

L^AT_EX

Jonas Abeln, Jörg Harney und Mathias Rodemeier

6. März 2011

Inhaltsverzeichnis

1	Einleitung	3
1.1	Was ist überhaupt L ^A T _E X?	3
1.2	Unterschied zu Word	3
2	Das erste eigene Dokument	5
2.1	Allgemeines	5
2.2	Dokumentenklassenbestimmung	5
2.3	Dokument beginnen und unterteilen	5
2.4	Bilder und Graphiken	5
2.5	Hervorheben und Trennen	6
2.6	Dokument beenden	6
2.7	Fertig!	6
3	Dokumentenklassen	7
4	Packete (Usepackages)	8
5	Mathematische Formeln	11
5.1	Grundprinzip	11
5.1.1	Mathematische Formeln im Text	11
5.1.2	Mathematische Formeln in gesonderten Zeilen	11
5.2	Grundrechenarten	13
5.2.1	Addition, Subtraktion, Multiplikation	13
5.2.2	Division	13
5.3	Wurzeln, Exponenten und Indizes	14
5.3.1	Wurzeln	14
5.3.2	Exponenten und Indizes	14
5.4	Matritzen	15
5.5	Symbole, Sonderzeichen und andere mathematische Operatoren	15

6	Tabellen	16
6.1	Grundprinzip	16
6.2	Spaltendefinitionen	16
6.3	Befehle innerhalb der <i>tabular</i> -Umgebung	17
6.4	Zusammenfassen mehrerer Boxen	17
6.4.1	Zusammenfassen von Spalten	17
6.4.2	Zusammenfassen von Zeilen	18
7	Bilder nebeneinander anordnen (subfigure)	20
8	Nützliche Links	21
9	Template für die Seminararbeit	21

1 Einleitung

Herzlich Willkommen zur Einführung in das Textverarbeitungsprogramm \LaTeX . Wahrscheinlich fragen Sie sich jetzt, was ist \LaTeX überhaupt? Genauso ging es unserem Kurs auch als wir 2009 zum ersten mal zum Seminarfach kamen. Natürlich assoziiert man mit Latex erst einmal etwas ganz anderes. Aber dieses unscheinbare und unbekannte Programm sollte es unserem Kurs deutlich einfacher machen unsere Facharbeiten zu verfassen. Wir können nur jedem empfehlen die auch vor der Facharbeit stehen, diese mit \LaTeX zu schreiben. Daher wünschen wir jedem viel Spaß beim Lesen dieses Tutorials!

1.1 Was ist überhaupt \LaTeX ?

\LaTeX ist wie oben schon angedeutet ein Textverarbeitungsprogramm, das aber anders arbeitet als zum Beispiel Microsoft Word. Es ist ein „what-you-mean-is-what-you-get“ Textprogramm, das heißt, dass die Textdatei sozusagen programmiert wird. Daher es werden Befehle eingegeben und das Programm führt diese als Textdatei aus. Auf den ersten Blick mag dies sehr kompliziert klingen, doch es bietet sehr viele Vorteile. Zum Beispiel macht \LaTeX sinnige Seitenumbrüche von selbst oder setzt in einem Inhaltsverzeichnis oder Literaturverzeichnis die Zahlen und Beziehungen von selbst. Außerdem braucht man sich keine weiteren Gedanken über das Hervorheben von Überschriften gemacht werden, da auch diese vom Programm selbst erledigt werden.

