

6 Tabellen

6.1 Tabellen und Abbildungen

Informationen jeglicher Art lassen sich am effektivsten in Tabellen oder Abbildungen darstellen. Dabei ist zu beachten, dass Tabellen in der Regel reines Datenmaterial aufbereiten, während sich Abbildungen in erster Linie durch Grafiken, Wörter und deren Zusammenhang auszeichnen. Allerdings ist die Unterscheidung in Tabellen und Abbildungen nicht immer trennscharf. In $\text{\LaTeX}$ ist die grundsätzliche Struktur und der Aufbau von Tabellen und Abbildungen gleich. Nur durch den darzustellende Inhalt kann zwischen diesen beiden Kategorien von Darstellungen unterschieden werden. Dennoch ist wie wir noch sehen werden die Unterscheidung bei der Einbindung von Grafiken mit Hilfe einer Abbildungsumgebung wichtig (vgl. Abschnitt 7).

In den folgenden beiden Kapiteln wird das grundlegende Verständnis für die Erstellung von Darstellungen aufgezeigt. Zunächst werden Tabellen mit Hilfe von `tabular`-Umgebungen besprochen. Darauf aufbauend werden dann Erweiterungen und spezielle Problembereiche behandelt. Abbildungen sind das Thema des Kapitels 7.

6.2 Umgebungen zur Erzeugung von Tabellen

Die einfachste Form Tabellen zu erzeugen ist die Verwendung einer `tabular`- und `array`-Umgebung, die jede für sich weitere Vorteile besitzt. Ersteres findet Anwendung im Text-Modus, letzteres im Mathe-Modus.¹ Beide Grundformen sind in dem von FRANK MITTEL-BACH erstellten `array`-Paket enthalten und lassen sich über den Befehl `\usepackage{array}` in der Präambel des Dokuments laden, welches in der Standardversion von $\text{\LaTeX}$ enthalten ist. Alle `tabular`-Befehle sind also auch im `array`-Paket anwendbar, wobei letzteres noch über mehr Optionen verfügt.

Mit Hilfe einer `tabular`-Umgebung ist es möglich Darstellungen flexibel zu positionieren, Absätze zu erstellen, sowie Spalten und Zeilen in vielfältiger Weise zu formatieren. Auch komplexe Darstellungen können erstellt werden. Die Syntax dieser Grundformen ist wie folgt:

```
\begin{array}[Position]{Spaltenoptionen} Zeilen \end{array}
```

```
\begin{tabular}[Position]{Spaltenoptionen} Zeilen \end{tabular}
```

Jede Zeile besteht aus einer Reihe von Spalten, die durch das `&`-Zeichen voneinander getrennt werden. Die Zeilen einer Tabelle werden durch `\\` voneinander getrennt. Jede Zeile sollte genauso viele Spalteneinträge aufweisen, wie im Spaltenkopf festgelegt sind. Nachfolgende Tabellenumgebung erstellt eine „Box“, bestehend aus einer beliebigen Anzahl an Zeilen und Spalten:

¹Auf die Darstellung und Verwendung der `tabbing`-Umgebung, die ebenfalls einige Vorteile besitzt wird an dieser Stelle verzichtet. Für einen Überblick siehe [Goossens, Mittelbach, und Samarin \(2002, S. 104–106\)](#).

```

\begin{tabular}[Position]{Spaltenoptionen}
  Spalte 1 & Spalte 2 & ... & Spalte n \\
  . . .
\end{tabular}

```

Achtung: Die Umgebungen `tabular` und `table` sind voneinander zu unterscheiden. Während erstere Tabellentext in Spalten anordnet, hat letztere zur Aufgabe einen definierten Inhalt zusammen zu halten und im Text bzw. Dokument gleiten zu lassen. Dabei können mehrere `tabular`-Umgebungen in einer `table`-Umgebung enthalten sein. Dazu später in Abschnitt 6.3.1 mehr.

