

## Inhaltsverzeichnis

<b>1</b>	<b>Gleitobjekte</b>	<b>1</b>
1.1	Allgemeines zu Gleitobjekten . . . . .	1
1.2	Beschriftung und Referenzierung . . . . .	2
1.3	Tabellen . . . . .	3
<b>2</b>	<b>Anordnung von Grafiken</b>	<b>6</b>
<b>3</b>	<b>Unterabbildungen</b>	<b>7</b>
<b>4</b>	<b>Abbildungen im Textfluss</b>	<b>7</b>

## 1 Gleitobjekte

### 1.1 Allgemeines zu Gleitobjekten

In L<sup>A</sup>T<sub>E</sub>X werden üblicherweise Objekte, die nicht direkt zum Text gehören, als Gleitobjekt behandelt. Dies sind zum Beispiel Abbildungen oder Tabellen. Ein Gleitobjekt zeichnet sich dadurch aus, dass es nicht unbedingt an dem Ort im Dokument, an dem es definiert wurde, ausgegeben wird. Stattdessen versucht L<sup>A</sup>T<sub>E</sub>X, eine möglichst gute Position für das Gleitobjekt zu finden. Dabei folgt L<sup>A</sup>T<sub>E</sub>X ein paar Regeln: Das Gleitobjekt wird möglichst früh ausgegeben, aber nicht vor dem Ort, an dem es definiert wurde. Außerdem versucht L<sup>A</sup>T<sub>E</sub>X, Gleitobjekte in dieser Reihenfolge oben auf einer Seite, unten auf einer Seite oder mit anderen nicht positionierten Gleitobjekten gesammelt auf einer Seite auszugeben. Die Reihenfolge der Gleitobjekte bleibt in jedem Fall erhalten. Die Position auf einer Seite kann mit den Optionen aus Tabelle 1 beeinflusst werden.

Bei größeren Arbeiten sollte man die Feinpositionierung, sofern man nicht mit der Ausgabe von L<sup>A</sup>T<sub>E</sub>X einverstanden ist, allerdings erst am Ende vornehmen, da Änderungen am Text auch Positionsänderungen nach sich ziehen können.

Neben der automatischen Positionierung von Gleitobjekten nummeriert L<sup>A</sup>T<sub>E</sub>X alle Gleitobjekte eines Typs durch. Dabei sammelt es auch Informationen, um bei Bedarf ein entsprechendes Verzeichnis zu erstellen. Für Abbildungen erhält man dieses beispielsweise durch den Befehl `\listoffigures`. Zusätzlich ist es natürlich möglich, Gleitobjekte zu referenzieren, ohne sich um die konkrete Nummer, die z.B. eine Abbildung bekommen

Option	Position
l	links
r	rechts
h	hier
p	eigene Gleitobjekt-Seite
!	Positionierungsregeln ignorieren (z.B. h!)

Tabelle 1: Positionierungsoptionen für Gleitobjekte

Präfix	Typ	Präfix	Typ
ch:	Kapitel	sec:	Abschnitt
subsec:	Unterabschnitt	fig:	Abbildung
tab:	Tabelle	eq:	Gleichung
lst:	Code Listing	app:	Anhang

Tabelle 2: Konventionsgemäße Präfixe

hat, zu kümmern. Aufgrund der Funktionsweise von L<sup>A</sup>T<sub>E</sub>X funktioniert die Referenz und das Verzeichnis aber erst, wenn nach Einfügen des Gleitobjektes ein zweites Mal kompiliert wurde. Das liegt daran, dass L<sup>A</sup>T<sub>E</sub>X in jedem Durchlauf eine Liste der Gleitobjekte erzeugt, für die Ausgabe aber die Liste vom vorherigen Durchlauf benutzt.

Weiterhin sei noch darauf hingewiesen, dass der Inhalt von Gleitobjekten prinzipiell beliebig ist. In einer `figure`-Umgebung darf also auch Text oder eine Tabelle stehen, die Beschriftung beginnt aber weiterhin mit „Abbildung“.

Um eine Ausgabe von Gleitobjekten an einem gewissen Punkt zu erzwingen, gibt es den Befehl `\clearpage`, der alle ausstehenden Gleitobjekte ausgibt und danach eine neue Seite beginnt. Alternativ kann man den Befehl `\FloatBarrier` aus dem Paket `placeins` verwenden. Dieser gibt alle ausstehenden Gleitobjekte aus, erzeugt aber keinen Seitenumbruch.