1.2 Unterschied zu Word

Durch die oben beschriebenen Vorteile hebt sich \LaTeX von „What-you-see-is-what-you-get“ Programmen wie Word oder Open Office deutlich ab. Bei solchen Textverarbeitungsprogrammen sieht man zwar sofort was später gedruckt wird, jedoch ist die Fehlerquote deutlich höher. Beispiele hierfür wären, dass irgendwo ein Absatz zu groß, ein Seitenumbruch nicht stimmig oder durch einen Fehler sich die Schriftart verändert hat. Außerdem kann die Datei in vielen verschiedenen Formaten herausgegeben werden, wie zum Beispiel als PDF Format. Dieses erlaubt es das Schriftdokument auf fast jeden Rechner zu öffnen, auch wenn dieser nicht über \LaTeX verfügt. Ein weiteres Aus-

gabemöglichkeit ist HTML. So wird es auch schnell möglich Webseiten mit Texten zu füllen. Außerdem gibt \LaTeX bei jeder Ausarbeitung ein Editor-Dokument aus, dass von beinahe allen PC's geöffnet werden kann. Aber der wohl größte Vorteil bei \LaTeX liegt in dem sehr einfachen Formeleditor. Im Gegensatz zu beispielsweise Word, wo es sehr umständlich ist mit dem Formeleditor längere Formeln zu erstellen und es überhaupt nicht möglich ist alle Arten von Formeln darzustellen, ist es vergleichsweise einfach bei \LaTeX . Als Beispiel hierfür:

```
\begin{equation}
f(x)=\sqrt[4]{38} \sin(2x) \frac{2x^2}{9}
\end{equation}
```

Würde im Dokument zu sehen sein als:

$$f(x) = \sqrt[4]{38} \sin(2x) \frac{2x^2}{9} \quad (1)$$

Um etwas dergleichen bei Word darzustellen würde man viel länger brauchen, beziehungsweise sind viele von den bei \LaTeX möglichen Formeln bei Word gar nicht vorhanden (Hierfür gibt es noch ein weiteres Kapitel, welches sich nochmal mit den mathematischen Formeln auseinander setzen wird). Daher sollte bei größeren Texten, Aufsätzen oder Büchern, aber auch bei kurzen gut ausgearbeiteten oder mathematischen Texten, \LaTeX , als Textverarbeitungsprogramm, die erste Wahl sein. Bei kurzen Texten oder schnell zu erledigenden Texten hat natürlich zum Beispiel Word die Nase vorn. Dennoch sollte jeder mal \LaTeX ausprobieren um die wunderbaren Vorteile kennen zu lernen.

2 Das erste eigene Dokument

2.1 Allgemeines

Nachdem man \LaTeX erfolgreich installiert hat, zeigen wir nun wie man sein erstes Dokument erstellt. Allgemein ist zu sagen, dass alle Befehle mit einem Backslash beginnen. Direkt dahinter schreibt man den Befehl. Wenn ein Befehl etwas anzeigen soll wie [author], setzt man dahinter geschwungene Klammern `\{..Text..}` und schreibt den entsprechenden Text in die Klammern. Die grundlegendsten Befehle zur Erstellung eines einfachen Dokumentes werden hier nun erläutert.

2.2 Dokumentenklassenbestimmung

Der erste Befehl ist die Bestimmung der Dokumentenklasse wie Artikel, Brief, Tabelle etc. Wir beginnen hier mit dem Artikel, der einfachsten Form, der Befehl ist wie folgt: `\documentclass{article}`. Danach folgen verschiedene Einstellungen des Zeilenabstand, Rand oder Ähnliches, was wir hier aber der Einfachheit halber erst vernachlässigen.

2.3 Dokument beginnen und unterteilen

Mit dem Befehl `\begin{document}` leitet man den Abschnitt ein um den ersten Text zu verfassen. Auch gibt es hier wieder wichtige Befehle wie `\section{Überschrift}`, `\subsection{Unterüberschrift}`, `\subsubsection{Unterunterüberschrift}`, usw. Damit kann man Abschnitte einteilen und so den Text gliedern. Das Schöne an \LaTeX ist, dass man sich um die Konfiguration keine Gedanken machen muss wie in Word. Man gibt einfach nur den Befehl ein und nachher sieht's klasse aus.