Die optionalen Argumente haben dabei folgende Eigenschaften:

- **Position** Dieser Befehl legt die *Position der Tabelle* auf der Seite fest. Voreingestellt ist eine mittige Ausrichtung auf dem Blatt (`c`). Weitere Optionen sind `t` (top, Ausrichtung oben) und `b` (bottom, Ausrichtung unten).
- **Spaltenoptionen** Hiermit wird der *Inhalt der Spalten* formatiert. Dazu existieren eine Reihe von Optionen, die in jeder Spalte verschieden sein dürfen. Für jede Spalte sind im Spaltenkopf die entsprechenden Parameter anzugeben (vgl. [Kopka \(1992, S. 35\)](#) und [Goossens, Mittelbach, und Samarin \(2002, S. 107\)](#)):
 1. `l` linksbündige Ausrichtung des Inhaltes einer Spalte
 2. `r` rechtsbündige Ausrichtung des Inhaltes einer Spalte
 3. `c` mittige Ausrichtung des Inhaltes einer Spalte (Voreinstellung)
 4. `m{Breite}` Spalte mit der Breite `Breite`. (Mehrzeiliger) Text wird auf die vertikale Mitte zentriert.
 5. `p{Breite}` vertikale Ausrichtung des Inhaltes oben `Breite`.
 6. `b{Breite}` vertikale Ausrichtung des Inhaltes unten.
 7. `>{Deklaration}` Kann direkt vor 1–6 stehen. Als Option kann ein Befehl, Text oder eine Mischung verwendet werden. Dies wird dann vor jedem Spalteneintrag eingefügt bzw. der Spalteneintrag so formatiert.
 8. `>{$}Spaltenoptionen 1-6<{$}` Hiermit sind Eintragungen im mathematischen Modus möglich, allerdings nur in einer `tabular`-Umgebung. Die `array`-Umgebung hingegen wird ja ohnehin im Mathe-Modus verwendet.
 9. `@{Text}` Jede Zeile in dieser Spalte wird mit dem Inhalt `Text` gefüllt. Zu beachten ist, dass der Befehl `\@` den „natürlichen“ Abstand zwischen den Spalten ignoriert. Möchte man dennoch einen Abstand haben, so kann man diesen herbeiführen indem man im Text ein Leerzeichen einfügt. Also, z. B. `@{Text }` oder indem man mit dem Befehl `\extracolsep{Breite}` einen Abstand (in Zentimeter) einfügt: `@{\extracolsep{Breite} Text}`.
 10. `|` vertikale Linie. Diese kann mit obigen Optionen kombiniert werden.
 11. `!{Text}` erstellt anstatt einer vertikalen Linie den Text `Text`. Diese Option kann mit obigen Optionen kombiniert werden und unterdrückt nicht den Spaltenabstand wie der Befehl `@{Text}`.

Bei der Formatierung der Spalten mittels der Formate 4–6 kann der Spaltentext aus mehreren Absätzen bestehen. Dazu ist intern `\parindent0pt` eingestellt, so dass die ersten Zeilen eines jeden Absatzes *nicht* eingerückt werden, was in Darstellungen in der Regel nicht erwünscht ist. Durch Formatierung des Befehls im entsprechenden Spaltenkopf, z.B. durch `>\parindent1cm`p, kann dies geändert werden.

Eine Tabelle stellt rein strukturmäßig eine vertikale Box vom Typ `parbox` oder `minipage` dar, so dass sie horizontal mit anderen `parbox`– oder `minipage`–Umgebungen positioniert werden kann. Häufig möchte man eine Darstellung mittig auf einer Seite platzieren. Dazu kann man die Tabelle in eine `center`–Umgebung einschließen. dabei ist allerdings zu berücksichtigen, dass diese Umgebung nicht nur den Inhalt zentriert, sondern auch vorher und nachher zusätzliche vertikale Abstände einfügt. Ist dies, z. B. in einer `table`–Umgebung nicht erwünscht, sollte man das zentrieren lieber mittels des Befehls `\centering` bewerkstelligen, dass als Schalter wirkt, also ab dem Befehl alles Folgende bis zum Ende der Gruppe zentriert. Nachstehendes Beispiel verdeutlicht den dargestellten Sachverhalt:

a.	Gibt die Unterschrift Auskunft über die Frage: was? wo? wann? Sind überflüssige Worte eliminiert?
b.	Sind alle Angaben und Zeichen gut zu unterscheiden?
c.	Ist die Quellenangabe vorhanden und vollständig?
d.	Ist der dargestellte Sachverhalt nicht trivial? Erfüllt die Darstellung die Anforderungen der Vollständigkeit, Genauigkeit und Anschaulichkeit?