## 1.2 Beschriftung und Referenzierung

Die Beschriftung eines Gleitobjekts wird über den Befehl `\caption{Beschriftung}` festgelegt. Wenn das Gleitobjekt referenziert werden soll, ist dies ein guter Ort, um eine entsprechende Markierung zu setzen. Dies geschieht mit dem Befehl `\label{Markierung}`. Um den Überblick zu behalten, sollten die Markierung nach Art des Gleitobjekts gruppiert werden, beispielsweise indem man alle Abbildungsmarkierungen mit „fig:“ beginnt. Eine Übersicht über die üblichen Präfixe, findet ihr in Tabelle 2. Die sieht dann wie so aus:

```
\begin{figure}
  \includegraphics [width=3cm]{Unix-AG}
  \caption{Logo der Unix-AG\label{fig:logo}}
\end{figure}
```

Auf eine solche Markierung kann mit dem Befehl `\ref{Markierung}` zugegriffen werden. Dieser gibt aber nur die Nummer aus. Das Beispiel würde dann so referenziert:

In `Abbildung~\ref{fig:logo}` sieht man das Logo der Unix-AG.

Falls dabei eine nicht vorhandene Markierung referenziert wird, gibt L<sup>A</sup>T<sub>E</sub>X beim Kompilieren eine Warnung aus und ersetzt die Nummer durch „??“. Zwischen die Bezeichnung, z.B. „Abbildung“, und den `\ref`-Befehl sollte ein nichtumbrechendes Leerzeichen („~“) gesetzt werden um unschöne Umbrüche zu vermeiden.

Option	Bedeutung
<code>margin</code>	Rand
<code>width</code>	Breite
<code>labelfont</code>	Schriftart für „Abbildung“ o.ä.
<code>textfont</code>	Schriftart für Beschriftung
<code>labelstep</code>	Abstand zwischen Bezeichnung und Text
<code>format</code>	Ausrichtung des Texts an der Bezeichnung
<code>labelformat</code>	Format der Bezeichnung

Tabelle 3: Wichtige Optionen des `caption`-Paket

Zeichen	Bedeutung
<code>c</code>	zentriert
<code>l</code>	linksbündig
<code>r</code>	rechtsbündig
<code>p{Breite}</code>	Absatz (Paragraph)
<code>*{n}</code>	n-fache Wiederholung
<code> </code>	vertikale Linie
<code>@{Text}</code>	fester Text in jeder Zeile

Tabelle 4: Spaltendefinitionen von `Tabular`

Um das Aussehen der Beschriftung zu beeinflussen, bietet sich das Paket `caption` an. Dieses bietet sehr viele Optionen, die unter <https://www.ctan.org/pkg/caption> nachgelesen werden können. Einige wichtige Optionen sind in Tabelle 3 wiedergegeben.

### 1.3 Tabellen

Tabellen werden in L<sup>A</sup>T<sub>E</sub>X analog zur aus dem letzten Vortrag bekannten `align`-Umgebung gesetzt, als Spaltentrenner dient also `&` und als Zeilentrenner `\\`. Die Tabelle selbst wird in der `tabular`-Umgebung gesetzt. Diese braucht als Argument eine Liste der Spalten, wobei jede Spalte durch einen Buchstaben, der die Ausrichtung definiert, repräsentiert wird. Eine Liste der Ausrichtungen findet sich in Tabelle 4. Senkrechte Linien erhält man durch „|“ zwischen zwei Spalten und horizontale Linien durch den Befehl `\hline` nach

Befehl	Bedeutung
<code>\endfirsthead</code>	Ende des Headers der ersten Seite
<code>\endhead</code>	Ende des Headers der folgenden Seiten
<code>\endlastfoot</code>	Ende des Footers der letzten Seite
<code>\endfoot</code>	Ende des Footers der vorherigen Seiten