2.4 Bilder und Graphiken

In den Text kann man auch Bilder einfügen, dabei sollte man aber auch darauf achten, dass im Pfad der Graphik keine Leerzeichen vorhanden ist. Am besten speichert man die Graphik direkt innerhalb einer Festplatte (E//: Graphik).

Mit folgenden Befehlen fügt man ein Bild ein:

```
\begin{figure}[h]
\centering
\includegraphics[width=0.80\textwidth]{Quellordner}
\caption{Untertitel}
\label{fig:Titel des Bildes}
\end{figure}
```

Im L^AT_EX-Programm kann man aber auch auf „Einfügen“- „Graphik“ gehen, was das Einfügen einer Graphik erleichtert.

2.5 Hervorheben und Trennen

Mit dem Befehl [clearpage] wird der Rest der Seite freigelassen. Das verhindert ein ungewolltes Trennen von Textabschnitten, die man zusammenbehalten möchte. Um Schriftzüge hervorzuheben gibt es die Möglichkeit sie größer zu schreiben. Dies tut man mit dem Befehl [large]. Dazu schreibt man den zu vergrößerten Schriftzug in die Zeile nach dem Befehl.

2.6 Dokument beenden

Wenn man seinen Text beendet hat, fügt man den Befehl `\end{document}` ein.

2.7 Fertig!

Schon hat man sein erstes Dokument erstellt!

Auf dieser Seite haben wir zur Erleichterung verschiedene Templates (Befehlzusammenstellung), mit denen man die verschiedene Dokumentenklassen erstellen kann, dann muss man nicht immer mühevoll Eigene schreiben.

3 Dokumentenklassen

Bei \LaTeX gibt es verschiedene sogenannte „Dokumentenklassen“, denn es macht einen Unterschied, ob man einen Brief verfassen will oder ob man ein Buch schreiben möchte. Die Dokumentenklasse wird wie folgt angegeben:

```
\documentclass[Optionen]{Klasse}
```

Hier sind einmal die verschiedenen Dokumentenklassen von \LaTeX angegeben:

Klasse	definiert die Dokumentenklasse
Übersicht der Klassen	
<code>article</code>	Artikel (z.B. in wissenschaftlichen Zeitschriften), kurze Berichte, etc.
<code>report</code>	längere Berichte (mit mehreren Kapiteln), Diplomarbeiten, Dissertationen, etc.
<code>book</code>	für Bücher
<code>scrartcl</code> , <code>scrreprt</code> , <code>scrbook</code>	Varianten der o. g. Klassen mit Anpassung an DIN-Papierformate
<code>letter</code>	für Briefe
<code>foils</code>	für Folien oder Präsentationen
Optionen	
Optionen	definiert die Optionen für die bestehende Dokumentenklasse
Übersicht der Optionen	
<code>10pt/11pt/12pt</code>	Option für die normale Schriftgröße (10pt ist etwas kleiner, 12pt größer)
<code>a4paper</code>	Papierformat in DIN-A4 (ohne diesen Zusatz wird amerikanisches Papierformat verwendet)
<code>fleqn</code>	linksbündige mathematische Gleichungen
<code>leqno</code>	Gleichungsnummern auf der linken Seite
<code>titlepage/notitlepage</code>	Festlegung für Titel und Zusammenfassung auf einer eigenen oder gesonderten Seite
<code>onecolumn/twocolumn</code>	ein- oder zweispaltiger Satz (Voreinstellungen: <code>onecolumn</code> ; gilt nicht für Dokumentenklasse “letter“)

4 Packete (Usepackages)

Die sogenannten „Usepackages“ sind ein sehr wichtiger Bestandteil von \LaTeX . Sie dienen dem Programm um weitere Befehle, die es im normalen Betrieb nicht verwendet, zu aktivieren. Daher setzen viele Befehle dementsprechende Usepackages voraus. Wenn in einem Dokument besondere Aktionen, wie zum Beispiel Bild neben Bild, ausgeführt werden sollen, wird das Package „subfigure“ benötigt. Für eine bessere Übersicht werden

Usepackages generell am Anfang des Dokuments aufgeführt. Im Folgenden wird erklärt wie sie eingebunden werden und die meist benutzten Packages aufgelistet.

