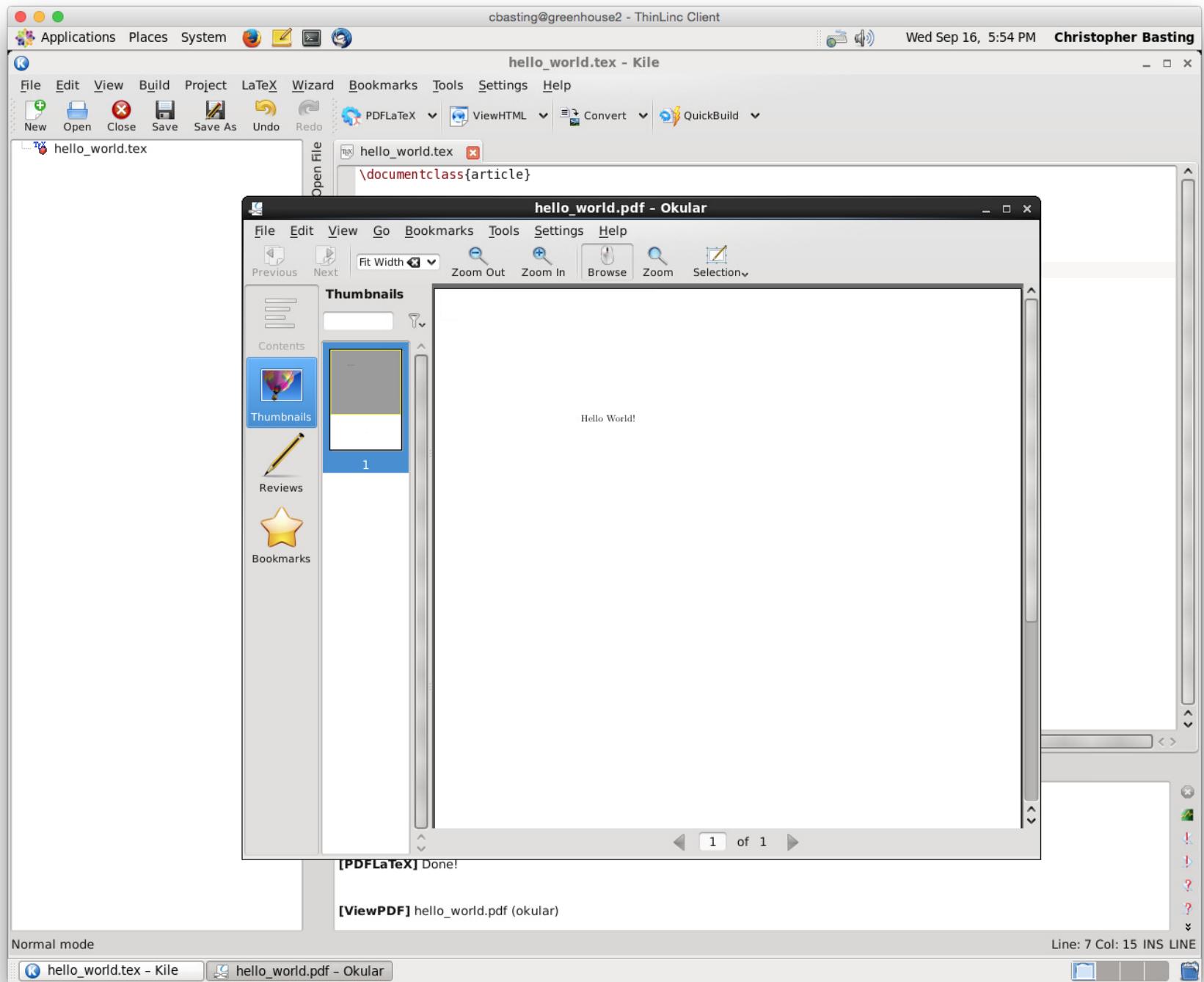












Hello World!

Ein erstes \LaTeX -Dokument

```
% <-- Kommentarzeichen  
  
\documentclass{article}  
  
% Praeamble  
  
\begin{document}  
    Hello World!  
\end{document}
```

Hello World!

1

Generierung einer PDF-Datei

Um eine PDF-Datei aus einem L^AT_EX Dokument **datei.tex** zu erzeugen, muss der L^AT_EX-Compiler aufgerufen werden:

```
pdflatex datei.tex
```

Alternativ kann man wie folgt vorgehen:

1 **latex datei.tex**

Dieser Befehl ruft den L^AT_EX-Compiler auf und erzeugt die DVI-Datei **datei.dvi**.

2 **dvips datei.dvi**

Konvertiert die **datei.dvi** in eine Post-Script Datei **datei.ps**.

3 **ps2pdf** oder direkt **dvipdf**

Erzeugt eine PDF-Datei **datei.pdf**.

Eingabe eines einfachen Textes

Quelltext

```
\documentclass{article}
\begin{document}
Normaler Text kann einfach eingegeben werden.
Er wird dann nach Vorgabe im Blocksatz gesetzt.

Eine Leerzeile kennzeichnet einen neuen Absatz.

Um eine Zeile im Absatz umzubrechen, \\
benutzt man zwei Backslashes.
\end{document}
```

Normaler Text kann einfach eingegeben werden. Er wird dann nach Vorgabe
im Blocksatz gesetzt.
Eine Leerzeile kennzeichnet einen neuen Absatz.
Um eine Zeile im Absatz umzubrechen,
benutzt man zwei Backslashes.

Befehle und Umgebungen

- **Befehle** haben in L^AT_EX die Form `\befehl`
- Beispiel: `\textit{kursiv}` schaltet auf *Kursivschrift* um
- Manche Befehle benötigen zusätzliche Parameter, z. B.
`\textbf{Text, der fett gedruckt sein soll}`
oder optionale Parameter, z. B.
`\documentclass[a4paper]{article}`
- **Umgebungen** beeinflussen den gesamten enthaltenen Text:
`\begin{center}`
zentrierter Text
`\end{center}`
Auch hier ist die Angabe von Parametern möglich.

Aufgabe 1 (Hello World)

Erstellen Sie folgendes "Hello-World" Beispiel.

```
\documentclass{article}  
\begin{document}  
    Hello World!  
\end{document}
```

Spezielle Zeichen

- bei Verwendung des ASCII-Zeichensatzes: Buchstaben ohne Umlaute, Zahlen und einige Sonderzeichen
- manche Zeichen sind \LaTeX -Steuerzeichen und daher reserviert (\$, _, {, }, \)
- solche Sonderzeichen können durch \ maskiert werden:

\$	\\$	%	\%
{	\{	}	\}
#	\#	_	_
&	\&		

Deutsche Texte - Sprachpakete

Ohne weitere Angaben nimmt L^AT_EX an, dass der eingegebene Text in englischer Sprache ist. Daher muss ggf. ein zusätzliches Sprachpaket eingebunden werden:

Beispiel-Header: deutsches Sprachpaket

```
\documentclass[a4paper]{article}          % DIN-A4 Papierformat
\usepackage[ngerman]{babel}                % deutsche Benennung
\usepackage[utf8]{inputenc}
\begin{document}
...
\end{document}
```

- **babel** sorgt für Unterstützung anderer Sprachen (Formate, Umlaute, Benennungen, Silbentrennung)
- **inputenc** unterstützt die direkte Eingabe von Zeichen über die Tastatur

Deutsche Texte – Umlaute und Anführungszeichen

Eingabe deutscher Texte:

- Umlaute: \ "a, \ "o, \ "u, \ss für ä, ö, ü, ß
- Anführungszeichen: \glqq, \grqq bzw. \glqqq, \grqqq für ‚einfache‘ bzw. „doppelte“ Anführungszeichen