Diese (noch unvollständige) Darstellung wird wie folgt erzeugt:

```
\begin{center}
\begin{tabular}{|>\bf \small}c<{.}|>\sf m{12cm}|}
a & Gibt die Unterschrift Auskunft über die Frage: was? wo?
    wann? Sind überflüssige Worte eliminiert? \\
b & Sind alle Angaben und Zeichen gut zu unterscheiden? \\
c & Ist die Quellenangabe vorhanden und vollständig? \\
d & Ist der dargestellte Sachverhalt nicht trivial? Erfüllt
    die Darstellung die Anforderungen der
    Vollständigkeit, Genauigkeit und Anschaulichkeit? \\
\end{tabular}
\end{center}
```

Dokumentation: [Mittelbach und Carlisle \(2001\)](#).

6.2.1 Weitere Stilparameter und Pakete

Da die erzeugte Darstellung noch nicht vollständig ist und zugegebenermaßen auch nicht sonderlich schön aussieht, werden im folgenden Abschnitt weitere wesentliche Stilparameter und Pakete angesprochen.

6.2.1.1 Ausrichtung am Dezimalpunkt

Eine weitere Option für die Ausrichtung des Inhaltes von Tabellen ist die Ausrichtung der Zahlen am Dezimalpunkt. Das Paket `dcolumn` von DAVID P. CARLISLE dient dazu innerhalb

einer `array`- oder `tabular`-Umgebung Spalten zu erzeugen, deren Inhalt (Zahlen) am Dezimalzeichen ausgerichtet wird. Die Argumente, die im Spaltenkopf anzugeben sind, sind: `D{Eingabetrennzeichen}{Ausgabetretnnzeichen}{Kommastellen}`.

Es kann sinnvoll sein diesen Befehl nicht direkt in die Präambel zu schreiben, sondern an gegebener Stelle eine eigene Option zu deklarieren. Mit dem Befehl

```
\newcolumntype{Name}{D{Eingabetrennzeichen}{Ausgabetretnnzeichen}{Kommastellen}}
```

definiert man eine neue Formatierung. In den entsprechenden Spaltenkopf setzt man dann `Name` ein (vgl. [Goossens, Mittelbach, und Samarin, 2002](#), S. 131).

Dokumentation: [Carlisle \(2001\)](#).

6.2.1.2 Horizontale Linien

Weiterhin möchte man häufig die Zeilen durch eine Linie von einander trennen. Ebenso die Tabelle durch eine Linie am Anfang und Ende begrenzen. Hierzu gibt es mehrere Möglichkeiten, die sich neben dem abzugrenzenden Inhalt nach der Komplexität der Tabelle richten. Alle drei Möglichkeiten sind am Ende einer Zeile direkt im Anschluss an das Zeichen für den Zeilenumbruch `\\` zu setzen.

`\hline` Fügt eine Linie ein und zwar über alle Spalten. Die Länge der Linie wird durch die Tabellenbreite bestimmt.

`\hhline{Optionen}` Ähnlich wie der Befehl `\hline`. Um diesen Befehl benutzen zu können, muss das Paket `hhline` von DAVID P. CARLISLE geladen werden. Dieses Paket enthält eine Reihe von Optionen, die die Stärke der Doppellinie und das Zusammenspiel von horizontalen und vertikalen Linien innerhalb einer Darstellung bestimmen (vgl. [Goossens, Mittelbach, und Samarin, 2002](#), S. 132 f.).

`\cline{n-m}` Durch die Angabe derjenigen Spalten, die mit einer Linie versehen werden sollen, lassen sich auch komplexe Darstellungen erzeugen. Es können mehrere `\cline`-Befehle hintereinander geschaltet werden. Anzugeben sind jeweils die Spalten in der die Linie beginnen soll und die Spalte mit der Endpunkt der Linie. Die Zahlen der Anfangs- und Endspalte sind durch einen Trennungsstrich von einander zu trennen. Also z. B. `\cline{1-1}` oder `\cline{1-2}`, hier gleichbedeutend mit `\hline`.