Zum Benutzen eines Usepackages wird der folgende Befehl benutzt:

```
\usepackage[Optionen]{Packet}
```

Packete: Hier wird der Name des Usepackage eingegeben.

Hier sind einmal die am häufigsten benutzten Usepackages aufgelistet:

<code>alltt</code>	Variante der verbatim-Umgebung
<code>amsmath</code> , <code>amssymb</code>	erweiterter mathematischer Formelsatz
<code>babel[german]</code>	Anpassung für jeweilige Sprachen mit Sprache als „Option“ (hier: <code>german</code> ; für die deutsche Sprache)
<code>color</code>	Unterstützung für die Farbausgabe
<code>dcolumn</code>	für Spalten mit Ausrichtung auf Dezimaltrennzeichen (für <code>array</code> - und <code>tabular</code> -Umgebungen)
<code>fancyhdr</code>	flexible Gestaltung der Kopf- bzw. Fußzeilen
<code>fontenc</code>	für Schriften mit unterschiedlicher Codierung
<code>float</code>	Erzwingt Bilder und Tabellen an der eingegebenen Stelle zu setzen
<code>geometry</code>	manuelle Bestimmung des Layouts mit den Maßen als Optionen
<code>german</code> , <code>ngerman</code>	Anpassung an die traditionelle (<code>german</code>) bzw. die neue Rechtschreibung (<code>ngerman</code>)
<code>graphicx</code>	Einbindung von Graphiken
<code>hyperref</code>	Einbindung von Hyperlinks
<code>inputenc</code>	Deklaration der Zeichenkodierung im Eingabefile
<code>latexsym</code>	Einbindung einiger Symbole
<code>longtable</code>	Einbindung von Tabellen über mehrere Seiten mit automatischem Seitenumbruch
<code>makeidx</code>	Unterstützung zum Erstellen eines Index
<code>multicol</code>	Mehrspaltiger Satz mit Kolumnenausgleich
<code>multirow</code>	wird benötigt um Tabellenzeilen miteinander zu verbinden
<code>subfigure</code>	Wird gebraucht um zwei Bilder nebeneinander zu setzen
<code>textcomp</code>	Einbindung von Schriften mit zusätzlichen Textsymbolen

5 Mathematische Formeln

5.1 Grundprinzip

Im Folgenden wird erklärt, wie man mit \LaTeX mathematische Formeln erstellt. Dazu gibt es grundsätzlich zwei verschiedene Möglichkeiten: Zum Einen kann man mathematische Formeln im Text wiedergeben, zum Anderen kann man sie in einer gesonderten Zeile erscheinen lassen.

5.1.1 Mathematische Formeln im Text

Möchte man mathematische Formeln im Text wiedergeben, so setzt man den Teil, der im Mathematikmodus erscheinen soll, entweder zwischen $\$$ und $\$$, zwischen $\backslash($ und $\backslash)$ oder zwischen $\text{\texttt{begin\{math\}}}$ und $\text{\texttt{end\{math\}}}$. Am einfachsten und empfehlenswerten ist es jedoch die Zeichen $\$$ zu verwenden.

Ein Beispiel

Der Satz des Pythagoras $a^2 + b^2 = c^2$ gibt an, dass die Hypothenuse im rechtwinkligen Dreieck mit Hilfe der beiden Katheten berechnet werden kann.

Diese Formel sieht im \TeX -Dokument so aus:

Der Satz des Pythagoras $\$ a^{\wedge}\{2\} + b^{\wedge}\{2\} = c^{\wedge}\{2\} \$$ gibt an, dass ...