Beispiel: deutsche Umlaute

```
\glqq Zw\"olf gro\ss e Boxk\"ampfer jagen Viktor quer \"uber  
den Sylter Deich.\grqqq
```

„Zwölf große Boxkämpfer jagen Viktor quer über den Sylter Deich.“

Deutsche Texte - Silbentrennung

- erfolgt automatisch
- mögliche Trennstellen können durch \- auch angegeben werden, z. B.
Donau\‐dampf\‐schiff\‐fahrts\‐gesell\‐schaft
oder für das gesamte Dokument in der Präambel:
\hyphenation{Donau\‐dampf\‐schiff\‐fahrts\‐gesell\‐schaft}

Textformatierung – Schriftstil

Familie	Befehle	Beispiel
normal (mit Serifen)	<code>\rmfamily \text{r}</code>	normal
serifenfrei	<code>\sfamily \text{sf}</code>	serifenfrei
Schreibmaschine	<code>\ttfamily \text{tt}</code>	Schreibmaschine
Varianten	Befehle	Beispiel
aufrecht	<code>\upshape \text{up}</code>	aufrecht
italic	<code>\itshape \text{it}</code>	<i>italic</i>
Kapitälchen	<code>\scshape \text{sc}</code>	KAPITÄLCHEN
fett	<code>\bfseries \text{bf}</code>	fett
unterstrichen	<code>\underline \text{underline}</code>	<u>unterstrichen</u>

Textformatierung – Schriftgröße

\tiny	winzig
\small	klein
\footnotesize	Fußnotengröße
\normalsize	normale Größe
\large	groß
\Large	größer
\huge	riesig
\Huge	Riesig

- für einige Wörter: {\huge riesig}
- für ganze Absätze: \begin{tiny} ... \end{tiny}
- alternativ: punktgenau durch {\fontsize{40}{48}\selectfont{}Test}
das erste Argument gibt die Schriftgröße an, das zweite den Grundlinienabstand

Textformatierung – minipages

- die `minipage`-Umgebung wird genutzt, wenn in Boxen komplexere Inhalte dargestellt werden sollen
- es muss die Breite angegeben werden

Beispiel: minipages

```
\begin{minipage}{0.45\textwidth}
Minipage 1 \\
\ldots \\
noch mehr Text
\end{minipage}
\begin{minipage}{0.45\textwidth}
Minipage 2
\end{minipage}
```

Minipage 1

...

noch mehr Text

Minipage 2

Textformatierung – Absatzausrichtung

Beispiel: Absatzausrichtung

```
\begin{flushright}  
    rechts  
\end{flushright}  
\begin{flushleft}  
    links  
\end{flushleft}  
\begin{center}  
    zentriert  
\end{center}
```



- Standard: Blocksatz
- alternativ kann man auch die Befehle `\raggedleft`, `\raggedright` und `\centering` verwenden (statt den obigen Umgebungen)

Textformatierung - Zitate und Gedichte

Beispiel: Zitate und Gedichte

```
\begin{quote}
    Phantasie ist wichtiger als Wissen, denn
        Wissen ist begrenzt.
\end{quote}
\begin{quotation}
    Mehr als die Vergangenheit interessiert mich
        die Zukunft, denn in ihr gedenke ich zu
        leben.\end{quotation}
\begin{verse}
    Umsonst ist's nicht, dass die Natur
        Uns schenke eine Zung' nicht nur.
        Sondern dazu die Faehigkeit
        Sie rauszustrecken ziemlich weit!
\end{verse}
```

Phantasie ist wichtiger als Wissen, denn Wissen ist begrenzt.

Mehr als die Vergangenheit interessiert mich die Zukunft, denn in ihr gedenke ich zu leben.

Umsonst ist's nicht, dass die Natur Uns schenke eine Zung' nicht nur. Sondern dazu die Faehigkeit Sie rauszustrecken ziemlich weit!

Text ohne Formatierung

- um Programmcode oder Beispiele ohne Formatierung auszugeben, gibt es die **verbatim**-Umgebung
- innerhalb eines Textes kann man den Befehl **\verb** nutzen; das erste Zeichen nach dem Befehl wird dann als Endmarker interpretiert

Beispiel: **verbatim**-Umgebung

```
\begin{verbatim}
Leer    zeichen und Umbrueche
       werden uebernommen.

\end{verbatim}
```

Leer zeichen und Umbrueche
 werden uebernommen.

Randnotizen und Fußnoten

Beispiel: Randnotiz und Fußnote

```
\documentclass[a4paper]{article}
\begin{document}
\ldots ganz viel Text \marginpar{
    Anmerkung}
und\footnote{einer Fußnote}
\end{document}
```

... ganz viel Text und¹

Anmerkung

¹einer Fußnote

Aufgabe 2 (Ein erstes Dokument)

Erstellen Sie ein einfaches L^AT_EX-Dokument. Teilen Sie den Text in Kapitel und Unterkapitel ein. Setzen Sie einige Wörter im Text kursiv, im Fettdruck oder in Schreibmaschinenschrift und in verschiedenen Schriftgrößen. Verwenden Sie zum Beispiel das Paket `blindtext` zur Erzeugung von Text-Passagen.

Aufgabe 3 (Umlaute)

Geben Sie in einem L^AT_EX-Dokument die deutschen Umlaute Ä, Ö, Ü, ä, ö, ü, ß aus, indem Sie:

- die Umlaute maskieren,
- die Umlaute im Quellcode direkt verwenden und den Zeichensatz entsprechend anpassen.

Aufgabe 4 (Zusammensetzen des Dokuments aus mehreren Dateien)

Erstellen Sie ein Dokument aus den bisherigen Aufgaben, welches aus mehreren Dateien besteht. In der Hauptdatei sollen die Präambel, die Titelseite und das Inhaltsverzeichnis stehen. Jedes Kapitel der obersten Ebene soll in einer eigenen Datei stehen und in die Hauptdatei mittels `\include` eingebunden werden. Hierzu müssen die Dateien der vorherigen Aufgaben in ein neues Verzeichnis kopiert werden und die Befehle `\documentclass` und die `document`-Umgebung aus den vorherigen Aufgaben entfernt werden, da diese sonst doppelt vorhanden wären.

Aufzählungen

drei Grundarten von Aufzählungen:

- itemize** einfache Aufzählung
- enumerate** nummerierte Aufzählung
- description** Beschreibung

itemize

- A
- B
- C

```
\begin{itemize}
  \item A
  \item B
  \item C
\end{itemize}
```

enumerate

- 1 A
- 2 B
- 3 C

```
\begin{enumerate}
  \item A
  \item B
  \item C
\end{enumerate}
```

description

- A ...
- B ...
- C ...

```
\begin{description}
  \item[A] \ldots
  \item[B] \ldots
  \item[C] \ldots
\end{description}
```

Verschachtelte Aufzählungen

Beispiel: verschachtelte Aufzählung

```
\begin{description}
  \item[Zutaten] \begin{itemize}
    \item 200g Mehl
    \item 5 Eier
    \item \ldots
  \end{itemize}
  \item[Zubereitung]\hfill\\ \
    \begin{enumerate}
      \item Mehl und Milch in
            eine Schuessel
      \item glatt ruehren
      \item \ldots
    \end{enumerate}
\end{description}
```

Zutaten ■ 200g Mehl
■ 5 Eier
■ ...

Zubereitung

- 1 Mehl und Milch in eine Schuessel
- 2 glatt ruehren
- 3 ...