Tabelle 5: Befehle in der `longtable`-Umgebung

Tabelle 6: schöne Tabelle mit `booktabs`

Absatz	links	links	ver-bunden
1	2	3	4-5
lang	lang	kurz	kurz-kurz

dem Zeilenumbruch. Wird dieser Befehl doppelt gesetzt, werden die senkrechten Linien nach der entsprechenden Zeile unterbrochen. Wenn die horizontale Linie nicht über die gesamte Breite gehen soll, kann der Befehl `\cline{n-m}` verwendet werden, der die Linie nur über Spalten `n` bis `m` erzeugt. Dies ist im folgenden Code gezeigt. Dort wird auch `*` verwendet, um die Spaltendefinition abzukürzen. Das Ergebnis ist in Tabelle 7 zu sehen.

```
\begin{table}
  \centering
  \begin{tabular}{|*{5}{l|}}\hline
    1 & 2 & 3 & 4 & 5 \\ \cline{2-4}
    6 & 7 & 8 & 9 & 10 \\ \hline
  \end{tabular}
  %Code der Tabellen in der Mitte und Rechts sowie \caption
  {}.
\end{table}
```

Mit dem Paket `booktabs` kann man Tabellen erzeugen, die optisch ansprechender gestaltet sind. Diese sind auch ohne vertikale oder doppelte Linien gut lesbar und fügen über dem Text etwas mehr Abstand ein, dass das ganze nicht so gequetscht aussieht.

Für die Linien über der Tabelle wird der Befehl `\toprule` verwendet. Um den Kopf von Tabellenkörper abzutrennen und für die Fußzeile gibt es `\midrule` bzw. `\bottomrule`. Die `\cline` wird durch `\cmidrule{n-m}` ersetzt.

```
\begin{tabular}{p{2cm}*{2}{l} r@{-}l}\toprule
  Absatz & links & links & ver&bunden\\ \midrule
  1 & 2 & 3 & 4 & 5 \\
  lang & lang & kurz & kurz & kurz \\ \bottomrule
\end{tabular}
```

Sollen mehrere Zellen zusammengefasst werden, gibt es die Befehle `\multirow{Anzahl}{Breite}{Inhalt}` und `\multicolumn{Anzahl}{Ausrichtung}{Inhalt}`, wobei `\multirow` das Paket `multirow` benötigt. Der folgende Beispielcode ist für beide Fälle in Tabelle 7 wiedergegeben.

```
\begin{table}
  %Code der Tabelle Links
  \begin{tabular}{|l|l|l|l|}\hline
    A1 & & & & A2 & A3 \\ \hline
    \multirow{2}{*}{B1 C1} & & & & B2 & B3 \\ \cline{2-3}
  \end{tabular}
\end{table}
```



Table 1: lange Tabelle			Table 1: lange Tabelle (Fortsetzung)		
erste	Kopf	zeile	weitere	Kopf	zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
ganz	viele	Zeilen	ganz	viele	Zeilen
weitere	Fuss	zeilen	letzte	Fuss	zeile

Abbildung 1: Beispiel zu Longtable

zu können, die schönere Trennlinien als `\hline` ergeben.

## 2 Anordnung von Grafiken

Die hier genannten Positionierungsbefehle sind generell anwendbar, aber speziell für Grafiken interessant. Die einfachste und wichtigste Form, die Anordnung zu beeinflussen, ist die `center`-Umgebung, die einfach ihren Inhalt horizontal zentriert. In Gleitobjekten sollte allerdings besser der `\centering`-Befehl verwendet werden, da die `center`-Umgebung eventuell ungewollte vertikale Abstände erzeugt. Sollen mehrere Objekte zusammengefasst werden, kann mit der `minipage`-Umgebung eine virtuelle Seite erzeugt werden. Als Option kann die Positionierung wie bei Tabellen angegeben werden, als Argument muss die Breite gesetzt sein. Sollen Grafiken übereinander gesetzt werden, reicht meist ein einfacher Zeilenumbruch (`\\`) aus, damit Grafiken zunächst wie Text positioniert werden. Um eine feinere Positionierung zu erreichen, gibt es einige weitere hilfreiche Befehle. Die Befehle `\hfill` und `\vfill` füllen einfach in horizontaler bzw. vertikaler Richtung mit Abständen auf. Dies ist nützlich, um gleiche Abstände zwischen mehreren Objekten zu erhalten oder zwei Objekte so zu positionieren, dass eins links und eins rechts gesetzt wird. In vertikaler Richtung kann der Abstand am einfachsten damit beeinflusst werden, dass hinter dem Zeilenumbruch ein zusätzlicher Abstand eingegeben wird. So erzeugt `\\[3mm]` einen Umbruch mit 3 mm zusätzlichem Abstand. Die Befehle