5.1.2 Mathematische Formeln in gesonderten Zeilen

Möchte man mathematische Formeln in eigenen Zeilen erscheinen lassen, so geschieht dies auf folgende Weise:

Die Befehle

$\backslash\text{\texttt{begin\{displaymath\}}}$ und $\backslash\text{\texttt{end\{displaymeth\}}}$

oder

$\backslash[$ und $\backslash]$

geben die Formeln in eigenen Zeilen, aber ohne zugehörigen Gleichungsnummern an.

Die Befehle

`\begin{equation}` und `\end{equation}`

geben die Formeln in eigenen Zeilen und mit zugehörigen Gleichungsnummern an.

Mit den Befehlen `\label` und `\ref` kann sich im Text auf eine jeweilige Gleichung bezogen werden, indem sie durch `\label{z.B. Gleichungsname, Gleichungsnummer}` ein sogenanntes *Label* zugesprochen bekommt. Dies ist im einfachsten Fall die Gleichungsnummer oder der Gleichungsname, um später die Übersicht zu behalten, kann aber auch beliebig umbenannt werden. Der Befehl `label` folgt direkt nach dem Befehl `\begin{equation}`: `\begin{equation} \label{pythagoras}`. Dann wird die Gleichung im Text durch den Befehl `\ref{Gleichungsname, Gleichungsnummer}` angesprochen: Der Satz des Pythagoras `\ref{pythagoras}` gibt an, dass ...

Ein Beispiel

ohne Gleichungsnummer

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

Der Satz des Pythagoras (s.o.) gibt an, dass ...

Diese Formel sieht im T_EX-Dokument so aus:

```
\begin{displaymath}
a^{2} + b^{2} = c^{2}
\end{displaymath}
```

Der Satz des Pythagoras (s.o.) gibt an, dass \ldots

mit Gleichungsnummer

$$a^2 + b^2 = c^2 \tag{2}$$

Der Satz des Pythagoras (2) gibt an, dass ...

Diese Formel sieht im \TeX -Dokument so aus:

```
\begin{equation} \label{pythagoras}
a^{2} + b^{2} = c^{2}
\end{equation}
```

Der Satz des Pythagoras (`\ref{pythagoras}`) gibt an, dass `\ldots`

Die Formeln können auch über das Menü erstellt werden: Dazu wird in der Kopfzeile des TeXnicCenter unter *Einfügen/Formeln* die jeweilige Option ausgesucht.

5.2 Grundrechenarten

5.2.1 Addition, Subtraktion, Multiplikation

Das Addieren, Subtrahieren und Multiplizieren ist mit \LaTeX sehr einfach darzustellen:

Addition, Subtraktion

Um zu addieren oder zu subtrahieren werden im Mathematikmodus die Operatoren $+$ und $-$ verwendet.

Multiplikation

Um zu multiplizieren wird im Mathematikmodus der Befehl `\cdot` verwendet.

5.2.2 Division

Für einfache Brüche wird der Operator $/$ verwendet.

Für kompliziertere Brüche verwendet man den Befehl `\frac{}{}`, wobei der Zähler in der ersten Klammer und der Nenner in der zweiten Klammer steht.

Ein Beispiel

$$\frac{(a+b) \cdot (a+b)}{(c-d) \cdot (c-d)}$$

Diese Formel sieht im \TeX -Dokument so aus:

```
\begin{displaymath}
\frac{(a + b) \cdot (a + b)}{(c-d) \cdot (c-d)}
\end{displaymath}
```

5.3 Wurzeln, Exponenten und Indizes

5.3.1 Wurzeln

Mit dem Befehl `\sqrt{}` lässt sich eine Wurzel erstellen. Der Ausdruck der unter der Wurzel stehen soll muss im \TeX -Dokument zwischen den beiden Klammern stehen. Bei der Wurzel aus der Wurzel eines Ausdrucks wendet man innerhalb des *sqr*t-Befehls diesen noch ein weiteres mal an: `\sqrt{\sqrt{a}}` $\sqrt{\sqrt{a}}$. Um die x -te Wurzel zu ziehen stellt man den geschweiften Klammern zwei eckige Klammern voraus: `\sqrt[x]{a}` $\sqrt[x]{a}$.