Aufgabe 5 (Listen und Aufzählungen)

Erstellen Sie ein Kochrezept. Verwenden Sie für die Zutatenliste die Aufzählungs-Umgebung `itemize` und für die Zubereitungshinweise die `enumerate`-Umgebung.

Tabellen - ein einfaches Beispiel

Beispiel: einfache Tabelle

```
\begin{center} \begin{tabular}{|c +c{$=$$}c|} \hline $x$ & $y$ & $\sum$ \\ \hline 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 5 \\ \hline \end{tabular} \end{center}
```

$x+y=\Sigma$
$1+1=2$
$1+2=3$
$2+3=5$

Tabellenformatierung:

- l linksbündig ausrichten | einfacher vertikaler Trennstrich
- c zentriert ausrichten || doppelter vertikaler Trennstrich
- r rechtsbündig ausrichten @{text} benutzerdefiniertes Trennzeichen
- p{n} Spalte mit fester Breite n

Tabellen - mehrspaltige Zellen

Beispiel: mehrspaltige Zellen

```
\begin{tabular}{lll}
  eins & zwei & drei \\
  vier \vline{} f\"unf & sechs &
  \\
  \cline{1-2}
  \multicolumn{3}{|c|}{sieben}
\end{tabular}
```

eins	zwei	drei
vier	fünf	sechs
sieben		

Tabellenformatierung:

- \hline** horizontale Linie über die ganze Breite
- \vline** vertikaler Line innerhalb einer Zeile
- \cline{m-n}** horizontale Linie von Spalte **m** bis Spalte **n**
- \multicolumn{n}{format}{Inhalt}** Zelle über **n** Spalten

Aufgabe 6 (Tabellen)

Erstellen Sie folgende Tabellen:

rechts	zentriert	links
Hier viel	steht unwichtiger	ganz Text

.	1	2	3	4	x	y
1	1	2	3	4	?	?
2	2	4	6	8	*	*
3	3	6	9	12	?	
4	4	8	12	16	$4x$:

Mathematik – Formeln im Fließtext

- Formeln müssen in \LaTeX markiert werden, damit sie korrekt interpretiert werden
- im Mathematik-Modus werden Leerzeichen ignoriert und Buchstabenketten als einzelne Zeichen betrachtet

Beispiel: Formeln im Fließtext

```
Für die reellen Zahlen  $\begin{aligned} &x \end{aligned}$  und  $\begin{aligned} &y \end{aligned}$  sei  
 $x \cdot y < 1$ .
```

Achtung! \glqq keine Leerzeichen\grqq~ wird im Mathematik-Modus zu \glqq\$keine Leerzeichen\$\grqq.

Für die reellen Zahlen x und y sei $x \cdot y < 1$.

Achtung! „keine Leerzeichen“ wird im Mathematik-Modus zu „keineLeerzeichen“.

Mathematik – Formeln in eigenem Absatz

- soll eine Formel abgesetzt dargestellt werden, so benutzt man die `displaymath`-Umgebung oder die Kurzschreibweise `\[. . . \]`

Beispiel: Formeln in eigenem Absatz

Im rechtwinkligen Dreieck gilt:

$$\begin{aligned} & \text{\textbackslash begin\{displaymath\}} \\ & \quad a^2 + b^2 = c^2 \\ & \text{\textbackslash end\{displaymath\}} \end{aligned}$$

Es gilt also `\[c=\sqrt{a^2+b^2}.\]`

Im rechtwinkligen Dreieck gilt:

$$a^2 + b^2 = c^2$$

Es gilt also

$$c = \sqrt{a^2 + b^2}.$$

Mathematik – nummerierte Gleichungen

- Gleichungen können mit der `equation`-Umgebung automatisch fortlaufend nummeriert werden
- mittels der `align`-Umgebung können mehrzeilige Formeln nummeriert und ausgerichtet werden
- `\nonumber` unterdrückt in der `align`-Umgebung die Nummerierung einer Zeile

Beispiel: mehrzeilige Gleichungen

```
\begin{equation}
n! = \prod_{i=1}^n i
\end{equation}
\begin{align}
f(n) &= f(n-1) + f(n-2) \\
&\sum\limits_{k=1}^n k &= 1 + 2 + \cdots + n \\
&\nonumber \\
&\quad &= \frac{n(n+1)}{2}
\end{align}
```

$$n! = \prod_{i=1}^n i \quad (1)$$

$$f(n) = f(n - 1) + f(n - 2) \quad (2)$$

$$\begin{aligned} \sum_{k=1}^n k &= 1 + 2 + \cdots + n \\ &= \frac{n(n+1)}{2} \end{aligned} \quad (3)$$

Mathematik – Unterschiede bei der Darstellung

- Formeln werden innerhalb der `math`-Umgebung anders dargestellt als in der `displaymath`-Umgebung

$\text{\LaTeX}-\text{Befehl}$	<code>math</code>	<code>displaymath</code>
<code>\lim_{x \rightarrow \infty} \rightarrow 0</code>	$\lim_{x \rightarrow \infty} \rightarrow 0$	$\lim_{x \rightarrow \infty} \rightarrow 0$

- mit dem Befehl `\displaystyle` bzw. `\textstyle` kann das jeweils andere Verhalten erzwungen werden

Beispiel: `textstyle` und `displaystyle`

Es gilt: `$\lim_{x \rightarrow 0} = \{\displaystyle \lim_{x \rightarrow 0}\}$,`
`\[\lim_{x \rightarrow 0} = \{\textstyle \lim_{x \rightarrow 0}\}. \]`

Es gilt: $\lim_{x \rightarrow 0} = \lim_{x \rightarrow 0},$
 $\lim_{x \rightarrow 0} = \lim_{x \rightarrow 0}.$

Mathematik – Symbole

Logik

\exists	\exists	\forall	\forall	\neg	\neq	\in	\in	\emptyset	\emptyset
\not\in	\notin	\ni	\ni	\land	\wedge	\lor	\vee	\varnothing	\emptyset
\setminus	\backslash	\implies	\implies	\iff	\iff	\rightarrow	\rightarrow	\top	\top

Pfeile

\leftarrow	\leftarrow	\rightarrow	\rightarrow	\uparrow	\uparrow	\downarrow	\downarrow
\Leftarrow	\Leftarrow	\Rightarrow	\Rightarrow	\Updownarrow	\Updownarrow	\Downarrow	\Downarrow
\leftrightsquigarrow	\leftrightsquigarrow	\Leftrightsquigarrow	\Leftrightsquigarrow	\nearrow	\nearrow	\searrow	\searrow
\updownarrow	\updownarrow	\Updownarrow	\Updownarrow	\swarrow	\swarrow	\nwarrow	\nwarrow
\longleftarrow	\longleftarrow	\Longleftarrow	\Longleftarrow	\mapsto	\mapsto	\leadsto	\leadsto
\longrightarrow	\longrightarrow	\Longrightarrow	\Longrightarrow				

Mathematik – Symbole

Sonstige

\partial	∂	\nabla	∇	\infty	∞	\ell	ℓ	\imath	\imath
\jmath	\jmath	\Re	\Re	\Im	\Im	\mathfrak{I}	\mathfrak{I}	\cdots	\cdots
\vdots	\vdots	\ddots	\ddots	\ldots	\ldots	\pm	\pm	\mp	\mp