Dokumentation: [Carlisle \(1994\)](#).

6.2.1.3 Spaltenübergreifende Zeilen

Die Stilooption `\multicolumn` erlaubt eine n -spaltige Aufteilung der Darstellung, bei der der Text gleichmäßig auf die n Spalten aufgeteilt wird.² Dabei kann innerhalb einer Darstellung mehrfach zwischen ein- und mehrspaltiger Formatierung gewechselt werden. Der Befehl ist wie folgt: `\multicolumn{Spaltenanzahl}{Spaltenoptionen}{Text}`. Je nach Anzahl der zusammenfügenden Spalten ersetzt dieser Befehl das `&`-Zeichen. Die Zeile wird wie gewohnt mit `\\` abgeschlossen. Zu beachten ist, dass die mit dem `\multicolumn`-Befehl angegebenen Spaltenoptionen die Spaltenoptionen im Spaltenkopf der `array`- und `tabular`-Umgebung überschreiben. Siehe hierzu nachstehendes Beispiel.

²Ebenso existiert das Paket `multirow` eines unbekannten Autors, das einen mehrzeiligen Spalteneintrag ermöglicht. Siehe hierzu [Goossens, Mittelbach, und Samarin \(2002](#), S. 137).

Titel	
1.	Gibt die Unterschrift Auskunft über die Frage: was? wo? wann?
Darstellung	
2.	Sind alle Angaben und Zeichen gut zu unterscheiden?
3.	Sind nicht zu viele überflüssige Details enthalten?
4.	Ist die Beschriftung auch bei einer Verkleinerung gut lesbar?
Qualität	
5.	Ist die Quellenangabe vorhanden und vollständig?
6.	Ist der dargestellte Sachverhalt nicht trivial? Erfüllt die Darstellung die Anforderungen der Vollständigkeit, Genauigkeit und Anschaulichkeit?

Die Syntax für diese Tabelle ist wie folgt:

```
\begin{center}
\begin{tabular}{||c|>{\sf}p{12cm}||} \hline{t==:t|}
\multicolumn{2}{||c||}{\large Titel} \\\hline{||~-||}
1. & Gibt die Unterschrift Auskunft über die Frage:
was? wo? wann? \\\hline{||~-||}
\multicolumn{2}{||c||}{\large Darstellung} \\\hline{||~-||}
2. & Sind alle Angaben und Zeichen gut zu unterscheiden? \\\
3. & Sind nicht zu viele überflüssige Details
enthalten? \\\
4. & Ist die Beschriftung auch bei einer Verkleinerung gut
lesbar? \\\hline{||~-||}
\multicolumn{2}{||c||}{\large Qualität} \\\hline{||~-||}
5. & Ist die Quellenangabe vorhanden und vollständig? \\\
6. & Ist der dargestellte Sachverhalt nicht trivial?
Erfüllt die Darstellung die Anforderungen der
Vollständigkeit, Genauigkeit und Anschaulichkeit? \\\
\hline{b==:b|}
\end{tabular}
\end{center}
```

6.2.1.4 Mehrseitige Tabellen

Manchmal kann es vorkommen, dass eine zu erstellende Tabelle nicht auf eine Seite passt. Normalerweise überschreibt der Text in diesem Fall den unteren Seitenrand und man erhält eine Fehlermeldung. Möchte man mehrseitige Tabellen erstellen, so ist das Paket **supertab** von THEO JURRIENS und JOHANNES BRAAMS oder das Paket **longtable** von DAVID P. CARLISLE sehr hilfreich. Beide Pakete sind in ihrer Funktionalität sehr ähnlich, aber dennoch unterschiedlich in ihrer Handhabung. Das **supertab**-Paket besticht in erster Linie durch seine einfachere Handhabung, während mit dem **longtable**-Paket auch komplexe, mehrseitige Tabellen mit gleichbleibender Tabellenbreite möglich sind. Nachfolgend wird das **longtable**-Paket näher erläutert.