5.3.2 Exponenten und Indizes

Um mit \LaTeX Exponenten darzustellen verwendet man innerhalb des Mathematikmodus (z.B. mit `$ \dots $`) den Operator `^` gefolgt von zwei geschweiften Klammern `{}`. Um Indizes darzustellen verwendet man den Operator `_` gefolgt von zwei geschweiften Klammern.

$$x^2 \qquad \text{\code{x}\^{2}}$$

$$x_2 \qquad \text{\code{x}_{2}}$$

$$\frac{\sqrt[x]{z^{\frac{1}{2}}}}{e^{-\frac{1}{3}x^2} \cdot k}$$

```
\frac{\sqrt[x]{z^{\frac{1}{2}}}}{e^{-\frac{1}{3}x^2} \cdot k}
```

$$\frac{\sqrt[x]{z_4}}{e^{x^2} \cdot k}$$

```
\frac{\sqrt{x}{z_{4}}}{e^{x_{2}}} \cdot k
```

5.4 Matritzen

Das Erstellen von Matritzen hat mit L^AT_EX Ähnlichkeiten mit dem Erstellen von Tabellen.

Hierzu verwendet man die Befehle

```
\begin{array}
\end{array}
```

Ein Beispiel

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \dots \\ 4 & 5 & 6 & \dots \\ 7 & 8 & 9 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

```
\begin{displaymath}
\mathbf{A} =
\left[ \begin{array}{cccc}
1 & 2 & 3 & \ldots \\
4 & 5 & 6 & \ldots \\
7 & 8 & 9 & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{array} \right]
\end{displaymath}
```

5.5 Symbole, Sonderzeichen und andere mathematische Operatoren

Andere mathematische Operatoren sowie Symbole und Sonderzeichen sind über die Kopfzeile des *TeXnicCenter* unter *Mathe* einzufügen. Der Umgang mit mathematischen Operatoren erfolgt stets nach dem gleichen Schema (z.B. mit \$... \$).

6 Tabellen

6.1 Grundprinzip

Im Folgenden wird erklärt, wie man mit \LaTeX Tabellen erstellt.

Am einfachsten funktioniert dies mit der `Tabular`-Umgebung:

Mit dem Befehl

```
\begin{tabular}{Spaltendefinition}
```

mit der entsprechenden Spaltendefinition hinter dem *tabular*-Befehl.

Und dem Befehl

```
\end{tabular}
```

zum Beenden der laufenden Tabelle.

Alle restlichen Tabelleninhalte müssen sich grundsätzlich zwischen den beiden (o.g.) Befehlen zum Beginnen und Beenden einer jeweiligen Tabelle befinden.

6.2 Spaltendefinitionen

<code>l</code>	der Text innerhalb der Spalte wird linksbündig ausgerichtet
<code>c</code>	der Text innerhalb der Spalte wird zentriert ausgerichtet
<code>r</code>	der Text innerhalb der Spalte wird rechtsbündig ausgerichtet
<code> , </code>	einfache, doppelte senkrechte Trennlinie zwischen den Spalten
<code>p{Breite}</code>	manuelle Festlegung der Spaltenbreite, kürzere Spalte ist oben ausgerichtet
<code>m{Breite}</code>	manuelle Festlegung der Spaltenbreite, kürzere Spalte ist mittig ausgerichtet
<code>b{Breite}</code>	manuelle Festlegung der Spaltenbreite, kürzere Spalte ist unten ausgerichtet

Ein Beispiel

```
\begin{tabular}{|l||c|r|}
```

Dies ist eine Tabelle mit drei Spalten und vertikalen Trennlinien. Mit einer doppelten Trennlinie nach der ersten Spalte und einfachen Trennlinien vor der ersten, nach der zweiten und nach der dritten Spalte. Mit linksbündigem Text in der ersten, zentriertem Text in der zweiten und rechtsbündigem Text in der dritten Spalte.