Mathematik – Griechische Buchstaben

\alpha	α	\xi	ξ	\Gamma	Γ
\beta	β	\pi	π	\Delta	Δ
\gamma	γ	\varpi	ϖ	\Theta	Θ
\delta	δ	\rho	ρ	\Lambda	λ
\epsilon	ϵ	\varrho	ϱ	\Xi	Ξ
\varepsilon	ε	\sigma	σ	\Pi	Π
\zeta	ζ	\varsigma	ς	\Sigma	Σ
\eta	η	\tau	τ	\Upsilon	Υ
\theta	θ	\upsilon	υ	\Phi	Φ
\vartheta	ϑ	\phi	ϕ	\Psi	Ψ
\iota	ι	\varphi	φ	\Omega	Ω
\kappa	κ	\chi	χ		
\lambda	λ	\psi	ψ		
\mu	μ	\omega	ω		
\nu	ν				

Mathematik – binäre Operatoren

<code>\leq</code>	\leq	<code>\geq</code>	\geq	<code>\equiv</code>	\equiv
<code>\models</code>	\models	<code>\prec</code>	\prec	<code>\succ</code>	\succ
<code>\sim</code>	\sim	<code>\perp</code>	\perp	<code>\preceq</code>	\preceq
<code>\succeq</code>	\succeq	<code>\simeq</code>	\simeq	<code>\mid</code>	\mid
<code>\ll</code>	\ll	<code>\gg</code>	\gg	<code>\asymp</code>	\asymp
<code>\parallel</code>	\parallel	<code>\subset</code>	\subset	<code>\supset</code>	\supset
<code>\approx</code>	\approx	<code>\bowtie</code>	\bowtie	<code>\subseteqq</code>	\subseteqq
<code>\supseteqq</code>	\supseteqq	<code>\cong</code>	\cong	<code>\sqsubset</code>	\sqsubset
<code>\sqsupset</code>	\sqsupset	<code>\neq</code>	\neq	<code>\smile</code>	\smile
<code>\sqsubseteqq</code>	\sqsubseteqq	<code>\sqsupseteqq</code>	\sqsupseteqq	<code>\doteq</code>	\doteq
<code>\frown</code>	\frown	<code>\in</code>	\in	<code>\ni</code>	\ni
<code>\proto</code>	\propto	<code>=</code>	$=$	<code>\vdash</code>	\vdash
<code>\dashv</code>	\dashv	<code><</code>	$<$	<code>></code>	$>$

Mathematik – Negation von Operatoren

- oftmals kann ein Operator negiert werden, indem der Befehl `\not` vorgestellt wird

$$\not< \neq | \not\leq \not\leq | \not\subseteq \not\subseteq$$

- einige Operatoren haben dafür jedoch spezielle Symbole

$$\not= \neq | \not\in \not\in$$

Mathematik – gestapelte Operatoren

- manchmal werden Anmerkungen an Relationen geschrieben oder eigene Operatoren definiert
- dazu eignet sich der `\stackrel` Befehl

Beispiel: `\stackrel` Befehl

Zeigen Sie:

```
\[ f(x) > 0 \stackrel{!}{\stackrel{\iff}{\rightarrow}} f'(x) \geq 0 \]
\[ x \stackrel{\text{inj.}}{\longmapsto} g(x) \]
```

Zeigen Sie:

$$\begin{aligned} f(x) > 0 &\stackrel{!}{\iff} f'(x) \geq 0 \\ x &\stackrel{\text{inj.}}{\longmapsto} g(x) \end{aligned}$$

Mathematik – Funktionen

\arccos \arcsin \arctan \arg \cos
\cosh \cot \coth \csc \deg
\det \dim \exp \gcd \hom
\inf \ker \lg \lim \liminf
\limsup \ln \log \max \min
\Pr \sec \sin \sinh \sup
\tan \tanh

Mathematik – Hoch- und Tiefstellung

- um Exponenten oder Indizes anzugeben, benutzt man zur Hochstellung das Zeichen \wedge und zur Tiefstellung das Zeichen \backslash
- möchte man mehr als ein Zeichen hoch- oder tiefstellen, muss man den Term in geschweiften Klammern angeben
- die Reihenfolge ist egal
- mit den Befehlen `\limits` und `\nolimits` kann man die Darstellung für das Hoch- bzw. Tiefstellen beinflussen

Beispiel: Hoch- und Tiefstellung

```
\[ a_i^{\wedge n} \cdot a^{\{n^2+1\}}_{\backslash j\_ell}\]
```

$$a_i^n \cdot a_{j_\ell}^{n^2+1}$$

Mathematik – Brüche

- der `\frac` Befehl dient der Darstellung von Brüchen und erfordert zwei Argumente

Beispiel: Brüche

```
\begin{align*}
    \frac{1}{2} < \frac{n+1}{n}
    \\
    \frac{1}{1 + \frac{1}{\frac{1}{1+x}}}
    \\
    \frac{1}{1 + \frac{1}{x}}
\end{align*}
```

$$\frac{1}{2} < \frac{n+1}{n}$$
$$\frac{1}{1 + \frac{1}{\frac{1}{1+x}}}$$
$$\frac{1}{1 + \frac{1}{x}}$$

Mathematik – Wurzel und Binomialkoeffizient

Wurzel

- der Wurzelbefehl heißt `\sqrt`
- optional kann der Parameter `n` angeben werden ($n = 2$: Quadratwurzel)

Binomialkoeffizient

- zur Darstellung des Binomialkoeffizienten dient der Befehl `\choose`
- das erste Argument wird oben, das zweite unten gesetzt

Beispiel: Wurzeln, Binomialkoeffizienten

```
\[
    \sqrt{16} = \sqrt[2]{16}
    = {4 \choose 1}
\]
```

$$\sqrt{16} = \sqrt[2]{16} = \binom{4}{1}$$

Mathematik – Klammerung

- Klammerzeichen: () [] {} || \langle\rangle \lfloor\rfloor \lfloor\rfloor \lceil\rfloor \rceil

- die Größe der Klammern kann manuell durch die Befehle \big, \Big, \bigg, \Bigg variiert werden

$$((\left(\right)))$$

- besser: die Klammergröße kann automatisch gesetzt werden
- dazu setzt man den einzuklammernden Begriff zwischen die Befehle \left und \right; direkt darauf folgt das zu verwendende Klammerzeichen
- möchte man auf einer Seite keine Klammern haben, verwendet man als Klammerzeichen . (Punkt)
- Beispiel: \left\{ 1 - \left| \frac{1}{2} \right| \dots \right.

$$\left\{ 1 - \left| \frac{1}{2} \right| \dots$$

Mathematik – Matrizen

- die `array`-Umgebung entspricht der `tabular`-Umgebung im Mathematik-Modus
- die Inhalte der Zellen sind automatisch im Mathematik-Modus gesetzt
- bekannte Befehle für Tabellen (z. B. `\multicolumn`, `\hline`, ...) können verwendet werden

Beispiel: Matrizen

```
\[ \left[ \begin{array}{ccc}
  a & b & c \\
  d & e & f \\
  g & i & h
\end{array} \right] \]
```

$$\left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & i & h \end{array} \right]$$

Mathematik – Fallunterscheidungen

- für Fallunterscheidungen steht die `cases`-Umgebung zur Verfügung
- kann auch mittels Klammerung und der `array`-Umgebung selbst erstellt bzw. individualisiert werden

Beispiel: Fallunterscheidungen

```
\[
  x = \begin{cases}
    0, & \text{falls } y \leq 0 \\
    y^2, & \text{sonst}
  \end{cases}
\]
```

$$x = \begin{cases} 0, & \text{falls } y \leq 0 \\ y^2, & \text{sonst} \end{cases}$$

Mathematik – Schriftgröße

- die Schriftgröße kann im Mathematik-Modus durch folgende Befehl geändert werden:

\displaystyle $\frac{1}{2}$

\textstyle $\frac{1}{2}$

\scriptstyle $\frac{1}{2}$

Mathematik – Akzente

a'	a'	a''	a''	a'''	a'''
\bar{a}	\bar{a}	\overline{a}	\bar{a}	\underline{a}	\underline{a}
\hat{a}	\hat{a}	\widehat{a}	\widehat{a}	\check{a}	\check{a}
\tilde{a}	\tilde{a}	\widetilde{a}	\widetilde{a}	\vec{a}	\vec{a}
\dot{a}	\dot{a}	\ddot{a}	\ddot{a}		

Mathematik – horizontale Abstände

Befehl	Breite	Beschreibung
\qquad	—	$2 \times \text{quad}$
\quad	—	so breit wie ein Zeichen hoch ist
(Leerzeichen)	-	Zeichenabstand
\,	-	$\frac{3}{18} \times \text{quad}$
\:	-	$\frac{4}{18} \times \text{quad}$
\;	-	$\frac{4}{18} \times \text{quad}$
\!	-	$-\frac{3}{18} \times \text{quad}$

Beispiel: horizontaler Abstand

```
\[
  \int_0^{\pi} \sin(x); \mathrm{d}x
\]
```

$$\int_0^{\pi} \sin(x) \, dx$$

Aufgabe 7 (Formelsatz)

Erstellen Sie folgenden Text:

Sei I ein reelles Intervall und $f : I \rightarrow \mathbb{R}$ eine $(n + 1)$ -mal stetig differenzierbare Funktion. Dann gilt für alle $a, x \in I$: $f(x) = T_n(x) + R_n(x)$ mit dem n -ten Taylorpolynom an der Entwicklungsstelle a

$$\begin{aligned}T(x) &= \sum_{k=0}^n \frac{f^{(k)}(x)}{k!}(x - a)^k \\&= f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \cdots + \frac{f^{(n)}(x)}{n!}(x - a)^n\end{aligned}$$

und dem n -ten Restglied

$$R_n(x) = \int_a^x \frac{(x - t)^n}{n!} f^{(n+1)}(t) dt$$

In den Formeln stehen f' , f'' , ..., $f^{(n)}$ für die erste, zweite, ..., n -te Ableitung der Funktion f .

Hinweis: Verwenden Sie zur Erzeugung der Mengensymbole \mathbb{R} und \mathbb{N} den im Paket `amssymb` enthaltenen Befehl `\mathbb{R}` und für die ausgerichteten Gleichungen die Umgebung `align` aus dem `amsmath` Paket.

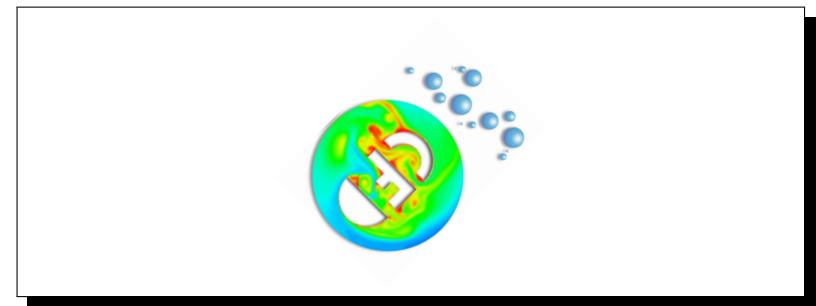
Grafiken

- Grafiken können mit dem Befehl `\includegraphics` eingebunden werden
- dieser benötigt das Paket `graphicx`, dass in der Präambel geladen werden muss:
`\usepackage{graphicx}`
- benutzt man `pdflatex` zum Kompilieren des Dokuments, so müssen die verwendeten Grafiken in PDF- oder Bitmap-Grafiken umgewandelt werden
- alternativ kann man auch das Paket `epstopdf` verwendet, dann muss `pdflatex` aber so gestartet werden:
`pdflatex -shell-escape`

Grafiken – Beispiel

Beispiel: `\includegraphics`

```
\begin{center}
  \includegraphics[width=3cm ,
    angle=225]{examples/
  Grafiken_Beispiel/cfd.pdf}
\end{center}
```



Optionale Parameter

`width=b`

Breite vorgeben (z. B. in `cm`)

`height=h`

Höhe vorgeben (z. B. in `cm`)

`keepaspectratio=k`

Seitenverhältnis beibehalten (`true` oder `false`)

`scale=s`

skalieren (Skalierungsfaktor angeben)

`angle=a`

rotieren (Winkel in Grad angeben)

`trim=l b r t`

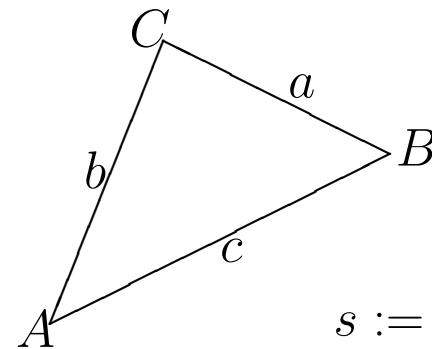
Bild zuschneiden

Grafiken – picture-Umgebung

Beispiel: picture-Umgebung

```
\setlength{\unitlength}{1.5cm}
\begin{picture}(6,5)
\thicklines
\put(1,0.5){\line(2,1){3}}
\put(4,2){\line(-2,1){2}}
\put(2,3){\line(-2,-5){1}}
\put(0.7,0.3){$A$}
\put(4.05,1.9){$B$}
\put(1.7,2.95){$C$}
\put(3.1,2.5){$a$}
\put(1.3,1.7){$b$}
\put(2.5,1.05){$c$}
\put(0.3,4){$F=\sqrt{s(s-a)(s-b)(s-c)}$}
\put(3.5,0.4){$\displaystyle s:=\frac{a+b+c}{2}$}
\end{picture}
```

$$F = \sqrt{s(s-a)(s-b)(s-c)}$$



$$s := \frac{a + b + c}{2}$$

Gleitende Objekte (Floats)

- gleitende Objekte müssen nicht an der Stelle im Text auftauchen, an der sie definiert wurden
- L^AT_EX entscheidet “selbstständig” wo sie plaziert werden
- Beispielumgebungen (die gleitende Objekte darstellen):
 - `\begin{figure}[htbp] ... \end{figure}`
 - `\begin{table}[htbp] ... \end{table}`
- optional kann die gewünschte Platzierung angegeben werden:
 - h** *here* Objekt dort einbauen, wo es definiert wird
 - t** *top* Objekt am Anfang der Seite platzieren
 - b** *bottom* Objekt am Ende der Seite platzieren
 - p** *page* Gleitobjekte auf einer Seite sammeln
- zur Beschriftung dient der `\caption`-Befehl

Inhaltsverzeichnis ausgeben

- um Standardverzeichnisse auszugeben, muss man einfach die entsprechenden Schlüsselwörter angeben:
 - \tableofcontents
 - \listoffigures
 - \listoftables
- die maximale Tiefe des Inhaltsverzeichnisses kann mit folgendem Befehl geändert werden:
 - \setcounter{tocdepth}{1}
- manuelle Einträge können so eingefügt werden:
 - \addcontentsline{toc}{subsection}{Titel}
- mögliche Werte für das Zielverzeichnis:

toc	<i>table of contents</i>	chapter, section, subsection subsubsection, paragraph
lof	<i>list of figures</i>	figure
lot	<i>list of tables</i>	table

Querverweise

■ Befehle zur Benutzung von Querverweisen:

- \label{marker}** Verbindet die momentane Textstelle mit **marker**. So kann von einer anderen Textstelle aus auf diese Stelle verwiesen werden.
- \ref{marker}** Erzeugt einen Querverweis auf eine Stelle, die zuvor mittels **label** gekennzeichnet wurde. Der Querverweis gibt die Gliederungsnummer der betreffenden Textstelle an.
- \pageref{marker}** Wie **\ref**, gibt jedoch die Seitennummer der Textstelle zurück.