Ein Vorteil der **longtable**-Umgebung ist, dass alle Tabellenteile die gleiche Breite aufweisen. Dazu wird die Breite einer **longtable**-Umgebung in der **.aux**-Datei gespeichert, um die maximale Breite für den jeweiligen Tabellenteil (einer jeden Seite) zu finden. Diese Funktion

muss mit dem Befehl `\setlongtables` außerhalb der `longtable`-Umgebung aktiviert werden. Mögliche Probleme einer späteren Änderung der Formatierung sind dabei nicht ausgeschlossen (vgl. [Goossens, Mittelbach, und Samarin, 2002](#), S. 124).

Weiterhin besteht die Möglichkeit einzelne Tafeln mit eigenen Über- und Unterschriften und mit weiteren Optionen zu versehen. Soll ein Tabellenverzeichnis erstellt werden, so wird ein kombiniertes Verzeichnis der `table`- und `longtable`-Umgebungen erstellt.

Dokumentation: [Carlisle \(2000\)](#).

6.2.2 Darstellungen im Querformat

Manchmal ist es wünschenswert wenn man eine Darstellung im Querformat erstellen kann. Das $\text{\LaTeX}$ -Bundle `graphics` enthält das Paket `landscape`, welches eine `landscape`-Umgebung zur Verfügung stellt. Alles was innerhalb dieser Umgebung steht wird im Querformat wiedergegeben. D. h. dass der Inhalt dieser Umgebung um 90 Grad gegen den Uhrzeigersinn gedreht und auf eine eigene Seite platziert wird. Kopf- und Fußzeilen bleiben dabei im voreingestellten Portrait-Format erhalten. Zu Beginn und am Ende der Umgebung wird ein `\clearpage` ausgeführt:

```
\usepackage{landscape}

\begin{landscape}
  \begin{figure}
    \caption{gedrehte Überschrift}
    Inhalt der Abbildung
  \end{figure}
\end{landscape}
```

Dokumententeile können aber nur gedreht werden, wenn dies auch vom verwendeten dvi-Gerätetreiber unterstützt wird. In den meisten Fällen bedeutet dies, dass die dvi-Datei nach Postscript zu konvertieren ist. Zum Betrachten und Ausdrucken ist ein Postscript-Viewer wie zum Beispiel GSView geeignet. Oder man erstellt aus der Postscript-Datei eine PDF-Datei.

6.3 Gleitobjekte (1)

6.3.1 Platzierung von Gleitobjekten

Große Bilder und lange Tabellen können oft nicht da positioniert werden, wo sie eigentlich hingehören. Möchte man Darstellungen anschließend an einen Text und fortsetzend mit dem nachfolgenden Text platzieren, so passen sie häufig nicht vollständig auf die aktuelle Seite. Damit die Tabellen und Abbildungen nun aber nicht irgendwo im Text erscheinen oder gar durch einen Seitenwechsel „zerrissen“ werden, gibt es die Möglichkeit die Objekte an eine bestimmte bzw. geeignete Stelle „gleiten“ zu lassen. Hierfür kennt $\text{\LaTeX}$ die beiden Umgebungen `table` und `figure`.

Zwar existieren eigene Umgebungen für Tabellen und Abbildungen, tatsächlich ist aber die Erzeugung der Gleitobjekte unabhängig davon. Dennoch sollte man für die Beibehaltung seiner (logischen) Struktur die passende Umgebung wählen, da ansonsten spätestens bei der Erstellung des Tabellen- und/oder Abbildungsverzeichnisses falsche Einträge generiert werden (vgl. Abschnitt [6.3.1.1](#)).