Sind keine vertikalen Trennlinien gewünscht, so sind die Abtrennungen („|“) durch Leerzeichen zu ersetzen.

6.3 Befehle innerhalb der *tabular*-Umgebung

<code>&</code>	zum Trennen der Spalten
<code>\\</code>	Zeilenumbruch
<code>\hline</code>	horizontale Linie
<code>\hline \hline</code>	doppelte, horizontale Linie
<code>\cline{1-2}</code>	horizontale Linie von einer Spalte x bis zur jeweiligen Spalte y (hier von Spalte 1 bis Spalte 2)

6.4 Zusammenfassen mehrerer Boxen

Um einzelne Tabellenboxen zu einer Box zusammenzufassen - also eine große Box über zwei oder mehrere kleine Boxen zu ziehen - wendet man folgende Befehle an:

6.4.1 Zusammenfassen von Spalten

```
\multicolumn{Spaltenzahl}{Spaltendefinition}{Text}
```

Mit dem *multicolumn*-Befehl lässt sich eine Spalte über mehrere kleine Spalten ziehen. Für die Spaltenzahl wird die Anzahl an Spalten angegeben, über die die jeweilige große Spalte laufen soll. In der Spaltendefinition wird die Ausrichtung des Spalteninhalts angegeben - wie in der Definition

```
\begin{tabular}{Spaltendefinition}:
```

- l linksbündig
- c zentriert
- r rechtsbündig

In die letzte Klammer wird der Text dieser Spalte geschrieben.

Ein Beispiel

Vokabeln			
Englisch		Deutsch	
1.	door	1.	die Tür
2.	waiter, waitress	2.	der Kellner, die Kellnerin; die Bedienung; der Ober
3.	(magic) wand	3.	der Zauberstab
4.	potato	4.	die Kartoffel
5.	...	5.	...

Diese Tabelle sieht im \TeX -Dokument so aus:

```
\begin{tabular}{|l|l|l|l|}
\hline
\multicolumn{4}{|c|}{Vokabeln} \\
\hline \hline
\multicolumn{2}{|c|}{Englisch} & \multicolumn{2}{|c|}{Deutsch} \\
\hline
1. & door & 1. & die Tür \\
2. & waiter, waitress & 2. & der Kellner, die Kellnerin; \\
& & & die Bedienung; der Ober \\
3. & (magic) wand & 3. & der Zauberstab \\
4. & potato & 4. & die Kartoffel \\
5. & \ldots & 5. & \ldots \\
\end{tabular}
```

6.4.2 Zusammenfassen von Zeilen

```
\multirow{Zeilenzahl}{Zeilenbreite}{Text}
```

Mit dem *multirow*-Befehl lässt sich eine Zeile über mehrere kleine Zeilen ziehen. Dies erfolgt entsprechend analog zum *multicolumn*-Befehl; allerdings wird hier keine Zeilendefinition, sondern nur eine Zeilenbreite angegeben. Der Text steht wiederum in der letzten Klammer. Um den Befehl ausführen zu können ist das `\usepackage{multirow}` notwendig.

Ein Beispiel

Verkehrsaufkommen				
Personenverkehr	Individualverkehr			
	motorisiert		nicht motorisiert	
	öffentlicher Personenverkehr			
	Straßenverkehr	Schienenverkehr	Schiffsverkehr	Luftverkehr
Güterverkehr	40 Tonnen und weniger			
	Straßenverkehr	Schienenverkehr	Schiffsverkehr	Luftverkehr
	über 40 Tonnen			
	Straßenverkehr	Schienenverkehr	Schiffsverkehr	Luftverkehr