Querverweise

- Durch ein Präfix (optional) kann angegeben werden, was referenziert wird, z. B.:
`\label{sec:Querverweise}`
- mögliche Präfixe sind:

<code>chap</code>	chapter
<code>sec</code>	section
<code>fig</code>	figure
<code>tab</code>	table
<code>eq</code>	equation
<code>lst</code>	listing

- in `figure`- und `table`-Umgebung muss das Label innerhalb des `\caption`-Befehls gesetzt werden

Querverweise – Beispiel

Querverweise – Beispiel

```
\begin{equation}
    \label{eq:pythagoras}
    a^2 + b^2 = c^2
\end{equation}
Satz des Pythagoras (siehe \ref{
    eq:pythagoras})
```

$$a^2 + b^2 = c^2 \quad (4)$$

Satz des Pythagoras (siehe 4)

Aufgabe 8 (Inhaltsverzeichnis und Querverweise)

Ändern Sie das Dokument aus Aufgabe 2 so ab, dass eine Titelseite und ein Inhaltsverzeichnis ausgegeben wird. Bauen Sie einige Querverweise mit den Befehlen `\label`, `\ref` und `\pageref` in das Dokument ein.

Aufgabe 9 (Bilder)

Binden Sie ein Bild in ein das L^AT_EX-Dokument aus Aufgaben 4 ein. Es soll eine Breite von `6 cm` haben und in einer Figure-Umgebung mit Bezeichnung stehen. Verändern Sie die Positionierung der gleitenden Umgebung `figure`. Erstellen Sie einen Querverweis auf das Bild.

Eigene L^AT_EX-Befehle

- eigene Befehle können mittels `\newcommand` angelegt werden
- allgemeine Form:
 - `\newcommand{befehlsname}{definition}`
 - `\newcommand{befehlsname}[n]{definition}`
 - `\newcommand{befehlsname}[n][default]{definition}`

befehlsname Name des Befehls (muss mit \ beginnen)
n Anzahl der Parameter
default Vorgabewert für optionale Parameter
definition alles was beim Aufruf ausgeführt werden soll
- mit `\renewcommand` kann ein bereits vorhandener Befehl ersetzt werden

Eigene L^AT_EX-Befehle – Beispiel

Eigene L^AT_EX-Befehle – Beispiel

```
\newcommand{\ds}{\ensuremath{\\
  displaystyle}}
Bruch in Display-Style: \$\ds \frac{a}{b} \$\\

\newcommand{\meinbefehl}[1]{\\
  \texttt{\textbackslash#1}}
\meinbefehl{meinbefehl}\\

\newcommand{\todo}[2]{\colorbox{orange}{\\
  \textbf{\#1}: \#2}}
\todo{ToDo-Box}{Das ist eine ToDo-Box.}
```

Bruch in Display-Style: $\frac{a}{b}$

\meinbefehl

ToDo-Box: Das ist eine ToDo-Box.

Eigene L^AT_EX-Umgebungen

- eigene Umgebungen können mittels `\newenvironment` angelegt werden
- allgemeine Form:

- `\newenvironment{umgebung}{vorher}{nachher}`
- `\newenvironment{umgebung}[n]{vorher}{nachher}`
- `\newenvironment{umgebung}[n][default]{vorher}{nachher}`

umgebung Name der Umgebung (ohne \)

n Anzahl der Parameter

vorage Vorgabewert für optionale Parameter

vorher Befehle, die vor Beginn der Umgebung ausgeführt werden

nachher Befehle, die nach Ende der Umgebung ausgeführt werden

- mit `\renewenvironment` kann eine bereits vorhandene Umgebung ersetzt werden

Eigene \LaTeX -Umgebungen – Beispiel

Eigene \LaTeX -Umgebungen – Beispiel

```
\newenvironment{determinante}[1]{  
    \ensuremath\left|\begin{array}{#1}\right.{  
    \end{array}\right|}  
\[  
    \begin{determinante}{ll}  
        1 & 2 \\ 3 & 4  
    \end{determinante}  
\]
```

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$$

Eigene Zähler

\newcounter{zaehler}	neuen Zähler namens <code>zaehler</code> anlegen
\newcounter{zaehler}[depend]	neuen Zähler namens <code>zaehler</code> anlegen, der bei Veränderung von <code>depend</code> zurückgesetzt wird
\stepcounter{zaehler}	Zähler <code>zaehler</code> um eins erhöhen
\setcounter{zaehler}{wert}	Zähler <code>zaehler</code> auf Wert <code>wert</code> setzen
\addtocounter{zaehler}{wert}	Zähler <code>zaehler</code> um <code>wert</code> erhöhen
\value{zaehler}	den Wert von <code>zaehler</code> auslesen
\arabic{zaehler}	Ziffer 1, 2, 3, 4, ...
\roman{zaehler}	Römische Zahl (klein) i, ii, iii, iv, ...
\Roman{zaehler}	Römische Zahl (groß) I, II, III, IV, ...
\alph{zaehler}	Kleinbuchstaben a, b, c, d, ...
\Alpha{zaehler}	Großbuchstaben A, B, C, D, ...

Vordefinierte Zähler

Dokument page

Kapitel part, chapter, section, subsection, subsubsection
paragraph, subparagraph

Aufzählung enumi, enumii, enumiii, enumiv

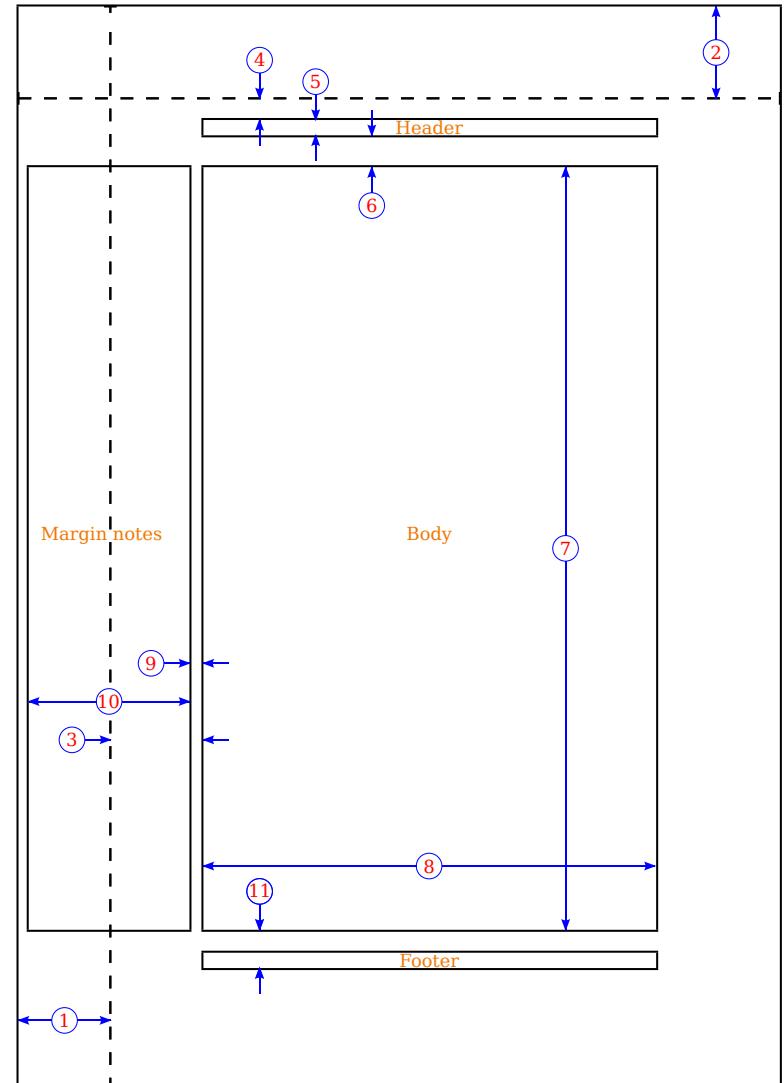
Umgebungen equation, figure, table, footnote