Die Umgebungen lauten demnach `\begin{table}[Position]` und `\end{table}` für Tabellen, sowie entsprechend `\begin{figure}[Position]` und `\end{figure}` für Abbildungen. Alles, was nun in einer solchen Umgebung steht wird dann automatisch an eine Stelle gesetzt, an der das Objekt vollständig hin passt. Die Tabelle muss also nicht zwangsläufig durch eine `tabular`-Umgebung erzeugt worden sein. Die Position eines Objekts kann dabei durch optionale Argumente beeinflusst werden:

- **h** für „here“ (hier: genau an dieser Stelle). Sollte nie alleine verwendet werden.
- **t** für „top“ (oben auf der Seite)
- **b** für „bottom“ (unten auf der Seite)
- **p** für „page“ (auf einer eigenen Seite)
- **!** für „bang“ (bei einer Platzierung durch **h**, **t**, **b** wird ein Mindestanteil der Seiten für normalen Text reserviert)

Die Voreinstellung für `report` und `article` lautet `tbp`. Werden keine Argumente angegeben, so versucht L^AT_EX das Objekt zuerst oben auf der Seite, dann unten auf der Seite oder auf einer eigenen Seite zu platzieren. Einzelne Argumente können bzw. sollten kombiniert werden, wobei die Reihenfolge entscheidend ist. Wird z.B. `tb` angegeben, so wird L^AT_EX das Objekt nur oben oder unten auf der aktuellen Seite platzieren, je nachdem wo zuerst genügend Platz vorhanden ist.

Manchmal kann es vorkommen, dass es L^AT_EX nicht möglich ist das Objekt nach Angabe zu positionieren. In diesem Fall wird das Objekt immer weiter nach hinten verschoben, maximal solange bis ein neues Kapitel beginnt oder das Dokument zu Ende ist (oder der Befehl `\clearpage` erscheint). Alle nachfolgenden Objekte werden dabei ebenfalls nach hinten verschoben oder der Platz zwischen Objekten verringert sich (je nach Positionsangabe). Es ist wichtig sich klar zu machen, dass durch das optionale Argument der `table`- und `figure`-Umgebung die Plazierungsmöglichkeiten für eine Darstellung auf die angegebenen Orte eingeschränkt werden. Aus diesem Grund sollte man mehrere Plazierungsangaben verwenden (empfehlenswert!), sie ganz am Schluss bei der Endfassung des Dokuments einsetzen (nicht empfehlenswert!!) oder auf sie gänzlich verzichten (bedingt empfehlenswert!!!).

6.3.1.1 Überschriften, Labels und Verzeichnisse

Tabellen und Abbildungen können mit einem Titel versehen werden. Die Bezeichnung wird entweder immer über (als Überschrift) oder immer unter (als Unterschrift) die Darstellung gesetzt.³ Der erforderliche Befehl dazu lautet `\caption{Titel}`.

Möchte man im Text Bezug auf eine Darstellung nehmen, so ist dieser mit den Befehlen `\label{AbkuerzungTitel}` und `\ref{AbkuerzungTitel}` herzustellen, wie man sie von den Gliederungsebenen her kennt. L^AT_EX erzeugt dann die entsprechende Nummer der Darstellung. Analog kann auch der Befehl `\pageref{AbkuerzungTitel}` verwendet werden, der dann anstatt einer Nummer die entsprechende Seitenzahl in den Text einfügt. Zu beachten ist, dass das Bezugswort zur Nummer (Abschnitt, Tabelle, Darstellung) selbst hinzugefügt werden muss. Angenommen, man möchte folgenden Satz im Text schreiben: *Die Ergebnisse sind in Tabelle*

³Dazu ob eher Über- oder Unterschriften zu verwenden sind siehe Theisen (2000, S. 162–167)

6.1 dargestellt. So muss dieser wie folgt eingegeben werden:

```
\textit{Die Ergebnisse sind in Tabelle \ref{tab:darstellung} dargestellt}.
```

Insbesondere bei langen Texten ist es vorteilhaft, eindeutige Labels zu vergeben, indem man dem diesen schon ansieht, ob es sich um eine Tabelle, Abbildungen, Grafik oder einen Abschnitt handelt. Z.B. `\label{tab:titel}` oder `\label{abb:titel}` (vgl. Abschnitt 3.6).

Die Befehle `\caption{Titel}` und `\label{AbkuerzungTitel}` müssen innerhalb der Umgebung des Gleitobjektes stehen. Je nach dem ob eine Überschrift oder ein Untertitel gewählt wird, sollte man die Befehle direkt nach der Eröffnung oder kurz vor der Schließung der `table`- oder `figure`-Umgebung einfügen. Auf jeden Fall sollte der `\label{AbkuerzungTitel}`-Befehl direkt dem `\caption{Titel}`-Befehl folgen. So kann sichergestellt, werden, dass die Befehle in einem sinnvollem Zusammenhang stehen.