Diese Tabelle sieht im T_FX-Dokument so aus:

```
\begin{tabular}{|p{3cm}|c|c|c|c|}
\hline
\multicolumn{5}{|c|}{Verkehrsaufkommen} \\
\hline
\hline
\multirow{4}{8cm}{Personenverkehr}
& \multicolumn{4}{|c|}{Individualverkehr} \\
\cline{2-5}
& \multicolumn{2}{|c|}{motorisiert} & & \\
\multicolumn{2}{|c|}{nicht motorisiert} & & & \\
\cline{2-5} \cline{2-5}
& \multicolumn{4}{|c|}{öffentlicher Personenverkehr} \\
```

```

\cline{2-5}
& Straßenverkehr & Schienenverkehr & Schiffsverkehr & Luftverkehr \\
\hline
\hline
\multirow{4}{8cm}{Güterverkehr}
& \multicolumn{4}{|c|}{40 Tonnen und weniger} \\
\cline{2-5}
& Straßenverkehr & Schienenverkehr & Schiffsverkehr & Luftverkehr \\
\cline{2-5}
& \multicolumn{4}{|c|}{über 40 Tonnen} \\
\cline{2-5}
& Straßenverkehr & Schienenverkehr & Schiffsverkehr & Luftverkehr \\
\hline
\end{tabular}

```

Einstellungen zum Schriftformat wurden hier der Übersichtlichkeit halber außer Acht gelassen.

7 Bilder nebeneinander anordnen (subfigure)

Wie bei den Usecackages schon beschrieben, wird um Bilder nebeneinander darstellen zu lassen, dass Usecpackage „subfigure“ benötigt. Im Teil „Das erstes eigenes Dokument“ wurde schon beschrieben wie ein Bild eingefügt wird. Dieses funktioniert ähnlich bei den Nebeneinander stehenden Bildern. So ist es möglich direkte Bildervergleiche anzustellen. Ein Vorteil gegenüber Word ist, dass die Bilder von der Größe genau angepasst werden können. Hier einmal die Befehlsstruktur für solche Bilder:

```

\usepackage{subfigure}
\begin{figure}[...]
\centering
\subfigure[Untertitel des 1. Bildes]{\includegraphics[width=...,

```

```
height=...]{Quellordner}}  
\subfigure[Untertitel des 2. Bildes]{\includegraphics[width=...,  
height=...]{Quellordner}}  
\caption{gemeinsamer Untertitel}  
\label{fig:Titel des Bildes}  
\end{figure}
```

8 Nützliche Links

Beim erstellen unserer Facharbeiten sind wir auf ein paar sehr nützliche Internetseiten gestoßen. Diese möchten wir hier für Sie auch noch einmal auflisten:

<http://www.latex-project.org/>

L^AT_EX-Hauptseite: Hier kann L^AT_EX heruntergeladen werden

<http://latex.mschoeder.net/>

Ein weiteres Tutorial für L^AT_EX: Sehr gut für Anfänger geeignet

<http://www.golatex.de/>

Deutschsprachiges L^AT_EX-Forum

<http://www.weinelt.de/latex/index.html>

Eine Übersicht der L^AT_EX-Befehle für Fortgeschrittene

<http://www.heiner-lamprecht.net/uploads/media/Handbuch.pdf>

Großes und sehr umfangreiches 284 Seitiges L^AT_EX-Handbuch

<ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>

Ein weiteres und etwas kleineres L^AT_EX-Handbuch mit 146 Seiten

9 Template für die Seminararbeit

Nach diesem Tutorial werden Sie sich wahrscheinlich fragen, wie lange wir an der Grundstruktur unserer Templates für die Seminararbeit gearbeitet haben. An dieser Stelle

möchten wir uns nochmal bei Herrn Dr. Preuß bedanken, der unserem Kurs ein Grund-Template geschrieben hatte. Dieses finden Sie auch hier zum Download unter folgendem Link:

Ansonsten wünschen wir allen viel Spaß beim erfolgreichen Arbeiten mit \LaTeX und freuen uns über eine Weiterempfehlung von der Homepage und \LaTeX !