<code>A\hfill a:</code> A <span style="float: right;">a</span>	<code>C\\[5mm]</code> c: C  c	<code>D\\...\\D\\</code> <code>vskip -15mm d:</code> D d. D
<code>B\hspace{1cm} b:</code> B <span style="margin-left: 1cm;">b</span>		

Abbildung 2: Beispiele zur Positionierung

`\hspace[Abstand]` und `\vspace[Abstand]` fügen ebenfalls zusätzliche Abstände ein. Einen ähnlichen Effekt erhält man mit `\hskip Abstand` und `\vskip Abstand`. Damit können Objekte auch in negativer Richtung verschoben werden. Diese Befehle greifen aber stark in den Textsatz ein, daher sollten sie sehr vorsichtig und nur wo sie wirklich nötig sind verwendet werden. Abstände können generell relativ zu einer Größe oder absolut angegeben werden `0.5\textwidth` stellt beispielsweise 50% der Textbreite dar. Absolute Größen werden verwendet, wenn eine Einheit angegeben wird. Üblicherweise sind `cm`, `mm`, `pt` (Punkt) und `em` (Breite eines „m“) praktische Einheiten. Insbesondere bei Vorträgen ist natürlich zu beachten, dass `1cm` nicht einem realen `cm` entspricht, sondern L<sup>A</sup>T<sub>E</sub>X von einer vordefinierten Höhe und Breite der Folien ausgeht. Einige Beispiele sind in [Abbildung 2](#) angegeben.

## 3 Unterabbildungen

Für Unterabbildungen steht das Paket `subfigure` zur Verfügung. Es bietet den Befehl `\subfigure[Beschriftung]{Inhalt}` an, der nummerierte Unterabbildungen erzeugt. Analog funktioniert auch der `\subtable`-Befehl. In der Praxis sieht das Ergebnis eines solchen Codes dann wie in [Abbildung 3](#) aus:

```
\begin{figure}
  \centering
  \subfigure[Erstes Logo]
    {\includegraphics[width=2.5cm]{UnixAG}}
  \subfigure[Zweites Logo]
    {\includegraphics[width=2.5cm]{UnixAG}}
  \caption{Logos\label{fig:subfigures}}
\end{figure}
```

## 4 Abbildungen im Textfluss



Abbildung 4: Logo

Manchmal ist es gewünscht, Abbildungen von Text umfließen zu lassen, insbesondere wenn die Abbildungen relativ klein sind. In solchen Fall kann das Paket `wrapfig` verwendet werden. Es bietet für Abbildungen die `warpfigure`- und für Tabellen die



(a) Erstes Logo (b) Zweites Logo

Abbildung 3: Logos

`wraptable`-Umgebungen an. Diese verhalten sich analog. Die Umgebung hat ein paar Argumente und Optionen: `\begin{wrapfigure}[Zeilen]{Ausrichtung}[Überhang]{Breite}`, die genauer unter <http://mirrors.ctan.org/macros/latex/contrib/wrapfig/wrapfig-doc.pdf> beschrieben sind. Oft wird die Höhe falsch berechnet, weshalb es ratsam sein kann, die Anzahl der Zeilen von Hand festzulegen. Falls dann auf einer Seite der Abstand zu groß ist, kann der Überhang genutzt werden, um dies zu korrigieren. Die Ausrichtung erfolgt hier in horizontaler Richtung also links im Text (`l`) oder rechts im Text (`r`). In Abbildung 4 ist der folgende Quelltext umgesetzt.

```
\begin{wrapfigure}{l}{.2\textwidth}
  \includegraphics[width=.2\textwidth]{UnixAG}
  \caption{Logo\label{fig:wrapfig}}
\end{wrapfigure}
```

Wie man auf dieser Seite sieht, kann es bei der Verwendung von `wrapfigure` aber passieren, dass die Reihenfolge der Abbildungen nicht mehr richtig ist, da die `wrapfigure` nicht als Gleitobjekt positioniert wird.