Wird der Befehl `\caption{Titel}` verwendet, so wird die fortlaufende Nummer automatisch von L^AT_EX hinzugefügt. Dabei werden Tabellen und Abbildungen getrennt voneinander durchnummeriert, so dass später auch ein Tabellen- und ein Abbildungsverzeichnis erstellt werden kann. Für ein zu erstellendes Tabellenverzeichnis verwendet man den Befehl `\listoftables` und für ein Abbildungsverzeichnis schreibt man `\listoffigures`. Geschickterweise fügt man beide Befehle nach dem Befehl `\tableofcontents`, also nach dem Inhaltsverzeichnis, in seinem Quelltext ein (vgl. Abschnitt 3.5).

Zusammenfassend für den bisherigen Sachverhalt die nun weiter modifizierte, aber syntxmäßig vollständige Tabelle 6.1. Zuerst als Syntax, danach in kompilierter Form. Ebenso erfolgt jetzt auch ein Eintrag im Tabellenverzeichnis.

```
\begin{table}[ht]
  \centering
  \begin{tabular}{|>{\sf}c<{.}|p{15cm}|} \hline
    \multicolumn{2}{|c|}{Anleitung zum Erstellen von Darstellungen} \\
    nach \citet[S. 11]{riedwyl:1987} (Auszug) \\ \hline
    a & Gibt die Unterschrift Auskunft über die Frage: was? wo? \\
       & wann? Sind überflüssige Worte eliminiert? \\ \hline
    b & Sind alle Angaben und Zeichen gut zu unterscheiden? \\ \hline
    c & Ist die Quellenangabe vorhanden und vollständig? \\ \hline
    d & Ist der dargestellte Sachverhalt nicht trivial? Erfüllt \\
       & die Darstellung die Anforderungen der Vollständigkeit, \\
       & Genauigkeit und Anschaulichkeit? \\ \hline
  \end{tabular}
  \caption{Checkliste Darstellungen}\label{tab:darstellung}
\end{table}
```

Das ergibt die Tabelle 6.1.

Literaturverzeichnis

CARLISLE, D. P. (1994): *The hhline package* ../texmf/doc/latex/tools/hhline.dvi.

——— (2000): *The longtable package* ../texmf/doc/latex/tools/longtable.dvi.

——— (2001): *The dcolumn package* ../texmf/doc/latex/tools/dcolumn.dvi.

Anleitung zum Erstellen von Darstellungen nach Riedwyl (1987, S. 11) (Auszug)	
a.	Gibt die Unterschrift Auskunft über die Frage: was? wo? wann? Sind überflüssige Worte eliminiert?
b.	Sind alle Angaben und Zeichen gut zu unterscheiden?
c.	Ist die Quellenangabe vorhanden und vollständig?
d.	Ist der dargestellte Sachverhalt nicht trivial? Erfüllt die Darstellung die Anforderungen der Vollständigkeit, Genauigkeit und Anschaulichkeit?

Tabelle 6.1: Checkliste Darstellungen

GOOSSENS, M., F. MITTELBACH, UND A. SAMARIN (2002): *Der $\text{\LaTeX}$ Begleiter*. Pearson Studium, München, ISBN: 3-8273-7044-2, 600 S., 39,95 €.

KOPKA, H. (1992): *$\text{\LaTeX}$: Erweiterungsmöglichkeiten*, Bd. 3. Addison–Wesley, München, ISBN: 3-89319-461-4, 552 S.

MITTELBACH, F., UND D. P. CARLISLE (2001): *A new implementation of $\text{\LaTeX}$'s *tabular* and *array* environment* `../texmf/doc/latex/tools/array.dvi`.

RIEDWYL, H. (1987): *Graphische Gestaltung von Zahlenmaterial*. Haupt, Stuttgart, 3 Aufl.

THEISEN, M. R. (2000): *Wissenschaftliches Arbeiten*. Verlag Vahlen, München, 10 Aufl., ISBN: 3-8006-2628-4, 292 S., 13,00 €.