Eigene Längen

\newlength{\laenge}	neue Längenvariable anlegen
\setlength{\laenge}{x}	Länge auf x setzen
\addtolength{\laenge}{x}	Länge um x erhöhen
\settowidth{\laenge}{text}	Länge von text speichern
\settoheight{\laenge}{text}	Höhe von text speichern
\settodepth{\laenge}{text}	Größter Abstand zur Basislinie

Vordefinierte Längen

- 1 \hoffset
- 2 \voffset
- 3 \oddsidemargin
- 4 \topmargin
- 5 \headheight
- 6 \headsep
- 7 \textheight
- 8 \textwidth
- 9 \marginparsep
- 10 \marginparwidth
- 11 \marginpar
- 12 \footskip
- \paperwidth, \paperheight



Theorem-Umgebung

- der `\newtheorem` Befehl dient der Erzeugung von Umgebungen für Theoreme, Sätze, Definitionen etc.

- `\newtheorem{name}{beschriftung}`
 - `\newtheorem{name}{beschriftung} [zaehler]`

name Name der Theorem-Umgebung

beschriftung die Bezeichnung der Umgebung im Dokument

zaehler der Zähler der zur Nummerierung verwendet wird

Theorem-Umgebung – Beispiel

Theorem-Umgebung – Beispiel

```
\newcounter{zsaetze}[section]
\newtheorem{defi}{Definition}[
  zsaetze]
\newtheorem{satz}{Satz}[zsaetze]

\begin{defi}[Taylorpolynom] ... \
\end{defi}
\begin{satz} ... \end{satz}
\begin{satz} ... \end{satz}
```

Definition (Taylorpolynom)

...

Satz

...

Satz

...

Bibliographie

- L^AT_EX stellt mit BibTeX ein sehr mächtiges System zur Verwaltung von Literaturverweisen bereit.
 - Um BibTeX nutzen zu können, muss zunächst eine Datenbank angelegt werden.
 - Die Datenbank ist eine einfache Textdatei mit Einträgen für die verschiedenen zitierten Quellen.
 - allgemeiner Aufbau eines Eintrags:
 - @literaturtyp{kennung, name1="Wert1", name2="Wert2", ...}
- literaturtyp
- spezifiziert die Art der Quelle
unterschiedliche Eintragstypen erfordern unterschiedliche Angaben zur Quelle
- kennung
- damit kann auf den Eintrag referenziert werden
- name="wert"
- Zuweisung von Werten an die verschiedenen Felder

Bibliographie – Unterstützte Literatur-Typen

<code>article</code>	Veröffentlichung in einer Zeitschrift
<code>book</code>	Buch
<code>booklet</code>	Buch ohne Verleger
<code>inbook</code>	Teil eines Buches
<code>incollection</code>	Teil einer Buchreihe
<code>inproceedings</code>	Teil einer Veröffentlichung zu einer Konferenz
<code>manual</code>	Technische Dokumentation
<code>masterthesis</code>	Masterarbeit
<code>phdthesis</code>	Doktorarbeit
<code>proceedings</code>	Veröffentlichung zu einer Konferenz
<code>techreport</code>	Technischer Bericht
<code>unpublished</code>	unveröffentlicht
<code>misc</code>	falls alles andere nicht passt

Bibliographie – Unterstützte Felder

- address** die Adresse des Verlags oder einer anderen Institution
- annotate** Anmerkungen
- author** Namen der Autoren (in BibTeX Format)
 - mehrere Namen werden durch **AND** getrennt
 - zwei Möglichkeiten Namen zu schreiben:
Donald E. Knuth oder Knuth, Donald E.
- booktitle** Titel des Buchs
- chapter** Kapitel- oder Abschnitt-Nummer
- crossref** Datenbank-Schlüssel
- edition** Auflage (z. B. eines Buchs)
- editor** Namen der Editoren (analog dem **author**-Feld)
- howpublished** Veröffentlichungsart
- institution** fördernde Institution eines technischen Reports
- journal** Zeitschriftenname (häufig abgekürzt)

Bibliographie – Unterstützte Felder

- key** Feld zur Sortierung und Erstellung von Labels
- month** Monat der Veröffentlichung/Erscheinung
- note** zusätzliche Information
- number** Nummer einer Zeitschrift, eines Reports oder eines Bandes
- organization** fördernde Organisation
- pages** Seitenzahlen oder Seitenzahlbereich (z. B. 7–33)
- publisher** Name des Verlags
- school** Name der “Schule”, an der eine Abschlussarbeit geschrieben wurde
- series** Name einer Reihe
- title** Titel des Werkes
- type** Typ eines technischen Reports
- volume** Band einer Zeitschrift oder eines Buches
- year** Jahr der Veröffentlichung/Erscheinung

Bibliographie einbinden

- verschiedene Felder sind bei verschiedenen Literaturtypen vorgeschrieben
- es gibt noch weitere Felder wie `ISBN`, `doi`, `abstract`, ...
- Sonderzeichen und Umlaute müssen speziell maskiert werden
- BibTeX muss eigens aufgerufen werden:

`latex Beispiel`

`bibtex Beispiel`

`latex Beispiel`

`latex Beispiel`

- BibTeX prüft die Syntax der Einträge und erzeugt eine Datei, die in das \LaTeX -Dokument eingebunden werden kann
- nach Aufruf von BibTeX muss \LaTeX die Datei zwei mal kompilieren, damit alle Referenzen korrekt gesetzt werden

Bibliographie einbinden

- um die Bibliographie anzuzeigen, verwendet man den folgenden Befehl:
`\bibliography{literatur}`
- `literatur` bezeichnet dabei die Datei mit den entsprechenden Literaturangaben
- aufgelistet werden alle Quellen, die im Dokument zitiert wurden (mittels dem Befehl `\cite{kennung}`)
- durch `\nocite{kennung}` wird auch der `kennung` entsprechende Eintrag aufgeführt, auch wenn er nicht zitiert wurde
- die Formatierung hängt von `\bibliographystyle` ab
- es gibt verschiedene Pakete zur Gestaltung der Literaturverweise, z. B. `natbib`

Bibliographie mit `natbib`

- `natbib` wird durch `\usepackage{natbib}` (in der Präambel) eingebunden
- als Format wird dann `plainnat` gewählt: `\bibliographystyle{plainnat}`
- `natbib` unterstützt mehrere zusätzliche Literaturangaben wie

`ISBN` ISBN-Nummer eines Buches

`ISSN` ISSN-Nummer einer Zeitschrift

`URL` Internet-Adresse für Online-Dokumente

`DOI` Digital Object Identifier

Zitieren mit `natbib`

```
\citet{knuth93}  
\citep{knuth93}  
\citep[siehe]{knuth93}  
\citeauthor{knuth93}  
\citeyear{knuth93}
```

Knuth et al. (1993)
(Knuth et al., 1993)
(siehe Knuth et al., 1993, S. 13)
Knuth et al.
1993

\LaTeX Beamer

- Klasse zur Erstellung von Präsentationen mit **\LaTeX**
- Benutzerhandbuch zur Klasse:
<http://ftp.fau.de/ctan/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
- Grundkonzept: eine **frame** entspricht einer Seite (Folie)
- innerhalb der Frames können (fast alle) normalen **\LaTeX**-Befehle verwendet werden

\LaTeX Beamer – Präambel

\LaTeX Beamer – Beispiel - Präambel

```
\documentclass{beamer}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}

\usetheme{Luebeck}
\usecolortheme{orchid}
\usefonttheme{default}
\useinnertheme{rounded}
\useoutertheme{shadow}
```

LaTeX Beamer – Hello World

Hello World Frame

```
\documentclass{beamer}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\begin{document}
    \begin{frame}
        \frametitle{Hello world!}
        \begin{itemize}
            \item ABC
        \end{itemize}
    \end{frame}
\end{document}
```



LATEX Beamer – Frames

LATEX Beamer – Frames – Allgemeine Form

```
\begin{frame}[Overlay][Optionen]{Titel}{Untertitel}  
    Inhalt  
\end{frame}
```

Overlay <+> sorgt dafür, dass Listen schrittweise aufgebaut werden

Optionen Anzeigeoptionen für diese Folie (mehrere Optionen müssen durch Kommas abgetrennt werden)

Titel Titel der Folie; kann auch mit `\frametitle` gesetzt werden

Untertitel Untertitel der Folie; kann auch mit `\framesubtitle` gesetzt werden

LaTeX Beamer – Frames – Optionen

- t** Ausrichtung des Inhalts oben (*top*)
- c** Ausrichtung des Inhalts mittig (*center*)
- b** Ausrichtung des Inhalts unten (*bottom*)

label=name Label für die Folie setzen

plain Kopf- und Fußzeile unterdrücken

squeeze Inhalt zusammenrücken

fragile notwendig für Folien mit Quelltext (**verbatim**)

LaTeX Beamer – Blöcke

Block

erzeugt durch:

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

Beispiel

erzeugt durch `exampleblock`-Umgebung

Wichtig

erzeugt durch `alertblock`-Umgebung

- nützlich um thematisch zusammenzufassen
- das Aussehen variiert je nach Themenvorlage und eigenen Einstellungen

LaTeX Beamer – Spalten

Spalte 1

...

Quellcode

```
\begin{columns}
  \column{0.5\textwidth}{%
    \begin{block}{Spalte 1}
      ...
    \end{block}
  \column{0.5\textwidth}{%
    \begin{block}{Spalte 2}
      ...
    \end{block}
  \end{columns}
```

Spalte 2

...

- auch mehr als zwei Spalten möglich

\LaTeX Beamer – Overlays

- Frames können mehrere Overlays enthalten.
- Overlays sorgen dann für das stückweise "Aufbauen" einer Folie.
- Der **\LaTeX**-Seitenzähler wird dabei angehalten.
- Inhalte können nach und nach Erscheinen oder nur zu bestimmten Zeiten sichtbar sein.

Quelltext

```
Erster Teil  
\pause \\  
Zweiter Teil
```

Vorschau

Erster Teil

\LaTeX Beamer – Overlays

- Frames können mehrere Overlays enthalten.
- Overlays sorgen dann für das stückweise "Aufbauen" einer Folie.
- Der **\LaTeX**-Seitenzähler wird dabei angehalten.
- Inhalte können nach und nach Erscheinen oder nur zu bestimmten Zeiten sichtbar sein.

Quelltext

```
Erster Teil
\pause \\
Zweiter Teil
```

Vorschau

```
Erster Teil
Zweiter Teil
```

LaTeX Beamer – Overlays – Beispiele Quellcode

Beispiele für Overlay Steuerung

```
\begin{itemize}
  \visible<1>{\item Dieser Text erscheint nur auf Overlay 1.}
  {\color<1-3>{red}{\item Dieser Text ist auf Overlays 1 bis 3
    rot.}}
  {\color<2->{blue}{\item Dieser Text ist ab Overlay 2 blau.}}
  \only<-3>{\item Dieser Text erscheint nur bis Overlay 3.}
  \textbf<1,3,5>{\item Dieser Text erscheint auf Overlays 1, 3
    und 5 im Fettdruck.}
  \alt<2>{\item Dieser Text erscheint nur auf Overlay 2.}{\item
    Sonst erscheint dieser Text.}
\end{itemize}
```

LaTeX Beamer – Overlays – Beispiele Quellcode

Overlay 1 / 5

- Dieser Text erscheint nur auf Overlay 1.
- Dieser Text ist auf Overlays 1 bis 3 rot.
- Dieser Text ist ab Overlay 2 blau.
- Dieser Text erscheint nur bis Overlay 3.
- Dieser Text erscheint auf Overlays 1, 3 und 5 im Fettdruck.
- Sonst erscheint dieser Text.

LaTeX Beamer – Overlays – Beispiele Quellcode

Overlay 2 / 5

- Dieser Text ist auf Overlays 1 bis 3 rot.
- Dieser Text ist ab Overlay 2 blau.
- Dieser Text erscheint nur bis Overlay 3.
- Dieser Text erscheint auf Overlays 1, 3 und 5 im Fettdruck.
- Dieser Text erscheint nur auf Overlay 2.

\LaTeX Beamer – Overlays – Beispiele Quellcode

Overlay 3 / 5

- Dieser Text ist auf Overlays 1 bis 3 rot.
- Dieser Text ist ab Overlay 2 blau.
- Dieser Text erscheint nur bis Overlay 3.
- **Dieser Text erscheint auf Overlays 1, 3 und 5 im Fettdruck.**
- Sonst erscheint dieser Text.

LaTeX Beamer – Overlays – Beispiele Quellcode

Overlay 4 / 5

- Dieser Text ist auf Overlays 1 bis 3 rot.
- Dieser Text ist ab Overlay 2 blau.
- Dieser Text erscheint auf Overlays 1, 3 und 5 im Fettdruck.
- Sonst erscheint dieser Text.

LaTeX Beamer – Overlays – Beispiele Quellcode

Overlay 5 / 5

- Dieser Text ist auf Overlays 1 bis 3 rot.
- Dieser Text ist ab Overlay 2 blau.
- **Dieser Text erscheint auf Overlays 1, 3 und 5 im Fettdruck.**
- Sonst erscheint dieser Text.

Aufgabe 10 (Präsentationen)

Erstellen Sie eine einfache Präsentation mit mehreren Folien. Benutzen Sie dazu das \LaTeX `beamer`-Paket. Folgende Merkmale sollen in der Präsentation enthalten sein:
Titelseite, Aufzählungen, Blöcke, zwei Spalten auf einer Folie und Overlays.

Literaturangaben

- Thomas Rohkämper (TU Dortmund): *Einführung in LATEX*, Sommersemester 2014
 - Deutschsprachige Anwendervereinigung TeX e. V.
<http://www.dante.de>
 - Helmut Kopka. *LATEX, Bd. 1: Einführung*. Pearson Studium, 2003.
 - Joachim Schlosser. *Wissenschaftliche Arbeiten schreiben mit LATEX: Leitfaden für Einsteiger*. mitp, 2009.
 - Petra Schlager und Manfred Thibud. *Wissenschaftlich mit LATEX arbeiten*. Pearson Studium, 2005.
 - Wikibooks: LaTeX
<http://en.wikibooks.org/wiki/LaTeX>
- Thomas Rohkämper (TU Dortmund): *Einführung in LATEX – Präsentationen mit beamer*, Sommersemester 2014